On functional encryption : new constructions and some relations

Chen, Jie

2013

Chen, J. (2013). On functional encryption : new constructions and some relations. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/51194>

<https://doi.org/10.32657/10356/51194>

**ON FUNCTIONAL ENCRYPTION:
NEW CONSTRUCTIONS AND SOME
RELATIONS**

CHEN JIE

School of Physical and Mathematical Sciences

**A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy**

2013

Acknowledgements

Through the course of my study in NTU, I have gained much more than I originally anticipated. The period of time I spent at NTU certainly brought major transformations to my entire life as a result of friendship, team work, persistence and intellectual education. Many thanks to my parents for their unconditional love, support, and encouragement. I would never have been able to finish my pursuit of PhD without their continuous support and best wishes. They were always there cheering me up and stood by me through the good times and bad.

My deepest gratitude goes first to my supervisor Huaxiong Wang who provided me with the amazing opportunity to study and do research in his lab, who expertly guided me through my PhD education and who shared the excitement of four years of research. His expertise in cryptology improved my research skills and prepared me for future challenges. I would especially like to express my sincere thanks for his invaluable constructive feedback on my work and for guiding me through the countless difficult times that I encountered during my study.

I am much grateful to Professor San Ling who took time out of his full schedule and gave me much guidance throughout my PhD education. Though his suggestions, I have obtained a much clearer research direction and more efficient research methods. I will keep his earnest guidance in mind forever.

I would also especially like to sincerely thank Hoeteck Wee for his invaluable supervision, unfailing enthusiasm and respectful professionalism in guiding me through the difficulties during the long research period. Without his bountiful help, I would not have achieved so much progress so fast.

I would also like to extend my sincere thanks to all other teachers who gave me so much comprehensive knowledge during the past four years. I have benefited so much from their courses and encouragement. I would never forget their edification and enlightening.

I would like to express my special thanks to Hoon Wei Lim who helped me a lot in discussion and participated in almost all my work in NTU, and gave me much help in thesis writing as well.

I would also like to thank all my dear friends: Yang Ding, Jian Guo, Le Su, Lei Wei, Liangfeng Zhang, Yun Zhang, Jun Chen, Lei Jiang, Li Li, Yaqiong Liu, Yuling Liu, Zhongliang Ma, Hao Sun, et al for their good company and encouragement. I have profited greatly from the assistance and wisdom of these friends. I would like to take this opportunity to thank them all and wish they achieve more progress during their study, research, and work period.

Many thanks go to the three anonymous reviewers of this thesis for their time and comments, which help significantly in improving a preliminary version of this thesis.

Finally, I am very grateful to the Division of Mathematical Sciences for providing excellent facilities and perfect environment for us post graduate students. I feel privileged to have been able to study as a post-graduate student of this division. I should like to express my sincere thanks to all the members of this division. Their company has certainly lightened up my life. I would never forget their caring and friendship.

List of Works

Below is the list of work done during my PhD studies in NTU.

1. Shorter ibe and signatures via asymmetric pairings
J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee
Pairing 2012
2. Anonymous and adaptively secure revocable ibe with constant-size public parameters
J. Chen, H. W. Lim, S. Ling, L. Su, and H. Wang
In submission
3. Revocable identity-based encryption from lattices
J. Chen, H. W. Lim, S. Ling, H. Wang, and K. Nguyen
ACISP 2012
4. The relation and transformation between hierarchical inner product encryption and spatial encryption
J. Chen, H. W. Lim, S. Ling, and H. Wang
Des. Codes Cryptography 2012
5. Spatial encryption supporting non-monotone access structure
J. Chen, H. W. Lim, S. Ling, L. Su, and H. Wang

Des. Codes Cryptography 2013, accepted

6. Efficient fully secure spatial encryption schemes under standard assumptions

J. Chen, H. W. Lim, S. Ling, and H. Wang

In submission

7. Achieving short ciphertexts/signatures and full security for attribute-based systems with threshold access structures

C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, and H. Wang

CT-RSA 2013

8. Combined public-key schemes: The case of abe and abs

C. Chen, J. Chen, H. W. Lim, Z. Zhang, and D. Feng

ProvSec 2012

I played a lead or active role in finding and writing all the above results in [1-8]. The results in [1-4] are involved in this thesis.

Table of Contents

1	Introduction	1
1.1	Public Key Encryption	1
1.2	More on Function Encryption	5
1.3	Scope of Thesis	9
1.4	Organization	10
2	Definitions	15
2.1	Definition of Functional Encryption	15
2.2	Instantiations of Functional Encryption	18
2.3	Revocable Identity-Based Encryption	23
2.4	Notations	26
3	Background on Pairings	29
3.1	Dual Pairing Vector Spaces	29
3.2	Complexity Assumptions	31
4	Background on Lattices	33
4.1	Integer Lattices	33
4.2	The Gram-Schmidt Norm and Trapdoors for Lattices	34

4.3	Discrete Gaussian Distribution	35
4.4	Sampling Algorithms	35
4.5	The LWE Hardness Assumption	36
4.6	Encoding Identities as Matrices	37
5	Shorter IBE via Asymmetric Pairings	39
5.1	Motivation	40
5.2	Subspace Assumptions via SXDH	41
5.3	Statistical Indistinguishability Lemmas	45
5.4	Identity-Based Encryption	45
5.5	Extensions	55
5.6	Discussion	81
6	Efficient Revocable IBE Schemes	85
6.1	Binary-Tree Data Structure	85
6.2	Revocable IBE from Bilinear Pairings	87
6.3	Revocable IBE from Lattices	117
7	The Relation and Transformation between HIPE and SE	131
7.1	Motivation	132
7.2	Concepts from Linear Algebra	135
7.3	Transformation from HIPE to SE	136
7.4	Transformation from SE to HIPE	141
7.5	Working in $\mathbb{Z}_N$	146
7.6	Discussion	147
8	Conclusions	151
8.1	Concluding Remarks	151
8.2	Suggestions for Future Work	152
	Bibliography	156

A Estimated Bit Sizes of Elements in Bilinear Groups	167
B Nominally Semi-Functional Keys	169

List of Figures

6.1	A graphical description of the KUNodes algorithm.	87
7.1	Partial Relations in FE.	132

List of Tables

2.1	Notations.	26
5.1	Comparisons between existing and our IBE schemes.	82
6.1	Comparisons between existing and our RIBE schemes.	117
7.1	Comparisons between existing and derived HIPE/SE schemes. . .	148
A.1	Estimated bit sizes of elements in bilinear groups.	167

Abbreviations

ABE	Attribute-Based Encryption
BHIBE	Broadcast Hierarchical Identity-Based Encryption
CIIBE	Co-Inclusive Identity-Based Encryption
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
CP-FE	Ciphertext-Policy Functional Encryption
DBDH	Decisional Bilinear Diffie-Hellman
DBDHE	Decisional Bilinear Diffie-Hellman Exponent
DDH	Decisional Diffie-Hellman
DDH1	Decisional Diffie-Hellman Assumption in G_1
DDH2	Decisional Diffie-Hellman Assumption in G_2
DLIN	Decisional Linear
DPBDHE	Decisional Parallel Bilinear Diffie-Hellman Exponent
DPVS	Dual Pairing Vector Spaces
DS1	Decisional Subspace Assumption in G_1
DS2	Decisional Subspace Assumption in G_2
ECC	Elliptic-Curve Cryptography
FE	Functional Encryption
FIBE	Fuzzy Identity-Based Encryption

FRD	Full-Rank Differences
FSIBE	Forward-Secure Identity-Based Encryption
GDS1	Generalized Decisional Subspace Assumption in G_1
GDS2	Generalized Decisional Subspace Assumption in G_2
HIBE	Hierarchical Identity-Based Encryption
HIPE	Hierarchical Inner Product Encryption
HVE	Hidden-Vector Encryption
IBE	Identity-Based Encryption
IIBE	Inclusive Identity-Based Encryption
IPE	Inner Product Encryption
KP-ABE	Key-Policy Attribute-Based Encryption
KP-FE	Key-Policy Functional Encryption
LSSS	Linear Secret-Sharing Scheme
LWE	Learning With Errors
PKC	Public-Key Cryptography
PKE	Public-Key Encryption
PKE-CNF	Public-Key Encryption schemes supporting Conjunctive Normal Form
PKE-CQ	Public-Key Encryption schemes supporting Comparison Queries
PKE-DNF	Public-Key Encryption schemes supporting Disjunctive Normal Form
PKE-EQ	Public-Key Encryption schemes supporting Equality Queries
PKE-KS	Public-Key Encryption supporting Keyword Search
PKE-PE	Public-Key Encryption schemes supporting Polynomial Evaluation
PKE-SQ	Public-Key Encryption schemes supporting Subset Queries
PPT	Probabilistic Polynomial-Time
RIBE	Revocable Identity-Based Encryption
SE	Spatial Encryption
SXDH	Symmetric External Diffie-Hellman

Abstract

In recent years, a new, generalized notion of public-key primitive called *Functional Encryption* (FE) has emerged. In the FE setting, public keys and their matching secret keys can be associated with identities, attributes, policies, predicates and so on. FE covers a large number of primitives, notably, *Identity-Based Encryption* (IBE), *Inner Product Encryption* (IPE), *Spatial Encryption* (SE), *Attribute-Based Encryption* (ABE) and has numerous applications, particularly in the domains of access control, content distribution, mail filtering, data searching, broadcasting, tracing, and biometrics. Many works on FE in the literature have been targeting on improving the efficiency of constructions, simplifying the relevant assumptions, and achieving stronger security (e.g. full security and adaptively attribute-hiding). In this thesis, we examine three aspects of FE: efficiency of IBE, key revocation in IBE, and the relation between two major classes of FE, namely, *Hierarchical IPE* (HIPE) and SE.

It is desirable that an IBE scheme be as efficient as possible, if it were to have any impact on practical applications. Ideally, we would like to have constant-size global public key, secret keys, and ciphertexts. Moreover, the scheme should ideally achieve full security, namely to be resilient even against an adversary that adaptively selects an identity to attack based on previous secret keys. There are a

few IBE schemes that achieve these features and rely on the *Decisional Bilinear Diffie-Hellman* (DBDH) and *Decisional Linear* (DLIN) assumptions. However, from a practical stand-point, these schemes are still not very efficient as they have relatively large global public key, ciphertexts, and secret keys. In Chapter 5, we will present an efficient IBE scheme under the *Symmetric External Diffie-Hellman* (SXDH) assumption via asymmetric pairings. In our IBE scheme, all parameters have constant numbers of group elements, and are shorter than those of previous constructions based on DLIN assumption.

Key revocation in IBE systems is known to be non-trivial. This is so since a user's identity is itself a public key. Moreover, the private key corresponding to the identity needs to be obtained from a trusted key authority through an authenticated and secrecy protected channel. So far, there exist only a very small number of *Revocable IBE* (RIBE) schemes that support non-interactive key revocation, in the sense that the user is not required to interact with the key authority or some kind of trusted hardware to renew his private key without changing his public key (or identity). These schemes are either proven to be only selectively secure or have global public key which grow linearly in a given security parameter. In Chapter 6, we will present two constructions of non-interactive RIBE that satisfy all the following three attractive properties: (i) proven to be adaptively secure under the SXDH and the DLIN assumptions, respectively; (ii) have constant-size global public key; and (iii) preserve the anonymity of ciphertexts—a property that has not yet been achieved in all the current schemes. On the other hand, we note that all the previous RIBE schemes are constructed from bilinear pairings. In recent years, however, lattice-based IBE has received considerable attention from the cryptographic research community. In the spirit of expanding the study of lattice-based IBE, we also show, how to construct an RIBE scheme in the lattice setting. This will also be presented in Chapter 6.

HIPE and SE are two important classes of FE that have numerous applications. Although HIPE and SE both involve some notions of linear algebra,

the former works in vectors while the latter is based on (affine) spaces. Moreover, they currently possess different properties in terms of security, anonymity (payload/attribute-hiding) and ciphertext sizes, for example. In Chapter 7, we will study the relation and transformation between HIPE and SE. We discover some interesting and novel property-preserving transformation techniques that enable a generic construction of an SE scheme from an HIPE scheme, and vice versa. The “bridging” between HIPE and SE using our techniques converts the problem of constructing an HIPE scheme with certain properties into a problem of constructing an SE scheme with the required (similar) properties, and vice versa.

Chapter

1

Introduction

Contents

1.1	Public Key Encryption	1
1.2	More on Function Encryption	5
1.3	Scope of Thesis	9
1.4	Organization	10

This chapter gives an overview of the thesis. We provide the motivation for our research and describe the scope of this thesis. In this chapter, we also present the overall structure of the thesis.

1.1 Public Key Encryption

In cryptography, the goal of encryption is generally for two parties, namely the sender and the receiver, to communicate while protecting the confidentiality, integrity, and authenticity of their information. In traditional or symmetric cryptography, both the sender and the receiver share a same secret key, i.e., the sender uses the secret key to encrypt the message while the receiver uses the same secret key to decrypt the message. However, this method presented a problem: how to

get the sender and the receiver to agree on a same secret key if they cannot even secretly communicate? They would need another way to communicate the secret key, such as meeting in-person in a secure location. Clearly, this is impractical if they are in distant and separate physical locations, although many schemes assume that secret keys can be distributed safely.

In order to solve this problem, the concept of *Public-Key Cryptography* (PKC) was introduced by Diffie and Hellman [43] in 1976. In a PKC system, each party generates and has a pair of keys: a public key and a secret key (by running some algorithm, see below). The keys live up to their names: the public key is published while the secret key is kept secret. PKC can be used not only for privacy (encryption), but also for authentication (digital signatures) and other various techniques [35]. In this thesis, we focus on *Public-Key Encryption* (PKE) systems and take PKE for example: a message encrypted using the public key can only be decrypted by the user who possesses the corresponding secret key; in this way, all communications involve only public keys while the need for the sender and receiver to share secret information is eliminated. Unlike symmetric cryptography, the security of PKC relies on the intractability of certain mathematical problems. In other words, the secret key is linked mathematically to the public key, it is always possible to attack a PKC system by deriving the secret key from the public key. Typically, the defense against this is to make the problems of deriving the secret key from the public key as challenging as possible. Currently, there are two major algorithms for PKC systems: (i) RSA [85] (standing for Ron Rivest, Adi Shamir, and Leonard Adleman), which is based on the presumed hardness of factoring large integers, is the most widely deployed and used public-key system in real world applications; (ii) ECC [70, 60] (referring to Elliptic-Curve Cryptography), which assumes that finding the discrete logarithm of a random elliptic curve element with respect to a known base point is infeasible, has been studied for many years before the emergence of pairing-based systems.

Identity-Based Encryption Moving beyond the traditional boundaries of PKE, the idea of using any arbitrary string as a public key was conceived by Shamir [87] and such a primitive is known as *Identity-Based Encryption* (IBE) [87, 22, 40]. It allows a public (encryption) key to be constructed based on some public identification information (or identifier). To encrypt a message, one needs only the intended recipient’s identifier and a set of public system parameters. The secret (decryption) key corresponding to a public key is generated by and obtained from a trusted authority. IBE has been extensively studied particularly over the last decade. We now have constructions of IBE schemes from a large class of assumptions, namely pairings, quadratic residuosity and lattices, starting with the early constructions in the (idealized) random oracle model [22, 40, 52], to more recent constructions in the (plain) standard model [37, 20, 21, 39, 2]. The emergence of IBE simplifies various conventional systems that manage a large number of public keys, rather than storing a big database of public keys the system can either derive these public keys from usernames, or simply use the integers $1, \dots, N$ as distinct public keys [22].

Functional Encryption Over the years, the concept of IBE has been generalized. Since identifiers in the context of IBE can be any arbitrary strings, public keys and their matching private keys can also be associated with attributes, policies, predicates and so on. In recent endeavors towards a more generalized notion of encryption, a new class of cryptographic primitive called *Functional Encryption* (FE) [28, 86, 81, 53, 25, 58, 63, 76, 33] emerged. Roughly speaking, FE supports restricted secret keys that enable a secret key holder to learn a specific function of encrypted data, but learn nothing else about the data. For example, given an encrypted program the secret key may enable its holder to learn the output of the program on a specific input without learning anything else about the program. Generally speaking, in an FE system for a functionality $\hat{F}(\cdot, \cdot)$ defined over some key space $\mathcal{K}$ and some plaintext space $\mathcal{X}$, an authority holding a master secret

key can generate a secret key $\text{SK}_{\hat{v}}$ associated with $\hat{v} \in \mathcal{K}$ that enables the computation of the function $\hat{F}(\hat{v}, \cdot)$ on encrypted data (namely, the decryptor can compute $\hat{F}(\hat{v}, \hat{x})$ from an encryption of $\hat{x} \in \mathcal{X}$ using $\text{SK}_{\hat{v}}$) [33].

We note that the above concept from [33] is more general and different from traditional encryption concept [76, 25, 58] in the sense that it is not restricted to be the model of learning “all-or-nothing” for the encrypted plaintext/message any more. Actually, the contributions made in this thesis are not in the direction of FE of [33], but rather in that of [76, 25, 58]. Namely, we focus on the following classes of FE: a plaintext $\hat{x} \in \mathcal{X}$ is itself a pair $(\widehat{\text{ind}}, \widehat{\text{msg}}) \in \mathcal{I} \times \mathcal{M}$ where $\widehat{\text{ind}}$ is an element of an index space $\mathcal{I}$ and $\widehat{\text{msg}}$ is an element of a payload message space $\mathcal{M}$.¹ There are two types of secrecy in these applications: *payload-hiding* (with public index or non-anonymous in the IBE setting) and *attribute-hiding* (with hidden index or anonymous) [58]. Roughly speaking, the former requires that only the message $\widehat{\text{msg}}$ be concealed from an adversary, while the latter requires that both the index $\widehat{\text{ind}}$ and the message $\widehat{\text{msg}}$ be concealed from the adversary. We note that when an FE scheme has $\hat{v}$ and $\widehat{\text{ind}}$ as strings and $\hat{F}(\hat{v}, (\widehat{\text{ind}}, \widehat{\text{msg}}))$ outputs the corresponding payload message $\widehat{\text{msg}}$ iff $\widehat{\text{ind}} = \hat{v}$, it is an IBE scheme. So far as we know, the widest class of FE systems supporting the attribute-hiding security is *Inner Product Encryption* (IPE) [58], which was proposed by Katz, Sahai, and Waters for problems of searching on encrypted data. In the IPE setting, each of $\hat{v}$ and $\widehat{\text{ind}}$ corresponds to a vector over a field or ring while $\hat{F}(\hat{v}, (\widehat{\text{ind}}, \widehat{\text{msg}})) = \widehat{\text{msg}}$ iff $\widehat{\text{ind}} \cdot \hat{v} = 0$, where $\widehat{\text{ind}} \cdot \hat{v}$ is the inner-product of the vectors $\widehat{\text{ind}}$ and $\hat{v}$. On the other hand, the widest class of FE systems supporting expressive access control is *Attribute-Based Encryption* (ABE) [86, 53, 79, 18, 63, 76], which can be expressed using AND, OR, and Threshold gates. However, such FE systems do not satisfy the attribute-hiding security. In the ABE setting, either one of $\hat{v}$ and $\widehat{\text{ind}}$ is a

¹In this thesis, we distinguish the notion of “plaintext” and “message”. In addition to a message $\widehat{\text{msg}}$, a plaintext $\hat{x}$ also contains an index $\widehat{\text{ind}}$. Here, $\widehat{\text{ind}}$ plays a similar role of $\hat{v}$ in a secret key, in other words, a ciphertext is an encryption of a message $\widehat{\text{msg}}$ under an index $\widehat{\text{ind}}$. It could be an identity, an attribute, etc. Take IBE for example, both $\hat{v}$ and $\widehat{\text{ind}}$ are identities.

set of attributes and the other is an access structure or span program along with a set of attributes, while $\hat{F}(\hat{v}, (\widehat{\text{ind}}, \widehat{\text{msg}})) = \widehat{\text{msg}}$ iff the set of attributes satisfies the access role defined by the access structure. Given such flexibility and generalization, FE has numerous applications, particularly in the domains of access control, content distribution, mail filtering, data searching, broadcasting, tracing, and biometrics [81, 53, 79, 18, 90, 28, 25, 24, 32, 86].

1.2 More on Function Encryption

Here, we review more related notions for FE, such as security, complexity assumptions, construction tools, security proof techniques, key revocation etc., which make a better understanding of our work.

Full/Selective Security There are two types of security model in the applications of FE: *full/adaptive security* and *selective security*. Roughly speaking, the former allows the adversary to adaptively query some secret keys for $\hat{v}_1, \hat{v}_2, \dots \in \mathcal{K}$ before submitting a challenge $\widehat{\text{ind}}^* \in \mathcal{I}$ while the latter is a weaker model that requires the adversary to declare a challenge $\widehat{\text{ind}}^* \in \mathcal{I}$ before seeing the system's global public key.

Complexity Assumption The security of a PKC system is always based on the intractability of certain mathematical problems. The underlying hard problems, on which the security of the cryptographic schemes provably rely, are called complexity assumptions. It is desirable that the underlying assumptions in the cryptographic schemes are well-established or standard, such as the *Decisional Bilinear Diffie-Hellman* (DBDH) [22], *Decisional Linear* (DLIN) [26], *Symmetric External Diffie-Hellman* (SXDH) [14] assumptions from bilinear pairings and *Learning With Errors* (LWE) [84, 80] assumption from lattices. There are also some less standard (and non-static) assumptions that are depended on the query complexity of the adversary, such as *Decisional Bilinear Diffie-Hellman Exponent*

(DBDHE) [30] and *Decisional Parallel Bilinear Diffie-Hellman Exponent* (DPB-DHE) [94] assumptions.

Construction Tools In the literature, we now have constructions of FE schemes from a large class of assumptions, namely pairings [22], quadratic residuosity [40], and lattices [52]. We will focus on pairings but also involve a construction from lattices in this thesis. All of our pairing-based constructions rely on *Dual Pairing Vector Spaces* (DPVS) proposed by Okamoto and Takashima [74, 75], which is a technique to achieve orthogonality in prime order bilinear groups. DPVS has been one of the main tools to obtain fully secure and attribute-hiding instantiations of FE and has numerous applications [74, 75, 63, 76, 77, 78, 61]. Another important pairing-based tool is composite order bilinear groups [31, 62], which is also a technique to achieve orthogonality but lacking in efficiency [47, 61].

Security Proof Techniques To reduce the security of a cryptographic scheme to the underlying hard problems (in other words, to prove that breaking a cryptographic scheme is at least as hard as solving the hard problems), we traditionally use *partitioning strategies* [20, 21, 92, 86, 53, 18]. In such a strategy, the simulator partitions the key space $\mathcal{K}$ into two pieces: $\mathcal{K}_1$, secret keys that the simulator can generate and $\mathcal{K}_2$, those that it cannot. The index space $\mathcal{I}$ is also partitioned into two pieces accordingly: $\mathcal{I}_1$, a challenge ciphertext that the simulator cannot generate and $\mathcal{I}_2$, those that it can. To ensure that all keys the adversary queries all fall in $\mathcal{K}_1$ and that the challenge index falls in $\mathcal{I}_2$, most prior works have to rely on the selective security model [20, 21, 86, 53, 18] or have global public key which grow linearly in a given security parameter [92].

The *Dual System Encryption* framework was proposed by Waters [93] with the goal of moving beyond the constraints of the traditional partitioning reduction techniques, which has become a powerful tool to obtain full security for various classes of FE [62, 63, 76]. Briefly, in a dual system security proof, the simulator could generate any secret key and any challenge ciphertext. There are two types

of keys and ciphertexts: normal and semi-functional. The normal secret keys and ciphertexts are used in the real system; while the semi-functional objects are only provided for definitional purposes, and are not part of the system, which are only introduced in the hybrid security proof. Moreover, a ciphertext can be properly decrypted using a secret key unless both the secret key and the ciphertext are semi-functional, in which case decryption will fail with all but negligible probability. In the process of security proof, we first transform the ciphertext from normal to semi-functional, and then transform the secret keys from normal to semi-functional one by one. Finally, we arrive at a security game that only requires the simulator to generate semi-functional secret keys and ciphertexts while security can be proved directly.

Key Revocation in IBE IBE allows one's identity to be directly used as a public key. This obviates the need for a public key certificate that attests the binding between the identity and a (seemingly) random key, as with the more traditional certificate-based public key systems. Thus, IBE systems have simpler public key management than that of certificate-based systems. In IBE, however, a private key (corresponding to an identity) needs to be generated by a trusted key authority. This and the fact that a user's identity is itself a public key complicates key renewal or revocation—one cannot simply change his public key, as this will change his identity as well. Moreover, a system user's public key may need to be removed for various reasons. For example, the private key corresponding to the public key has been stolen; the user has lost his private key; or the user is no longer a legitimate system user. In these cases, it is important that the public/private key pair be revoked and replaced by new keys. While there has been a great deal of work on IBE in recent years, not much work has been devoted to *key revocation*.

One simple way to alleviate the key revocation problem in the IBE setting is to append a validity period to a target identity during encryption [22]. This results in a public key with a limited validity period, and hence, restricting the

window of exposure should the corresponding private key be compromised. If the validity period is sufficiently short, one may not require an explicit key revocation mechanism since an exposed private key is of little value to an adversary beyond the specified validity period. However, one major drawback of this approach is that each user has to periodically renew his private key. As a consequence, the key authority's workload increases linearly in the number of non-revoked users. Further, we must ensure that each transmission of a new private key between the key authority and a non-revoked user is performed through some form of authenticated and secure channel.

There exist some improved key revocation techniques in the literature, for example [64, 56]. However, they require interactions either between the user and the key authority, as before, or between the user and some kind of trusted hardware. These may not always be practical. The first non-interactive, *revocable IBE* scheme (namely, RIBE scheme) that neither presupposes the existence of trusted hardware nor requires a secure channel between the user and the key authority, is due to Boldyreva et al. [19]. Their scheme borrows the concept of *Fuzzy IBE* (FIBE) [86] in which encryption of a message is associated with two “attributes”, namely identity of the receiver and time period. The corresponding decryption key is also split into two private components, matching the identity and the time period, respectively. The private component that corresponds to the identity is essentially similar to a regular private key in IBE and it is issued to a user by the key authority through a secure channel. On the other hand, the private key component corresponding to the time period is regarded as a key update and is published by the key authority to all users. (Here the key update is public information and does not require secrecy protection.) Thus, to revoke a user, the key authority simply stops distributing the key updates for that user. Moreover, by making use of the binary-tree data structure [6, 73], every update of the revocation list only requires the key authority to perform logarithmic work in the maximal number of users and linear in the number of revoked users.

1.3 Scope of Thesis

Motivated by the versatile features of FE, in this thesis, we examine three aspects of FE: efficiency of IBE, key revocation in IBE, and the relation between two major classes of FE, namely, *Hierarchical IPE* (HIPE) [75] and *Spatial Encryption* (SE) [23]. More precisely:

1. Firstly, we present an efficient IBE scheme under the SXDH assumption in asymmetric bilinear groups. In our IBE scheme, all parameters have constant numbers of group elements, and are shorter than those of previous constructions based on DLIN assumption [61]. Our constructions use both Dual System Encryption and DPVS. Specifically, we show how to adapt the recent DLIN-based instantiations of Lewko [61] to the SXDH assumption. Furthermore, we show that our work can be extended to various FE schemes. Particularly, we demonstrate how to instantiate IPE and *Key-Policy FE* (KP-FE) [76] of which all parameters are shorter than those constructed under the DLIN assumption [76, 77]. These instantiations are syntactically simpler than those relying on the DLIN assumption [76], which save us roughly a factor of 1/3 in both the sizes of secret keys and ciphertexts, while the reductions for the global public key and the master secret key are even more substantial.
2. Secondly, we show some new constructions of RIBE [19] that are from bilinear pairings and lattices respectively. More precisely:
 - We present two constructions of RIBE from bilinear pairings that satisfy all the following three attractive properties: (i) proven to be adaptively secure under the SXDH and the DLIN assumptions, respectively; (ii) have constant-size global public key; and (iii) preserve the anonymity of ciphertexts—a property that has not yet been achieved in all the current schemes [19, 66]. So far, there exist only a very small number of RIBE schemes that support non-interactive key revocation,

in the sense that the user is not required to interact with the key authority or some kind of trusted hardware to renew his private key without changing his public key (or identity). These schemes are either proven to be only selectively secure or have global public key which grow linearly in a given security parameter.

- We present an efficient IBE scheme from lattices with efficient key revocation, so far, all the previous RIBE schemes are constructed from bilinear pairings [19, 66]. We adopt multiple trapdoors from the lattice IBE schemes of [2, 52] to realize key revocation, which in turn, makes use of binary-tree data structure [6, 73, 19, 66]. We prove that our scheme is selective secure in the standard model and under the LWE assumption, which is as hard as the worst-case approximating short vectors on arbitrary lattices [84, 80].
3. Thirdly, we show the relation and transformation between HIPE and SE. HIPE and SE are two important classes of FE that have numerous applications. Although HIPE and SE both involve some notions of linear algebra, the former works in vectors while the latter is based on (affine) spaces. Moreover, they currently possess different properties in terms of security, anonymity (payload/attribute-hiding) and ciphertext sizes, for example. We discover some interesting and novel property-preserving transformation techniques that enable generic construction of an SE scheme from an HIPE scheme, and vice versa. The “bridging” between HIPE and SE using our techniques converts the problem of constructing an HIPE scheme with certain properties into a problem of constructing an SE scheme with the required (similar) properties, and vice versa.

1.4 Organization

The remainder of this thesis is organized as follows:

Preliminaries We present background material on FE, tools of constructions in Chapters 2 and 3, 4, respectively. These are prerequisites for understanding our work. In Chapter 2, we first borrow the definition and the game-based security model for FE from [33] which are adequate to define all encryption systems in this thesis. We then define some concrete instantiations (such as IBE, IPE, SE, and KP-FE) that are related to our work using the syntax of FE. Subsequently, we also give a formal definition of RIBE, which involves some additional algorithms, and its associated security model. In addition, notations of this thesis are given.

In Chapter 3, we provide background on bilinear pairings. We first briefly describe bilinear pairing groups and how to generate random dual orthonormal bases for DPVS, on which almost of all our constructions are based. We also describe the definitions of SXDH, DLIN assumptions on which the security of our cryptographic schemes provably relies.

In Chapter 4, we provide background on lattices for our construction of lattice-based RIBE scheme in chapter 6. We introduce the definitions of integer lattices, the Gram-Schmidt norm, discrete Gaussians and also sampling algorithms to generate trapdoors for lattices. Subsequently, the core hard problem—LWE for the security proof our RIBE scheme is given.

Shorter IBE via Asymmetric Pairings In Chapter 5, we present the proposal of an efficient IBE scheme under the SXDH assumption in asymmetric bilinear groups. We begin by presenting a framework—Subspace assumptions derived from the SXDH assumption, which are analogues of the DLIN-based Subspace assumptions given in [61, 76], and then derive statistical indistinguishability lemmas. The Subspace assumptions and statistical indistinguishability lemmas are our main tools for the forthcoming security proofs of encryption schemes in this chapter and fully secure and anonymous RIBE scheme in Chapter 6. Subsequently, we show our efficient IBE scheme and extend our work to various FE schemes—IPE and KP-FE, all

parameters of which are shorter than those constructed under the DLIN assumption [61, 76, 77]. Finally, we give comparisons between existing and our IBE schemes followed by some discussions.

Efficient Revocable IBE Schemes In Chapter 6, we present our new constructions of RIBE that are from bilinear pairings and lattices respectively. We first describe what is the binary-tree data structure [19], which is used in RIBE schemes to achieve non-interactive key revocation with logarithmic complexity in key updates, and how does it work.

Subsequently, we present two constructions of adaptively secure and anonymous RIBE with constant-size global public key. We first give the construction under the SXDH assumption, the security proof of which will borrow some background from Chapter 5—the Subspace assumptions and statistical indistinguishability lemmas. We then briefly describe the DLIN-based scheme and finally give comparisons between existing and our RIBE schemes.

After that, we present an efficient RIBE scheme from lattices. Since we use the Agrawal et al. lattice IBE scheme [2] as a building block for our construction, we first briefly recall their scheme. We then show our construction and security reduction under the LWE assumption.

The Relation and Transformation between HIPE and SE In Chapter 7, we study the relation and transformation between HIPE and SE. We first derive some lemmas, which are essential for our forthcoming property-preserving transformation. Subsequently, we discover the techniques for constructing an SE scheme from an HIPE scheme, and vice versa that work in $\mathbb{Z}_q$ for some prime q (e.g. prime order bilinear groups). We also explain how an SE or HIPE scheme is defined and why our transformation techniques also work in $\mathbb{Z}_N$ for some hard factoring composite number N (e.g. composite order bilinear groups). Finally, we compare some selected existing

HIPE and SE schemes with schemes derived using our property-preserving transformation techniques.

Conclusions In the final chapter of this thesis, Chapter 8, we give concluding remarks about our proposals in Chapters 5, 6, and 7. These include the problems that we have studied and a summary of our research findings. We also provide some suggestions for future work related to our proposals.

Chapter 2

Definitions

Contents

2.1	Definition of Functional Encryption	15
2.2	Instantiations of Functional Encryption	18
2.3	Revocable Identity-Based Encryption	23
2.4	Notations	26

In this chapter, we first borrow the definition and the game-based security model for FE from [33] which are adequate to define all encryption systems in this thesis. We then define some concrete instantiations that are related to our work using the syntax of FE. Subsequently, we also give a formal definition of RIBE, which involves some additional algorithms, and its associated security model.

2.1 Definition of Functional Encryption

As in [33], we first describe a functionality $\hat{F}$ of the syntactic definition of FE. The functionality $\hat{F}$ describes the functions of a plaintext that can be learned from the ciphertext given a secret key:

Definition 2.1. *A functionality $\hat{F}$ defined over $(\mathcal{K}, \mathcal{X})$ is a function $\hat{F} : \mathcal{K} \times \mathcal{X} \rightarrow$*

$\{0, 1\}^*$ described as a (deterministic) Turing Machine. The set $\mathcal{K}$ is called the key space and the set $\mathcal{X}$ is called the plaintext space.

An FE scheme for the functionality $\hat{F}$ enables one to evaluate $\hat{F}(\hat{v}, \hat{x})$ given the encryption of $\hat{x}$ and a secret key $\mathbf{SK}_{\hat{v}}$ for $\hat{v}$ (namely, the decryptor can compute $\hat{F}(\hat{v}, \hat{x})$ from an encryption of $\hat{x} \in \mathcal{X}$ using $\mathbf{SK}_{\hat{v}}$). The algorithm for evaluation $\hat{F}(\hat{v}, \hat{x})$ using $\mathbf{SK}_{\hat{v}}$ is called *decrypt*. More precisely, an FE scheme is defined as follows:

Definition 2.2. A Functional Encryption (FE) scheme for a functionality $\hat{F}$ defined over $(\mathcal{K}, \mathcal{X})$ is a tuple of four Probabilistic Polynomial-Time (PPT) algorithms (Setup, KeyGen, Enc, Dec) and an additional, but optional PPT algorithm KeyDel satisfying the following correctness condition for all $\hat{v} \in \mathcal{K}$ and $\hat{x} \in \mathcal{X}$:

$$\begin{aligned}
 (\text{GPK}, \text{MSK}) &\leftarrow \text{Setup}(\lambda) && (\text{generate a global public key and a master secret key}) \\
 \mathbf{SK}_{\hat{v}} &\leftarrow \text{KeyGen}(\text{GPK}, \text{MSK}, \hat{v}) && (\text{generate a secret key for } \hat{v}) \\
 \text{CT}_{\hat{x}} &\leftarrow \text{Enc}(\text{GPK}, \hat{x}) && (\text{encrypt plaintext } \hat{x}) \\
 y &\leftarrow \text{Dec}(\text{GPK}, \mathbf{SK}_{\hat{v}}, \text{CT}_{\hat{x}}) && (\text{use } \mathbf{SK}_{\hat{v}} \text{ to compute } \hat{F}(\hat{v}, \hat{x}) \text{ from } \text{CT}_{\hat{x}}) \\
 \mathbf{SK}_{\hat{v}'} &\leftarrow \text{KeyDel}(\text{GPK}, \hat{v}, \mathbf{SK}_{\hat{v}}, \hat{v}') && (\text{generate a secret key for } \hat{v}' \text{ that satisfies} \\
 &&& \text{a partial order relation denoted as } \hat{v}' \preceq \hat{v})
 \end{aligned}$$

then we require that $y = \hat{F}(\hat{v}, \hat{x})$ with probability 1.

Security Model for Functional Encryption For the plaintext pair $(\hat{x}_{(0)}^*, \hat{x}_{(1)}^*)$ of an adversary's choice, we need the following requirement to make the experiment non-trivial:

$$\hat{F}(\hat{v}, \hat{x}_{(0)}^*) = \hat{F}(\hat{v}, \hat{x}_{(1)}^*) \text{ for all } \hat{v} \text{ for which the attacker has } \mathbf{SK}_{\hat{v}}. \quad (2.1)$$

Then we define a security game for an FE scheme as follows:

Definition 2.3. For $\beta = 0, 1$ define an experiment β , played by a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$, as follows:

- **Setup:** $\mathcal{B}$ runs $(\text{GPK}, \text{MSK}) \leftarrow \text{Setup}(\lambda)$ and gives GPK to $\mathcal{A}$. It also initializes a set $\Pi := \emptyset$ to maintain the secret key query list.
- **Query:** $\mathcal{A}$ adaptively makes repeated key queries of one of three types:
 - **KeyGen:** $\mathcal{A}$ submits a key generation query $\hat{v} \in \mathcal{K}$. $\mathcal{B}$ generates a secret key $\text{SK}_{\hat{v}}$ for $\hat{v}$, but does not give it to $\mathcal{A}$. It instead adds the secret key to the set Π and gives the adversary a reference to the secret key.
 - **KeyDel:** $\mathcal{A}$ specifies a secret key $\text{SK}_{\hat{v}}$ in the set Π for $\hat{v} \in \mathcal{K}$, then it submits a key delegation query for $\hat{v}' \in \mathcal{K}$, where $\hat{v}' \preceq \hat{v}$. $\mathcal{B}$ runs the $\text{KeyDel}(\text{GPK}, \hat{v}, \text{SK}_{\hat{v}}, \hat{v}')$ algorithm to get a new secret key $\text{SK}_{\hat{v}'}$ and adds this to the set Π .
 - **KeyRev:** $\mathcal{A}$ specifies an element of the set Π for a secret key $\text{SK}_{\hat{v}}$. $\mathcal{B}$ removes the item from the set Π and gives $\mathcal{A}$ the secret key. We note at this point there is no need for the challenger to allow more delegate queries on the secret key since $\mathcal{A}$ can run them itself.
- **Challenge:** $\mathcal{A}$ submits two plaintexts $\hat{x}_{(0)}^*, \hat{x}_{(1)}^* \in \mathcal{X}$ satisfying requirement (2.1) and in return, it receives $\text{Enc}(\text{GPK}, \hat{x}_{(\beta)}^*)$.
- **Guess:** $\mathcal{A}$ continues to issue secret key queries as before subject to requirement (2.1) and eventually outputs a bit in $\{0, 1\}$.

For $\beta = 0, 1$ let W_β be the event that $\mathcal{A}$ outputs 1 in Experiment β and define

$$\text{Adv}_{\mathcal{A}}^{\text{FE}}(\lambda) := |\Pr[W_0] - \Pr[W_1]|.$$

Definition 2.4. An FE scheme is secure if for all PPT adversaries $\mathcal{A}$ the function $\text{Adv}_{\mathcal{A}}^{\text{FE}}(\lambda)$ is negligible.

In all encryption systems of this paper, a plaintext $\hat{x} \in \mathcal{X}$ is itself a pair $(\widehat{\text{ind}}, \widehat{\text{msg}}) \in \mathcal{I} \times \mathcal{M}$ where $\widehat{\text{ind}}$ is called an index and $\widehat{\text{msg}}$ is called the payload message. Let $\hat{x}_{(0)}^* = (\widehat{\text{ind}}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$, $\hat{x}_{(1)}^* = (\widehat{\text{ind}}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*) \in \mathcal{X}$ be the adversary's

choice of plaintext pair. The security game can then be defined using Definition 2.3 with the following variations:

- If the adversary outputs challenge indices $\widehat{\text{ind}}_{(0)}^*, \widehat{\text{ind}}_{(1)}^*$ before the **Setup** phase, the security game is then under the *selective security* model. Otherwise it is under the *full/adaptive security* model.
- For anonymity, there are two types of adversaries subject to the following restrictions:
 1. $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)) \neq \widehat{\text{msg}}_{(0)}^*$ and $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)) \neq \widehat{\text{msg}}_{(1)}^*$ for all the key queries, $\hat{v}_i$.
 2. Two challenge messages are equal, i.e., $\widehat{\text{msg}}_{(0)}^* = \widehat{\text{msg}}_{(1)}^*$, and any key query $\hat{v}_i$ satisfies $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)) = \hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*))$, i.e., one of the following conditions:
 - (a) $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)) = \widehat{\text{msg}}_{(0)}^* = \widehat{\text{msg}}_{(1)}^* = \hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*))$;
 - (b) $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)) \neq \widehat{\text{msg}}_{(0)}^*$ and $\hat{F}(\hat{v}_i, (\widehat{\text{ind}}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)) \neq \widehat{\text{msg}}_{(1)}^*$.

The above security game is under *adaptively attribute-hiding* model. If the adversary's queries subject to restriction 1 above, the security game is under the *weakly attribute-hiding* model. Furthermore, if we require $\widehat{\text{ind}}_{(0)}^* = \widehat{\text{ind}}_{(1)}^*$, the security game is then under the *payload-hiding* model. In the IBE setting, we typically use the term “anonymous” instead of attribute-hiding.

2.2 Instantiations of Functional Encryption

We now define some concrete instantiations using the syntax of FE proposed in Section 2.1.

(Hierarchical) Identity-Based Encryption In the *Identity-Based Encryption* (IBE) setting, a functionality $\hat{F}$ is defined over a key space and an index

space using sets of identities. The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for IBE then correspond to all identities id (resp. id'). Here

$$\hat{F}(\text{id}, (\text{id}', \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \text{id}' = \text{id} \\ \perp & \text{otherwise.} \end{cases}$$

A generalization of IBE that reflects an organizational hierarchy is called *Hierarchical IBE* (HIBE) [57, 51]. In the HIBE setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of hierarchical identities with maximal depth d . The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for HIBE then corresponds to all hierarchical identities $(\text{id}_1, \dots, \text{id}_{\ell_s})$ (resp. $(\text{id}'_1, \dots, \text{id}'_{\ell_c})$) of depth at most d . Here

$$\hat{F}((\text{id}_1, \dots, \text{id}_{\ell_s}), ((\text{id}'_1, \dots, \text{id}'_{\ell_c}), \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \ell_s \leq \ell_c \text{ and } \text{id}'_i = \text{id}_i \text{ for all } i \in [\ell_s] \\ \perp & \text{otherwise.} \end{cases}$$

Moreover, $(\text{id}'_1, \dots, \text{id}'_{\ell'_s}) \preceq (\text{id}_1, \dots, \text{id}_{\ell_s})$ iff $\ell_s \leq \ell'_s$ and $\text{id}'_i = \text{id}_i$ for all $i \in [\ell_s]$. Namely $(\text{id}_1, \dots, \text{id}_{\ell_s}, \text{id}_{\ell_s+1}, \dots, \text{id}_{\ell'_s}) \preceq (\text{id}_1, \dots, \text{id}_{\ell_s})$.

(Hierarchical) Inner Product Encryption In the *Inner Product Encryption* (IPE) setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of vectors. The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for IPE then corresponds to all non-zero vectors $\mathbf{v}$ (resp. $\mathbf{x}$). Here

$$\hat{F}(\mathbf{v}, (\mathbf{x}, \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \mathbf{x} \cdot \mathbf{v} = 0 \\ \perp & \text{otherwise.} \end{cases}$$

A generalization of IPE that mirrors an organizational hierarchy is called *Hierarchical IPE* (HIPE or “delegatable” IPE) [75]. In the HIPE setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of hierarchical vectors. Let a hierarchy of depth d vector spaces have the form of $\vec{\mu} := (n, d; \mu_1, \dots, \mu_d)$ where $\mu_0 = 0 < \mu_1 < \mu_2 < \dots < \mu_d = n$. Let $\Phi_i := \mathbb{Z}_q^{\mu_i - \mu_{i-1}} \setminus \{\mathbf{0}\}$

for $i \in [d]$ be the sets of vectors. Let $\Phi := \bigcup_{i=1}^d (\Phi_1 \times \dots \times \Phi_d)$, where the union is a disjoint union. The key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) for HIPE then corresponds to all hierarchical vectors of the form $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ (resp. $(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c})$) of depth at most d in Φ . Here

$$\hat{F}((\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), ((\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}), \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \ell_s \leq \ell_c \text{ and } \mathbf{x}_i \cdot \mathbf{v}_i = 0 \text{ for all } i \in [\ell_s] \\ \perp & \text{otherwise.} \end{cases}$$

Moreover, $(\mathbf{v}'_1, \dots, \mathbf{v}'_{\ell'_s}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ iff $\ell_s \leq \ell'_s$ and $\mathbf{v}'_i = \mathbf{v}_i$ for all $i \in [\ell_s]$. Namely $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$.

Spatial Encryption The notion of *Spatial Encryption* (SE) was proposed as a general framework for constructing identity-based and broadcast encryption systems [23]. In the SE setting, a functionality $\hat{F}$ is defined over a key space and an index space using sets of spaces and vectors respectively. In an n -dimensional SE scheme, the key space $\mathcal{K}$ corresponds to all affine spaces S in $\mathbb{Z}_q^n$ and the index space $\mathcal{I}$ corresponds to all vectors $\mathbf{x}$ in $\mathbb{Z}_q^n$. Here

$$\hat{F}(S, (\mathbf{x}, \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \mathbf{x} \in S \\ \perp & \text{otherwise.} \end{cases}$$

Moreover, $S' \preceq S$ iff S' is a subspace of S .

Functional Encryption with General Relations Okamoto and Takashima [76] recently proposed FE schemes for a wide class of relations that are specified by non-monotone access structures [79] in combination with inner product relations. As with ABE, they proposed two types of FE schemes, the *Key-Policy FE* (KP-FE) and *Ciphertext-Policy FE* (CP-FE) schemes. We first describe the concept of span programs typically required by ABE.

Definition 2.5 (Span Programs [17]). *Let $\{p_1, \dots, p_n\}$ be a set of variables. A span program over $\mathbb{Z}_q$ is a labeled matrix $(\hat{\mathbf{A}}, \hat{\rho})$ where $\hat{\mathbf{A}}$ is an $(\hat{a} \times \hat{b})$ matrix over*

$\mathbb{Z}_q$ and $\hat{\rho}$ is a labeling of the rows of $\hat{\mathbf{A}}$ by literals from $\{p_1, \dots, p_n, \neg p_1, \dots, \neg p_n\}$ (every row is labeled by one literal), i.e., $\hat{\rho} : [\hat{a}] \rightarrow \{p_1, \dots, p_n, \neg p_1, \dots, \neg p_n\}$.

A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0, 1\}^n$ define the submatrix $\hat{\mathbf{A}}_\delta$ of $\hat{\mathbf{A}}$ consisting of those rows whose labels are set to 1 by the input, i.e., either rows labeled by some p_i such that $\delta_i = 1$ or rows labeled by some $\neg p_i$ such that $\delta_i = 0$. (i.e., $\hat{\gamma} : [\hat{a}] \rightarrow \{0, 1\}$ is defined by $\hat{\gamma}(j) = 1$ if $[\hat{\rho}(j) = p_i] \wedge [\delta_i = 1]$ or $[\hat{\rho}(j) = \neg p_i] \wedge [\delta_i = 0]$, and $\hat{\gamma}(j) = 0$ otherwise. Hence, we have $\hat{\mathbf{A}}_\delta := (\hat{\mathbf{A}}^{[j]})_{\hat{\gamma}(j)=1}$, where $\hat{\mathbf{A}}^{[j]}$ is the j -th row of $\hat{\mathbf{A}}$.)

The span program $(\hat{\mathbf{A}}, \hat{\rho})$ accepts δ iff $\mathbf{1} \in \text{span}\langle \hat{\mathbf{A}}_\delta \rangle$, i.e., some linear combination of the rows of $\hat{\mathbf{A}}_\delta$ gives the vector $\mathbf{1} = (1, 0, \dots, 0)$. A span program computes a Boolean function $\hat{f}$ if it accepts exactly those inputs δ where $\hat{f}(\delta) = 1$.

A span program is called monotone if the labels of the rows are only the positive literals $\{p_1, \dots, p_n\}$. Otherwise, it is non-monotone.

We first give the notion of a non-monotone access structure with evaluating map γ in Definition 2.5 by using inner-products of attribute vectors.

Definition 2.6 (Inner Products of Attribute Vectors and Access Structures [76]). $\mathcal{U}_i$ ($i = 1, \dots, d$ and $\mathcal{U}_i \subset \{0, 1\}^*$) is a sub-universe, a set of attributes, each of which is expressed by a pair of sub-universe id and n_i -dimensional vector, i.e., $(i, \mathbf{v})$, where $i \in [d]$ and $\mathbf{v} \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\}$. We denote such structure as $\vec{n} := (d; n_1, \dots, n_d)$.

We define such an attribute to be a variable p of a span program $(\hat{\mathbf{A}}, \hat{\rho})$, i.e., $p := (i, \mathbf{x})$. An access structure $\mathbb{A}$ is a span program $(\hat{\mathbf{A}}, \hat{\rho})$ along with variables $p := (i, \mathbf{x})$, $p' := (i', \mathbf{x}')$, ..., i.e., $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ such that $\hat{\rho} : [\hat{a}] \rightarrow \{(i, \mathbf{x}), (i', \mathbf{x}'), \dots, \neg(i, \mathbf{x}), \neg(i', \mathbf{x}'), \dots\}$.

Let Γ be a set of attributes, i.e., $\Gamma := \{(i, \mathbf{v}_i) | \mathbf{v}_i \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\}, 1 \leq i \leq d\}$, where $1 \leq i \leq d$ means that i is an element of some subset of $[d]$.

When Γ is given the access structure $\mathbb{A}$, map $\hat{\gamma} : [\hat{a}] \rightarrow \{0, 1\}$ for span program $(\hat{\mathbf{A}}, \hat{\rho})$ is defined as follows: For all $j \in [\hat{a}]$, set $\hat{\gamma}(j) = 1$ if $[\hat{\rho}(j) = (i, \mathbf{x}_j)] \wedge [(i, \mathbf{v}_i) \in \Gamma] \wedge [\mathbf{x}_j \cdot \mathbf{v}_i = 0]$ or $[\hat{\rho}(j) = \neg(i, \mathbf{x}_j)] \wedge [(i, \mathbf{v}_i) \in \Gamma] \wedge [\mathbf{x}_j \cdot \mathbf{v}_i \neq 0]$. Set $\hat{\gamma}(j) = 0$

otherwise.

Access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ accepts Γ iff $\mathbf{1} \in \text{span}\langle (\hat{\mathbf{A}}^{[j]})_{\hat{\rho}(j)=1} \rangle$.

We use the following secret-sharing scheme for a non-monotone access structure or span program.

Definition 2.7. A secret-sharing scheme for access structure $\mathbb{A}$ is a Linear Secret-Sharing Scheme (LSSS) in $\mathbb{Z}_q$ and is represented by $(\hat{\mathbf{A}}, \hat{\rho})$ if it consists of two efficient algorithms:

Lin.Share _{$(\hat{\mathbf{A}}, \hat{\rho})$} : Let $\hat{\mathbf{A}}$ be $\hat{a} \times \hat{b}$ share-generating matrix. Let $\mathbf{w} := (w_1, \dots, w_{\hat{b}}) \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{\hat{b}}$. Then, $s_0 := \mathbf{1} \cdot \mathbf{w}^\top$ is the secret to be shared, and $\mathbf{s}^\top := (s_1, \dots, s_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}^\top$ is the vector of $\hat{a}$ shares of the secret s_0 and the share s_j belongs to $\hat{\rho}(j)$.

Lin.Recon _{$(\hat{\mathbf{A}}, \hat{\rho})$} : If the span program $(\hat{\mathbf{A}}, \hat{\rho})$ accept δ , or access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ accepts Γ , i.e., $\mathbf{1} \in \text{span}\langle (\hat{\mathbf{A}}^{[j]})_{\hat{\rho}(j)=1} \rangle$ with $\hat{\gamma} : [\hat{a}] \rightarrow \{0, 1\}$, then there exist constants $\{\alpha_j \in \mathbb{Z}_q \mid j \in \Pi\}$ such that $\Pi \subseteq \{j \in [\hat{a}] \mid \hat{\gamma}(j) = 1\}$ and $\sum_{j \in \Pi} \alpha_j s_j = s_0$. Furthermore, these constants $\{\alpha_j\}$ can be computed in time polynomial in the size of matrix $\hat{\mathbf{A}}$.

In a KP-FE scheme supporting non-monotone access structure, a functionality $\hat{F}$ is defined over a key space and an index space using sets of non-monotone access structures and attribute vector tuples respectively (see Definition 2.6). The key space $\mathcal{K}$ corresponds to all non-monotone access structures $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$, while the index space $\mathcal{I}$ corresponds to all attribute vector sets Γ . Here,

$$\hat{F}(\mathbb{A}, (\Gamma, \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho}) \text{ accepts } \Gamma \\ \perp & \text{otherwise.} \end{cases}$$

In a CP-FE scheme supporting non-monotone access structure, a functionality $\hat{F}$ is defined over a key space and an index space using sets of attribute vector tuples and non-monotone access structures, respectively (see Definition 2.6). The

key space $\mathcal{K}$ corresponds to all attribute vector sets Γ , while the index space $\mathcal{I}$ corresponds to all non-monotone access structures $\mathbb{A} := (\hat{\mathbb{A}}, \hat{\rho})$. Here,

$$\hat{F}(\Gamma, (\mathbb{A}, \widehat{\text{msg}})) := \begin{cases} \widehat{\text{msg}} & \text{if } \mathbb{A} := (\hat{\mathbb{A}}, \hat{\rho}) \text{ accepts } \Gamma \\ \perp & \text{otherwise.} \end{cases}$$

2.3 Revocable Identity-Based Encryption

We recall the definition of IBE to support key revocation. In comparison with FE, we require some additional algorithms to facilitate key update and key revocation. We first recall the definition of RIBE and its security from [19] and then define an appropriate security model for our constructions.

Definition 2.8. *An Identity-Based Encryption with efficient revocation or simply Revocable IBE (RIBE) scheme has seven PPT algorithms Setup, PriKeyGen, KeyUpd, DecKeyGen, Enc, Dec, and KeyRev with associated message space $\mathcal{M}$, identity space $\mathcal{I}$ and time space $\mathcal{T}$. We assume that the size of $\mathcal{T}$ is polynomial in the security parameter. Each algorithm is run by either one of three types of parties—key authority, sender or receiver. Key authority maintains a revocation list RL and state ST. In what follows, an algorithm is called stateful if it updates RL or ST. We treat time as discrete as opposed to continuous.*

- **Setup**($1^\lambda, N_{\max}$) takes as input a security parameter λ and a maximal number of users $N_{\max}$. It outputs global public key GPK, a master secret key MSK, a revocation list RL (initially empty), and a state ST. (This is run by the key authority.)
- **PriKeyGen**(GPK, MSK, id, ST) takes as input the global public key GPK, the master secret key MSK, an identity $\text{id} \in \mathcal{I}$, and the state ST. It outputs a private key SK_{id} and an updated state ST. (This is stateful and run by the key authority.)
- **KeyUpd**(GPK, MSK, t, RL, ST) takes as input the global public key GPK, the

master secret key MSK , a key update time $t \in \mathcal{T}$, the revocation list RL , and the state ST . It outputs a key update KU_t . (This is run by the key authority.)

- $\text{DecKeyGen}(\text{SK}_{\text{id}}, \text{KU}_t)$ takes as input a private key SK_{id} and key update KU_t . It outputs a decryption key $\text{DK}_{\text{id},t}$ or a special symbol $\perp$ indicating that id was revoked. (This is deterministic and run by the receiver.)
- $\text{Enc}(\text{GPK}, \text{id}, t, \widehat{\text{msg}})$ takes as input the global public key GPK , an identity $\text{id} \in \mathcal{I}$, an encryption time $t \in \mathcal{T}$, and a message $\widehat{\text{msg}} \in \mathcal{M}$. It outputs a ciphertext $\text{CT}_{\text{id},t}$. (This is run by the sender.)
- $\text{Dec}(\text{GPK}, \text{DK}_{\text{id},t}, \text{CT}_{\text{id},t})$ takes as input the global public key GPK , a decryption key $\text{DK}_{\text{id},t}$, and a ciphertext $\text{CT}_{\text{id},t}$. It outputs a message $\widehat{\text{msg}} \in \mathcal{M}$. (This is deterministic and run by the receiver.)
- $\text{KeyRev}(\text{id}, t, \text{RL}, \text{ST})$ takes as input an identity to be revoked $\text{id} \in \mathcal{I}$, a revocation time $t \in \mathcal{T}$, the revocation list RL , and the state ST . It outputs an updated revocation list RL . (This is stateful and run by the key authority.)

The consistency condition requires that for all $\lambda \in \mathbb{N}$ and polynomials (in λ) $N_{\max}$, all GPK and MSK output by setup algorithm Setup , all $\widehat{\text{msg}} \in \mathcal{M}$, $\text{id} \in \mathcal{I}$, $t \in \mathcal{T}$ and all possible valid states ST and revocation lists RL , if identity id was not revoked before or, at time t then the following experiment returns 1 except for a negligible probability:

$(\text{SK}_{\text{id}}, \text{ST}) \leftarrow_{\text{R}} \text{PriKeyGen}(\text{GPK}, \text{MSK}, \text{id}, \text{ST});$
 $\text{KU}_t \leftarrow_{\text{R}} \text{KeyUpd}(\text{GPK}, \text{MSK}, t, \text{RL}, \text{ST})$
 $\text{DK}_{\text{id},t} \leftarrow \text{DecKeyGen}(\text{SK}_{\text{id}}, \text{KU}_t); \text{CT}_{\text{id},t} \leftarrow_{\text{R}} \text{Enc}(\text{GPK}, \text{id}, t, \widehat{\text{msg}})$
 If $\text{Dec}(\text{GPK}, \text{DK}_{\text{id},t}, \text{CT}_{\text{id},t}) = \widehat{\text{msg}}$ then return 1 else return 0.

Boldyreva et al. formalized and defined the selective-ID security for RIBE. Their definition captures not only the standard notion of selective-ID security but also takes into account key revocation. The following definition extends the security property expressed in [19] to the adaptive-ID and anonymous setting.

- **Setup:** It is run to generate public parameters GPK , a master secret key MSK , a revocation list RL (initially empty), and a state ST . Then GPK is given to $\mathcal{A}$.
- **Query:** $\mathcal{A}$ may adaptively make a polynomial number of queries of the following oracles (the oracles share state):
 - The private key generation oracle $\text{PriKeyGen}(\cdot)$ takes as input an identity id and runs $\text{PriKeyGen}(\text{GPK}, \text{MSK}, \text{id}, \text{ST})$ to return a private key SK_{id} .
 - The key update generation oracle $\text{KeyUpd}(\cdot)$ takes as input time t and runs $\text{KeyUpd}(\text{GPK}, \text{MSK}, t, \text{RL}, \text{ST})$ to return a key update KU_t .
 - The revocation oracle $\text{KeyRev}(\cdot)$ takes as input an identity id and time t and runs $\text{KeyRev}(\text{id}, t, \text{RL}, \text{ST})$ to update RL .
- **Challenge:** $\mathcal{A}$ outputs the two challenge pair $(\text{id}_{(0)}^*, t_{(0)}^*, \widehat{\text{msg}}_{(0)}^*), (\text{id}_{(1)}^*, t_{(1)}^*, \widehat{\text{msg}}_{(1)}^*) \in \mathcal{I} \times \mathcal{T} \times \mathcal{M}$. A random bit β is chosen. $\mathcal{A}$ is given $\text{Enc}(\text{GPK}, \text{id}_{(\beta)}^*, t_{(\beta)}^*, \widehat{\text{msg}}_{(\beta)}^*)$.
- **Guess:** The adversary may continue to make queries of the oracles as in Query phase and outputs a bit β' , and succeeds if $\beta' = \beta$.

The following restrictions must always hold:

1. $\text{KeyUpd}(\cdot)$ and $\text{KeyRev}(\cdot, \cdot)$ can be queried on time which is greater than or equal to the time of all previous queries, i.e., the adversary is allowed to query only in non-decreasing order of time. Also, the oracle $\text{KeyRev}(\cdot, \cdot)$ cannot be queried at time t if $\text{KeyUpd}(\cdot)$ was queried on t .
2. For $\beta = 0, 1$, if $\text{PriKeyGen}(\cdot)$ was queried on identity $\text{id}_{(\beta)}$ then $\text{KeyRev}(\cdot, \cdot)$ must be queried on $(\text{id}_{(\beta)}^*, t)$ for some $t \leq t_{(\beta)}^*$, i.e., identity $\text{id}_{(\beta)}^*$ must be in RL when $\text{KeyUpd}(\cdot)$ is queried at time $t_{(\beta)}^*$.

For $\beta = 0, 1$ let W_β be the event that the adversary outputs 1 in Experiment β

and define

$$\text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) := |\Pr[W_0] - \Pr[W_1]|.$$

Definition 2.9. An RIBE scheme is *adaptive-ID secure and anonymous* if for all PPT adversaries $\mathcal{A}$ the function $\text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda)$ is negligible.

Remark: The security notion of *non-anonymous* RIBE is defined as above with restriction that $\text{id}_{(0)}^* = \text{id}_{(1)}^*$ and $\text{t}_{(0)}^* = \text{t}_{(1)}^*$. On the other hand, if the adversary $\mathcal{A}$ outputs $(\text{id}_{(0)}^*, \text{t}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \text{t}_{(1)}^*)$ before the **Setup** phase, it is *selective-ID* security. We note that $\text{t}_{(0)}^*, \text{t}_{(1)}^*$ can be guessed with high probability by the security reduction algorithm, they do not need to be specified by the adversary.

2.4 Notations

$r \leftarrow_{\text{R}} \Pi$	r is chosen uniformly at random from the set Π
$r \leftarrow_{\text{R}} \Psi$	r is chosen at random from the distribution Ψ
$[\ell]$	the set $\{1, \dots, \ell\}$
$[\ell, \ell']$	the set $\{\ell, \ell + 1, \dots, \ell'\}$
$S(\mathbf{v}_1, \dots, \mathbf{v}_\ell)$	the space spanned by $\{\mathbf{v}_1, \dots, \mathbf{v}_\ell\}$
$S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_\ell)$	the orthogonal space of $S(\mathbf{v}_1, \dots, \mathbf{v}_\ell)$, i.e., the space spanned by all $\mathbf{x}$ where $\mathbf{x} \cdot \mathbf{v}_i = 0$ for $i \in [\ell]$
$B(S)$	a basis of the space S
$B^\perp(S)$	a basis of the space $S^\perp$
$\dim(S)$	the dimension of the space S
$\mathbf{A}^\top$	the transpose of the matrix $\mathbf{A}$
$\mathbf{A}^{[i]}$	the i -th row of the matrix $\mathbf{A}$
$S(\mathbf{A})$	the space spanned by the row vectors of the matrix $\mathbf{A}$
$S(\mathbf{A}, \mathbf{x})$	the affine space $\{\mathbf{a}\mathbf{A} + \mathbf{x} : \mathbf{a} \in \mathbb{Z}_q^\ell\}$, where $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$

Table 2.1: Notations.

In the remainder of the thesis, if not explicitly specified, we assume that all vectors are row vectors in $\mathbb{Z}_q^n$ for some integer n and prime q and spaces are Euclidean spaces spanned by row vectors. Particularly, in the lattice setting, we assume that all vectors are column vectors. Table 2.1 summarizes some notations used in the remainder of the thesis. Moreover, we may use subscript to indicate the set type of a scheme. For example, we use $\mathcal{K}_{\text{HIPE}}$ (resp. $\mathcal{I}_{\text{HIPE}}$) to denote the key space $\mathcal{K}$ (resp. index space $\mathcal{I}$) with regards to HIPE.

Chapter

3

Background on Pairings

Contents

3.1 Dual Pairing Vector Spaces	29
3.2 Complexity Assumptions	31

In this chapter, we present some background and basic concepts related to groups with efficiently computable bilinear maps and define the required complexity assumptions.

3.1 Dual Pairing Vector Spaces

All of our pairing-based constructions rely on *Dual Pairing Vector Spaces* (DPVS) proposed by Okamoto and Takashima [74, 75]. In this thesis, we concentrate on the asymmetric version [76]. We only briefly describe how to generate random dual orthonormal bases. See [74, 75, 76] for a full definition of DPVS.

Definition 3.1 (Asymmetric Bilinear Pairing Groups). *Asymmetric bilinear pairing groups $(q, G_1, G_2, G_T, g_1, g_2, e)$ are a tuple of a prime q , cyclic (multiplicative) groups G_1, G_2 and G_T of order q , $g_1 \neq 1 \in G_1$, $g_2 \neq 1 \in G_2$, and a polynomial-time computable nondegenerate bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$ i.e., $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ and $e(g_1, g_2) \neq 1$.*

In addition to referring to individual elements of G_1 or G_2 , we will also consider “vectors” of group elements. For $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{Z}_q^n$ and $g_\beta \in G_\beta$, we write $g_\beta^{\mathbf{v}}$ to denote a n -tuple of elements of G_β for $\beta = 1, 2$:

$$g_\beta^{\mathbf{v}} := (g_\beta^{v_1}, \dots, g_\beta^{v_n}).$$

For any $a \in \mathbb{Z}_q$ and $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_q^n$, we have:

$$g_\beta^{a\mathbf{v}} := (g_\beta^{av_1}, \dots, g_\beta^{av_n}), \quad g_\beta^{\mathbf{v}+\mathbf{w}} := (g_\beta^{v_1+w_1}, \dots, g_\beta^{v_n+w_n}).$$

Then we define

$$e(g_1^{\mathbf{v}}, g_2^{\mathbf{w}}) := \prod_{i=1}^n e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\mathbf{v} \cdot \mathbf{w}}.$$

Here, the dot product is taken modulo q .

Dual Pairing Vector Spaces For a fixed (constant) dimension n , we will choose two random bases $\mathbb{B} := (\mathbf{b}_1, \dots, \mathbf{b}_n)$ and $\mathbb{B}^* := (\mathbf{b}_1^*, \dots, \mathbf{b}_n^*)$ of $\mathbb{Z}_q^n$, subject to the constraint that they are “dual orthonormal”, meaning that

$$\mathbf{b}_j \cdot \mathbf{b}_k^* = 0 \pmod{q}$$

whenever $j \neq k$, and

$$\mathbf{b}_j \cdot \mathbf{b}_j^* = \psi \pmod{q}$$

for all j , where ψ is a random element of $\mathbb{Z}_q$. We denote such algorithm as $\text{Dual}(\mathbb{Z}_q^n)$.

Then for generators $g_1 \in G_1$ and $g_2 \in G_2$, we have

$$e(g_1^{\mathbf{b}_j}, g_2^{\mathbf{b}_k^*}) = 1$$

whenever $j \neq k$, where 1 here denotes the identity element in G_T .

More generally, we can sample multiple tuple of “dual orthonormal” bases. Namely, for fixed (constant) dimension $n_1, \dots, n_d$, we will choose d tuples of two random bases $\mathbb{B}_i := (\mathbf{b}_{1,i}, \dots, \mathbf{b}_{n_i,i})$ and $\mathbb{B}_i^* := (\mathbf{b}_{1,i}^*, \dots, \mathbf{b}_{n_i,i}^*)$ of $\mathbb{Z}_q^{n_i}$, subject to the constraint that they are “dual orthonormal”, meaning that

$$\mathbf{b}_{j,i} \cdot \mathbf{b}_{k,i}^* = 0 \pmod{q}$$

whenever $j \neq k$, and

$$\mathbf{b}_{j,i} \cdot \mathbf{b}_{j,i}^* = \psi \pmod{q}$$

for all j and i , where ψ is a random element of $\mathbb{Z}_q$. We denote such an algorithm as $\text{Dual}(\mathbb{Z}_q^{n_1}, \dots, \mathbb{Z}_q^{n_d})$.

The following lemma is from [61], which is used throughout this thesis.

Lemma 3.1. *For any fixed positive integers $m \leq n$, any fixed invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times m}$, pick $(\mathbb{B}, \mathbb{B}^*) \leftarrow_{\mathbf{R}} \text{Dual}(\mathbb{Z}_q^n)$, set*

$$\begin{aligned} \mathbb{F} &:= (\mathbf{d}_1, \dots, \mathbf{d}_{n-m}, \mathbf{d}_{n-m+1}\mathbf{A}, \dots, \mathbf{d}_n\mathbf{A}), \\ \mathbb{F}^* &:= (\mathbf{d}_1^*, \dots, \mathbf{d}_{n-m}^*, \mathbf{d}_{n-m+1}^*(\mathbf{A}^{-1})^\top, \dots, \mathbf{d}_n^*(\mathbf{A}^{-1})^\top). \end{aligned}$$

Then $(\mathbb{F}, \mathbb{F}^)$ is also distributed as a random sample from $\text{Dual}(\mathbb{Z}_q^n)$. In particular, the distribution of $(\mathbb{F}, \mathbb{F}^*)$ is independent of $\mathbf{A}$.*

3.2 Complexity Assumptions

SXDH Assumption The *Symmetric External Diffie-Hellman* (SXDH) assumption was formally defined by Ballard et al. [14], evidence for the validity of which were presented in the works of Verheul [91] and Galbraith and Rotger [48]. Informally, the SXDH assumption states that the *Decisional Diffie-Hellman* (DDH) assumption holds in both G_1 and G_2 . To define the SXDH assumption, we first define DDH problems in G_1 and G_2 .

Definition 3.2 (DDH1: Decisional Diffie-Hellman Assumption in G_1). *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned} \mathbb{G} &:= (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathbf{R}} \mathcal{G}, \\ a, b, c &\leftarrow_{\mathbf{R}} \mathbb{Z}_q, \\ D &:= (\mathbb{G}; g_1, g_2, g_1^a, g_1^b). \end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DDH1}}(\lambda) := |\Pr[\mathcal{A}(D, g_1^{ab})] - \Pr[\mathcal{A}(D, g_1^{ab+c})]|$$

is negligible in the security parameter λ .

The dual of above assumption is DDH assumption in G_2 (denoted as DDH2), which is identical to Definitions 3.4 with the roles of G_1 and G_2 reversed. We say that:

Definition 3.3. *The SXDH assumption holds if DDH problems are intractable in both G_1 and G_2 .*

DLIN Assumption We define the *Decisional Linear* (DLIN) assumption [26] in symmetric bilinear pairing groups (namely $G_1 = G_2$).

Definition 3.4 (DLIN: Decisional Linear Assumption). *Given a group generator $\mathcal{G}$, we define the following distribution:*

$$\begin{aligned} \mathbb{G} &:= (q, G, G_T, g, e) \leftarrow_{\mathbf{R}} \mathcal{G}, \\ a_1, a_2, b_1, b_2, c &\leftarrow_{\mathbf{R}} \mathbb{Z}_q, \\ D &:= (\mathbb{G}; g, g^{a_1}, g^{a_2}, g^{a_1 b_1}, g^{a_2 b_2}). \end{aligned}$$

We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DLIN}}(\lambda) := |\Pr[\mathcal{A}(D, g^{b_1+b_2})] - \Pr[\mathcal{A}(D, g^{b_1+b_2+c})]|$$

is negligible in the security parameter λ .

Chapter 4

Background on Lattices

Contents

4.1	Integer Lattices	33
4.2	The Gram-Schmidt Norm and Trapdoors for Lattices	34
4.3	Discrete Gaussian Distribution	35
4.4	Sampling Algorithms	35
4.5	The LWE Hardness Assumption	36
4.6	Encoding Identities as Matrices	37

In this chapter, we describe the required concepts for our lattice-based RIBE scheme. We introduce the definitions of integer lattices, the Gram-Schmidt norm, discrete Gaussians and also sampling algorithms to generate trapdoors for lattices. Subsequently, the core hard problem—LWE for the security proof our RIBE scheme is given. We assume that all vectors are column vectors.

4.1 Integer Lattices

Let $\mathbf{B} := [\mathbf{b}_1 | \dots | \mathbf{b}_m] \in \mathbb{R}^{m \times m}$ be an $m \times m$ matrix whose columns are linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbb{R}^m$. The m -dimensional full-rank lattice Λ

generated by $\mathbf{B}$ is the set,

$$\Lambda := \mathcal{L}(\mathbf{B}) := \left\{ \mathbf{y} \in \mathbb{R}^m \text{ s.t. } \exists \mathbf{s} \in \mathbb{Z}^m, \mathbf{y} = \mathbf{B}\mathbf{s} = \sum_{i=1}^m \mathbf{s}_i \mathbf{b}_i \right\}$$

Here, we are interested in integer lattices, i.e., when $\mathcal{L}$ is a subset of $\mathbb{Z}^m$.

Definition 4.1. For a prime q , $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, define:

$$\begin{aligned} \Lambda_q^\perp(\mathbf{A}) &:= \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{0} \pmod{q} \} \\ \Lambda_q^{\mathbf{u}}(\mathbf{A}) &:= \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{u} \pmod{q} \} \end{aligned}$$

4.2 The Gram-Schmidt Norm and Trapdoors for Lattices

Let S be a set of vectors $S := \{\mathbf{s}_1, \dots, \mathbf{s}_k\}$ in $\mathbb{R}^m$, we use $\|S\|$ to denote the Euclidean norm of the longest vector in S , i.e., $\|S\| := \max_i \sqrt{s_{i,1}^2 + \dots + s_{i,m}^2}$ for $1 \leq i \leq k$, where $\mathbf{s}_i := (s_{i,1}, \dots, s_{i,m})$. We use $\tilde{S} := \{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_k\} \subset \mathbb{R}^m$ to denote the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, \dots, \mathbf{s}_k$ in that order. We refer to $\|\tilde{S}\|$ as the Gram-Schmidt norm of S .

The problem of generating a random lattice $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a full short basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$ (satisfying $\|\tilde{\mathbf{T}}_\mathbf{A}\| \leq \mathcal{O}(\sqrt{n \log q})$ $\|\mathbf{T}_\mathbf{A}\| \leq \mathcal{O}(n \log q)$) has been previously investigated by [8, 10]. Here we use a better result with tighter parameters which was recently discovered by Micciancio and Peikert [69].

Theorem 4.1. Let $n \geq 1$, $q \geq 2$ be integers and $m = \lceil 2n \log q \rceil$. There is an efficient PPT algorithm $\text{TrapGen}(q, n)$ that outputs a pair $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{T}_\mathbf{A} \in \mathbb{Z}_q^{m \times m})$ such that $\mathbf{A}$ is statistically close to a uniform matrix in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{T}_\mathbf{A}$ is a basis for $\Lambda_q^\perp(\mathbf{A})$ satisfying $\|\tilde{\mathbf{T}}_\mathbf{A}\| \leq \mathcal{O}(\sqrt{n \log q})$ and $\|\mathbf{T}_\mathbf{A}\| \leq \mathcal{O}(n \log q)$ with all but negligible probability in n .

4.3 Discrete Gaussian Distribution

Let Λ be an m -dimensional lattice. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, define:

- $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) := \exp\left(-\pi \frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$: a Gaussian-shaped function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ ,
- $\rho_{\sigma,\mathbf{c}}(\Lambda) := \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$: the (always converging) sum of $\rho_{\sigma,\mathbf{c}}$ over Λ ,
- $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$: the discrete Gaussian distribution over Λ with parameters σ and center $\mathbf{c}$,

$$\forall \mathbf{y} \in \Lambda, \quad \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) := \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}.$$

For notational convenience, we abbreviate $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ as ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$.

The following lemmas from [52] are essential for our security proof.

Lemma 4.1. *There is an efficient PPT algorithm **SampleGaussian** that, given a basis $\mathbf{B}$ of an m -dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, a parameter $\sigma \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$, and a center $\mathbf{c} \in \mathbb{R}^m$, outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$.*

Let $\mathbf{B}_z$ be the standard basis for $\mathbb{Z}^m$, we use the **SampleGaussian**($\mathbf{B}_z, \sigma, 0$) algorithm to sample from distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma}$.

Lemma 4.2. *Let n and q be positive integers with q prime, and let $m \geq 2n \log q$. Then for all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and for any $\sigma \geq \omega(\sqrt{\log m})$, the distribution of the syndrome $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$, where $\mathbf{e}$ is from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$.*

4.4 Sampling Algorithms

The following **SampleLeft** [39, 2] and **SampleRight** [2] algorithms will be used to sample short vectors in our construction and in the simulation, respectively. Let

$\mathbf{A}$ and $\mathbf{C}$ be matrices in $\mathbb{Z}_q^{n \times m}$ and let $\mathbf{R}$ be a matrix in $\{-1, 1\}^{m \times m}$. By using either a trapdoor for $\Lambda_q^\perp(\mathbf{A})$ or a trapdoor for $\Lambda_q^\perp(\mathbf{C})$, we can sample a short vector $\mathbf{e}$ in $\Lambda_q^\perp(\mathbf{F})$ for some $\mathbf{u}$ in $\mathbb{Z}_q^n$, where $\mathbf{F} := (\mathbf{A} | \mathbf{A}\mathbf{R} + \mathbf{C}) \in \mathbb{Z}_q^{n \times 2m}$. With appropriate parameters, the distribution of $\mathbf{e}$ produced by these two algorithms are statistically indistinguishable from $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}), \sigma}$.

Theorem 4.2. *Let $q > 2$ and $m > n$. Then there is an efficient PPT algorithm `SampleLeft` that takes as input a rank n matrix $\mathbf{A}$ in $\mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{M}$ in $\mathbb{Z}_q^{n \times m_1}$, a “short” basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a gaussian parameter $\sigma > \|\widetilde{\mathbf{T}}_\mathbf{A}\| \cdot \omega(\sqrt{\log(m + m_1)})$. It outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}_1), \sigma}$ where $\mathbf{F}_1 := (\mathbf{A} | \mathbf{M})$. In particular, $\mathbf{e} \in \Lambda_q^\perp(\mathbf{F}_1)$.*

Theorem 4.3. *Let $q > 2$ and $m > n$. There is an efficient PPT algorithm `SampleRight` that takes as input matrices $\mathbf{A}, \mathbf{C}$ in $\mathbb{Z}_q^{n \times m}$ where $\mathbf{C}$ is rank n , a uniform random matrix $\mathbf{R} \in \{-1, 1\}^{m \times m}$, a basis $\mathbf{T}_\mathbf{C}$ of $\Lambda_q^\perp(\mathbf{C})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a gaussian parameter $\sigma > \|\widetilde{\mathbf{T}}_\mathbf{C}\| \cdot \sqrt{m} \omega(\log(m))$. It outputs a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}_2), \sigma}$ where $\mathbf{F}_2 := (\mathbf{A} | \mathbf{A}\mathbf{R} + \mathbf{C})$. In particular, $\mathbf{e} \in \Lambda_q^\perp(\mathbf{F}_2)$.*

We will also need the following lemma, a generalization of the left over hash lemma due to Dodis et al. [44], in our proof.

Lemma 4.3. *Suppose that $m > (n + 1) \log q + \omega(\log n)$ and that q is prime. Let $\mathbf{A}, \mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n \times m}$ and let $\mathbf{R}$ be an $m \times m$ matrix chosen uniformly in $\{1, -1\}^{m \times m} \bmod q$. Then, for all vectors $\mathbf{w}$ in $\mathbb{Z}_q^m$, the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{R}^\top \mathbf{w})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$.*

4.5 The LWE Hardness Assumption

The security of our construction relies on the *Learning With Errors* (LWE) problem defined by Regev [84].

Definition 4.2. Consider a prime q , a positive integer n , and a distribution χ over $\mathbb{Z}_q$, all public. An $(\mathbb{Z}_q, n, \chi)$ -LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being, either, a noisy pseudo-random sampler $\mathcal{O}_s$ carrying some constant random secret key $\mathbf{s} \in \mathbb{Z}_q^n$, or, a truly random sampler $\mathcal{O}_\$,$ whose behaviors are respectively as follows:

- $\mathcal{O}_s$: outputs samples of the form $(\mathbf{u}_i, v_i) = (\mathbf{u}_i, \mathbf{u}_i^\top \mathbf{s} + x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where, $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed persistent value invariant across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample from χ , and $\mathbf{u}_i$ is uniform in $\mathbb{Z}_q^n$.
- $\mathcal{O}_\$$: outputs truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The $(\mathbb{Z}_q, n, \chi)$ -LWE problem allows repeated queries to the challenge oracle $\mathcal{O}$. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$ -LWE problem if $|\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]|$ is non-negligible for a random $\mathbf{s} \in \mathbb{Z}_q^n$.

Regev [84] and Peikert [80] showed that for some noise distribution χ , denoted by $\bar{\Psi}_\alpha$, the LWE problem is at least as hard as the worst-case SIVP and GapSVP under a quantum reduction if the parameters are appropriately set.

Definition 4.3. Consider a real parameter $\alpha = \alpha(n) \in (0, 1)$ and a prime q . Let $\mathbb{T} := \mathbb{R}/\mathbb{Z}$ be the group of reals $[0, 1)$ with addition modulo 1. Let Ψ_α be the distribution over $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ then reduced modulo 1. Let $\lfloor x \rfloor := \lfloor x + \frac{1}{2} \rfloor$ be the nearest integer to the real $x \in \mathbb{R}$. We then define $\bar{\Psi}_\alpha$ as the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor xX \rfloor \bmod q$ where the random variable $X \in \mathbb{T}$ has distribution Ψ_α .

4.6 Encoding Identities as Matrices

In our construction and security proof, we require an injective encoding function $H : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^{n \times n}$ to map identities in $\mathbb{Z}_q^n$ to matrices in $\mathbb{Z}_q^{n \times n}$. Concrete construction of such a function can be found in [2, 42].

Definition 4.4. *Let q be a prime and n a positive integer. We say that a function $H : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^{n \times n}$ is an encoding with Full-Rank Differences (FRD) if:*

1. *for all distinct $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, the matrix $H(\mathbf{u}) - H(\mathbf{v}) \in \mathbb{Z}_q^{n \times n}$ is full rank;*
2. *H is computable in polynomial time in $n \log q$.*

Chapter 5

Shorter IBE via Asymmetric Pairings

Contents

5.1	Motivation	40
5.2	Subspace Assumptions via SXDH	41
5.3	Statistical Indistinguishability Lemmas	45
5.4	Identity-Based Encryption	45
5.5	Extensions	55
5.6	Discussion	81

In this chapter, we show a shorter IBE scheme under the SXDH assumption via asymmetric pairings. We begin by presenting a framework—Subspace assumptions derived from the SXDH assumption and then derive statistical indistinguishability lemmas, which are our main tools for the forthcoming security proofs of encryption schemes in this chapter. Subsequently, we present our IBE scheme and also demonstrate how to instantiate IPE and KP-FE. In our schemes, all parameters are shorter than those of previous constructions based on DLIN assumption [61, 76, 77].

5.1 Motivation

Short IBE It is desirable that an IBE scheme be as efficient as possible, if it were to have any impact on practical applications.¹ Ideally, we would like to have constant-size global public key, secret keys, and ciphertexts. Moreover, the scheme should ideally achieve full security, namely to be resilient even against an adversary that adaptively selects an identity to attack based on previous secret keys. We note that the IBE scheme of [92] is fully secure, however, its global public key has $\mathcal{O}(\lambda)$ group elements, where λ denotes the security parameter (which is very large, e.g., 128, 256). The first fully secure efficient IBE with constant-size global public key and ciphertexts under standard assumptions was obtained by Waters [93] in 2009; this scheme relied on the DBDH and DLIN assumptions. Since then, Lewko and Waters [62] and Lewko [61] gave additional fully secure efficient IBE schemes that achieve incomparable guarantees. Prior to these works, all known IBE schemes (in the standard model) were either selectively secure [37, 20, 39, 2], or require long parameters [21, 92, 39, 2], or were based on less standard assumptions that depended on the query complexity of the adversary [49]. From a practical stand-point, Waters' fully secure IBE [93] is still not very efficient as it has relatively large ciphertexts and secret keys, i.e., eleven and nine group elements,² respectively. Lewko's scheme [61] improved on both of these parameters at the cost of larger global public key and master secret key.

¹Generally, efficiency of pairing-based FE system is estimated in terms of global public key size, secret key size, ciphertext size, and the number of pairing operation for decryption. Here, the sizes are in terms of group elements (roughly speaking, the less, the better). Moreover, there are two computational operations for pairing-based system—pairing and scalar multiplication. However, compared with pairing operation, the computational cost for scalar multiplication is very low, thus we may care more about the number of pairing operation for decryption (generally, encryption do not involve pairing operation). Actually, the number of scalar multiplication for encryption is essentially the size of ciphertext while decryption only involves a very few number of scalar multiplication.

²Here, we do not separately consider group elements from target groups of pairings, although a ciphertext typically has a group element that is from an associated target group.

Shorter IBE? In his work, Waters also suggested obtaining even more efficient IBE schemes by turning to asymmetric bilinear groups:

Using the SXDH assumption we might hope to shave off three group elements from both ciphertexts and private keys.

In fact, improving the efficiency of a scheme using asymmetric pairings was first observed by Boneh, Boyen and Shacham [26]. At a fixed security level, group elements in the asymmetric setting are smaller and pairings can be computed more efficiently [47]. (Estimated bit sizes of group elements for bilinear group generators are given in Appendix A.) The SXDH assumption was formally defined by Ballard et al. [14] in their construction of a searchable encryption scheme, and has since been used in a number of different contexts, including secret-handshake schemes [11], anonymous IBE [45], continual leakage-resilience [34], and most notably, Groth-Sahai proofs [54].

5.2 Subspace Assumptions via SXDH

We now present Subspace assumptions derived from the SXDH assumption, which are analogues of the DLIN-based Subspace assumptions given in [61, 76].

Decisional Subspace Assumption We first consider the case of single bases:

Definition 5.1 (DS1: Decisional Subspace Assumption in G_1). *Given a group generator $\mathcal{G}(\cdot)$, define the following distribution:*

$$\begin{aligned} \mathbb{G} &:= (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathcal{R}} \mathcal{G}(1^\lambda), \\ (\mathbb{B}, \mathbb{B}^*) &\leftarrow_{\mathcal{R}} \text{Dual}(\mathbb{Z}_q^N), \tau_1, \tau_2, \mu_1, \mu_2 \leftarrow_{\mathcal{R}} \mathbb{Z}_q, \\ U_1 &:= g_2^{\mu_1 \mathbf{b}_1^* + \mu_2 \mathbf{b}_{K+1}^*}, \dots, U_K := g_2^{\mu_1 \mathbf{b}_K^* + \mu_2 \mathbf{b}_{2K}^*}, \\ V_1 &:= g_1^{\tau_1 \mathbf{b}_1}, \dots, V_K := g_1^{\tau_1 \mathbf{b}_K}, \\ W_1 &:= g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{K+1}}, \dots, W_K := g_1^{\tau_1 \mathbf{b}_K + \tau_2 \mathbf{b}_{2K}}, \\ D &:= (\mathbb{G}; g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_K^*}, g_2^{\mathbf{b}_{2K+1}^*}, \dots, g_2^{\mathbf{b}_N^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_N}, U_1, \dots, U_K, \mu_2), \end{aligned}$$

where K, N are fixed positive integers that satisfy $2K \leq N$. We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda) := |\Pr[\mathcal{A}(D, V_1, \dots, V_K) = 1] - \Pr[\mathcal{A}(D, W_1, \dots, W_K) = 1]|$$

is negligible in the security parameter λ .

Lemma 5.1. *If the DDH assumption in G_1 holds, then the Subspace assumption in G_1 stated in Definition 5.1 also holds. More precisely, for any adversary $\mathcal{A}$ against the Subspace assumption in G_1 , there exist probabilistic algorithms $\mathcal{B}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{\text{DDH1}}(\lambda).$$

Proof. We assume there exists a PPT algorithm $\mathcal{A}$ breaking the Subspace assumption with non-negligible advantage $\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda)$ (for some fixed positive integers K, N satisfying $N \geq 2K$). We create a PPT algorithm $\mathcal{B}$ which breaks the DDH assumption in G_1 with non-negligible advantage $\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda)$. $\mathcal{B}$ is given $g_1, g_2, g_1^a, g_1^b, T$, where T is either g_1^{ab} or T is a uniformly random element of G_1 .

$\mathcal{B}$ first samples random dual orthonormal bases, denoted by $\mathbf{f}_1, \dots, \mathbf{f}_N$ and $\mathbf{f}_1^*, \dots, \mathbf{f}_N^*$. From the definition, $\mathcal{B}$ chooses vectors $\mathbf{f}_1, \dots, \mathbf{f}_N, \mathbf{f}_1^*, \dots, \mathbf{f}_N^*$ randomly, subject to the constraints that $\mathbf{f}_i \cdot \mathbf{f}_j^* \equiv 0 \pmod{q}$ when $j \neq i$, and $\mathbf{f}_j \cdot \mathbf{f}_j^* \equiv \psi \pmod{q}$ for all j from 1 to N , where ψ is a random element of $\mathbb{Z}_q$. Then, $\mathcal{B}$ implicitly sets:

$$\mathbf{b}_1 := \mathbf{f}_1 + a\mathbf{f}_{K+1}, \dots, \mathbf{b}_K := \mathbf{f}_K + a\mathbf{f}_{2K},$$

$$\mathbf{b}_{K+1} := \mathbf{f}_{K+1}, \dots, \mathbf{b}_N := \mathbf{f}_N.$$

$\mathcal{B}$ also sets the dual basis as:

$$\mathbf{b}_1^* := \mathbf{f}_1^*, \dots, \mathbf{b}_K^* := \mathbf{f}_K^*,$$

$$\mathbf{b}_{K+1}^* := \mathbf{f}_{K+1}^* - a\mathbf{f}_1^*, \dots, \mathbf{b}_{2K}^* := \mathbf{f}_{2K}^* - a\mathbf{f}_K^*,$$

$$\mathbf{b}_{2K+1}^* := \mathbf{f}_{2K+1}^*, \dots, \mathbf{b}_N^* := \mathbf{f}_N^*.$$

We observe that under these definitions, $\mathbf{b}_j \cdot \mathbf{b}_k^* \equiv 0 \pmod{q}$ when $j \neq k$, and $\mathbf{b}_j \cdot \mathbf{b}_j^* \equiv \psi \pmod{q}$ for all j from 1 to N . We note that $\mathcal{B}$ can produce all of $g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_N}$ (given g_1, g_1^a) as well as $g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_K^*}$ and $g_2^{\mathbf{b}_{2K+1}^*}, \dots, g_2^{\mathbf{b}_N^*}$ (given g_2). However, $\mathcal{B}$ cannot produce $g_2^{\mathbf{b}_{K+1}^*}, \dots, g_2^{\mathbf{b}_{2K}^*}$ (these require knowledge of g_2^a). It is not difficult to check that $\mathbf{b}_1, \dots, \mathbf{b}_N$ and $\mathbf{b}_1^*, \dots, \mathbf{b}_N^*$ are properly distributed.

Now $\mathcal{B}$ creates $U_1, \dots, U_K$ by choosing random values $\mu'_1, \mu'_2 \in \mathbb{Z}_q$ and setting:

$$U_1 := g_2^{\mu'_1 \mathbf{b}_1^* + \mu'_2 \mathbf{f}_{K+1}^*} := g_2^{(\mu'_1 + a\mu'_2) \mathbf{b}_1^* + \mu'_2 \mathbf{b}_{K+1}^*}.$$

In other words, $\mathcal{B}$ has implicitly set $\mu_1 := \mu'_1 + a\mu'_2$ and $\mu_2 := \mu'_2$. We note that these values are uniformly random, and μ_2 is known to $\mathcal{B}$. $\mathcal{B}$ can then form $U_2, \dots, U_K$ as:

$$U_2 := g_2^{\mu'_1 \mathbf{b}_2^* + \mu'_2 \mathbf{f}_{K+2}^*}, \dots, U_K := g_2^{\mu'_1 \mathbf{b}_K^* + \mu'_2 \mathbf{f}_{2K}^*}.$$

$\mathcal{B}$ implicitly sets $\tau_1 := b, \tau_2 := c$ and computes:

$$T_1 := T^{\mathbf{f}_{K+1}} \cdot (g_1^b)^{\mathbf{f}_1}, \dots, T_K := T^{\mathbf{f}_{2K}} \cdot (g_1^b)^{\mathbf{f}_K}.$$

If $T = g_1^{ab}$, then these are distributed as $V_1, \dots, V_K$, since

$$T^{\mathbf{f}_{K+j}} \cdot (g_1^b)^{\mathbf{f}_j} = g_1^{\tau_1 \mathbf{b}_j}.$$

If $T = g_1^{ab+c}$, then these are distributed as $W_1, \dots, W_K$, since

$$T^{\mathbf{f}_{K+j}} \cdot (g_1^b)^{\mathbf{f}_j} = g_1^{\tau_1 \mathbf{b}_j + \tau_2 \mathbf{b}_{K+j}}.$$

$\mathcal{B}$ then gives

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_K^*}, g_2^{\mathbf{b}_{2K+1}^*}, \dots, g_2^{\mathbf{b}_N^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_N}, U_1, \dots, U_K, \mu_2)$$

to $\mathcal{A}$, along with $T_1, \dots, T_K$. $\mathcal{B}$ can then leverage $\mathcal{A}$'s advantage $\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda)$ in distinguishing between the distributions $(V_1, \dots, V_K)$ and $(W_1, \dots, W_K)$ to achieve an advantage $\text{Adv}_{\mathcal{B}}^{\text{DDH1}}(\lambda)$ in distinguishing $T = g_1^{ab}$ from $T = g_1^{ab+c}$, hence violating the DDH assumption in G_1 . $\square$

The dual of the Subspace assumption in G_1 is Subspace assumption in G_2 (denoted as DS2), which is identical to Definitions 5.1 with the roles of G_1 and G_2 reversed. Similarly, we can prove that the Subspace assumption holds in G_2 if the DDH assumption in G_2 holds.

Generalized Decisional Subspace Assumption We also generalize the Decisional Subspace assumption for multiple tuple of dual orthonormal bases.

Definition 5.2 (GDS1: Generalized Decisional Subspace Assumption in G_1).

Given a group generator $\mathcal{G}(\cdot)$, define the following distribution:

$$\begin{aligned} \mathbb{G} &:= (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathcal{R}} \mathcal{G}(1^\lambda), \\ \{(\mathbb{B}_i, \mathbb{B}_i^*)\}_{i \in [d]} &\leftarrow_{\mathcal{R}} \text{Dual}(\mathbb{Z}_q^{N_1}, \dots, \mathbb{Z}_q^{N_d}), \\ \tau_1, \tau_2, \mu_1, \mu_2 &\leftarrow_{\mathcal{R}} \mathbb{Z}_q, \\ \left\{ U_{1,i} &:= g_2^{\mu_1 \mathbf{b}_{1,i}^* + \mu_2 \mathbf{b}_{K+1,i}^*}, \dots, U_{K,i} := g_2^{\mu_1 \mathbf{b}_{K,i}^* + \mu_2 \mathbf{b}_{2K,i}^*} \right\}_{i \in [d]}, \\ \left\{ V_{1,i} &:= g_1^{\tau_1 \mathbf{b}_{1,i}}, \dots, V_{K,i} := g_1^{\tau_1 \mathbf{b}_{K,i}} \right\}_{i \in [d]}, \\ \left\{ W_{1,i} &:= g_1^{\tau_1 \mathbf{b}_{1,i} + \tau_2 \mathbf{b}_{K+1,i}}, \dots, W_{K,i} := g_1^{\tau_1 \mathbf{b}_{K,i} + \tau_2 \mathbf{b}_{2K,i}} \right\}_{i \in [d]}, \end{aligned}$$

$D :=$

$$\left(\mathbb{G}; \left\{ g_2^{\mathbf{b}_{1,i}^*}, \dots, g_2^{\mathbf{b}_{K,i}^*}, g_2^{\mathbf{b}_{2K+1,i}^*}, \dots, g_2^{\mathbf{b}_{N_i,i}^*}, g_1^{\mathbf{b}_{1,i}}, \dots, g_1^{\mathbf{b}_{N_i,i}}, U_{1,i}, \dots, U_{K,i} \right\}_{i \in [d]}, \mu_2 \right),$$

where K_i, N_i are fixed positive integers that satisfy $2K_i \leq N_i$ for $i \in [d]$. We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda) :=$$

$$\left| \Pr[\mathcal{A}(D, \{V_{1,i}, \dots, V_{K,i}\}_{i \in [d]}) = 1] - \Pr[\mathcal{A}(D, \{W_{1,i}, \dots, W_{K,i}\}_{i \in [d]}) = 1] \right|$$

is negligible in the security parameter λ .

Lemma 5.2. If the DDH assumption in G_1 holds, then the Generalized Subspace assumption in G_1 stated in Definition 5.2 also holds. More precisely, for any adversary $\mathcal{A}$ against the Generalized Subspace assumption in G_1 , there exist

probabilistic algorithms $\mathcal{B}$ whose running times are essentially the same as that of $\mathcal{A}$, such that

$$\text{Adv}_{\mathcal{A}}^{\text{GDS1}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{\text{DDH1}}(\lambda).$$

The proof for above lemma is essentially the same as those of Lemma 5.1. We could similarly define Generalized Subspace assumption in G_2 (denoted as GDS2), which is identical to Definitions 5.2 with the roles of G_1 and G_2 reversed. Similarly, we can prove that the Generalized Subspace assumption holds in G_2 if the DDH assumption in G_2 holds.

5.3 Statistical Indistinguishability Lemmas

We require the following lemmas for our security proofs, which is derived from [76].

Lemma 5.3. *For $p \in \mathbb{Z}_q$, let $C_p := \{(\mathbf{x}, \mathbf{v}) | \mathbf{x} \cdot \mathbf{v} = p, \mathbf{0} \neq \mathbf{x}, \mathbf{0} \neq \mathbf{v} \in \mathbb{Z}_q^n\}$. For all $(\mathbf{x}, \mathbf{v}) \in C_p$, for all $(\mathbf{z}, \mathbf{w}) \in C_p$, and $\mathbf{A} \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n \times n}$ ($\mathbf{A}$ is invertible with overwhelming probability),*

$$\Pr[\mathbf{x}\mathbf{A}^\top = \mathbf{z} \wedge \mathbf{v}\mathbf{A}^{-1} = \mathbf{w}] = \frac{1}{\#C_p}.$$

Lemma 5.4. *For $p_1, p_2 \in \mathbb{Z}_q$, let*

$$C_{p_1, p_2} := \left\{ (\mathbf{x}, \mathbf{v}_1, \mathbf{v}_2) \left| \begin{array}{l} \mathbf{x} \neq \mathbf{0}, \mathbf{x} \cdot \mathbf{v}_1 = p_1, \mathbf{x} \cdot \mathbf{v}_2 = p_2, \text{ where } \mathbf{x}, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}_q^n \\ \{\mathbf{v}_1, \mathbf{v}_2\} \text{ are linearly independent over } \mathbb{Z}_q \end{array} \right. \right\}.$$

For all $(\mathbf{x}, \mathbf{v}_1, \mathbf{v}_2) \in C_{p_1, p_2}$, for all $(\mathbf{z}, \mathbf{w}_1, \mathbf{w}_2) \in C_{p_1, p_2}$, and $\mathbf{A} \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n \times n}$ ($\mathbf{A}$ is invertible with overwhelming probability),

$$\Pr[\mathbf{x}\mathbf{A}^\top = \mathbf{z} \wedge \mathbf{v}_1\mathbf{A}^{-1} = \mathbf{w}_1 \wedge \mathbf{v}_2\mathbf{A}^{-1} = \mathbf{w}_2] = \frac{1}{\#C_{p_1, p_2}}.$$

5.4 Identity-Based Encryption

We now present our IBE construction along with our proof of its security under the SXDH assumption. Our constructions use both Dual System Encryption [93]

and DPVS [74, 75]. Specifically, we show how to adapt the recent DLIN-based instantiations of Lewko [61] to the SXDH assumption.

Construction We begin with our IBE scheme:

- **Setup**(1^λ) This algorithm takes in the security parameter λ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q . The algorithm samples random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathcal{R}} \text{Dual}(\mathbb{Z}_q^4)$. Let $\mathbf{d}_1, \dots, \mathbf{d}_4$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \dots, \mathbf{d}_4^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathcal{R}} \mathbb{Z}_q$, computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$, and outputs the global public key as

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\},$$

and the master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}\}.$$

- **KeyGen**(GPK, MSK, id) This algorithm picks $r \leftarrow_{\mathcal{R}} \mathbb{Z}_q$. The secret key is computed as

$$\text{SK}_{\text{id}} := g_2^{\alpha \mathbf{d}_1^* + r(\text{id} \mathbf{d}_1^* - \mathbf{d}_2^*)}.$$

- **Enc**(GPK, id, $\widehat{\text{msg}}$) This algorithm picks $z \leftarrow_{\mathcal{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\text{CT}_{\text{id}} := \left\{ C := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, \quad C_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2)} \right\}.$$

- **Dec**(GPK, SK_{id} , CT_{id}) This algorithm computes the message as

$$\widehat{\text{msg}} := C / e(C_0, \text{SK}_{\text{id}}).$$

Correctness Correctness is straight-forward:

$$\begin{aligned} e(C_0, \text{SK}_{\text{id}}) &= e(g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2)}, g_2^{\alpha \mathbf{d}_1^* + r(\text{id} \mathbf{d}_1^* - \mathbf{d}_2^*)}) \\ &= e(g_1, g_2)^{\alpha z \mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{z r \text{id} \mathbf{d}_1 \cdot \mathbf{d}_1^* - z r \text{id} \mathbf{d}_2 \cdot \mathbf{d}_2^*} \\ &= g_T^{\alpha z}. \end{aligned}$$

Proof of Security We prove the following theorem by showing a series of lemmas.

Theorem 5.1. *The IBE scheme is fully secure and anonymous under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the IBE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{\text{IBE}}(\lambda) \leq \text{Adv}_{\mathcal{B}_0}^{\text{DDH1}}(\lambda) + \sum_{\kappa=1}^{q_n} \text{Adv}_{\mathcal{B}_{\kappa}}^{\text{DDH2}}(\lambda) + (6q_n + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

We adopt the Dual System Encryption methodology to prove the security of our IBE scheme. We use the concepts of *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them. We note that these algorithms are only provided for definitional purposes, and are not part of the IBE system. In particular, they do not need to be efficiently computable from the public parameters and the master secret key.

KeyGenSF The algorithm picks $r, \nu_1, \nu_2 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and forms a semi-functional secret key as

$$\text{SK}_{\mathbf{v}}^{(\text{SF})} := g_2^{\alpha \mathbf{d}_1^* + r(\text{id} \mathbf{d}_1^* - \mathbf{d}_2^*) + [\nu_1 \mathbf{d}_3^* + \nu_2 \mathbf{d}_4^*]}. \quad (5.1)$$

EncryptSF The algorithm picks $z, \chi_1, \chi_2 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and forms a semi-functional ciphertext as

$$\text{CT}_{\mathbf{x}}^{(\text{SF})} := \left\{ \mathbf{C} := \widehat{\text{msg}} \cdot (g_T^{\alpha})^z, \quad \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2) + [\chi_1 \mathbf{d}_3 + \chi_2 \mathbf{d}_4]} \right\}. \quad (5.2)$$

We observe that if one applies the decryption procedure with a semi-functional key and a normal ciphertext, decryption will succeed because $\mathbf{d}_3^*, \mathbf{d}_4^*$ are orthogonal to all of the vectors in exponent of $\mathbf{C}_0$, and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext by a normal key will also

succeed because $\mathbf{d}_3, \mathbf{d}_4$ are orthogonal to all of the vectors in the exponent of the key. When both the ciphertext and key are semi-functional, the result of $e(\mathbf{C}_0, \mathbf{SK}_{\mathbf{v}})$ will have an additional term, namely

$$e(g_1, g_2)^{\nu_1 \chi_1 \mathbf{d}_3^* \cdot \mathbf{d}_3 + \nu_2 \chi_2 \mathbf{d}_4^* \cdot \mathbf{d}_4} = g_T^{(\nu_1 \chi_1 + \nu_2 \chi_2)}.$$

Decryption will then fail unless $\nu_1 \chi_1 + \nu_2 \chi_2 \equiv 0 \pmod{q}$. If this modular equation holds, we say that the key and ciphertext pair is *nominally semi-functional* (c.f. Appendix B).

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbf{v}_1, \dots, \mathbf{v}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- **Game_{Real}**: is the real security game.
- **Game₀**: is the same as **Game_{Real}** except that the challenge ciphertext is semi-functional.
- **Game _{κ}** : for κ from 1 to q_n , **Game _{κ}** is the same as **Game₀** except that the first κ keys are semi-functional and the remaining keys are normal.
- **Game_{Final}**: is the same as **Game _{q_n}** , except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$. We denote the challenge ciphertext in **Game_{Final}** as $\mathbf{CT}_{\text{id}(\mathbf{R})}^{(\mathbf{R})}$.

We prove following lemmas to show the above games are indistinguishable by following an analogous strategy of [61, 63]. Our main arguments are computational indistinguishability (guaranteed by the Subspace assumptions, which are implied by the SXDH assumption) and statistical indistinguishability. The advantage gap between **Game_{Real}** and **Game₀** is bounded by the advantage of the Subspace assumption in G_1 . Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext remains the same from the adversary's view. For κ from 1 to q_n , the advantage gap between **Game _{$\kappa-1$}** and **Game _{κ}** is bounded by the advantage of Subspace assumption

in G_2 . Similarly, we require a statistical indistinguishability argument to show that the distribution of the κ -th semi-functional key remains the same from the adversary's view. Finally, we statistically transform Game_{q_n} to $\text{Game}_{\text{Final}}$ in one step, i.e., we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]}) \quad \text{and} \quad (\text{GPK}, \text{CT}_{\text{id}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

are equivalent for the adversary's view.

We let $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 5.5. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - 2/q$, with $K = 2$ and $N = 4$.*

Proof. $\mathcal{B}_0$ is given

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, g_2^{\mathbf{b}_2^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_4}, U_1, U_2, \mu_2).$$

along with T_1, T_2 . We require that $\mathcal{B}_0$ decides whether T_1, T_2 are distributed as $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}$ or $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_3}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_4}$.

$\mathcal{B}_0$ simulates $\text{Game}_{\text{Real}}$ or Game_0 with $\mathcal{A}$, depending on the distribution of T_1, T_2 . To compute the global public key and master secret key, $\mathcal{B}_0$ first chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{2 \times 2}$. We implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, \mathbf{d}_2 := \mathbf{b}_2, & (\mathbf{d}_3, \mathbf{d}_4) &:= (\mathbf{b}_3, \mathbf{b}_4)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, \mathbf{d}_2^* := \mathbf{b}_2^*, & (\mathbf{d}_3^*, \mathbf{d}_4^*) &:= (\mathbf{b}_3^*, \mathbf{b}_4^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}_3^*}, g_2^{\mathbf{d}_4^*}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ chooses random value $\alpha \in \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. It then gives $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ two pairs $(\widehat{\text{msg}}_{(0)}^*, \text{id}_{(0)}^*)$ and $(\widehat{\text{msg}}_{(1)}^*, \text{id}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $\text{id}_{(\beta)}^*$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := T_1 \cdot T_2^{\text{id}_{(\beta)}^*},$$

where $\mathcal{B}_0$ has implicitly set $z := \tau_1$. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

Now, if T_1, T_2 are equal to $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}$, then this is a properly distributed normal encryption of $\widehat{\text{msg}}_{(\beta)}^*$. In this case, $\mathcal{B}_0$ has properly simulated $\text{Game}_{\text{Real}}$. If T_1, T_2 are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_3}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_4}$ instead, then the ciphertext element $\mathbf{C}_0$ has an additional term of

$$\tau_2(\mathbf{b}_3 + \text{id}_{(\beta)}^* \mathbf{b}_4)$$

in its exponent. The coefficients here in the basis $\mathbf{b}_3, \mathbf{b}_4$ form the vector $\tau_2(1, \text{id}_{(\beta)}^*)$. To compute the coefficients in the basis $\mathbf{d}_3, \mathbf{d}_4$, we multiply the matrix $\mathbf{A}^{-1}$ by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*)^\top$. Since $\mathbf{A}$ is random (everything else given to $\mathcal{A}$ has been distributed independently of $\mathbf{A}$), these coefficients are uniformly random except with probability $2/q$ (more precisely, we set $\mathbf{x} := \tau_2(1, \text{id}_{(\beta)}^*)$, $\mathbf{v} := (\text{id}_\kappa, -1)$ and thus $(\mathbf{A}^{-1} \mathbf{x}^\top, \mathbf{A}^\top \mathbf{v}^\top)$ is uniformly distributed over $\{C_{\tau_2(\text{id}_\kappa - \text{id}_{(\beta)}^*)}\}_{\tau_2 \neq 0}$ from Lemma 5.3 as long as $\tau_2 \neq 0$, which in turn implies that the coefficients $(\mathbf{A}^{-1} \mathbf{x}^\top, 0 \mathbf{A}^\top \mathbf{v}^\top) = (\mathbf{A}^{-1} \mathbf{x}^\top, \mathbf{0})$ in terms of the bases $\mathbf{b}_3, \mathbf{b}_4$ and $\mathbf{b}_3^*, \mathbf{b}_4^*$ are uniformly distributed over $\mathbb{Z}_q^2 \times \{0\}^2$ except for the cases τ_2 defined in Subspace problem is zero and (χ_3, χ_4) defined in Equation 5.2 is the zero vector). Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0 . This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\text{Game}_{\text{Real}}$ and Game_0 to achieve an advantage $\epsilon - \frac{2}{q}$ against the Subspace assumption in G_1 , namely $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - \frac{2}{q}$. $\square$

Lemma 5.6. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_\kappa$ such that $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon -$*

$6/q$, with $K = 2$ and $N = 4$.

Proof. $\mathcal{B}_\kappa$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_4^*}, U_1, U_2, \mu_2)$$

along with T_1, T_2 . We require that $\mathcal{B}_\kappa$ decides whether T_1, T_2 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_3^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_4^*}$.

$\mathcal{B}_\kappa$ simulates Game_κ or $\text{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2 . To compute the global public key and master secret key, $\mathcal{B}_\kappa$ chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{2 \times 2}$. We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, \mathbf{d}_2 := \mathbf{b}_2, & (\mathbf{d}_3, \mathbf{d}_4) &:= (\mathbf{b}_3, \mathbf{b}_4)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, \mathbf{d}_2^* := \mathbf{b}_2^*, & (\mathbf{d}_3^*, \mathbf{d}_4^*) &:= (\mathbf{b}_3^*, \mathbf{b}_4^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). $\mathcal{B}_\kappa$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}$ can give $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}\}$$

is known to $\mathcal{B}_\kappa$, which allows $\mathcal{B}_\kappa$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_\kappa$ also knows $g_2^{\mathbf{d}_3^*}, g_2^{\mathbf{d}_4^*}$, it can easily produce semi-functional keys. To answer the first $\kappa - 1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_\kappa$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ -th key query for id_κ , $\mathcal{B}_\kappa$ responds with:

$$\text{SK}_{\text{id}_\kappa} := (g_2^{\mathbf{b}_1^*})^\alpha T_1^{\text{id}_\kappa} \cdot T_2^{-1}.$$

This implicitly sets $r := \tau_1$. If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}$, then this is a properly distributed normal key. If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_3^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_4^*}$, then this is a

semi-functional key, whose exponent vector includes

$$\tau_2(\text{id}_\kappa \mathbf{b}_3^* - \mathbf{b}_4^*) \quad (5.3)$$

as its component in the span of $\mathbf{b}_3^*, \mathbf{b}_4^*$. To respond to the remaining key queries, $\mathcal{B}_\kappa$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_\kappa$ two pairs $(\widehat{\text{msg}}_{(0)}^*, \text{id}_{(0)}^*)$ and $(\widehat{\text{msg}}_{(1)}^*, \text{id}_{(1)}^*)$. $\mathcal{B}_\kappa$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $\text{id}_{(\beta)}^*$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1 U_2^{\text{id}_{(\beta)}^*},$$

where $\mathcal{B}_\kappa$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2(\mathbf{b}_3 + \text{id}_{(\beta)}^* \mathbf{b}_4). \quad (5.4)$$

We observe that if $\text{id}_{(\beta)}^* = \text{id}_\kappa$ (which is not allowed), then vectors 5.3 and 5.4 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

We now argue that since $\text{id}_{(\beta)}^* \neq \text{id}_\kappa$, in $\mathcal{A}$'s view the vectors 5.3 and 5.4 are distributed as random vectors in the spans of $\mathbf{d}_3^*, \mathbf{d}_4^*$ and $\mathbf{d}_3, \mathbf{d}_4$ respectively. To see this, we take the coefficients of vectors 5.3 and 5.4 in terms of the bases $\mathbf{b}_3^*, \mathbf{b}_4^*$ and $\mathbf{b}_3, \mathbf{b}_4$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_3^*, \mathbf{d}_4^*$ and $\mathbf{d}_3, \mathbf{d}_4$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\text{id}_\kappa, -1)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*).$$

Since the distribution of everything given to $\mathcal{A}$ except for the κ -th key and the challenge ciphertext is independent of the random matrix $\mathbf{A}$ and $\text{id}_{(\beta)}^* \neq \text{id}_\kappa$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_1, χ_2) or (ν_1, ν_2) defined in Equations 5.2 and 5.1 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

If T_1, T_2 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}$, then the coefficients of the vector 5.4 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_1, χ_2) defined in Equations 5.2 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

In summary, $\mathcal{B}_\kappa$ has properly simulated either $\text{Game}_{\kappa-1}$ or Game_κ for $\mathcal{A}$, depending on the distribution of T_1, T_2 . It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - 6/q$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon - 6/q$. $\square$

Lemma 5.7. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) \leq \text{Adv}_{\mathcal{A}}^{\text{Game}_{q_n}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in Game_{q_n} and that of

$$(\text{GPK}, \text{CT}_{\text{id}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$ are equivalent for the adversary's view, where $\text{CT}_{\text{id}_{(\text{R})}}^{(\text{R})}$ is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$.

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_{\text{R}} \mathbb{Z}_q^{2 \times 2}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \dots, \mathbf{f}_4)$, and $\mathbb{F}^* := (\mathbf{f}_1^*, \dots, \mathbf{f}_4^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \xi_{1,1} & \xi_{1,2} & 1 & 0 \\ \xi_{2,1} & \xi_{2,2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}_1^* \\ \mathbf{f}_2^* \\ \mathbf{f}_3^* \\ \mathbf{f}_4^* \end{pmatrix} := \begin{pmatrix} 1 & 0 & -\xi_{1,1} & -\xi_{2,1} \\ 0 & 1 & -\xi_{1,2} & -\xi_{2,2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1^* \\ \mathbf{d}_2^* \\ \mathbf{d}_3^* \\ \mathbf{d}_4^* \end{pmatrix}.$$

It is easy to verify that $\mathbb{F}$ and $\mathbb{F}^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}$ and $\mathbb{D}^*$.

Then the global public key, challenge ciphertext, and queried secret keys, $(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$ in Game_{q_n} are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}\}, \\ \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id}_{(\beta)}^* \mathbf{d}_2) + [\chi_1 \mathbf{d}_3 + \chi_2 \mathbf{d}_4]} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} &:= g_2^{\alpha \mathbf{d}_1^* + r_\ell(\text{id}_\ell \mathbf{d}_1^* - \mathbf{d}_2^*) + [\nu_{1,\ell} \mathbf{d}_3^* + \nu_{2,\ell} \mathbf{d}_4^*]} \right\}_{\ell \in [q_n]}. \end{aligned}$$

Then we can express them over bases $\mathbb{F}$ and $\mathbb{F}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, g_1^{\mathbf{f}_2}\}, \\ \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{(z'_1 \mathbf{f}_1 + z'_2 \mathbf{f}_2) + [\chi_1 \mathbf{d}_3 + \chi_2 \mathbf{d}_4]} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} &:= g_2^{\alpha \mathbf{f}_1^* + r_\ell(\text{id}_\ell \mathbf{f}_1^* - \mathbf{f}_2^*) + [\nu'_{1,\ell} \mathbf{f}_3^* + \nu'_{2,\ell} \mathbf{f}_4^*]} \right\}_{\ell \in [q_n]}, \end{aligned}$$

where

$$\begin{aligned} z'_1 &:= z - \chi_1 \xi_{1,1} - \chi_2 \xi_{2,1}, \\ z'_2 &:= z \text{id}_{\beta}^* - \chi_1 \xi_{1,2} - \chi_2 \xi_{2,2}, \\ \left\{ \begin{array}{l} \nu'_{1,\ell} := \nu_{1,\ell} + \alpha \xi_{1,1} + r_\ell(\text{id}_\ell \xi_{1,1} - \xi_{1,2}), \\ \nu'_{2,\ell} := \nu_{2,\ell} + \alpha \xi_{1,2} + r_\ell(\text{id}_\ell \xi_{2,1} - \xi_{2,2}) \end{array} \right\}_{\ell \in [q_n]}, \end{aligned}$$

which are all uniformly distributed if (χ_1, χ_2) defined in Equation 5.2 is a non-zero vector since $z, \{\xi_{i,j}\}_{i \in [d], j \in [2]}, \{\nu_{1,\ell}, \nu_{2,\ell}\}_{\ell \in [q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients $z(1, \text{id}_{(\beta)}^*)$ of $\mathbf{d}_1, \mathbf{d}_2$ in the $\mathbf{C}_1$ term of the challenge ciphertext is changed to random coefficients $(z'_1, z'_2) \in \mathbb{Z}_q^n$ of $\mathbf{f}_1, \mathbf{f}_2$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$. Moreover, all coefficients $\{(\nu'_{1,\ell}, \nu'_{2,\ell})\}_{\ell \in [q_n]}$ of $\mathbf{f}_3^*, \mathbf{f}_4^*$ in the $\{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]}$ are all uniformly distributed since $\{(\nu_{1,\ell}, \nu_{2,\ell})\}_{\ell \in [q_n]}$ of $\mathbf{d}_3^*, \mathbf{d}_4^*$ are all independent random values. Thus

$$(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$(\text{GPK}, \text{CT}_{\text{id}_{(R)}}^{(R)}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$.

Here, we regard $(\mathbb{F}, \mathbb{F}^*)$ as fixed bases, and $(\mathbb{D}, \mathbb{D}^*)$ (over the choice of $\mathbf{A} := (\xi_{i,j})$) as a set of alternate equi-probable bases that are consistent with $\mathcal{A}$'s view (both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same global public key). Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}, \mathbb{D}^*)$ and in $\text{Game}_{\text{Final}}$ over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, Game_{q_n} and $\text{Game}_{\text{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $(\chi_1, \chi_2) = \mathbf{0}$). $\square$

Lemma 5.8. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $\text{Game}_{\text{Final}}$. Hence, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$. $\square$

In $\text{Game}_{\text{Final}}$, the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$, independent of the two messages and the challenge identities provided by $\mathcal{A}$. Thus, our IBE scheme is anonymous.

5.5 Extensions

We now present our IPE and KP-FE schemes under the SXDH assumption. Our construction of IPE can be regarded as an extension to our IBE scheme, in the sense that we extend the embedded equality relation to a general inner product relation. Our construction of KP-FE is analogues of the DLIN-based KP-FE scheme [76]

Inner Product Encryption We begin with construction of our IPE scheme:

- **Setup**(1^λ) This algorithm takes in the security parameter λ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q . The algorithm samples random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathbb{R}} \text{Dual}(\mathbb{Z}_q^{2n})$. Let $\mathbf{d}_1, \dots, \mathbf{d}_{2n}$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \dots, \mathbf{d}_{2n}^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$, and outputs the global public key as

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \dots, g_1^{\mathbf{d}_n}\},$$

and the master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, \dots, g_2^{\mathbf{d}_n^*}\}.$$

- **KeyGen**(GPK, MSK, $\mathbf{v} := (v_1, \dots, v_n)$) This algorithm picks $r \leftarrow_{\mathbb{R}} \mathbb{Z}_q$. The secret key is computed as

$$\text{SK}_{\mathbf{v}} := g_2^{\alpha \mathbf{d}_1^* + r(v_1 \mathbf{d}_1^* + \dots + v_n \mathbf{d}_n^*)}.$$

- **Enc**(GPK, $\mathbf{x} := (x_1, \dots, x_n), \widehat{\text{msg}}$) WLOG., we assume that $x_1 = 1$.³ This algorithm picks $z \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\text{CT}_{\mathbf{x}} := \left\{ C := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, \quad C_0 := g_1^{z(x_1 \mathbf{d}_1 + \dots + x_n \mathbf{d}_n)} \right\}.$$

- **Dec**(GPK, $\text{SK}_{\mathbf{v}}$, $\text{CT}_{\mathbf{x}}$) This algorithm computes the message as

$$\widehat{\text{msg}} := C / e(C_0, \text{SK}_{\mathbf{v}}).$$

Correctness is straight-forward:

$$\begin{aligned} e(C_0, \text{SK}_{\mathbf{v}}) &= e(g_1^{z(x_1 \mathbf{d}_1 + \dots + x_n \mathbf{d}_n)}, g_2^{\alpha \mathbf{d}_1^* + r(v_1 \mathbf{d}_1^* + \dots + v_n \mathbf{d}_n^*)}) \\ &= e(g_1, g_2)^{\alpha z x_1 \mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{z r (v_1 x_1 \mathbf{d}_1 \cdot \mathbf{d}_1^* + \dots + v_n x_n \mathbf{d}_n \cdot \mathbf{d}_n^*)} \\ &= g_T^{\alpha z} \cdot g_T^{z r \mathbf{v} \cdot \mathbf{x}} \\ &= g_T^{\alpha z}. \end{aligned}$$

³Observe that we can't compute $(g_T^\alpha)^z$ during decryption if $x_1 \neq 1$. In applications, we could fix $x_1 = 1$ for encryption and $v_1 = 0$ for key generation at the cost of one additional dimension if necessary.

The security of our IPE scheme is shown in the following theorem. The proof is essentially similar to that of Theorem 5.1 except that we use an n -dimensional semi-functional space here.

Theorem 5.2. *The IPE scheme is fully secure and weakly attribute-hiding under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the IPE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{\text{IPE}}(\lambda) \leq \text{Adv}_{\mathcal{B}_0}^{\text{DDH1}}(\lambda) + \sum_{\kappa=1}^{q_n} \text{Adv}_{\mathcal{B}_{\kappa}}^{\text{DDH2}}(\lambda) + (6q_n + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

We use the same proof techniques and strategy as with those for our IBE scheme. We first define *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them.

KeyGenSF The algorithm picks $r, \nu_1, \dots, \nu_n \leftarrow_{\mathbf{R}} \mathbb{Z}_q$ and forms a semi-functional secret key as

$$\text{SK}_{\mathbf{v}}^{(\text{SF})} := g_2^{\alpha \mathbf{d}_1^* + r(v_1 \mathbf{d}_1^* + \dots + v_n \mathbf{d}_n^*) + [\nu_1 \mathbf{d}_{n+1}^* + \dots + \nu_n \mathbf{d}_{2n}^*]}. \quad (5.5)$$

EncryptSF The algorithm picks $z, \chi_1, \dots, \chi_n \leftarrow_{\mathbf{R}} \mathbb{Z}_q$ and forms a semi-functional ciphertext as

$$\text{CT}_{\mathbf{x}}^{(\text{SF})} := \left\{ \mathbf{C} := \widehat{\text{msg}} \cdot (g_T^{\alpha})^z, \mathbf{C}_0 := g_1^{z(x_1 \mathbf{d}_1 + \dots + x_n \mathbf{d}_n) + [\chi_1 \mathbf{d}_{n+1} + \dots + \chi_n \mathbf{d}_{2n}]} \right\}. \quad (5.6)$$

We observe that if one applies the decryption procedure with a semi-functional key and a normal ciphertext, decryption will succeed because $\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*$ are orthogonal to all of the vectors in exponent of $\mathbf{C}_0$, and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext by a normal key will also succeed because $\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}$ are orthogonal to all of the vectors in the

exponent of the key. When both the ciphertext and key are semi-functional, the result of $e(C_0, SK_v)$ will have an additional term, namely

$$e(g_1, g_2)^{\nu_1 \chi_1 d_{n+1}^* \cdot d_{n+1} + \dots + \nu_n \chi_n d_{2n}^* \cdot d_{2n}} = g_T^{(\nu_1 \chi_1 + \dots + \nu_n \chi_n)}.$$

Decryption will then fail unless $\nu_1 \chi_1 + \dots + \nu_n \chi_n \equiv 0 \pmod{q}$. If this modular equation holds, we say that the key and ciphertext pair is *nominally semi-functional*.

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbf{v}_1, \dots, \mathbf{v}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- **Game_{Real}**: is the real security game.
- **Game₀**: is the same as **Game_{Real}** except that the challenge ciphertext is semi-functional.
- **Game _{κ}** : for κ from 1 to q_n , **Game _{κ}** is the same as **Game₀** except that the first κ keys are semi-functional and the remaining keys are normal.
- **Game_{Final}**: is the same as **Game _{q_n}** , except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$. We denote the challenge ciphertext in **Game_{Final}** as $CT_{\mathbf{x}(R)}^{(R)}$.

We let $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 5.9. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - 2/q$, with $K = n$ and $N = 2n$.*

Proof. $\mathcal{B}_0$ is given

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_n^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_{2n}}, U_1, \dots, U_n, \mu_2)$$

along with $T_1, \dots, T_n$. We require that $\mathcal{B}_0$ decides whether $T_1, \dots, T_n$ are distributed as $g_1^{\tau_1 \mathbf{b}_1}, \dots, g_1^{\tau_1 \mathbf{b}_n}$ or $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{n+1}}, \dots, g_1^{\tau_1 \mathbf{b}_n + \tau_2 \mathbf{b}_{2n}}$.

$\mathcal{B}_0$ simulates **Game_{Real}** or **Game₀** with $\mathcal{A}$, depending on the distribution of $T_1, \dots, T_n$. To compute the global public key and master secret key, $\mathcal{B}_0$ first

chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$. We implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, \dots, \mathbf{d}_n := \mathbf{b}_n, & (\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}) &:= (\mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n})\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, \dots, \mathbf{d}_n^* := \mathbf{b}_n^*, & (\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*) &:= (\mathbf{b}_{n+1}^*, \dots, \mathbf{b}_{2n}^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}_{n+1}^*}, \dots, g_2^{\mathbf{d}_{2n}^*}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ picks $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. It then gives $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \dots, g_1^{\mathbf{d}_n}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, \dots, g_2^{\mathbf{d}_n^*}\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ two pairs $(\widehat{\text{msg}}_{(0)}^*, \mathbf{x}_{(0)}^*)$ and $(\widehat{\text{msg}}_{(1)}^*, \mathbf{x}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $\mathbf{x}_{(\beta)}^* := (x_{1,\beta}^*, \dots, x_{n,\beta}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)} \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)} \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := T_1^{x_{1,\beta}^*} \dots T_n^{x_{n,\beta}^*},$$

where $\mathcal{B}_0$ has implicitly set $z := \tau_1$. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

Now, if $T_1, \dots, T_n$ are equal to $g_1^{\tau_1 \mathbf{b}_1}, \dots, g_1^{\tau_1 \mathbf{b}_n}$, then this is a properly distributed normal encryption of $\widehat{\text{msg}}_{(\beta)}$. In this case, $\mathcal{B}_0$ has properly simulated $\text{Game}_{\text{Real}}$. If $T_1, \dots, T_n$ are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{n+1}}, \dots, g_1^{\tau_1 \mathbf{b}_n + \tau_2 \mathbf{b}_{2n}}$ instead, then the ciphertext element $\mathbf{C}_0$ has an additional term of

$$\tau_2(x_{1,\beta}^* \mathbf{b}_{n+1} + \dots + x_{n,\beta}^* \mathbf{b}_{2n})$$

in its exponent. The coefficients here in the basis $\mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n}$ form the vector $\tau_2(x_{1,\beta}^*, \dots, x_{n,\beta}^*)$. To compute the coefficients in the basis $\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}$, we multiply the matrix $\mathbf{A}^{-1}$ by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1}(x_{1,\beta}^*, \dots, x_{n,\beta}^*)^\top$.

Since $\mathbf{A}$ is random (everything else given to $\mathcal{A}$ has been distributed independently of $\mathbf{A}$), these coefficients are uniformly random except with probability $2/q$ (namely, the cases τ_2 defined in Subspace problem is zero, $(\chi_1, \dots, \chi_n)$ defined in Equation 5.6 is the zero vector) from Lemma 5.3. Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0 . This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\text{Game}_{\text{Real}}$ and Game_0 to achieve an advantage $\epsilon - \frac{2}{q}$ against the Subspace assumption in G_1 , namely $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - \frac{2}{q}$. $\square$

Lemma 5.10. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa}$ such that $\text{Adv}_{\mathcal{B}_{\kappa}}^{\text{DS2}}(\lambda) = \epsilon - 6/q$, with $K = n$ and $N = 2n$.*

Proof. $\mathcal{B}_{\kappa}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_n}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_{2n}^*}, U_1, \dots, U_n, \mu_2)$$

along with $T_1, \dots, T_n$. We require that $\mathcal{B}_{\kappa}$ decides whether $T_1, \dots, T_n$ are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, \dots, g_2^{\tau_1 \mathbf{b}_n^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_{n+1}^*}, \dots, g_2^{\tau_1 \mathbf{b}_n^* + \tau_2 \mathbf{b}_{2n}^*}$.

$\mathcal{B}_{\kappa}$ simulates Game_{κ} or $\text{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of $T_1, \dots, T_n$. To compute the global public key and master secret key, $\mathcal{B}_{\kappa}$ chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$. We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, \dots, \mathbf{d}_n := \mathbf{b}_n, & (\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}) &:= (\mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n})\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, \dots, \mathbf{d}_n^* := \mathbf{b}_n^*, & (\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*) &:= (\mathbf{b}_{n+1}^*, \dots, \mathbf{b}_{2n}^*)(\mathbf{A}^{-1})^{\top}. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). $\mathcal{B}_{\kappa}$ picks $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute $g_T^{\alpha} := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}_{\kappa}$ can give $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^{\alpha}, g_1^{\mathbf{d}_1}, \dots, g_1^{\mathbf{d}_n}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, \dots, g_2^{\mathbf{d}_n^*}\}$$

is known to $\mathcal{B}_\kappa$, which allows $\mathcal{B}_\kappa$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_\kappa$ also knows $g_2^{\mathbf{d}_{n+1}^*}, \dots, g_2^{\mathbf{d}_{2n}^*}$, it can easily produce semi-functional keys. To answer the first $\kappa - 1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_\kappa$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ -th key query for $\mathbf{v}_\kappa := (v_1, \dots, v_n)$, $\mathcal{B}_\kappa$ responds with:

$$\text{SK}_{\mathbf{v}_\kappa} := (g_2^{\mathbf{b}_1^*})^\alpha T_1^{v_1} \dots T_n^{v_n}.$$

This implicitly sets $r := \tau_1$. If $T_1, \dots, T_n$ are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, \dots, g_2^{\tau_1 \mathbf{b}_n^*}$, then this is a properly distributed normal key. Otherwise, if $T_1, \dots, T_n$ are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_{n+1}^*}, \dots, g_2^{\tau_1 \mathbf{b}_n^* + \tau_2 \mathbf{b}_{2n}^*}$, then this is a semi-functional key, whose exponent vector includes

$$\tau_2(v_1 \mathbf{b}_{n+1}^* + \dots + v_n \mathbf{b}_{2n}^*) \quad (5.7)$$

as its component in the span of $\mathbf{b}_{n+1}^*, \dots, \mathbf{b}_{2n}^*$. To respond to the remaining key queries, $\mathcal{B}_\kappa$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_\kappa$ two pairs $(\widehat{\text{msg}}_{(0)}^*, \mathbf{x}_{(0)}^*)$ and $(\widehat{\text{msg}}_{(1)}^*, \mathbf{x}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $\mathbf{x}_{(\beta)}^* := (x_{1,\beta}^*, \dots, x_{n,\beta}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1^{x_{1,\beta}^*} \dots U_n^{x_{n,\beta}^*},$$

where $\mathcal{B}_\kappa$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2(x_{1,\beta}^* \mathbf{b}_{n+1} + \dots + x_{n,\beta}^* \mathbf{b}_{2n}). \quad (5.8)$$

We observe that if $\mathbf{x}_{(\beta)}^* \cdot \mathbf{v}_\kappa = 0$ (which is not allowed), then vectors 5.7 and 5.8 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

We now argue that since $\mathbf{x}_{(\beta)}^* \cdot \mathbf{v}_\kappa \neq 0$, in $\mathcal{A}$'s view the vectors 5.7 and 5.8 are distributed as random vectors in the spans of $\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*$ and $\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}$

respectively. To see this, we take the coefficients of vectors 5.7 and 5.8 in terms of the bases $\mathbf{b}_{n+1}^*, \dots, \mathbf{b}_{2n}^*$ and $\mathbf{b}_{n+1}, \dots, \mathbf{b}_{2n}$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*$ and $\mathbf{d}_{n+1}, \dots, \mathbf{d}_{2n}$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (v_1, \dots, v_n)^\top, \mu_2 \mathbf{A}^{-1} (x_{1,\beta}^*, \dots, x_{n,\beta}^*).$$

Since the distribution of everything given to $\mathbf{A}$ except for the κ -th key and the challenge ciphertext is independent of the random matrix $\mathbf{A}$ and $\mathbf{x}_{(\beta)}^* \cdot \mathbf{v}_\kappa \neq 0$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, $(\chi_1, \dots, \chi_n)$ or $(\nu_1, \dots, \nu_n)$ defined in Equations 5.6 and 5.5 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

If $T_1, \dots, T_n$ are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, \dots, g_2^{\tau_1 \mathbf{b}_n^*}$, then the coefficients of the vector 5.8 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, $(\chi_1, \dots, \chi_n)$ defined in Equation 5.6 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_\kappa$ has properly simulated Game_κ in this case.

In summary, $\mathcal{B}_\kappa$ has properly simulated either $\text{Game}_{\kappa-1}$ or Game_κ for $\mathcal{A}$, depending on the distribution of $T_1, \dots, T_n$. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - 6/q$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon - 6/q$. $\square$

Lemma 5.11. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) \leq \text{Adv}_{\mathcal{A}}^{\text{Game}_{q_n}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\mathbf{x}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in Game_{q_n} and that of

$$(\text{GPK}, \text{CT}_{\mathbf{x}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$ are equivalent for the adversary's view, where $\text{CT}_{\mathbf{x}_{(\text{R})}}^{(\text{R})}$ is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$.

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n \times n}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \dots, \mathbf{f}_{2n})$, and $\mathbb{F}^* := (\mathbf{f}_1^*, \dots, \mathbf{f}_{2n}^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \\ \mathbf{f}_{n+1} \\ \vdots \\ \mathbf{f}_{2n} \end{pmatrix} := \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ \xi_{1,1} & \cdots & \xi_{1,n} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \xi_{n,1} & \cdots & \xi_{n,n} & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_n \\ \mathbf{d}_{n+1} \\ \vdots \\ \mathbf{d}_{2n} \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}_1^* \\ \vdots \\ \mathbf{f}_n^* \\ \mathbf{f}_{n+1}^* \\ \vdots \\ \mathbf{f}_{2n}^* \end{pmatrix} := \begin{pmatrix} 1 & \cdots & 0 & -\xi_{1,1} & \cdots & -\xi_{n,1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & -\xi_{1,n} & \cdots & -\xi_{n,n} \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1^* \\ \vdots \\ \mathbf{d}_n^* \\ \mathbf{d}_{n+1}^* \\ \vdots \\ \mathbf{d}_{2n}^* \end{pmatrix}.$$

It is easy to verify that $\mathbb{F}$ and $\mathbb{F}^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}$ and $\mathbb{D}^*$.

Then the global public key, challenge ciphertext for $\mathbf{x}_{(\beta)}^* := (x_{1,\beta}^*, \dots, x_{n,\beta}^*)$, and queried secret keys for $\{\mathbf{v}_\ell := (v_{1,\ell}, \dots, v_{n,\ell})\}_{\ell \in [q_n]}$, $(\text{GPK}, \text{CT}_{\mathbf{x}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$ in Game_{q_n} are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, \dots, g_1^{\mathbf{d}_n}\}, \\ \text{CT}_{\mathbf{x}_{(\beta)}^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z(x_{1,\beta}^* \mathbf{d}_1 + \cdots + x_{n,\beta}^* \mathbf{d}_n) + [\chi_1 \mathbf{d}_{n+1} + \cdots + \chi_n \mathbf{d}_{2n}]} \right\}, \\ \left\{ \text{SK}_{\mathbf{v}_\ell}^{(\text{SF})} &:= g_2^{\alpha \mathbf{d}_1^* + r_\ell(v_{1,\ell} \mathbf{d}_1^* + \cdots + v_{n,\ell} \mathbf{d}_n^*) + [\nu_{1,\ell} \mathbf{d}_{n+1}^* + \cdots + \nu_{n,\ell} \mathbf{d}_{2n}^*]} \right\}_{\ell \in [q_n]}. \end{aligned}$$

Then we can express them over bases $\mathbb{F}$ and $\mathbb{F}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, \dots, g_1^{\mathbf{f}_n}\}, \\ \text{CT}_{\mathbf{x}_{(\beta)}^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{(z'_1 \mathbf{f}_1 + \cdots + z'_n \mathbf{f}_n) + [\chi_1 \mathbf{d}_{n+1} + \cdots + \chi_n \mathbf{d}_{2n}]} \right\}, \\ \left\{ \text{SK}_{\mathbf{v}_\ell}^{(\text{SF})} &:= g_2^{\alpha \mathbf{f}_1^* + r_\ell(v_{1,\ell} \mathbf{f}_1^* + \cdots + v_{n,\ell} \mathbf{f}_n^*) + [\nu'_{1,\ell} \mathbf{f}_{n+1}^* + \cdots + \nu'_{n,\ell} \mathbf{f}_{2n}^*]} \right\}_{\ell \in [q_n]}, \end{aligned}$$

where

$$\begin{aligned}
z'_1 &:= zx_{1,\beta}^* - \chi_1 \xi_{1,1} - \cdots - \chi_n \xi_{n,1}, \\
&\vdots \\
z'_n &:= zx_{n,\beta}^* - \chi_1 \xi_{1,n} - \cdots - \chi_n \xi_{n,n}, \\
&\left\{ \begin{array}{c} \nu'_{1,\ell} := \nu_{1,\ell} + \alpha \xi_{1,1} + r_\ell (v_{1,\ell} \xi_{1,1} + \cdots + v_{n,\ell} \xi_{1,n}), \\ \vdots \\ \nu'_{n,\ell} := \nu_{n,\ell} + \alpha \xi_{1,n} + r_\ell (v_{1,\ell} \xi_{n,1} + \cdots + v_{n,\ell} \xi_{n,n}) \end{array} \right\}_{\ell \in [q_n]},
\end{aligned}$$

which are all uniformly distributed if $(\chi_1, \dots, \chi_n)$ defined in Equation 5.6 is a non-zero vector since $z, \{\xi_{i,j}\}_{i \in [d], j \in [n]}, \{\nu_{1,\ell}, \dots, \nu_{n,\ell}\}_{\ell \in [q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients $z(x_{1,\beta}^*, \dots, x_{n,\beta}^*)$ of $\mathbf{d}_1, \dots, \mathbf{d}_n$ in the C_1 term of the challenge ciphertext is changed to random coefficients $(z'_1, \dots, z'_n) \in \mathbb{Z}_q^n$ of $\mathbf{f}_1, \dots, \mathbf{f}_n$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$. Moreover, all coefficients $\{(\nu'_{1,\ell}, \dots, \nu'_{n,\ell})\}_{\ell \in [q_n]}$ of $\mathbf{f}_{n+1}^*, \dots, \mathbf{f}_{2n}^*$ in the $\{\mathbf{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]}$ are all uniformly distributed since $\{(\nu_{1,\ell}, \dots, \nu_{n,\ell})\}_{\ell \in [q_n]}$ of $\mathbf{d}_{n+1}^*, \dots, \mathbf{d}_{2n}^*$ are all independent random values. Thus

$$(\text{GPK}, \text{CT}_{\mathbf{x}_{(\beta)}^*}^{(\text{SF})}, \{\mathbf{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$(\text{GPK}, \text{CT}_{\mathbf{x}_{(\text{R})}}^{(\text{R})}, \{\mathbf{SK}_{\mathbf{v}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$.

In the adversary's view, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same global public key. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}, \mathbb{D}^*)$ and in $\text{Game}_{\text{Final}}$ over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, Game_{q_n} and $\text{Game}_{\text{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $(\chi_1, \dots, \chi_n) = \mathbf{0}$). $\square$

Lemma 5.12. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $\text{Game}_{\text{Final}}$. Hence, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$. $\square$

In $\text{Game}_{\text{Final}}$, the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random vector in $\mathbb{Z}_q^n$, independent of the two messages and the challenge identities provided by $\mathcal{A}$. Thus, our IPE scheme is weakly attribute-hiding.

Key-Policy Functional Encryption We define function $\hat{\rho}_1$ by $\hat{\rho}_1(j) := i$ if $\hat{\rho}(j) = (i, \mathbf{v}_j)$ or $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, where $\hat{\rho}$ is given in access structure $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$. As with [76], we only deal with the case that $\hat{\rho}_1$ is injective for $\mathbb{A} := (\hat{\mathbf{A}}, \hat{\rho})$ with decryption key $\text{SK}_{\mathbb{A}}$ in the proposed KP-FE scheme, see [76] for how to relax the restriction.

- **Setup**($1^\lambda, \vec{n} := (d; n_1, \dots, n_d)$) This algorithm takes in the security parameter λ , a structure $\vec{n} := (d; n_1, \dots, n_d)$ and generates a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q . The algorithm samples random dual orthonormal bases, $\{(\mathbb{D}_i, \mathbb{D}_i^*)\}_{i=0, \dots, d} \leftarrow_{\text{R}} \text{Dual}(\mathbb{Z}_q^3, \mathbb{Z}_q^{2n_1}, \dots, \mathbb{Z}_q^{2n_d})$. For $i \in [0, d]$ let $\mathbf{d}_{1,i}, \dots, \mathbf{d}_{2n_i,i}$ denote the elements of $\mathbb{D}_i$ and $\mathbf{d}_{1,i}^*, \dots, \mathbf{d}_{2n_i,i}^*$ denote the elements of $\mathbb{D}_i^*$, where $2n_0 = 3$. It computes $g_T := e(g_1, g_2)^{\mathbf{d}_{1,0} \cdot \mathbf{d}_{1,0}^*}$ and outputs the global public key as

$$\text{GPK} := \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{ g_1^{\mathbf{d}_{1,i}}, \dots, g_1^{\mathbf{d}_{n_i,i}} \right\}_{i \in [d]} \right\},$$

and the master secret key

$$\text{MSK} := \left\{ g_1^{\mathbf{d}_{1,0}^*}, g_1^{\mathbf{d}_{3,0}^*}, \left\{ g_1^{\mathbf{d}_{1,i}^*}, \dots, g_1^{\mathbf{d}_{n_i,i}^*} \right\}_{i \in [d]} \right\}.$$

- **KeyGen**($\text{MSK}, \text{GPK}, \mathbb{A} := (\hat{\mathbf{A}} \in \mathbb{Z}_q^{\hat{\mathbf{a}} \times \hat{\mathbf{b}}}, \hat{\rho})$) This algorithm picks $\mathbf{w} \leftarrow_{\text{R}} \mathbb{Z}_q^{\hat{\mathbf{b}}}$ and sets $\mathbf{s}^\top := (s_1, \dots, s_{\hat{\mathbf{a}}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}^\top$, $s_0 := \mathbf{1} \cdot \mathbf{w}^\top$. It computes

$$\mathbf{K}_0 := g_2^{-s_0 \mathbf{d}_{1,0}^* + \mathbf{d}_{3,0}^*},$$

for $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \dots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\})$, it also picks $r_j \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes

$$K_j := g_2^{s_j \mathbf{d}_{1,0}^* + r_j (v_{1,j} \mathbf{d}_{1,j}^* + \dots + v_{n_i,j} \mathbf{d}_{n_i,j}^*)}.$$

- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$K_j := g_2^{s_j (v_{1,j} \mathbf{d}_{1,j}^* + \dots + v_{n_i,j} \mathbf{d}_{n_i,j}^*)}.$$

It returns the secret key $\text{SK}_{\mathbb{A}} := (\mathbb{A}, K_0, \{K_j\}_{j \in [\hat{a}]})$.

- $\text{Enc}(\text{GPK}, \Gamma := \{(i, \mathbf{x}_i := (x_{1,i}, \dots, x_{n_i,i})) \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\} \mid 1 \leq i \leq d, x_{1,i} := 1\}, \widehat{\text{msg}})$ This algorithm picks $s, s_0 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$. It computes

$$\begin{aligned} C &:= \widehat{\text{msg}} \cdot g_T^z, & C_0 &:= g_1^{z_0 \mathbf{d}_{1,0} + z \mathbf{d}_{3,0}}, \\ C_i &:= g_1^{z_0 (x_{1,i} \mathbf{d}_{1,i}^* + \dots + x_{n_i,i} \mathbf{d}_{n_i,i}^*)} & \text{for } (i, \mathbf{x}_i) \in \Gamma. \end{aligned}$$

It returns the ciphertext $\text{CT}_{\Gamma} := (\Gamma, C, C_0, \{C_i\}_{(i, \mathbf{x}_i) \in \Gamma})$.

- $\text{Dec}(\text{GPK}, \text{SK}_{\mathbb{A}}, \text{CT}_{\Gamma})$ This algorithm computes Π and $\{\alpha_j\}_{j \in \Pi}$ such that $s_0 = \sum_{j \in \Pi} \alpha_j s_j$, and

$$\begin{aligned} \Pi &\subseteq \{j \in [\hat{a}] \mid [\hat{\rho}(j) = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma \wedge \mathbf{x}_i \cdot \mathbf{v}_j = 0] \\ &\quad \vee [\hat{\rho}(j) = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i) \in \Gamma \wedge \mathbf{x}_i \cdot \mathbf{v}_j \neq 0]\}. \end{aligned}$$

It recovers the message as

$$\widehat{\text{msg}} := C / \left(e(C_0, K_0) \prod_{j \in \Pi \wedge \hat{\rho}(j) = (i, \mathbf{v}_j)} e(C_i, K_j)^{\alpha_j} \prod_{j \in \Pi \wedge \hat{\rho}(j) = \neg(i, \mathbf{v}_j)} e(C_i, K_j)^{\alpha_j / (\mathbf{x}_i \cdot \mathbf{v}_j)} \right).$$

Correctness is as following:

$$\begin{aligned}
& e(C_0, K_0) \prod_{j \in \Pi \wedge \rho(j) = (i, \mathbf{v}_j)} e(C_i, K_j)^{\alpha_j} \prod_{j \in \Pi \wedge \rho(j) = \neg(i, \mathbf{v}_j)} e(C_i, K_j)^{\alpha_j / (\mathbf{x}_i \cdot \mathbf{v}_j)} \\
&= g_T^{-s_0 z_0 + z} \prod_{j \in \Pi \wedge \rho(j) = (i, \mathbf{v}_j)} g_T^{\alpha_j s_j z_0} \prod_{j \in \Pi \wedge \rho(j) = \neg(i, \mathbf{v}_j)} g_T^{\alpha_j s_j z_0 (\mathbf{x}_i \cdot \mathbf{v}_j) / (\mathbf{x}_i \cdot \mathbf{v}_j)} \\
&= g_T^{z_0(-s_0 + \sum_{j \in \Pi} \alpha_j s_j) + z} \\
&= g_T^z.
\end{aligned}$$

The security of our KP-FE scheme is shown in the following theorem. Our proof strategy could be viewed as the combination of those from our IPE scheme and the KP-FE scheme of [76].

Theorem 5.3. *The KP-FE scheme is fully secure and weakly attribute-hiding under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the KP-FE scheme, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_{1-}, \mathcal{B}_1, \dots, \mathcal{B}_{q_n-}, \mathcal{B}_{q_n}$ whose running times are essentially the same as that of $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{\text{KP-FE}}(\lambda) \leq \text{Adv}_{\mathcal{B}_0}^{\text{DDH1}}(\lambda) + \sum_{\kappa=1}^{q_n} (\text{Adv}_{\mathcal{B}_{\kappa-}}^{\text{DDH2}}(\lambda) + \text{Adv}_{\mathcal{B}_{\kappa}}^{\text{DDH2}}(\lambda)) + (2dq_n + 6q_n + d + 3)/q$$

where q_n is the maximum number of $\mathcal{A}$'s key queries.

As before, we adopt the Dual System Encryption methodology to prove the security of our KP-FE scheme. Let $\mathbf{A}_i \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n_i \times n_i}$ ($\mathbf{A}_i$ is invertible with overwhelming probability) for $i \in [d]$:

KeyGenSF A semi-functional key will take on one of two forms. The algorithm first runs normal key generation algorithm to generates

$$\text{SK}_{\mathbb{A}} := (\mathbb{A}, K_0, \{K_j\}_{j \in [\hat{a}]}).$$

A semi-functional key of type 1

$$\text{SK}_{\mathbb{A}}^{(\text{SF1})} := (\mathbb{A}, K_0^{(\text{SF1})}, \{K_j^{(\text{SF1})}\}_{j \in [\hat{a}]})$$

and a semi-functional key of type 2

$$\text{SK}_{\mathbb{A}}^{(\text{SF2})} := (\mathbb{A}, \mathbf{K}_0^{(\text{SF2})}, \{\mathbf{K}_j^{(\text{SF2})}\}_{j \in [\hat{a}]})$$

are formed as follows. It picks $\theta_0 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, $\mathbf{y} \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{\hat{b}}$, sets $\mathbf{u} := (\theta_1, \dots, \theta_{\hat{a}})^{\top} := \mathbf{A} \cdot \mathbf{y}^{\top}$ and computes

$$\mathbf{K}_0^{(\text{SF1})} := \mathbf{K}_0 \cdot g_2^{[\theta_0 \mathbf{d}_{2,0}^*]}, \quad (5.9)$$

$$\mathbf{K}_0^{(\text{SF2})} := \mathbf{K}_0 \cdot g_2^{[\theta_0 \mathbf{d}_{2,0}^*]}, \quad (5.10)$$

for $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \dots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\})$, it also picks $r_j \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes

$$\mathbf{K}_j^{(\text{SF1})} := \mathbf{K}_j \cdot g_2^{[\nu_{1,j} \mathbf{d}_{n_i+1,i}^* + \dots + \nu_{n_i,j} \mathbf{d}_{2n_i,i}^*]}, \quad (5.11)$$

$$\mathbf{K}_j^{(\text{SF2})} := \mathbf{K}_j, \quad (5.12)$$

where $(\nu_{1,j}, \dots, \nu_{n_i,j}) := \mathbf{A}_i^{\top} \cdot (\theta_j + \gamma_j v_{1,j}, \gamma_j v_{2,j}, \dots, \gamma_j v_{n_i,j})^{\top}$ and $\gamma_j \leftarrow_{\mathbb{R}} \mathbb{Z}_q$.

- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$\mathbf{K}_j^{(\text{SF1})} := \mathbf{K}_j \cdot g_2^{[\nu_{1,j} \mathbf{d}_{n_i+1,i}^* + \dots + \nu_{n_i,j} \mathbf{d}_{2n_i,i}^*]}, \quad (5.13)$$

$$\mathbf{K}_j^{(\text{SF2})} := \mathbf{K}_j, \quad (5.14)$$

where $(\nu_{1,j}, \dots, \nu_{n_i,j}) := \theta_j \mathbf{A}_i^{\top} \cdot (v_{1,j}, \dots, v_{n_i,j})^{\top}$.

EncryptSF A semi-functional ciphertext will take on one of two forms. The algorithms first run normal key generation algorithm to generate

$$\text{CT}_{\Gamma} := (\Gamma, \mathbf{C}, \mathbf{C}_0, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i) \in \Gamma}).$$

A semi-functional key of type 1

$$\text{CT}_{\Gamma}^{(\text{SF1})} := (\Gamma, \mathbf{C}^{(\text{SF1})}, \mathbf{C}_0^{(\text{SF1})}, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i)^{(\text{SF1})} \in \Gamma})$$

and a semi-functional key of type 2

$$\text{CT}_\Gamma^{(\text{SF2})} := (\Gamma, \mathbf{C}^{(\text{SF2})}, \mathbf{C}_0^{(\text{SF2})}, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i)^{(\text{SF2})} \in \Gamma})$$

are formed as follows. It picks $z_0 \leftarrow_{\mathbf{R}} \mathbb{Z}_q$ and computes

$$\mathbf{C}^{(\text{SF1})} := \mathbf{C}; \quad \mathbf{C}^{(\text{SF2})} := \mathbf{C}; \quad (5.15)$$

$$\mathbf{C}_0^{(\text{SF1})} := \mathbf{C}_0 \cdot g_1^{[\zeta_0 \mathbf{d}_{2,0}]}, \quad \mathbf{C}_0^{(\text{SF2})} := \mathbf{C}_0 \cdot g_1^{[\zeta_0 \mathbf{d}_{2,0}]}, \quad (5.16)$$

$$\left\{ \mathbf{C}_k^{(\text{SF1})} := \mathbf{C}_k \cdot g_1^{[\chi_{1,i} \mathbf{d}_{n_i+1,i}^* + \dots + \chi_{n_i,i} \mathbf{d}_{2n_i,j}^*]} \right\}_{(i, \mathbf{x}_i) \in \Gamma}; \quad (5.17)$$

$$\left\{ \mathbf{C}_k^{(\text{SF2})} := \mathbf{C}_k \cdot g_1^{[\chi_{1,i} \mathbf{d}_{1,i}^* + \dots + \chi_{n_i,i} \mathbf{d}_{n_i,j}^*]} \right\}_{(i, \mathbf{x}_i) \in \Gamma}. \quad (5.18)$$

where

$$(\chi_{1,i}, \dots, \chi_{n_i,i}) := \zeta_0 \mathbf{A}_i^{-1} \cdot (x_{1,i}, \dots, x_{n_i,i})$$

in Equation 5.17 and

$$(\chi_{1,i}, \dots, \chi_{n_i,i}) \leftarrow_{\mathbf{R}} \mathbb{Z}_q^{n_i}$$

in Equation 5.18.

For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q_n key queries $\mathbb{A}_1, \dots, \mathbb{A}_{q_n}$, our proof of security consists of the following sequence of games between $\mathcal{A}$ and a challenger $\mathcal{B}$.

- **Game_{Real}**: is the real security game.
- **Game₀**: is the same as **Game_{Real}** except that the challenge ciphertext is semi-functional of type 2.
- **Game _{κ -}**: for κ from 1 to q_n , **Game _{κ}** is the same as **Game₀** except that the first $\kappa - 1$ keys are semi-functional of type 2, the κ -th key is semi-functional of type 1, and the remaining keys are normal.
- **Game _{κ}** : for κ from 1 to q_n , **Game _{κ}** is the same as **Game₀** except that the first κ keys are semi-functional of type 2 and the remaining keys are normal.

- $\text{Game}_{\text{Final}}$: is the same as Game_{q_n} , except that the challenge ciphertext is a semi-functional encryption of a random message in G_T . We denote the challenge ciphertext in $\text{Game}_{\text{Final}}$ as $\text{CT}_{\Gamma^*}^{(\text{R})}$.

We prove following lemmas to show the above games are indistinguishable by following an analogous strategy of [76]. Our main arguments are computational indistinguishability (guaranteed by the Subspace assumptions, which are implied by the SXDH assumption) and statistical indistinguishability. The advantage gap between $\text{Game}_{\text{Real}}$ and Game_0 is bounded by the advantage of the Subspace assumption in G_1 . Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext remains the same from the adversary's view. For κ from 1 to q_n , the advantage gap between $\text{Game}_{\kappa-1}$ and $\text{Game}_{\kappa-}$ is bounded by the advantage of Subspace assumption in G_2 . Similarly, we require a statistical indistinguishability argument to show that the distribution of the κ -th semi-functional key remains the same from the adversary's view. For κ from 1 to q_n , the advantage gap between $\text{Game}_{\kappa-}$ and Game_{κ} is bounded by the advantage of Subspace assumption in G_2 . Finally, we statistically transform Game_{q_n} to $\text{Game}_{\text{Final}}$ in one step, i.e., we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{SF2})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]}) \text{ and } (\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{R})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]})$$

are equivalent for the adversary's view.

We let $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 5.13. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - (d+2)/q$, with $K_0 = 1$, $\{K_i = n_1\}_{i \in [d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i \in [d]}$.*

Proof. $\mathcal{B}_0$ is given

$$D := \left(\mathbb{G}; g_2^{\mathbf{b}_{1,0}^*}, g_2^{\mathbf{b}_{3,0}^*}, g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{2,0}}, g_1^{\mathbf{b}_{3,0}}, U_{1,0}, \right. \\ \left. \{g_2^{\mathbf{b}_{1,i}^*}, \dots, g_2^{\mathbf{b}_{n_i,i}^*}, g_1^{\mathbf{b}_{1,i}}, \dots, g_1^{\mathbf{b}_{2n_i,i}}, U_{1,i}, \dots, U_{1,n_i}\}_{i \in [d]}, \mu_2 \right)$$

along with $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. We require that $\mathcal{B}_0$ decides whether

$$T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$$

are distributed as

$$g_1^{\tau_1 \mathbf{b}_{1,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i}}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}}\}_{i \in [d]}$$

or

$$g_1^{\tau_1 \mathbf{b}_{1,0} + \tau_2 \mathbf{b}_{2,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i} + \tau_2 \mathbf{b}_{n_i+1,i}}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i} + \tau_2 \mathbf{b}_{2n_i,i}}\}_{i \in [d]}.$$

$\mathcal{B}_0$ simulates $\text{Game}_{\text{Real}}$ or Game_0 with $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. To compute the global public key and master secret key, $\mathcal{B}_0$ implicitly set dual orthonormal bases $\mathbb{D}_0, \mathbb{D}_0^*$ to:

$$\begin{aligned} \mathbf{d}_{1,0} &:= \mathbf{b}_{1,0}, \mathbf{d}_{2,0} := \mathbf{b}_{2,0}, \mathbf{d}_{3,0} := \mathbf{b}_{3,0}, \\ \mathbf{d}_{1,0}^* &:= \mathbf{b}_{1,0}^*, \mathbf{d}_{2,0}^* := \mathbf{b}_{2,0}^*, \mathbf{d}_{3,0}^* := \mathbf{b}_{3,0}^*, \end{aligned}$$

for $i \in [d]$, $\mathcal{B}_0$ picks a random invertible matrix $\mathbf{A}_i \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n_i \times n_i}$ and implicitly sets dual orthonormal bases $\mathbb{D}_i, \mathbb{D}_i^*$ to:

$$\begin{aligned} \mathbf{d}_{1,i} &:= \mathbf{b}_{1,i}, \dots, \mathbf{d}_{n_i,i} := \mathbf{b}_{n_i,i}, \quad (\mathbf{d}_{n_i+1,i}, \dots, \mathbf{d}_{2n_i,i}) := (\mathbf{b}_{n_i+1,i}, \dots, \mathbf{b}_{2n_i,i}) \mathbf{A}_i, \\ \mathbf{d}_{1,i}^* &:= \mathbf{b}_{1,i}^*, \dots, \mathbf{d}_{n_i,i}^* := \mathbf{b}_{n_i,i}^*, \quad (\mathbf{d}_{n_i+1,i}^*, \dots, \mathbf{d}_{2n_i,i}^*) := (\mathbf{b}_{n_i+1,i}^*, \dots, \mathbf{b}_{2n_i,i}^*) (\mathbf{A}_i^{-1})^\top. \end{aligned}$$

We note that $\{\mathbb{D}_i, \mathbb{D}_i^*\}_{i \in [0,d]}$ are properly distributed, and reveal no information about $\{\mathbf{A}_i\}_{i \in [d]}$ (c.f. Lemma 3.1). Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}_{2,0}^*}, \{g_2^{\mathbf{d}_{n_i+1,i}^*}, \dots, g_2^{\mathbf{d}_{2n_i,i}^*}\}_{i \in [d]}$, but these will not be needed for creating normal keys. $\mathcal{B}_0$ sets $g_T := e(g_1, g_2)^{\mathbf{d}_{1,0} \cdot \mathbf{d}_{1,0}^*}$ and gives $\mathcal{A}$ the global public key

$$\text{GPK} := \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{ g_1^{\mathbf{d}_{1,i}}, \dots, g_1^{\mathbf{d}_{n_i,i}} \right\}_{i \in [d]} \right\},$$

The master secret key

$$\text{MSK} := \left\{ g_1^{\mathbf{d}_{1,0}^*}, g_1^{\mathbf{d}_{3,0}^*}, \left\{ g_1^{\mathbf{d}_{1,i}^*}, \dots, g_1^{\mathbf{d}_{n_i,i}^*} \right\}_{i \in [d]} \right\}.$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_0$ the challenge messages $\widehat{\text{msg}}_{(0)}^*, \widehat{\text{msg}}_{(1)}^*$ and attribute set $\Gamma^* := \{(i, \mathbf{x}_i := (x_{1,i}^*, \dots, x_{n_i,i}^*)) | 1 \leq i \leq d, x_{1,i} := 1\}$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$

and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under Γ^* as follows:

$$\begin{aligned} \mathbf{C} &:= \widehat{\text{msg}}_{(\beta)}^* \cdot g_T^s, & \mathbf{C}_0 &:= T_{1,0} \cdot g_1^{s\mathbf{d}_{3,0}}, \\ \mathbf{C}_i &:= T_1^{x_{1,i}^*} \cdots T_{n_i}^{x_{n_i,i}^*} & \text{for } (i, \mathbf{x}_i^*) \in \Gamma^*. \end{aligned}$$

where $s \leftarrow_{\mathcal{R}} \mathbb{Z}_q$, $\mathcal{B}_0$ has implicitly set $s_0 := \tau_1$. It gives the challenge ciphertext $(\Gamma^*, \mathbf{C}, \mathbf{C}_0, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i) \in \Gamma})$ to $\mathcal{A}$.

Now, if $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to $g_1^{\tau_1 \mathbf{b}_{1,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i}}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}}\}_{i \in [d]}$, then this is a properly distributed normal encryption of $\widehat{\text{msg}}_{\beta}$. In this case, $\mathcal{B}_0$ has properly simulated $\text{Game}_{\text{Real}}$. Otherwise, if $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to $g_1^{\tau_1 \mathbf{b}_{1,0} + \tau_2 \mathbf{b}_{2,0}}, \{g_1^{\tau_1 \mathbf{b}_{1,i} + \tau_2 \mathbf{b}_{n_i+1,i}}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i} + \tau_2 \mathbf{b}_{2n_i,i}}\}_{i \in [d]}$ instead, then the ciphertext elements $\mathbf{C}_0, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i) \in \Gamma}$ have additional terms of

$$\tau_2 \mathbf{b}_{2,0}, \{\tau_2(x_{1,i}^* \mathbf{b}_{n_i+1,i} + \cdots + x_{n_i,i}^* \mathbf{b}_{2n_i,i})\}_{(i, \mathbf{x}_i) \in \Gamma}$$

in their exponents. The coefficients here in the basis $\mathbf{b}_{2,0}, \{\mathbf{b}_{n_i+1,i}, \dots, \mathbf{b}_{2n_i,i}\}_{(i, \mathbf{x}_i) \in \Gamma}$ form the vectors $\tau_2, \{\tau_2(x_{1,i}^*, \dots, x_{n_i,i}^*)\}_{(i, \mathbf{x}_i) \in \Gamma}$. To compute the coefficients in the basis $\mathbf{d}_{2,0}, \{\mathbf{d}_{n_i+1,i}, \dots, \mathbf{d}_{2n_i,i}\}_{(i, \mathbf{x}_i) \in \Gamma}$, we multiply the matrices $\{\mathbf{A}_i^{-1}\}_{(i, \mathbf{x}_i) \in \Gamma}$ by the transpose of this vectors, obtaining $\tau_2, \{\tau_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*)^\top\}_{(i, \mathbf{x}_i) \in \Gamma}$. Since $\{\mathbf{A}_i\}_{i \in [d]}$ are random (everything else given to $\mathcal{A}$ has been distributed independently of $\{\mathbf{A}_i\}_{i \in [d]}$), these coefficients are uniformly random except with probability $(2+d)/q$ (namely, the cases τ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \dots, \chi_{n_i,i})$ defined in Equations 5.16 and 5.18 is zero or the zero vector) from Lemma 5.3. Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0 . This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\text{Game}_{\text{Real}}$ and Game_0 to achieve an advantage ϵ against the Subspace assumption in G_1 , namely $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon - (2+d)/q$. $\square$

Lemma 5.14. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa-}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa-}$ such that $\text{Adv}_{\mathcal{B}_{\kappa-}}^{\text{DS2}}(\lambda) = \epsilon - 1/q$, with $K_0 = 1$, $\{K_i = n_i\}_{i \in [d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i \in [d]}$.*

Proof. $\mathcal{B}_{\kappa^-}$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{3,0}}, g_1^{\mathbf{b}_{1,0}^*}, g_1^{\mathbf{b}_{2,0}^*}, g_1^{\mathbf{b}_{3,0}^*}, U_{1,0}, \right. \\ \left. \{g_1^{\mathbf{b}_{1,i}}, \dots, g_1^{\mathbf{b}_{n_i,i}}, g_2^{\mathbf{b}_{1,i}^*}, \dots, g_2^{\mathbf{b}_{2n_i,i}^*}, U_{1,i}, \dots, U_{n_i,i}\}_{i \in [d]}, \mu_2 \right)$$

along with $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. We require that $\mathcal{B}_{\kappa^-}$ decides whether

$$T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$$

are distributed as

$$g_1^{\tau_1 \mathbf{b}_{1,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^*}\}_{i \in [d]}$$

or

$$g_1^{\tau_1 \mathbf{b}_{1,0}^* + \tau_2 \mathbf{b}_{2,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^* + \tau_2 \mathbf{b}_{n_i+1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^* + \tau_2 \mathbf{b}_{2n_i,i}^*}\}_{i \in [d]}.$$

$\mathcal{B}_{\kappa^-}$ simulates Game_{κ^-} or $\text{Game}_{\kappa-1}$ with $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. To compute the global public key and master secret key, $\mathcal{B}_{\kappa^-}$ implicitly set dual orthonormal bases $\mathbb{D}_0, \mathbb{D}_0^*$ to:

$$\mathbf{d}_{1,0} := \mathbf{b}_{1,0}, \mathbf{d}_{2,0} := \mathbf{b}_{2,0}, \mathbf{d}_{3,0} := \mathbf{b}_{3,0}, \\ \mathbf{d}_{1,0}^* := \mathbf{b}_{1,0}^*, \mathbf{d}_{2,0}^* := \mathbf{b}_{2,0}^*, \mathbf{d}_{3,0}^* := \mathbf{b}_{3,0}^*,$$

for $i \in [d]$, $\mathcal{B}_{\kappa^-}$ picks a random invertible matrix $\mathbf{A}_i \leftarrow_{\mathbf{R}} \mathbb{Z}_q^{n_i \times n_i}$ and implicitly set dual orthonormal bases $\mathbb{D}_i, \mathbb{D}_i^*$ to:

$$\mathbf{d}_{1,i} := \mathbf{b}_{1,i}, \dots, \mathbf{d}_{n_i,i} := \mathbf{b}_{n_i,i}, \quad (\mathbf{d}_{n_i+1,i}, \dots, \mathbf{d}_{2n_i,i}) := (\mathbf{b}_{n_i+1,i}, \dots, \mathbf{b}_{2n_i,i}) \mathbf{A}_i, \\ \mathbf{d}_{1,i}^* := \mathbf{b}_{1,i}^*, \dots, \mathbf{d}_{n_i,i}^* := \mathbf{b}_{n_i,i}^*, \quad (\mathbf{d}_{n_i+1,i}^*, \dots, \mathbf{d}_{2n_i,i}^*) := (\mathbf{b}_{n_i+1,i}^*, \dots, \mathbf{b}_{2n_i,i}^*) (\mathbf{A}_i^{-1})^\top.$$

We note that $\{\mathbb{D}_i, \mathbb{D}_i^*\}_{i \in [0,d]}$ are properly distributed, and reveal no information about $\{\mathbf{A}_i\}_{i \in [d]}$ (c.f. Lemma 3.1). $\mathcal{B}_{\kappa^-}$ computes $g_T := e(g_1, g_2)^{\mathbf{d}_{1,0} \cdot \mathbf{d}_{1,0}^*}$ and then gives $\mathcal{A}$ the global public key

$$\text{GPK} := \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \left\{ g_1^{\mathbf{d}_{1,i}}, \dots, g_1^{\mathbf{d}_{n_i,i}} \right\}_{i \in [d]} \right\},$$

The master secret key

$$\text{MSK} := \left\{ g_1^{\mathbf{d}_{1,0}^*}, g_1^{\mathbf{d}_{3,0}^*}, \left\{ g_1^{\mathbf{d}_{1,i}^*}, \dots, g_1^{\mathbf{d}_{n_i,i}^*} \right\}_{i \in [d]} \right\}.$$

is known to $\mathcal{B}_{\kappa-}$, which allows $\mathcal{B}_{\kappa-}$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa-}$ also knows $g_2^{\mathbf{d}_{2,0}^*}$, it can easily produce semi-functional keys of type 2.

To answer the first $\kappa - 1$ key queries that $\mathcal{A}$ makes, $\mathcal{B}_{\kappa-}$ runs the semi-functional key generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ -th key query for $\mathbb{A}_\kappa := (\hat{\mathbf{A}}, \hat{\rho})$, $\mathcal{B}_{\kappa-}$ picks $\mathbf{w}', \mathbf{y}' \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{\hat{b}}$ and sets $\mathbf{s}^\top := (s'_1, \dots, s'_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{w}^\top, \mathbf{u}^\top := (\theta'_1, \dots, \theta'_{\hat{a}})^\top := \hat{\mathbf{A}} \cdot \mathbf{y}^\top, s'_0 := \mathbf{1} \cdot \mathbf{w}^\top, \theta'_0 := \mathbf{1} \cdot \mathbf{y}^\top$. It computes

$$\mathbf{K}_0 := T_{1,0}^{-\theta'_0} \cdot g_2^{-s'_0 \mathbf{d}_{1,0}^* + \mathbf{d}_{3,0}^*},$$

for $j \in [\hat{a}]$,

- if $\hat{\rho}(j) = (i, \mathbf{v}_j := (v_{1,j}, \dots, v_{n_i,j}) \in \mathbb{Z}_q^{n_i} \setminus \{\mathbf{0}\})$, it also picks $r'_j, \gamma'_j \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes

$$\mathbf{K}_j := T_{1,i}^{\theta'_j} \cdot (T_{1,i}^{v_{1,j}} \dots T_{n_i,i}^{v_{n_i,j}})^{\gamma'_j} \cdot g_2^{s'_j \mathbf{d}_{1,j}^* + r'_j(v_{1,j} \mathbf{d}_{1,j}^* + \dots + v_{n_i,j} \mathbf{d}_{n_i,j}^*)}.$$

- if $\hat{\rho}(j) = \neg(i, \mathbf{v}_j)$, it computes

$$\mathbf{K}_j := (T_{1,i}^{v_{1,j}} \dots T_{n_i,i}^{v_{n_i,j}})^{\theta'_j} \cdot g_2^{s'_j(v_{1,j} \mathbf{d}_{1,j}^* + \dots + v_{n_i,j} \mathbf{d}_{n_i,j}^*)}.$$

This implicitly sets

$$\mathbf{s}^\top := (s_1, \dots, s_{\hat{a}})^\top := \mathbf{s}' + \tau_1 \mathbf{u}', \quad s_0 := s'_0 + \tau_1 \theta'_0, \quad \{r_j := r'_j + \tau_1 \gamma'_j, \quad \gamma_j := \tau_2 \gamma'_j\}_{j \in [\hat{a}]}.$$

We note that if $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}^* + \tau_2 \mathbf{b}_{2,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^* + \tau_2 \mathbf{b}_{n_i+1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^* + \tau_2 \mathbf{b}_{2n_i,i}^*}\}_{i \in [d]},$$

whose exponent vectors include

$$\left. \begin{aligned} & -\tau_2 \theta'_0 \mathbf{b}_{2,0}^*, \\ & \left\{ \begin{array}{ll} \tau_2 (\theta'_j \mathbf{b}_{n_i+1,i}^* + \gamma'_j (v_{1,j} \mathbf{b}_{n_i+1,i}^* + \dots + v_{n_i,j} \mathbf{b}_{2n_i,i}^*)), & \text{if } \hat{\rho}(j) = (i, \mathbf{v}_j) \\ \tau_2 \theta'_j (v_{1,j} \mathbf{b}_{n_i+1,i}^* + \dots + v_{n_i,j} \mathbf{b}_{2n_i,i}^*), & \text{if } \hat{\rho}(j) = \neg(i, \mathbf{v}_j). \end{array} \right\}_{j \in [\hat{a}]} \end{aligned} \right\}. \quad (5.19)$$

Take the coefficients of vectors 5.19 in terms of the bases

$$\mathbf{b}_{2,0}^*, \{\mathbf{b}_{n_i+1,i}^*, \dots, \mathbf{b}_{2n_i,i}^*\}_{i \in [d]}$$

and translate them into coefficients in terms of the bases

$$\mathbf{d}_{2,0}^*, \{\mathbf{d}_{n_i+1,i}^*, \dots, \mathbf{d}_{2n_i,i}^*\}_{i \in [d]}.$$

Using the change of basis matrix $\{\mathbf{A}_i\}_{i \in [d]}$, we obtain the new coefficients (in vector form) as:

$$-\tau_2 \theta'_0, \left\{ \begin{array}{ll} \mathbf{A}_i^\top (\tau_2 \theta'_j + \tau_2 \gamma'_j v_{1,j}, \dots, \tau_2 \gamma'_j v_{n_i,j})^\top, & \text{if } \hat{\rho}(j) = (i, \mathbf{v}_j) \\ \tau_2 \theta'_j \mathbf{A}_i^\top (v_{1,j}, \dots, v_{n_i,j})^\top, & \text{if } \hat{\rho}(j) = \neg(i, \mathbf{v}_j) \end{array} \right\}_{j \in [\hat{a}]} . \quad (5.20)$$

We will argue that this is a semi-functional key of type 1. To respond to the remaining key queries, $\mathcal{B}_{\kappa^-}$ simply runs the normal key generation algorithm.

$\mathcal{A}$ sends $\mathcal{B}_{\kappa^-}$ the challenge messages $\widehat{\text{msg}}_{(0)}^*, \widehat{\text{msg}}_{(1)}^*$ and attribute set $\Gamma^* := \{(i, \mathbf{x}_i^* := (x_{1,i}^*, \dots, x_{n_i,i}^*)) | 1 \leq i \leq d, x_{1,i}^* := 1\}$. $\mathcal{B}_{\kappa^-}$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under Γ^* as follows:

$$\begin{aligned} \mathbf{C} &:= \widehat{\text{msg}}_{(\beta)}^* \cdot g_T^z, \quad \mathbf{C}_0 := U_{1,0} \cdot g_1^{z \mathbf{d}_{3,0}}, \\ \{\mathbf{C}_i &:= (U_{1,i})^{x_{1,i}^* \mathbf{d}_{1,i}} \dots (U_{n_i,i})^{x_{n_i,i}^* \mathbf{d}_{n_i,i}}\}_{(i, \mathbf{x}_i^*) \in \Gamma^*}. \end{aligned}$$

where $z \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, $\mathcal{B}_{\kappa^-}$ has implicitly set $z_0 := \mu_1$. It gives the challenge ciphertext $(\Gamma^*, \mathbf{C}, \mathbf{C}_0, \{\mathbf{C}_i\}_{(i, \mathbf{x}_i^*) \in \Gamma^*})$ to $\mathcal{A}$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_{2,0}, \left\{ \mu_2 (x_{1,i}^* \mathbf{b}_{n_i+1,i}^* + \dots + x_{n_i,i}^* \mathbf{b}_{2n_i,i}^*) \right\}_{(i, \mathbf{x}_i^*) \in \Gamma^*}. \quad (5.21)$$

Take the coefficients of vectors 5.21 in terms of the bases

$$\mathbf{b}_{2,0}, \{\mathbf{b}_{n_i+1,i}, \dots, \mathbf{b}_{2n_i,i}\}_{i \in [d]}$$

and translate them into coefficients in terms of the bases

$$\mathbf{d}_{2,0}, \{\mathbf{d}_{n_i+1,i}, \dots, \mathbf{d}_{2n_i,i}\}_{i \in [d]}.$$

Using the change of basis matrix $\{\mathbf{A}_i\}_{i \in [d]}$, we obtain the new coefficients (in vector form) as:

$$\mu_2, \left\{ \mu_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*) \right\}_{(i, \mathbf{x}_i^*) \in \Gamma^*}. \quad (5.22)$$

If $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^*}\}_{i \in [d]},$$

then the κ -th responding key query is a properly distributed normal key while the vectors 5.22 is a semi-functional ciphertext of type 2 except with probability $(2+d)/q$ (namely, the cases μ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \dots, \chi_{n_i,i})$ defined in Equations 5.16 and 5.18 is zero or the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{\kappa-}$ has properly simulated $\text{Game}_{\kappa-1}$.

If $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}^* + \tau_2 \mathbf{b}_{2,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^* + \tau_2 \mathbf{b}_{n_i+1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^* + \tau_2 \mathbf{b}_{2n_i,i}^*}\}_{i \in [d]},$$

we now argue that $\mathcal{B}_{\kappa-}$ has properly simulated $\text{Game}_{\kappa-}$. Observe that the vectors 5.20 and 5.22 are distributed as semi-functional key and ciphertext of type 1 except that $\tau_2 \theta'_0 := \tau_2 \cdot \mathbf{1} \cdot \mathbf{y}'$ is related to $\tau_2(\theta'_1, \dots, \theta'_a) := \tau_2 \hat{\mathbf{A}} \cdot \mathbf{y}'$ instead of a random value from $\mathbb{Z}_q$. We claim that since $\mathbb{A}_{\kappa}$ does not accept Γ^* , in $\mathcal{A}$'s view the vectors 5.19 and 5.21 are well distributed. Note that for any $j \in [\hat{a}]$, $(\mathbf{A}_i^\top, \mathbf{A}_i^{-1})$ with $i := \hat{\rho}_1(j)$ is independent from the other variables, since $\hat{\rho}_1$ is injective:

1. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \in \Gamma^* \wedge \mathbf{x}_i^* \cdot \mathbf{v}_j = 0]$.

Then, from Lemma 5.3, the joint distribution of

$$(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*), \mathbf{A}_i^\top(\tau_2 \theta'_j + \tau_2 \gamma'_j v_{1,j}, \dots, \tau_2 \gamma'_j v_{n_i,i})^\top)$$

is uniformly and independently distributed on $C_{\tau_2 \theta'_i \mu_2} := \{(\mathbf{z}, \mathbf{w}) | \mathbf{z} \cdot \mathbf{w} = \tau_2 \theta'_i \mu_2\}$.

2. $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \in \Gamma^* \wedge \mathbf{x}_i^* \cdot \mathbf{v}_j \neq 0]$.

Then, from Lemma 5.3, the joint distribution of

$$(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*), \tau_2 \theta'_j \mathbf{A}_i^\top(v_{1,j}, \dots, v_{n_i,i})^\top)$$

is uniformly and independently distributed on $C_{\tau_2\theta'_i\mu_2} := \{(\mathbf{z}, \mathbf{w}) | \mathbf{z} \cdot \mathbf{w} = \tau_2\theta'_i\mu_2\}$.

3. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \in \Gamma^* \wedge \mathbf{x}_i^* \cdot \mathbf{v}_j \neq 0]$.

Then, from Lemma 5.3, the joint distribution of

$$(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*), \mathbf{A}_i^\top(\tau_2\theta'_j + \tau_2\gamma'_j v_{1,j}, \dots, \tau_2\gamma'_j v_{n_i,i}))^\top$$

is uniformly and independently distributed on $C_{\mu_2\tau_2\theta'_i + \tau_2\theta'_j\mu_2\mathbf{x}_i^* \cdot \mathbf{v}_j}$. Since γ'_j is uniformly and independently distributed on $\mathbb{Z}_q$, the above joint distribution is uniformly and independently distributed over $\mathbb{Z}_q^{2n_i}$.

4. $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \in \Gamma^* \wedge \mathbf{x}_i^* \cdot \mathbf{v}_j = 0]$.

Then, from Lemma 5.3, the joint distribution of

$$(\mu_2 \mathbf{A}_i^{-1}(x_{1,i}^*, \dots, x_{n_i,i}^*), \tau_2\theta'_j \mathbf{A}_i^\top(v_{1,j}, \dots, v_{n_i,i}))^\top$$

is uniformly and independently distributed on C_0 .

5. $[\hat{\rho} = (i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \notin \Gamma^*]$ or $[\hat{\rho} = \neg(i, \mathbf{v}_j) \wedge (i, \mathbf{x}_i^*) \notin \Gamma^*]$.

Then, the distribution of

$$(\mathbf{A}_i^\top(\tau_2\theta'_j + \tau_2\gamma'_j v_{1,j}, \dots, \tau_2\gamma'_j v_{n_i,i}))^\top \text{ or } \tau_2\theta'_j \mathbf{A}_i^\top(v_{1,j}, \dots, v_{n_i,i})^\top$$

is uniformly and independently distributed on $\mathbb{Z}_q^{n_i}$.

We then observe the joint distribution (or relation) of 5.20 and 5.22. Those in cases 3-5 are obviously independent from $\tau_2\theta'_0$. Due to the restriction of adversary $\mathcal{A}$'s key queries, $\mathbb{A}_\kappa$ does not accept Γ^* . Therefore, $\tau_2\theta'_0 := \tau_2 \cdot \mathbf{1} \cdot \mathbf{y}'$ is independent from the joint distribution of $\tau_2(\theta'_1, \dots, \theta'_a) := \tau_2 \hat{\mathbf{A}} \cdot \mathbf{y}'$ (over the random selection of $\mathbf{y}'$). Thus, $\tau_2\theta'_0$ is uniformly and independently distributed from the other variables in the joint distribution of $\mathcal{B}_{\kappa^-}$'s simulation except with probability $1/q$ (namely, the case τ_2 defined in Subspace problem is zero) from Lemma 5.3.

In summary, $\mathcal{B}_{\kappa^-}$ has properly simulated either $\text{Game}_{\kappa-1}$ or Game_{κ^-} for $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. It can therefore leverage

$\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - (d+3)/q$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_{\kappa}^-}^{\text{DS2}}(\lambda) = \epsilon - (d+3)/q$. $\square$

Lemma 5.15. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa}^-}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa}$ such that $\text{Adv}_{\mathcal{B}_{\kappa}}^{\text{DS2}}(\lambda) = \epsilon - (d+3)/q$, with $K_0 = 1$, $\{K_i = n_i\}_{i \in [d]}$ and $N_0 = 3$, $\{N_i = 2n_i\}_{i \in [d]}$.*

Proof. $\mathcal{B}_{\kappa}$ is given

$$D := \left(\mathbb{G}; g_1^{\mathbf{b}_{1,0}}, g_1^{\mathbf{b}_{3,0}}, g_1^{\mathbf{b}_{1,0}^*}, g_1^{\mathbf{b}_{2,0}^*}, g_1^{\mathbf{b}_{3,0}^*}, U_{1,0}, \right. \\ \left. \left\{ g_1^{\mathbf{b}_{1,i}}, \dots, g_1^{\mathbf{b}_{n_i,i}}, g_2^{\mathbf{b}_{1,i}^*}, \dots, g_2^{\mathbf{b}_{2n_i,i}^*}, U_{1,i}, \dots, U_{n_i,i} \right\}_{i \in [d]}, \mu_2 \right)$$

along with

$$T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}.$$

We require that $\mathcal{B}_{\kappa}$ decides whether $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are distributed as

$$g_1^{\tau_1 \mathbf{b}_{1,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^*}\}_{i \in [d]}$$

or

$$g_1^{\tau_1 \mathbf{b}_{1,0}^* + \tau_2 \mathbf{b}_{2,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^* + \tau_2 \mathbf{b}_{n_i+1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^* + \tau_2 \mathbf{b}_{2n_i,i}^*}\}_{i \in [d]}.$$

$\mathcal{B}_{\kappa}$ acts in the same way as $\mathcal{B}_{\kappa}^-$ in the proof of Lemma 5.14 except that K_0 is responded as:

$$K_0 := T_{1,0}^{-\theta'_0} \cdot g_2^{-s'_0 \mathbf{d}_{1,0}^* + \mathbf{d}_{3,0}^*} \cdot g_2^{\theta''_0 \mathbf{d}_{2,0}^*},$$

where $\theta''_0 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$.

Now, if $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^*}\}_{i \in [d]},$$

then the κ -th responding key query is a properly distributed semi-functional key of type 2 while the challenge ciphertext is a semi-functional ciphertext of type 2 except with probability $(2+d)/q$ (namely, the cases μ_2 defined in Subspace problem is zero, ζ_0 or there exists $i \in [d]$ such that $(\chi_{1,i}, \dots, \chi_{n_i,i})$ defined in

Equations 5.16 and 5.18 is zero or the zero vector) from Lemma 5.3. In this case, $\mathcal{B}_\kappa$ has properly simulated Game_κ .

If $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$ are equal to

$$g_1^{\tau_1 \mathbf{b}_{1,0}^* + \tau_2 \mathbf{b}_{2,0}^*}, \{g_1^{\tau_1 \mathbf{b}_{1,i}^* + \tau_2 \mathbf{b}_{n_i+1,i}^*}, \dots, g_1^{\tau_1 \mathbf{b}_{n_i,i}^* + \tau_2 \mathbf{b}_{2n_i,i}^*}\}_{i \in [d]},$$

then the κ -th responding key query is a properly distributed semi-functional key of type 1 while the challenge ciphertext is a semi-functional ciphertext of type 1 except with probability $1/q$ (namely, the case τ_2 defined in Subspace problem is zero) from Lemma 5.3. Thus, $\mathcal{B}_\kappa$ has properly simulated $\text{Game}_{\kappa-}$.

In summary, $\mathcal{B}_\kappa$ has properly simulated either $\text{Game}_{\kappa-}$ or Game_κ for $\mathcal{A}$, depending on the distribution of $T_{1,0}, \{T_{1,i}, \dots, T_{n_i,i}\}_{i \in [d]}$. It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon - (d+3)/q$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon - (d+3)/q$. $\square$

Lemma 5.16. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) \leq \text{Adv}_{\mathcal{A}}^{\text{Game}_{q_n}}(\lambda) + 1/q$.*

Proof. To prove this lemma, we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{SF2})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]})$$

in Game_{q_n} and that of

$$(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{R})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$ are equivalent for the adversary's view, where $\text{CT}_{\Gamma^*}^{(\text{R})}$ is a semi-functional encryption of a random message in G_T .

By definition, we only need to consider elements based on $\mathbb{D}_0 := (\mathbf{d}_{1,0}, \mathbf{d}_{2,0}, \mathbf{d}_{3,0})$ and $\mathbb{D}_0^* := (\mathbf{d}_{1,0}^*, \mathbf{d}_{2,0}^*, \mathbf{d}_{3,0}^*)$. For this purpose, we pick $\xi \leftarrow_{\text{R}} \mathbb{Z}_q$ and define new dual orthonormal bases $\mathbb{F}_0 := (\mathbf{f}_{1,0}, \mathbf{f}_{2,0}, \mathbf{f}_{3,0})$, and $\mathbb{F}^* := (\mathbf{f}_{1,0}^*, \mathbf{f}_{2,0}^*, \mathbf{f}_{3,0}^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_{1,0} \\ \mathbf{f}_{2,0} \\ \mathbf{f}_{3,0} \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \xi \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{1,0} \\ \mathbf{d}_{2,0} \\ \mathbf{d}_{3,0} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{f}_{1,0}^* \\ \mathbf{f}_{2,0}^* \\ \mathbf{f}_{3,0}^* \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\xi & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{1,0}^* \\ \mathbf{d}_{2,0}^* \\ \mathbf{d}_{3,0}^* \end{pmatrix}.$$

It is easy to verify that $\mathbb{F}_0$ and $\mathbb{F}_0^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}_0$ and $\mathbb{D}_0^*$.

Then the global public key, challenge ciphertext for Γ^* , and queried secret keys for $\{\mathbb{A}_\ell\}_{\ell \in [q_n]}$, $(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{SF})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF})}\}_{\ell \in [q_n]})$ in Game_{q_n} are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned} \text{GPK} &:= \left\{ \mathbb{G}; g_T, g_1^{\mathbf{d}_{1,0}}, g_1^{\mathbf{d}_{3,0}}, \dots \right\} \\ \text{CT}_{\Gamma^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)} \cdot g_T^z, \quad \mathbf{C}_0 := g_1^{z_0 \mathbf{d}_{1,0} + \zeta_0 \mathbf{d}_{2,0} + z' \mathbf{d}_{3,0}}, \dots \right\}, \\ \left\{ \text{SK}_{\mathbb{A}_\ell}^{(\text{SF})} := \left\{ \mathbb{A}_\ell, \mathbf{K}_{0,\ell} := g_2^{-s_{0,\ell} \mathbf{d}_{1,0}^* + \theta_{0,\ell} \mathbf{d}_{2,0}^* + \mathbf{d}_{3,0}^*}, \dots \right\} \right\}_{\ell \in [q_n]}. \end{aligned}$$

Then we can express them over bases $\mathbb{F}_0$ and $\mathbb{F}_0^*$ as

$$\begin{aligned} \text{GPK} &:= \left\{ \mathbb{G}; g_T, g_1^{\mathbf{f}_{1,0}}, g_1^{\mathbf{f}_{3,0}}, \dots \right\} \\ \text{CT}_{\Gamma^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)} \cdot g_T^z, \quad \mathbf{C}_0 := g_1^{z_0 \mathbf{f}_{1,0} + \zeta_0 \mathbf{f}_{2,0} + z' \mathbf{f}_{3,0}}, \dots \right\}, \\ \left\{ \text{SK}_{\mathbb{A}_\ell}^{(\text{SF})} := \left\{ \mathbb{A}_\ell, \mathbf{K}_{0,\ell} := g_2^{-s_{0,\ell} \mathbf{f}_{1,0}^* + \theta'_{0,\ell} \mathbf{f}_{2,0}^* + \mathbf{f}_{3,0}^*}, \dots \right\} \right\}_{\ell \in [q_n]}. \end{aligned}$$

where

$$\begin{aligned} z' &:= z - \zeta_0 \xi, \\ \left\{ \theta'_{0,\ell} &:= \theta_{0,\ell} + \xi \right\}_{\ell \in [q_n]}, \end{aligned}$$

which are all uniformly distributed if $\zeta_0 \neq 0$ since $\xi, \{\theta_{0,\ell}\}_{\ell \in [q_n]}$ are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients (z_0, ζ_0, z) of $\mathbf{d}_{0,1}, \mathbf{d}_{0,2}, \mathbf{d}_{0,3}$ in the $\mathbf{C}_0$ term of the challenge ciphertext is changed to coefficients $(z_0, \zeta_0, z') \in \mathbb{Z}_q^3$ of $\mathbf{f}_{1,0}, \mathbf{f}_{2,0}, \mathbf{f}_{3,0}$. The challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T , since z in the $\mathbf{C}$ is hidden. Moreover, all other coefficients are all well distributed. Thus

$$(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{SF2})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]})$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$(\text{GPK}, \text{CT}_{\Gamma^*}^{(\text{R})}, \{\text{SK}_{\mathbb{A}_\ell}^{(\text{SF2})}\}_{\ell \in [q_n]})$$

in $\text{Game}_{\text{Final}}$.

In the adversary's view, both $(\mathbb{D}_0, \mathbb{D}_0^*)$ and $(\mathbb{F}_0, \mathbb{F}_0^*)$ are consistent with the same global public key. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{q_n} over bases $(\mathbb{D}_0, \mathbb{D}_0^*)$ and in $\text{Game}_{\text{Final}}$ over bases $(\mathbb{F}_0, \mathbb{F}_0^*)$. Thus, Game_{q_n} and $\text{Game}_{\text{Final}}$ are statistically indistinguishable except with probability $1/q$ (namely, the case $\zeta_0 = 0$). $\square$

Lemma 5.17. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $\text{Game}_{\text{Final}}$. Hence, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Final}}}(\lambda) = 0$. $\square$

5.6 Discussion

We presented a more efficient IBE scheme under the SXDH assumption; our scheme also achieves anonymity.⁴ The ciphertexts and secret keys consist of only five and four group elements, respectively. That is, we shave off two group elements from both ciphertexts and private keys in Lewko's DLIN-based IBE [61]. See Table 5.1 for a summary of comparisons between existing and our IBE schemes, which is compared in terms of master public key (GPK) size, master secret key (MSK) size, secret key (SK) size, ciphertext (CT) size, the number of pairing for decryption, anonymity, and assumption used. The sizes are in terms of group elements and λ denotes the security parameter.⁵

⁴It follows from our analysis that Lewko's IBE [61] is also anonymous, although this was not pointed out in her paper.

⁵We only compare against existing schemes in terms of the numbers of group elements. Note that these are just rough estimates of the sizes of different parameters. For more accurate comparisons in terms of bits, one can refer to Table A.1 for a specific chosen curve and for a specific level of security.

Source	Waters [92]	Waters [93]	Lewko [61]	RCS [82]	Ours
# GPK	$\mathcal{O}(\lambda)$	13	25	9	9
# MSK	1	5	30	7	9
# SK	2	9	6	7	4
# CT	3	11	7	9	5
# pairing	2	9	6	7	4
anonymity	No	No	Yes	No	Yes
assumptions	DBDH	DLIN DBDH	DLIN	XDH DLIN DBDH	SXDH

Table 5.1: Comparisons between existing and our IBE schemes.

As with all known fully secure efficient IBE schemes, our construction relies on Waters’ dual system encryption framework [93]. Following Lewko’s DLIN-based IBE [61], we instantiate Dual System Encryption under the SXDH assumption via DPVS [74, 75]. This is the first work to instantiate either Dual System Encryption or DPVS under the SXDH assumption. We proceed to highlight several salient features of our IBE scheme in relation to Lewko’s IBE [61]:

- Our scheme has an extremely simple structure, similar to the selectively secure IBE of Boneh and Boyen [20], as well as the fully secure analogues given by Lewko and Waters [62] and Lewko [61].
- By shifting from the DLIN assumption to the simpler SXDH assumption, we obtain IBE scheme that is syntactically simpler and achieve shorter parameters. Specifically, Lewko’s IBE scheme [61] relies on 6 basis vectors to simulate the subgroup structure in the Lewko-Waters IBE scheme [62], whereas our construction uses only 4 basis vectors. This means that we can use a 4-dimensional vector space instead of a 6-dimensional one. As a result, we save two group elements in both the secret key and the ciphertext, that

is, by a factor of $1/3$. The savings for the global public key and master secret key is even more substantial, because we use only two basis vectors for the main scheme, as opposed to four basis vectors in Lewko's scheme. In both our scheme and in Lewko's, the remaining two basis vectors are used for the semi-functional components in the proof of security.

- The final step of the proof of security (after switching to semi-functional secret keys and ciphertexts) is different from that of Lewko's. We rely on an information theoretic argument similar to that in [76] instead of computational arguments.

Finally, we believe that our SXDH instantiations constitute a simpler demonstration of the power of DPVS. We also showed how to instantiate IPE and KP-FE under the SXDH assumption. These instantiations are also syntactically simpler and have shorter parameters than those relying on the DLIN assumption [76].

Chapter 6

Efficient Revocable IBE Schemes

Contents

6.1	Binary-Tree Data Structure	85
6.2	Revocable IBE from Bilinear Pairings	87
6.3	Revocable IBE from Lattices	117

In this chapter, we present our new constructions of RIBE that are from bilinear pairings and lattices respectively. We begin with the binary-tree data structure [19], which is used in RIBE schemes to achieve non-interactive key revocation with logarithmic complexity in key update, and how it works. Subsequently, we present two constructions of adaptively secure and anonymous RIBE schemes with constant-size global public key under the SXDH and the DLIN assumptions, respectively. After that, we present the first efficient RIBE scheme from lattices.

6.1 Binary-Tree Data Structure

To reduce the number of key updates to be performed by the key authority, Boldyreva et al. organize and relate users' key updates in a binary tree [6, 73], which is specified as follows:

We denote the binary-tree by **BT** and its root node by **root**. If ν is a leaf node then $\text{Path}(\nu)$ denotes the set of nodes on the path from ν to **root** (both ν and **root** inclusive). If θ is a non-leaf node then θ_ℓ, θ_r denote the left and right child of θ , respectively. We assume that all nodes in the tree are uniquely encoded as strings, and the tree is defined by all of its node descriptions.

Each user is assigned to a leaf node ν . Upon registration, the key authority provides the user with a set of distinct private keys for each node in $\text{Path}(\nu)$.

At time t , the key authority determines the minimal set Y of nodes in **BT** such that none of the nodes in **RL** with corresponding time $\leq t$ (users revoked on or before t) have any ancestor (or, themselves) in the set Y , and all other leaf nodes (corresponding to non-revoked users) have exactly one ancestor (or, themselves) in the set. This algorithm, denoted by **KUNodes**, takes as input a binary tree **BT**, a revocation list **RL** and a time t) and can be formally specified as follows:

```

KUNodes(BT, RL,  $t$ )
   $X, Y \leftarrow \emptyset$ 
   $\forall (\nu_i, t_i) \in \text{RL}$ 
    if  $t_i \leq t$  then add  $\text{Path}(\nu_i)$  to  $X$ 
   $\forall \theta \in X$ 
    if  $\theta_\ell \notin X$  then add  $\theta_\ell$  to  $Y$ 
    if  $\theta_r \notin X$  then add  $\theta_r$  to  $Y$ 
  If  $Y = \emptyset$  then add root to  $Y$ 
  Return  $Y$ 

```

The **KUNodes** algorithm marks all the ancestors of revoked nodes as revoked and outputs all the non-revoked children of revoked nodes. A graphical description is illustrated in Figure 6.1.

The key authority then publishes a key update for all nodes of Y .

A user assigned to leaf ν is then able to form an effective decryption key for time t if the set Y contains a node in $\text{Path}(\nu)$. By doing so, every update of the

Figure 6.1: A graphical description of the KUNodes algorithm.

revocation list RL only requires the key authority to perform logarithmic work in the maximal number of users and linear in the number of revoked users.

6.2 Revocable IBE from Bilinear Pairings

In this section, we present two adaptively secure and anonymous RIBE schemes under the SXDH and the DLIN assumptions, respectively.

Motivation Boldyreva et al.’s RIBE scheme [19] is shown to be secure in the selective-ID model, which is, unfortunately, a rather weak model since the adversary must set the challenge identity and time at the beginning of a security game before receiving the relevant global public key. Libert and Vergnaud [66] recently proposed an RIBE scheme using similar key revocation techniques as with [19] but that is secure in the adaptive-ID model. That is, their scheme is resilient even against an adversary that is allowed to adaptively select an identity as the attack target based on the responses to the adversary’s queries in a security game. However, instead of building on FIBE, Libert and Vergnaud adopted a variant [65] of the Waters IBE scheme [92], which has a drawback in having global public key that comprise $\mathcal{O}(\lambda)$ group elements for security parameter λ . Consequently,

the Libert and Vergnaud RIBE scheme inherits a similar limitation. Clearly, it is desirable that a scheme has small or constant-size global public key, secret keys, and ciphertexts, if it were to be deployed in real world applications.

IBE systems achieving anonymity have ciphertexts that do not leak the identities of the respective target recipients. Hence, anonymous IBE preserves the privacy of ciphertext recipients. Moreover, it is known that anonymous IBE systems can be leveraged to construct Public-Key Encryption supporting Keyword Search (PKE-KS)¹ [27, 1]. Extending this to our context, anonymous and revocable IBE implies PKE-KS with *revocable keywords*, which seems to be a very useful property to have. However, all existing RIBE schemes do not consider this. Our work shows that it is indeed possible to construct such an anonymous RIBE scheme.

Intuition We construct our RIBE scheme from ABE but in a compact way of IBE. ABE provides more general boolean access structure (than that of FIBE) that seems to naturally fit the requirements of RIBE. To see this, we consider the binary-tree key update approach [19, 66] (as described before) in the ABE setting. As before, a ciphertext in our RIBE scheme is associated with two attributes: identity id_i and time period t_j . The ciphertext can be decrypted by a user if and only if the user possesses both the private key for identity id_i and the key update for time t_j on some node in the tree. Since the private keys and key updates associated with a specific node are not given to the users simultaneously, collusion among some (non-revoked) users on some attributes (i.e. time attribute) is possible. Hence from the view of ABE, all users can be regarded as “sharing” the same key (or private component) associated with access structure of the form

$$(\text{id}_1 \vee \cdots \vee \text{id}_{N_1}) \wedge (t_1 \vee \cdots \vee t_{N_2})$$

¹In PKE-KS, an encryptor is able to make a document searchable by keywords and where the capabilities to search on particular keywords are delegated by a central authority.

on each node in the tree for some integers N_1 and N_2 , but each user is given only some parts of the key for this access structure. That is, the parts of the key that the user gets correspond to access structure $\text{id}_* \wedge (t_1 \vee \dots \vee t_{N_2})$ if this node is in the path from the leaf node associated with id_* to the root node; while the key updates corresponding to $(t_1 \vee \dots \vee t_{N_2})$ are given to all users (not necessarily at the same time). Clearly, we require that the private keys are collusion-resistant on different nodes. Moreover, ABE supports a large universe attribute space that can be used to deal with exponential identity spaces required in RIBE. (For each node in the tree, we must generate a key and distribute it to all users associated to the node.)

There exist adaptively secure (key-policy and ciphertext-policy) ABE schemes, see for example Okamoto and Takashima's work [76]. We observe that, however, their schemes cannot be used directly for our purpose because the resulting RIBE somewhat unexpectedly has private keys and ciphertexts with sizes that grow linearly in the maximal number of users and the size of time space (even though they are polynomial in the security parameter).

We give two new constructions of RIBE by building on the concepts of ABE. We adopt the Waters dual system encryption methodology, which is a powerful tool to obtain adaptive security for various classes of FE but does not provide generic transformation from selective security to adaptive security. Further, as explained before, existing adaptively secure ABE schemes do not translate into efficient RIBE. To our knowledge, our work seems to be the first to instantiate the dual system encryption methodology with RIBE that has constant-size global public key.

Unlike the previous works, such as [62, 61, 63, 76], which make use of a single type of nominally semi-functional pair, our security proof requires two types of nominally semi-functional pairs (corresponding to the identity and time attributes). Hence, different from the standard security game for IBE, the adversary is allowed to query parts of the challenge identities and time periods in our

security game. Prior to the start of the game, we execute a preliminary game to locate the positions of the challenge identities and times. We then transform all the private keys and key updates associated with the non-challenge identities and times, respectively, into nominally semi-functional (we denote by Type I) one by one. We finally transform the challenge private keys and key updates (or simply keys) into nominally semi-functional (we denote by Type II) node by node. Note that the distribution of nominally semi-functional pairs of Type I for challenge identities and times can be detected by the adversary that they are different from the distribution of the semi-functional keys and ciphertexts. Moreover, nominally semi-functional pairs of Type II can be only generated for the last remaining keys; in other words, all the other keys must have been already semi-functional. This is why the preliminary game is needed. We also introduce some statistical indistinguishability arguments in our proof to show that the distributions of nominally semi-functional pair of both Types I & II remain the same as the distributions of semi-functional keys and ciphertexts from the adversary's view.

Construction We now begin with our RIBE scheme, at the end of each algorithm, we provide some intuition and/or remark (marked by the symbol “//”) about the algorithm.

- **Setup**(λ, N_{max}) This algorithm takes in a security parameter λ , a maximal number N_{max} of users and generate an asymmetric bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q . Next perform the following steps:
 1. Let RL be an empty set and BT be a binary-tree with at least N_{max} leaf nodes, set $ST = BT$.
 2. Sample random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathbb{R}} \text{Dual}(\mathbb{Z}_q^6)$. Let $\mathbf{d}_1, \dots, \mathbf{d}_6$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \dots, \mathbf{d}_6^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$

3. Output RL, ST, the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\},$$

and the master secret key MSK

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}.$$

- **PriKeyGen**(GPK, MSK, id, RL, ST) This algorithm takes in the global public key GPK, the master secret key MSK, an identity id, the revocation list RL, and the state ST, it picks an unassigned leaf node v from BT and stores id in that node. It then performs the following steps:

1. For any $\theta \in \text{Path}(v)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}$ are undefined, then pick $\alpha_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ ². Pick $r_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{\text{id},\theta} := g_2^{(\alpha_{\theta,1} + r_{\theta,1}\text{id})\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^*}.$$

2. Output $\text{SK}_{\text{id}} := \{(\theta, K_{\text{id},\theta})\}_{\theta \in \text{Path}(v)}$, ST.

//The algorithm computes the id-component of the decryption key for all the nodes on the path from the leaf node (corresponding to id) to root.

- **KeyUpd**(GPK, MSK, t, RL, ST) This algorithm takes in the global public key GPK, the master secret key MSK, a time t, the revocation list RL, and the state ST, it performs the following steps:

1. $\forall \theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}$ are undefined, then pick $\alpha_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,2} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{t,\theta} := g_2^{(\alpha_{\theta,2} + r_{\theta,2}t)\mathbf{d}_1^* - r_{\theta,2}\mathbf{d}_3^*}.$$

2. Output $\text{KU}_t := \{(\theta, K_{t,\theta})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)}$.

²To avoid having to store $\alpha_{\theta,1}, \alpha_{\theta,2}$ for each node, the authority can derive them from a pseudo-random function of using a shorter seed and re-compute them when necessary.

//The algorithm first finds a minimal set of nodes which contains an ancestor (or, the node itself) of all the non-revoked nodes. Then it computes the t-component of the decryption key for all the nodes in that set.

- **DecKeyGen**($\text{SK}_{\text{id}}, \text{KU}_t$) This algorithm takes in a private secret key $\text{SK}_{\text{id}} := \{(i, \text{K}_{\text{id},i})\}_{i \in I}$, $\text{KU}_t := \{(j, \text{K}_{t,j})\}_{j \in J}$ for some set of nodes I, J , it runs the following steps:

1. $\forall (i, \text{K}_{\text{id},i}) \in \text{SK}_{\text{id}}, (j, \text{K}_{t,j}) \in \text{KU}_t$, if $\exists (i, j)$ s.t. $i = j$ then $\text{DK}_{\text{id},t} \leftarrow (\text{K}_{\text{id},i}, \text{K}_{t,j})$; else (if SK_{id} and KU_t do not have any node in common) $\text{DK}_{\text{id},t} \leftarrow \perp$.
2. Output $\text{DK}_{\text{id},t}$.

- **Enc**($\text{GPK}, \text{id}, t, \widehat{\text{msg}}$) This algorithm takes in the global public key GPK , an identity id , a time t , and a message $\widehat{\text{msg}}$, pick $z \leftarrow_{\text{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\text{CT}_{\text{id},t} := \left\{ C := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, \quad C_0 := g_1^{z(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + t\mathbf{d}_3)} \right\}.$$

- **Dec**($\text{GPK}, \text{DK}_{\text{id},t}, \text{CT}_{\text{id},t}$) This algorithm takes in the global public key GPK , a decryption key $\text{DK}_{\text{id},t} := (\text{K}_{\text{id},\theta}, \text{K}_{t,\theta})$, and a ciphertext $\text{CT}_{\text{id},t} := (C, C_0)$, it computes the message as

$$\widehat{\text{msg}} := C / (e(C_0, \text{K}_{\text{id},\theta}) \cdot e(C_0, \text{K}_{t,\theta})).$$

- **KeyRev**($\text{id}, t, \text{RL}, \text{ST}$) This algorithm takes in an identity id , a time t , the revocation list RL , and the state ST , the algorithm adds (id, t) to RL for all nodes ν associated with identity id and returns RL .

Correctness Observe that

$$\begin{aligned} e(C_0, \text{K}_{\text{id},\theta}) &= e(g_1^{z(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + t\mathbf{d}_3)}, g_2^{(\alpha_{\theta,1} + r_{\theta,1}\text{id})\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^*}) \\ &= e(g_1, g_2)^{\alpha_{\theta,1}z\mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{zr_{\theta,1}\text{id}\mathbf{d}_1 \cdot \mathbf{d}_1^* - zr_{\theta,1}\text{id}\mathbf{d}_2 \cdot \mathbf{d}_2^*} \\ &= e(g_1, g_2)^{\alpha_{\theta,1}z\mathbf{d}_1 \cdot \mathbf{d}_1^*}. \end{aligned}$$

Similarly, $e(C_0, K_{t,\theta}) = e(g_1, g_2)^{\alpha_{\theta,2} z \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. The message is recovered as:

$$C / e(C_0, K_{\text{id},\theta}) \cdot e(C_0, K_{t,\theta}) = \widehat{\text{msg}} \cdot (e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*})^z / e(g_1, g_2)^{\alpha z \mathbf{d}_1 \cdot \mathbf{d}_1^*} = \widehat{\text{msg}}.$$

Proof of Security We prove the following theorem to show that our RIBE scheme is indeed secure.

Theorem 6.1. *The RIBE scheme is adaptively secure and anonymous under the SXDH assumption. More precisely, for any adversary $\mathcal{A}$ against the RIBE scheme, there exist probabilistic algorithms*

$$\begin{aligned} & \mathcal{B}_0, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=1, \dots, q_{n_1}, \kappa_2=1, \dots, \lceil \log N_{\max} \rceil}, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=q_{n_1}+1, \dots, q_{n_1}+q_{n_2}+1, \kappa_2=1, \dots, N_{\max}}, \\ & \{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}\}_{\kappa_2=1, \dots, 4N_{\max}} \end{aligned}$$

whose running times are essentially the same as that of $\mathcal{A}$, such that

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) & \leq (q_{n_1} q_{n_2})^2 \cdot \left(\text{Adv}_{\mathcal{B}_0}^{\text{DDH1}}(\lambda) + \sum_{\kappa_1=1}^{q_{n_1}} \sum_{\kappa_2=1}^{\lceil \log N_{\max} \rceil} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DDH2}}(\lambda) \right. \\ & \quad + \sum_{\kappa_1=q_{n_1}+1}^{q_{n_2}} \sum_{\kappa_2=1}^{N_{\max}} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DDH2}}(\lambda) + \sum_{\kappa_2=1}^{4N_{\max}} \text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DDH2}}(\lambda) \\ & \quad \left. + \frac{6(q_{n_1} \lceil \log N_{\max} \rceil + q_{n_2} N_{\max}) + 32N_{\max} + 6}{q} \right) \end{aligned}$$

where $q_{n_1}, q_{n_2} \geq 4$ are the maximum number of $\mathcal{A}$'s private key and key update queries respectively.

We adopt the dual system encryption methodology by Waters [93] to prove the security of our IBE scheme. We use the concepts of *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them. Moreover, in our RIBE system, semi-functional keys include two types—*semi-functional private keys* (for identity) and *semi-functional key updates* (for time).

We note that these algorithms are only provided for definitional purposes, and are not part of the RIBE system. In particular, they do not need to be efficiently computable from the public parameters and the master secret key.

PriKeyGenSF The algorithm picks $r_{\theta,1}, \nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1} \leftarrow_{\text{R}} \mathbb{Z}_q$ and forms a semi-functional private key for node θ as

$$\mathbf{K}_{\text{id},\theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [\nu_{\theta,4,1} \mathbf{d}_4^* + \nu_{\theta,5,1} \mathbf{d}_5^* + \nu_{\theta,6,1} \mathbf{d}_6^*]}. \quad (6.1)$$

KeyUpdSF The algorithm picks $r_{\theta,2}, \nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2} \leftarrow_{\text{R}} \mathbb{Z}_q$ and forms a semi-functional updated key for node θ as

$$\mathbf{K}_{\text{t},\theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [\nu_{\theta,4,2} \mathbf{d}_4^* + \nu_{\theta,5,2} \mathbf{d}_5^* + \nu_{\theta,6,2} \mathbf{d}_6^*]}. \quad (6.2)$$

EncryptSF The algorithm picks $z, \chi_4, \chi_5, \chi_6 \leftarrow_{\text{R}} \mathbb{Z}_q$ and forms a semi-functional ciphertext as

$$\text{CT}_{\text{id},\text{t}}^{(\text{SF})} := \left\{ \mathbf{C} := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + [\chi_4 \mathbf{d}_4 + \chi_5 \mathbf{d}_5 + \chi_6 \mathbf{d}_6]} \right\}. \quad (6.3)$$

We call a private key (resp. key update) semi-functional if all its parts are semi-functional, which is denoted as

$$\text{SK}_{\text{id}}^{(\text{SF})} := \{(\theta, \mathbf{K}_{\text{id},\theta}^{(\text{SF})})\}_{\theta \in \text{Path}(v)} \text{ (resp. } \text{KU}_{\text{t}}^{(\text{SF})} := \{(\theta, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t})}.$$

We observe that a normal ciphertext $\text{CT}_{\text{id},\text{t}}$ can be decrypted by a semi-functional key pair $(\mathbf{K}_{\text{id},\theta}^{(\text{SF})}, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})$ on some node θ , because $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ are orthogonal to all of the vectors in exponent of $\mathbf{C}_0$, and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext $\text{CT}_{\text{id},\text{t}}^{(\text{SF})}$ by a normal key pair $(\mathbf{K}_{\text{id},\theta}, \mathbf{K}_{\text{t},\theta})$ on some node θ will also succeed because $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ are orthogonal to all of the vectors in the exponent of the key. When both the ciphertext and key pair on some node are semi-functional, the result of $e(\mathbf{C}_0^{(\text{SF})}, \mathbf{K}_{\text{id},\theta}^{(\text{SF})}) \cdot e(\mathbf{C}_0^{(\text{SF})}, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})$ and will have an additional term, namely

$$e(g_1, g_2)^{\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \mathbf{d}_i^* \cdot \mathbf{d}_i} = e(g_1, g_2)^{\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \psi}.$$

Decryption will then fail unless $\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \psi \equiv 0 \pmod{q}$. If this modular equation holds, we say that the private key, key update and ciphertext pair is *nominally semi-functional*. In our security proof, there are two type of nominally semi-functional pairs:

Nominally semi-functional pair of Type I

$$\begin{aligned} K_{\text{id},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [\nu_{\theta,4,1} \text{id} \mathbf{d}_4^* - \nu_{\theta,4,1} \mathbf{d}_5^*]}, \\ K_{\text{t},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [\nu_{\theta,4,2} \text{t} \mathbf{d}_4^* - \nu_{\theta,4,2} \mathbf{d}_6^*]}, \\ \text{CT}_{\text{id},\text{t}}^{(\text{SF})} &:= \left\{ C := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, C_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + [\chi_4(\mathbf{d}_4 + \text{id} \mathbf{d}_5 + \text{t} \mathbf{d}_6)]} \right\}, \end{aligned}$$

where $r_{\theta,1}, \nu_{\theta,4,1}, r_{\theta,2}, \nu_{\theta,4,2}, z, \chi_4 \leftarrow_{\mathcal{R}} \mathbb{Z}_q$.

Nominally semi-functional pair of Type II

$$\begin{aligned} K_{\text{id},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [(\alpha_{\theta} + \nu_{\theta,4,1} \text{id}) \mathbf{d}_4^* - \nu_{\theta,4,1} \mathbf{d}_5^*]}, \\ K_{\text{t},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [(-\alpha_{\theta} + \nu_{\theta,4,2} \text{t}) \mathbf{d}_4^* - \nu_{\theta,4,2} \mathbf{d}_6^*]}, \\ \text{CT}_{\text{id},\text{t}}^{(\text{SF})} &:= \left\{ C := \widehat{\text{msg}} \cdot (g_T^\alpha)^z, C_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + [\chi_4(\mathbf{d}_4 + \text{id} \mathbf{d}_5 + \text{t} \mathbf{d}_6)]} \right\}, \end{aligned}$$

where $\alpha_{\theta}, r_{\theta,1}, \nu_{\theta,4,1}, r_{\theta,2}, \nu_{\theta,4,2}, z, \chi_4 \leftarrow_{\mathcal{R}} \mathbb{Z}_q$.

Note that nominally semi-functional pair of Type I is used to transform the non-challenge private key and key update queries into semi-functional ones while Type II is for the challenge private key and key update queries.

Assume that a probabilistic polynomial-time adversary $\mathcal{A}$ makes at most q_{n_1} private key queries $\text{id}_1, \dots, \text{id}_{q_{n_1}}$ and q_{n_2} key update queries $\text{t}_1, \dots, \text{t}_{q_{n_2}}$. Since there are many types of adversaries according to whether the challenges $\text{id}_{(0)}^*, \text{id}_{(1)}^*, \text{t}_{(0)}^*, \text{t}_{(1)}^*$ being queried and the restriction of queries. In order to simplify and unify reduction, we add four dumb queries $\text{id}_{q_{n_1}+1}, \text{id}_{q_{n_1}+2}, \text{t}_{q_{n_2}+1}, \text{t}_{q_{n_2}+2}$ (the keys for these queries will not be given to $\mathcal{A}$), which makes the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $\text{t}_{(0)}^*, \text{t}_{(1)}^*$ be included in the $q_{n_1} + 2$ private key queries and the $q_{n_2} + 2$ key update queries. For any adversary, we use values φ_1, φ_2 ($0 < \varphi_1 < \varphi_2 < q_{n_1} + 2$) to indicate the positions of $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ being queried, namely

either the φ_1 -th or φ_2 -th query is $\text{id}_{(0)}^*$ and the other is $\text{id}_{(1)}^*$. Similarly, we use values φ_3, φ_4 ($0 < \varphi_3 < \varphi_4 < q_{n_2} + 2$) to indicate the positions of $t_{(0)}^*, t_{(1)}^*$ being queried.

Our proof of security consists of the following sequence of games between the adversary $\mathcal{A}$ and challengers.

- **Game_{Real}**: is the real security game.
- **Game_{Real'}**: is a preliminary game, which is the same as **Game_{Real}** except that the challenger picks $\phi_1, \phi_2 \leftarrow_{\mathcal{R}} [q_{n_1} + 2]$ ($0 < \phi_1 < \phi_2 < q_{n_1} + 2$) and $\phi_3, \phi_4 \leftarrow_{\mathcal{R}} [q_{n_2} + 2]$ ($0 < \phi_3 < \phi_4 < q_{n_2} + 2$) before setup, and the game is aborted if $\phi_i \neq \varphi_i$ for any $i \in [4]$.

//Guess the positions of the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $t_{(0)}^*, t_{(1)}^*$.

If the guess is incorrect then the game aborts. Re-write

$$\begin{aligned}\Gamma_1 &:= \{\text{id}'_1, \dots, \text{id}'_{q_{n_1}}\} = \{\text{id}_1, \dots, \text{id}_{q_{n_1}+2}\} \setminus \{\text{id}_{\varphi_1}, \text{id}_{\varphi_2}\} \\ \Gamma_2 &:= \{t'_1, \dots, t'_{q_{n_2}}\} = \{t_1, \dots, t_{q_{n_2}+2}\} \setminus \{t_{\varphi_3}, t_{\varphi_4}\}.\end{aligned}$$

- **Game₀**: is the same as **Game_{Real'}** except that the challenge ciphertext is semi-functional.
- **Game _{κ_1, κ_2}** : for κ_1 from 1 to q_{n_1} , for κ_2 from 0 to $\lceil \log N_{max} \rceil$, **Game _{κ_1, κ_2}** is the same as **Game₀** except that the first $\kappa_1 - 1$ private keys and the first κ_2 components of the κ_1 -th private key for Γ_1 are semi-functional and the remaining keys are normal.

//Transform all private keys into semi-functional ones (one by one and node by node) except the ϕ_1 -th and ϕ_2 -th private queries. Namely, the private keys for the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ (if queried) are still normal. Note that the number of nodes associated with a private key is $\lceil \log N_{max} \rceil$. Moreover **Game_{1,0}** and **Game₀**, **Game _{$\kappa_1, \lceil \log N_{max} \rceil$}** and **Game _{$\kappa_1+1,0$}** are identical.

- **Game _{κ_1, κ_2}** : for κ_1 from $q_{n_1} + 1$ to q_{n_2} , for κ_2 from 0 to N_{max} , **Game _{κ_1, κ_2}**

is the same as $\text{Game}_{q_{n_1}, \lceil \log N_{max} \rceil}$ (namely all private keys for Γ_1 are semi-functional) except that the first $\kappa_1 - q_{n_1} - 1$ key updates and the first κ_2 components of the $(\kappa_1 - q_{n_1})$ -th key update for Γ_2 are semi-functional and the remaining key updates are normal.

//Transform all key updates into semi-functional ones (one by one and node by node) except the ϕ_3 -th and ϕ_4 -th key update queries. Namely, the key updates for the challenge times $t_{(0)}^*, t_{(1)}^*$ (if queried) are still normal. Note that a key update for a time updates at most N_{max} nodes. Moreover, $\text{Game}_{q_{n_1}, \lceil \log N_{max} \rceil}$ and $\text{Game}_{q_{n_1}+1, 0}$, $\text{Game}_{\kappa_1, N_{max}}$ and $\text{Game}_{\kappa_1+1, 0}$ are identical.

- $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$: for κ_2 from 0 to $4N_{max}$, $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ is the same as $\text{Game}_{q_{n_1}+q_{n_2}, N_{max}}$ (namely all private keys for Γ_1 and key updates for Γ_2 are semi-functional) except that the ϕ_1, ϕ_2 -th private keys, the ϕ_3, ϕ_4 -th key updates for the first κ_2 nodes are semi-functional and the remaining keys are normal.

//Transform the φ_1, φ_2 -th private key and the φ_3, φ_4 -th key update queries into semi-functional ones (node by node). Note that there are at most $2^{\lceil \log N_{max} \rceil} (\leq 4N_{max})$ nodes in the binary tree. Moreover, $\text{Game}_{q_{n_1}+q_{n_2}, N_{max}}$ and $\text{Game}_{q_{n_1}+q_{n_2}+1, 0}$ are identical, namely all keys are semi-functional in $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{max}}$.

- Game_{Final} : is the same as $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{max}}$, except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$ a random time in $\mathbb{Z}_q$. We denote the challenge ciphertext in Game_{Final} as $\text{CT}_{\text{id}(\mathbf{R}), t(\mathbf{R})}^{(\mathbf{R})}$.

We prove the following lemmas to show the above games are indistinguishable. The advantage gap between Game_{Real} and Game_0 is bounded by the advantage of the Subspace assumption in G_1 . Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext

remains the same from the adversary's view. Similarly, the advantage gap between any two consecutive games of $\text{Game}_{1,1}$ to $\text{Game}_{q_{n_1}+q_{n_2}+1,4N_{max}}$ is bounded by the advantage of Subspace assumption in G_2 . Finally, we statistically transform $\text{Game}_{q_{n_1}+q_{n_2}+1,4N_{max}}$ to Game_{Final} in one step, i.e., we show the joint distributions of parameters in these two games are equivalent for the adversary's view.

We let $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Real}}$ denote an adversary $\mathcal{A}$'s advantage in the real game.

Lemma 6.1. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Real}}(\lambda) \leq (q_{n_1}q_{n_2})^2 \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{Real'}}(\lambda)$.*

Proof. Since $\phi_1, \phi_2, \phi_3, \phi_4$ are uniformly and independently generated, which are hidden in the adversary $\mathcal{A}$'s view. The game is non-aborted with probability

$$\frac{4}{(q_{n_1}+2)(q_{n_1}+1)(q_{n_2}+2)(q_{n_2}+1)}.$$

Thus,

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{Game}_{Real}}(\lambda) &= \frac{(q_{n_1}+2)(q_{n_1}+1)(q_{n_2}+2)(q_{n_2}+1)}{4} \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{Real'}}(\lambda) \\ &\leq (q_{n_1}q_{n_2})^2 \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{Real'}}(\lambda). \end{aligned}$$

□

Lemma 6.2. *Suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{Real'}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_0$ such that $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon + \frac{2}{q}$, with $K = 3$ and $N = 6$.*

Proof. $\mathcal{B}_0$ is given

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, g_2^{\mathbf{b}_2^*}, g_2^{\mathbf{b}_3^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_6}, U_1, U_2, U_3, \mu_2).$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_0$ decides whether T_1, T_2, T_3 are distributed as $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}, g_1^{\tau_1 \mathbf{b}_3}$ or $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_4}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_5}, g_1^{\tau_1 \mathbf{b}_3 + \tau_2 \mathbf{b}_6}$.

$\mathcal{B}_0$ simulates $\text{Game}_{Real'}$ or Game_0 with $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master secret key, $\mathcal{B}_0$ chooses

a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ ($\mathbf{A}$ is invertible with overwhelming probability if it is uniformly picked) and implicitly sets dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). Moreover, $\mathcal{B}_0$ cannot generate $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}, g_2^{\mathbf{d}_6^*}$, but these will not be needed for creating normal private keys and key updates. $\mathcal{B}_0$ chooses random value $\alpha \in \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. It then gives $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_0$, which allows $\mathcal{B}_0$ to respond to all of $\mathcal{A}$'s queries by calling the normal private keys, key updates, and key revocation algorithms.

$\mathcal{A}$ sends $\mathcal{B}_0$ two pairs $(\text{id}_{(0)}^*, \text{t}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \text{t}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := T_1(T_2)^{\text{id}_{(\beta)}^*}(T_3)^{\text{t}_{(\beta)}^*},$$

where $\mathcal{B}_0$ has implicitly set $z := \tau_1$. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

Now, if T_1, T_2, T_3 are equal to $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}, g_1^{\tau_1 \mathbf{b}_3}$, then this is a properly distributed normal encryption of $\widehat{\text{msg}}_{(\beta)}^*$. In this case, $\mathcal{B}_0$ has properly simulated $\text{Game}_{\text{Real}}'$. If T_1, T_2, T_3 are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_4}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_5}, g_1^{\tau_1 \mathbf{b}_3 + \tau_2 \mathbf{b}_6}$ instead, then the ciphertext element $\mathbf{C}_0$ has an additional term of

$$\tau_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \tau_2 \mathbf{b}_5 + \text{t}_{(\beta)}^* \tau_2 \mathbf{b}_6$$

in its exponent. The coefficients here in the basis $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ form the vector $(\tau_2, \text{id}_{(\beta)}^* \tau_2, \text{t}_{(\beta)}^* \tau_2)$. To compute the coefficients in the basis $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$, we multiply

the matrix $\mathbf{A}^{-1}$ by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$. Since $\mathbf{A}$ is random (everything else given to $\mathcal{A}$ has been distributed independently of $\mathbf{A}$), these coefficients are uniformly random except with probability $2/q$ (namely, the cases τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equation 6.3 is the zero vector) from Lemma 5.3. Therefore, in this case, $\mathcal{B}_0$ has properly simulated Game_0 . This allows $\mathcal{B}_0$ to leverage $\mathcal{A}$'s advantage ϵ between $\text{Game}_{\text{Real}'}$ and Game_0 to achieve an advantage $\epsilon + \frac{2}{q}$ against the Subspace assumption in G_1 , namely $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon + \frac{2}{q}$. $\square$

Lemma 6.3. *For κ_1 from 1 to q_{n_1} , for κ_2 from 0 to $\lceil \log N_{\max} \rceil$, suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa_1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$, with $K = 3$ and $N = 6$.*

Proof. $\mathcal{B}_{\kappa_1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{\kappa_1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ simulates $\text{Game}_{\kappa_1, \kappa_2}$ or $\text{Game}_{\kappa_1, \kappa_2-1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . To compute the global public key and master secret key, $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, $\mathbf{A}$ is invertible). We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}$ can give $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{\kappa_1, \kappa_2}$, which allows $\mathcal{B}_{\kappa_1, \kappa_2}$ to respond to all of $\mathcal{A}$'s private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa_1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}$, $g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that $\mathcal{A}$ makes, $\mathcal{B}_{\kappa_1, \kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ_2 -th component of the κ_1 -th private key for id'_{κ_1} , $\mathcal{B}_{\kappa_1, \kappa_2}$ responds with:

$$\mathbf{K}_{\text{id}'_{\kappa_1}, \theta} := (g_2^{\mathbf{b}_1^*})^{\alpha_{\theta, 1}} T_1^{\text{id}'_{\kappa_1}} (T_2)^{-1}.$$

This implicitly sets $r_{\theta, 1} := \tau_1$. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then this is a properly distributed normal private key. Otherwise, if T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then this is a semi-functional key, whose exponent vector includes

$$\text{id}'_{\kappa_1} \tau_2 \mathbf{b}_4^* - \tau_2 \mathbf{b}_5^* \tag{6.4}$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$. To respond to the remaining key queries, $\mathcal{B}_{\kappa_1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_{\kappa_1, \kappa_2}$ two pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1 (U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{\kappa_1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \tag{6.5}$$

We observe that if $\text{id}_{(\beta)}^* = \text{id}'_{\kappa_1}$ (which is impossible), then vectors 6.4 and 6.5 would be orthogonal, resulting in a nominally semi-functional ciphertext and key

pair $(\mathcal{B}_{\kappa_1, \kappa_2})$ can also use T_1, T_2, T_3 to generate private key part for $t_{(\beta)}^*$ of Type I. It gives the ciphertext (C, C_0) to $\mathcal{A}$.

We now argue that since $\text{id}_{(\beta)}^* \neq \text{id}_{\kappa_1}'$, in $\mathcal{A}$'s view the vectors 6.4 and 6.5 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 6.4 and 6.5 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\text{id}_{\kappa_1}', -1, 0)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top.$$

Since the distribution of everything given to $\mathbf{A}$ except for the κ_2 -th component of the κ_1 -th private key $\mathbf{K}_{\text{id}_{\kappa_1}', \theta}$ and the challenge ciphertext (C, C_0) is independent of the random matrix $\mathbf{A}$ and $\text{id}_{(\beta)}^* \neq \text{id}_{\kappa_1}'$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) or $(\nu_{\theta, 4, 1}, \nu_{\theta, 5, 1}, \nu_{\theta, 6, 1})$ defined in Equations 6.3 and 6.1 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 6.5 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 6.3 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2-1}$ in this case.

In summary, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated either $\text{Game}_{\kappa_1, \kappa_2-1}$ or $\text{Game}_{\kappa_1, \kappa_2}$ for $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon + \frac{6}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$. $\square$

Lemma 6.4. *For κ_1 from $q_{n_1} + 1$ to $q_{n_1} + q_{n_2}$, for κ_2 from 0 to $N_{\max}$, suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa_1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$, with $K = 3$ and $N = 6$.*

Proof. $\mathcal{B}_{\kappa_1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{\kappa_1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ simulates $\text{Game}_{\kappa_1, \kappa_2}$ or $\text{Game}_{\kappa_1, \kappa_2 - 1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . To compute the global public key and master secret key, $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, $\mathbf{A}$ is invertible). We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{b}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}$ can give $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{\kappa_1, \kappa_2}$, which allows $\mathcal{B}_{\kappa_1, \kappa_2}$ to respond to all of $\mathcal{A}$'s private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa_1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that $\mathcal{A}$ makes, $\mathcal{B}_{\kappa_1, \kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$. To answer the κ_2 -th component of the $(\kappa_1 - q_{n_1})$ -th key update for $t'_{\kappa_1 - q_{n_1}}$, $\mathcal{B}_{\kappa_1, \kappa_2}$ responds with:

$$\mathbf{K}_{t'_{\kappa_1 - q_{n_1}}, \theta} := (g_2^{\mathbf{b}_1^*})^{\alpha_{\theta, 2}} T_1^{t'_{\kappa_1 - q_{n_1}}} (T_3)^{-1}.$$

This implicitly sets $r_{\theta, 2} := \tau_1$. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then this is a properly distributed normal key update. Otherwise, if T_1, T_2, T_3 are equal to

$g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then this is a semi-functional key update, whose exponent vector includes

$$\mathbf{t}'_{\kappa_1 - q_{n_1}} \tau_2 \mathbf{b}_4^* - \tau_2 \mathbf{b}_6^* \quad (6.6)$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$. To respond to the remaining key queries, $\mathcal{B}_{\kappa_1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_{\kappa_1, \kappa_2}$ two pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)$. $\mathcal{B}_0$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1(U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{\kappa_1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (6.7)$$

We observe that if $\mathbf{t}_{(\beta)}^* = \mathbf{t}'_{\kappa_1 - q_{n_1}}$ (which is impossible), then vectors 6.6 and 6.7 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair ($\mathcal{B}_{\kappa_1, \kappa_2}$ can also use T_1, T_2, T_3 to generate private key part for $\text{id}_{(\beta)}^*$) of Type I. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

We now argue that since $\mathbf{t}_{(\beta)}^* \neq \mathbf{t}'_{\kappa_1 - q_{n_1}}$, in $\mathcal{A}$'s view the vectors 6.6 and 6.7 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 6.6 and 6.7 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix $\mathbf{A}$, we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\mathbf{t}'_{\kappa_1 - q_{n_1}}, -1, 0)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top.$$

Since the distribution of everything given to $\mathbf{A}$ except for the κ_2 -th component of the $(\kappa_1 - q_{n_1})$ -th key update $\mathbf{K}_{\mathbf{t}'_{\kappa_1 - q_{n_1}}, \theta}$ and the challenge ciphertext $(\mathbf{C}, \mathbf{C}_0)$ is

independent of the random matrix $\mathbf{A}$ and $t_{(\beta)}^* \neq t'_{\kappa_1 - q_{n_1}}$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 6.3 and 6.2 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 6.7 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 6.3 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2 - 1}$ in this case.

In summary, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated either $\text{Game}_{\kappa_1, \kappa_2 - 1}$ or $\text{Game}_{\kappa_1, \kappa_2}$ for $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon + \frac{6}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$. $\square$

Lemma 6.5. *For κ_2 from 0 to $4N_{\max}$, suppose that there exists an adversary $\mathcal{A}$ where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2 - 1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{8}{q}$, with $K = 3$ and $N = 6$.*

Proof. $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ simulates $\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ or $\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2 - 1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . To compute the global public key and master secret key, $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, $\mathbf{A}$ is invertible). We then implicitly set dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed, and reveal no information about $\mathbf{A}$ (c.f. Lemma 3.1). $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. $\mathcal{B}$ can give $\mathcal{A}$ the global public key

$$\text{GPK} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master secret key

$$\text{MSK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$, which allows $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ to respond to all of $\mathcal{A}$'s private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}$, $g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that $\mathcal{A}$ makes, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to $\mathcal{A}$.

However, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ changes the strategy to respond all the components for the κ_2 -th node in the binary tree of private keys and key updates. All key queries for Γ_1 and Γ_2 are similar with the following process except that $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ uses $g_2^{\mathbf{d}_1^*}, \dots, g_2^{\mathbf{d}_6^*}$ to re-randomize the exponents. To answer the component for the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $t_{(0)}^*, t_{(1)}^*$ (namely, the ϕ_1, ϕ_2 -th private key and ϕ_3, ϕ_4 -th key update queries) on the κ_2 -th node, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ picks $\alpha'_{\theta,1}, \alpha''_{\theta,1} \in \mathbb{Z}_q$ and responds with:

$$\begin{aligned} \mathbf{K}_{\text{id}_{(0)}^*, \theta} &:= g_2^{\alpha'_{\theta,1} \mathbf{b}_1^*} (T_1^{\mathbf{b}_1^*})^{\alpha''_{\theta,1}} T_1^{r'_{\theta,1} \text{id}_{(0)}^*} (T_2)^{-r'_{\theta,1}}, \\ \mathbf{K}_{t_{(0)}^*, \theta} &:= g_2^{(\alpha - \alpha'_{\theta,1})} (T_1^{\mathbf{b}_1^*})^{-\alpha''_{\theta,1}} T_1^{r'_{\theta,2} t_{(0)}^*} (T_2)^{-r'_{\theta,2}}, \\ \mathbf{K}_{\text{id}_{(1)}^*, \theta} &:= g_2^{\alpha'_{\theta,1} \mathbf{b}_1^*} (T_1^{\mathbf{b}_1^*})^{\alpha''_{\theta,1}} T_1^{r''_{\theta,1} \text{id}_{(1)}^*} (T_2)^{-r''_{\theta,1}}, \\ \mathbf{K}_{t_{(1)}^*, \theta} &:= g_2^{(\alpha - \alpha'_{\theta,1})} (T_1^{\mathbf{b}_1^*})^{-\alpha''_{\theta,1}} T_1^{r''_{\theta,2} t_{(1)}^*} (T_2)^{-r''_{\theta,2}}, \end{aligned}$$

where $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ implicitly sets $\alpha_{\theta,1} := \alpha'_{\theta,1} + \alpha''_{\theta,1} \tau_1$ and $\alpha_{\theta,2} := \alpha - \alpha'_{\theta,1} - \alpha''_{\theta,1} \tau_1$ (note that $\alpha_{\theta,1} + \alpha_{\theta,2} = \alpha$). Note that from the restriction of queries for the challenge identities and times, only part of the keys are given to $\mathcal{A}$.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then these are properly distributed normal keys. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then these are semi-functional keys, whose exponent vector includes

$$(\alpha''_{\theta,1} \tau_2 + \text{id}_{(0)}^* \tau_2 r'_{\theta,1}) \mathbf{b}_4^* - \tau_2 r'_{\theta,1} \mathbf{b}_5^*, \quad (-\alpha''_{\theta,1} \tau_2 + \mathbf{t}_{(0)}^* \tau_2 r'_{\theta,2}) \mathbf{b}_4^* - \tau_2 r'_{\theta,2} \mathbf{b}_6^*, \quad (6.8)$$

$$(\alpha''_{\theta,1} \tau_2 + \text{id}_{(1)}^* \tau_2 r''_{\theta,1}) \mathbf{b}_4^* - \tau_2 r''_{\theta,1} \mathbf{b}_5^*, \quad (-\alpha''_{\theta,1} \tau_2 + \mathbf{t}_{(1)}^* \tau_2 r''_{\theta,2}) \mathbf{b}_4^* - \tau_2 r''_{\theta,2} \mathbf{b}_6^*, \quad (6.9)$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ respectively. To respond to the remaining key queries, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, $\mathcal{A}$ sends $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ two challenge pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*)$. $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\widehat{\text{msg}}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \widehat{\text{msg}}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \widehat{\text{msg}}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1(U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (6.10)$$

We observe that $((\mathbf{C}, \mathbf{C}_0), \mathbf{K}_{\text{id}_{(\beta)}^*, \theta}, \mathbf{K}_{\mathbf{t}_{(\beta)}^*, \theta})$ would result in a nominally semi-functional ciphertext and key pair of Type II. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to $\mathcal{A}$.

Since the adversary $\mathcal{A}$ is only allowed to query one of the following sets for the challenge identities and times:

$$\emptyset, \{\text{id}_{(0)}^*\}, \{\text{id}_{(1)}^*\}, \{\mathbf{t}_{(0)}^*\}, \{\mathbf{t}_{(1)}^*\}, \{\text{id}_{(0)}^*, \text{id}_{(1)}^*\}, \{\mathbf{t}_{(0)}^*, \mathbf{t}_{(1)}^*\}, \{\text{id}_{(0)}^*, \mathbf{t}_{(1)}^*\}, \{\text{id}_{(1)}^*, \mathbf{t}_{(0)}^*\},$$

we now argue that in $\mathcal{A}$'s view the given vectors 6.8, 6.9 and 6.10 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 6.8, 6.9 and 6.10 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix $\mathbf{A}$ and statistical indistinguishability lemmas, we obtain new random coefficients (in vector form), which are summarized in the following Table:

Case	Type of Adversary	New Coefficients
1	$\emptyset$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$
2	$\{\text{id}_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(0)}^* \tau_2 r'_{\theta,1}, -\tau_2 r'_{\theta,1}, 0)^\top$
3	$\{\text{id}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(1)}^* \tau_2 r''_{\theta,1}, -\tau_2 r''_{\theta,1}, 0)^\top$
4	$\{\mathbf{t}_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(0)}^* \tau_2 r'_{\theta,2}, 0, -\tau_2 r'_{\theta,2})^\top$
5	$\{\mathbf{t}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(1)}^* \tau_2 r''_{\theta,2}, 0, -\tau_2 r''_{\theta,2})^\top$
6	$\{\text{id}_{(0)}^*, \text{id}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(0)}^* \tau_2 r'_{\theta,1}, -\tau_2 r'_{\theta,1}, 0)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(1)}^* \tau_2 r''_{\theta,1}, -\tau_2 r''_{\theta,1}, 0)^\top$
7	$\{\mathbf{t}_{(0)}^*, \mathbf{t}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(0)}^* \tau_2 r'_{\theta,2}, 0, -\tau_2 r'_{\theta,2})^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(1)}^* \tau_2 r''_{\theta,2}, 0, -\tau_2 r''_{\theta,2})^\top$
8	$\{\text{id}_{(0)}^*, \mathbf{t}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(0)}^* \tau_2 r'_{\theta,1}, -\tau_2 r'_{\theta,1}, 0)^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(1)}^* \tau_2 r''_{\theta,2}, 0, -\tau_2 r''_{\theta,2})^\top$
9	$\{\text{id}_{(1)}^*, \mathbf{t}_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1}(1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$ $\mathbf{A}^\top(\alpha''_{\theta,1}\tau_2 + \text{id}_{(1)}^* \tau_2 r''_{\theta,1}, -\tau_2 r''_{\theta,1}, 0)^\top$ $\mathbf{A}^\top(-\alpha''_{\theta,1}\tau_2 + \mathbf{t}_{(0)}^* \tau_2 r'_{\theta,2}, 0, -\tau_2 r'_{\theta,2})^\top$

Since the distribution of everything given to $\mathbf{A}$ except for the coefficients of the vectors in above Table is independent of the random matrix $\mathbf{A}$, we can conclude that these coefficients are uniformly except with probability

- $2/q$, namely except for the cases:
 - μ_2 defined in Subspace problem is zero,

- (χ_4, χ_5, χ_6) defined in Equation 6.3 the zero vector,

from Lemma 5.3 for Case 1.

- $4/q$ namely except for the cases:

- μ_2 or τ_2 defined in Subspace problem is zero,
- (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 6.3, 6.1 and 6.2 is the zero vector,

from Lemma 5.4 for Cases 2-5, since $\alpha''_{\theta,1}$ is randomly picked from $\mathbb{Z}_q$.

- $6/q$ namely except for the cases:

- μ_2 or τ_2 defined in Subspace problem is zero,
- (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 6.3, 6.1 and 6.2 is the zero vector,
- $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ and $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 6.1 and 6.2 are linearly dependent,

from Lemma 5.3 for Cases 6-9, since $\alpha''_{\theta,1}$ is randomly picked from $\mathbb{Z}_q$ and the coefficients of vectors 6.8, 6.9 are linearly independent.

Thus, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 6.10 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 6.3 is the zero vector) from Lemma 5.3. Thus, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2-1}$ in this case.

In summary, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated either $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2-1}$ or $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ for $\mathcal{A}$, depending on the distribution of T_1, T_2, T_3 . It can therefore leverage $\mathcal{A}$'s advantage ϵ between these games to obtain an advantage $\epsilon + \frac{8}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{8}{q}$. $\square$

Lemma 6.6. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{max}}}(\lambda) \leq \text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) + \frac{1}{q}$.*

Proof. To prove this lemma, we show the joint distributions of

$$(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in Game_{ν} and that of

$$(\text{GPK}, \text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in Game_{Final} are equivalent for the adversary's view, where $\text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}$ is a semi-functional encryption of a random message in G_T and under a random identity $\text{id}_{(\text{R})}$ in $\mathbb{Z}_q$ and a random time $\text{t}_{(\text{R})}$ in $\mathbb{Z}_q$.

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_{\text{R}} \mathbb{Z}_q^{3 \times 3}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \dots, \mathbf{f}_6)$, and $\mathbb{F}^* := (\mathbf{f}_1^*, \dots, \mathbf{f}_6^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_6 \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \xi_{1,1} & \xi_{1,2} & \xi_{1,3} & 1 & 0 & 0 \\ \xi_{2,1} & \xi_{2,2} & \xi_{2,3} & 0 & 1 & 0 \\ \xi_{3,1} & \xi_{3,2} & \xi_{3,3} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \\ \mathbf{d}_5 \\ \mathbf{d}_6 \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}_1^* \\ \mathbf{f}_2^* \\ \mathbf{f}_3^* \\ \mathbf{f}_4^* \\ \mathbf{f}_5^* \\ \mathbf{f}_6^* \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & -\xi_{1,1} & -\xi_{2,1} & -\xi_{3,1} \\ 0 & 1 & 0 & -\xi_{1,2} & -\xi_{2,2} & -\xi_{3,2} \\ 0 & 0 & 1 & -\xi_{1,3} & -\xi_{2,3} & -\xi_{3,3} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1^* \\ \mathbf{d}_2^* \\ \mathbf{d}_3^* \\ \mathbf{d}_4^* \\ \mathbf{d}_5^* \\ \mathbf{d}_6^* \end{pmatrix}.$$

It is easy to verify that $\mathbb{F}$ and $\mathbb{F}^*$ are also dual orthonormal, and are distributed the same as $\mathbb{D}$ and $\mathbb{D}^*$.

Then the global public key, challenge ciphertext, queried private keys and key updates ($\text{GPK}, \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_2}]}$) in $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{\max}}$ are expressed over bases $\mathbb{D}$ and $\mathbb{D}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}, \\ \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)} \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + \text{t}\mathbf{d}_3) + \chi_4\mathbf{d}_4 + \chi_5\mathbf{d}_5 + \chi_6\mathbf{d}_6} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} &:= \left\{ \left(\theta, \mathbf{K}_{\text{id}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1}\text{id}_\ell)\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^* + \nu_{\theta,4,1}\mathbf{d}_4^* + \nu_{\theta,5,1}\mathbf{d}_5^* + \nu_{\theta,6,1}\mathbf{d}_6^*} \right) \right. \right. \\ &\quad \left. \left. \right\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\ \left\{ \text{KU}_{\text{t}_\ell}^{(\text{SF})} &:= \left\{ \left(\theta, \mathbf{K}_{\text{t}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2}\text{t}_\ell)\mathbf{d}_1^* - r_{\theta,2}\mathbf{d}_3^* + \nu_{\theta,4,2}\mathbf{d}_4^* + \nu_{\theta,5,2}\mathbf{d}_5^* + \nu_{\theta,6,2}\mathbf{d}_6^*} \right) \right. \right. \\ &\quad \left. \left. \right\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_\ell)} \right\}_{\ell \in [q_{n_2}]}. \end{aligned}$$

Then we can express them over bases $\mathbb{F}$ and $\mathbb{F}^*$ as

$$\begin{aligned} \text{GPK} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, g_1^{\mathbf{f}_2}, g_1^{\mathbf{f}_3}\}, \\ \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})} &:= \left\{ \mathbf{C} := \widehat{\text{msg}}_{(\beta)} \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z'\mathbf{f}_1 + z'_2\mathbf{f}_2 + z'_3\mathbf{f}_3 + \chi_4\mathbf{f}_4 + \chi_5\mathbf{f}_5 + \chi_6\mathbf{f}_6} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} &:= \left\{ \left(\theta, \mathbf{K}_{\text{id}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1}\text{id}_\ell)\mathbf{f}_1^* - r_{\theta,1}\mathbf{f}_2^* + \nu'_{\theta,4,1}\mathbf{f}_4^* + \nu'_{\theta,5,1}\mathbf{f}_5^* + \nu'_{\theta,6,1}\mathbf{f}_6^*} \right) \right. \right. \\ &\quad \left. \left. \right\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\ \left\{ \text{KU}_{\text{t}_\ell}^{(\text{SF})} &:= \left\{ \left(\theta, \mathbf{K}_{\text{t}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2}\text{t}_\ell)\mathbf{f}_1^* - r_{\theta,2}\mathbf{f}_3^* + \nu'_{\theta,4,2}\mathbf{f}_4^* + \nu'_{\theta,5,2}\mathbf{f}_5^* + \nu'_{\theta,6,2}\mathbf{f}_6^*} \right) \right. \right. \\ &\quad \left. \left. \right\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_\ell)} \right\}_{\ell \in [q_{n_2}]}. \end{aligned}$$

where

$$\begin{aligned} z'_1 &:= z - \chi_4\xi_{1,1} - \chi_5\xi_{2,1} - \chi_6\xi_{3,1}, \\ z'_2 &:= z\text{id}_{(\beta)}^* - \chi_4\xi_{1,2} - \chi_5\xi_{2,2} - \chi_6\xi_{3,2}, \\ z'_3 &:= z\text{t}_{(\beta)}^* - \chi_4\xi_{1,3} - \chi_5\xi_{2,3} - \chi_6\xi_{3,3} \end{aligned}$$

$$\left\{ \left\{ \begin{array}{l} \nu'_{\theta,4,1} := \nu_{\theta,4,1} + \xi_{1,1}(\alpha_{\theta,1} + r_{\theta,1}\text{id}_\ell) - r_{\theta,1}\xi_{1,2}, \\ \nu'_{\theta,5,1} := \nu_{\theta,5,1} + \xi_{2,1}(\alpha_{\theta,1} + r_{\theta,1}\text{id}_\ell) - r_{\theta,1}\xi_{2,2}, \\ \nu'_{\theta,6,1} := \nu_{\theta,6,1} + \xi_{3,1}(\alpha_{\theta,1} + r_{\theta,1}\text{id}_\ell) - r_{\theta,1}\xi_{3,2} \end{array} \right\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\
\left\{ \left\{ \begin{array}{l} \nu'_{\theta,4,2} := \nu_{\theta,4,2} + \xi_{1,1}(\alpha_{\theta,2} + r_{\theta,2}\text{t}_\ell) - r_{\theta,2}\xi_{1,3}, \\ \nu'_{\theta,5,2} := \nu_{\theta,5,2} + \xi_{2,1}(\alpha_{\theta,2} + r_{\theta,2}\text{t}_\ell) - r_{\theta,2}\xi_{2,3}, \\ \nu'_{\theta,6,2} := \nu_{\theta,6,2} + \xi_{3,1}(\alpha_{\theta,2} + r_{\theta,2}\text{t}_\ell) - r_{\theta,2}\xi_{3,3} \end{array} \right\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_\ell)} \right\}_{\ell \in [q_{n_2}]},$$

which are all uniformly distributed if (χ_4, χ_5, χ_6) defined in Equation 6.3 is a non-zero vector since

$$\begin{aligned}
& \{\xi_{ij}\}_{i \in [3], j \in [3]}, \\
& \left\{ \{\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1}\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\
& \left\{ \{\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2}\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_\ell)} \right\}_{\ell \in [q_{n_2}]}
\end{aligned}$$

are all uniformly picked from $\mathbb{Z}_q$.

In other words, the coefficients $(z, \text{id}_{(\beta)}^*, z\text{t}_{(\beta)}^*)$ of $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ in the $\mathbf{C}_0$ term of the challenge ciphertext is changed to random coefficients $(z'_1, z'_2, z'_3) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q$ of $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$ and a random time in $\mathbb{Z}_q$. Moreover, it is not different to check that all other coefficients are well distributed. Thus

$$(\text{GPK}, \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

expressed over bases $\mathbb{F}$ and $\mathbb{F}^*$ is properly distributed as

$$(\text{GPK}, \text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_\ell}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in $\text{Game}_{\text{Final}}$.

In the adversary's view, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same public key. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_ν over bases $(\mathbb{D}, \mathbb{D}^*)$

and in Game_{Final} over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, $\text{Game}_{q_{n_1}+q_{n_1}+1, 4N_{max}}$ and Game_{Final} are statistically indistinguishable. $\square$

Lemma 6.7. *For any adversary $\mathcal{A}$, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in Game_{Final} . Hence, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) = 0$. $\square$

In Game_{Final} , the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in $\mathbb{Z}_q$ and a random time in $\mathbb{Z}_q$, independent of the two messages, the challenge identities, and times provided by $\mathcal{A}$. Thus, our RIBE scheme is anonymous.

DLIN-based Construction We begin with our RIBE scheme:

- **Setup**(λ, N_{max}) This algorithm takes in a security parameter λ , a maximal number N_{max} of users and generate a symmetric bilinear pairing $\mathbb{G} := (q, G, G_T, g, e)$ for sufficiently large prime order q . Next perform the following steps:

1. Let RL be an empty set and BT be a binary-tree with at least N_{max} leaf nodes, set $\text{ST} = \text{BT}$.
2. Sample random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\text{R}} \text{Dual}(\mathbb{Z}_q^9)$. Let $\mathbf{d}_1, \dots, \mathbf{d}_9$ denote the elements of $\mathbb{D}$ and $\mathbf{d}_1^*, \dots, \mathbf{d}_9^*$ denote the elements of $\mathbb{D}^*$. It also picks $\alpha \leftarrow_{\text{R}} \mathbb{Z}_q$ and computes $g_T^\alpha := e(g, g)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$.
3. Output RL , ST , the global public key

$$\text{GPK} := \{ \mathbb{G}; g_T^\alpha, g^{\mathbf{d}_1}, \dots, g^{\mathbf{d}_6} \},$$

and the master secret key MSK

$$\text{MSK} := \{ \alpha, g^{\mathbf{d}_1^*}, \dots, g^{\mathbf{d}_6^*} \}.$$

- **PriKeyGen**(GPK, MSK, id, RL, ST) This algorithm takes in the global public key GPK, the master secret key MSK, an identity id, the revocation list RL, and the state ST, it picks an unassigned leaf node v from BT and stores id in that node. It then performs the following steps:

1. For any $\theta \in \text{Path}(v)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}, \alpha_{\theta,3}$ are undefined, then pick $\alpha_{\theta,1}, \alpha_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,1}, r_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{\text{id},\theta} := g^{(\alpha_{\theta,1}+r_{\theta,1}\text{id})\mathbf{d}_1^*-r_{\theta,1}\mathbf{d}_2^*+(\alpha_{\theta,3}+r_{\theta,3}\text{id})\mathbf{d}_4^*-r_{\theta,3}\mathbf{d}_5^*}.$$

2. Output $\text{SK}_{\text{id}} := \{(\theta, K_{\text{id},\theta})\}_{\theta \in \text{Path}(v)}$, ST.

- **KeyUpd**(GPK, MSK, t, RL, ST) This algorithm takes in the global public key GPK, the master secret key MSK, a time t, the revocation list RL, and the state ST, it performs the following steps:

1. $\forall \theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}, \alpha_{\theta,3}$ are undefined, then pick $\alpha_{\theta,1}, \alpha_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,2}, r_{\theta,4} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{t,\theta} := g^{(\alpha_{\theta,2}+r_{\theta,2}t)\mathbf{d}_1^*-r_{\theta,2}\mathbf{d}_3^*+(-\alpha_{\theta,3}+r_{\theta,4}t)\mathbf{d}_4^*-r_{\theta,4}\mathbf{d}_6^*}.$$

2. Output $\text{KU}_t := \{(t, \theta, K_{t,\theta})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)}$.

- **DecKeyGen**($\text{SK}_{\text{id}}, \text{KU}_t$) This algorithm takes in a private secret key $\text{SK}_{\text{id}} := \{(i, K_{\text{id},i})\}_{i \in \mathcal{I}}$, $\text{KU}_t := \{(j, K_{t,j})\}_{j \in \mathcal{J}}$ for some set of nodes $\mathcal{I}, \mathcal{J}$, it runs the following steps:

1. $\forall (i, K_{\text{id},i}) \in \text{SK}_{\text{id}}, (j, K_{t,j}) \in \text{KU}_t$, if $\exists (i, j)$ s.t. $i = j$ then $\text{DK}_{\text{id},t} \leftarrow (K_{\text{id},i}, K_{t,j})$; else (if SK_{id} and KU_t do not have any node in common) $\text{DK}_{\text{id},t} \leftarrow \perp$.
2. Output $\text{DK}_{\text{id},t}$.

- $\text{Enc}(\text{GPK}, \text{id}, t, \widehat{\text{msg}})$ This algorithm takes in the global public key GPK , an identity id , a time t , and a message $\widehat{\text{msg}}$, pick $z_1, z_2 \leftarrow_{\text{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\text{CT}_{\text{id},t} := \{C := \widehat{\text{msg}} \cdot (g_T^\alpha)^{z_1}, \quad C_0 := g^{z_1(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + t\mathbf{d}_3) + z_2(\mathbf{d}_4 + \text{id}\mathbf{d}_5 + t\mathbf{d}_6)}\}.$$

- $\text{Dec}(\text{GPK}, \text{DK}_{\text{id},t}, \text{CT}_{\text{id},t})$ This algorithm takes in the global public key GPK , a decryption key $\text{DK}_{\text{id},t} := (\text{K}_{\text{id},\theta}, \text{K}_{t,\theta})$, and a ciphertext $\text{CT}_{\text{id},t} := (C, C_0)$, it computes the message as

$$\widehat{\text{msg}} := C / (e(C_0, \text{K}_{\text{id},\theta}) \cdot e(C_0, \text{K}_{t,\theta})).$$

- $\text{KeyRev}(\text{id}, t, \text{RL}, \text{ST})$ This algorithm takes in an identity id , a time t , the revocation list RL , and the state ST , the algorithm adds (id, t) to RL for all nodes ν associated with identity id and returns RL .

Correctness is as following:

$$\begin{aligned} & e(C_0, \text{K}_{\text{id},\theta}) \\ &= e(g^{z_1(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + t\mathbf{d}_3) + z_2(\mathbf{d}_4 + \text{id}\mathbf{d}_5 + t\mathbf{d}_6)}, g^{(\alpha_{\theta,1} + r_{\theta,1}\text{id})\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^* + (\alpha_{\theta,3} + r_{\theta,3}\text{id})\mathbf{d}_4^* - r_{\theta,3}\mathbf{d}_5^*}) \\ &= e(g, g)^{\alpha_{\theta,1}z_1\mathbf{d}_1 \cdot \mathbf{d}_1^* + \alpha_{\theta,3}z_2\mathbf{d}_4 \cdot \mathbf{d}_4^*}. \end{aligned}$$

Similarly, $e(C_0, \text{K}_{t,\theta}) = e(g, g)^{\alpha_{\theta,2}z_1\mathbf{d}_1 \cdot \mathbf{d}_1^* - \alpha_{\theta,3}z_2\mathbf{d}_4 \cdot \mathbf{d}_4^*}$. Then

$$\begin{aligned} e(C_1, \text{K}_{\text{id},\theta}) \cdot e(C_1, \text{K}_{t,\theta}) &= e(g, g)^{\alpha_{\theta,1}z_1\mathbf{d}_1 \cdot \mathbf{d}_1^* + \alpha_{\theta,3}z_2\mathbf{d}_4 \cdot \mathbf{d}_4^*} \cdot e(g, g)^{\alpha_{\theta,2}z_1\mathbf{d}_1 \cdot \mathbf{d}_1^* - \alpha_{\theta,3}z_2\mathbf{d}_4 \cdot \mathbf{d}_4^*} \\ &= g_T^{\alpha_{\theta,1}z_1 + \alpha_{\theta,2}z_1} \\ &= (g_T^\alpha)^{z_1}. \end{aligned}$$

We show the RIBE scheme is secure by the following theorem, the proof techniques are essentially the same as those for Theorem 6.2 except that we use the DLIN-based Subspace assumption of [61].

Theorem 6.2. *The RIBE scheme is adaptively secure and anonymous under the DLIN assumption. More precisely, for any adversary $\mathcal{A}$ against the RIBE scheme, there exist probabilistic algorithms*

$$\begin{aligned} & \mathcal{B}_0, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=1, \dots, q_{n_1}, \kappa_2=1, \dots, \lceil \log N_{max} \rceil}, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=q_{n_1}+1, \dots, q_{n_1}+q_{n_2}+1, \kappa_2=1, \dots, N_{max}}, \\ & \{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}\}_{\kappa_2=1, \dots, 4N_{max}} \end{aligned}$$

whose running times are essentially the same as that of $\mathcal{A}$, such that

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) \leq & (q_{n_1} q_{n_2})^2 \cdot \left(\text{Adv}_{\mathcal{B}_0}^{\text{DLIN}}(\lambda) + \sum_{\kappa_1=1}^{q_{n_1}} \sum_{\kappa_2=1}^{\lceil \log N_{max} \rceil} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DLIN}}(\lambda) \right. \\ & + \sum_{\kappa_1=q_{n_1}+1}^{q_{n_2}} \sum_{\kappa_2=1}^{N_{max}} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DLIN}}(\lambda) + \sum_{\kappa_2=1}^{4N_{max}} \text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DLIN}}(\lambda) \\ & \left. + \frac{6(q_{n_1} \lceil \log N_{max} \rceil + q_{n_2} N_{max}) + 32N_{max} + 6}{q} \right) \end{aligned}$$

where $q_{n_1}, q_{n_2} \geq 4$ are the maximum number of $\mathcal{A}$'s private key and key update queries respectively.

Discussion In this work, we improved the previous work by achieving adaptive security with *constant-size global public key*. Moreover, our RIBE scheme is *anonymous*, which in turn, can be leveraged to construct a PKE-KS scheme supporting keyword revocation. Our constructions make use of the key revocation techniques of [19, 66]. Particularly, we employ binary-tree data structure to achieve key update with logarithmic complexity in the maximal number of users for the key authority.

We give a summary of comparisons between existing and our RIBE schemes in Table 6.1. Here, we use **GPK** to denote global public key, **SK** to denote private key, **KU** to denote key update, **CT** to denote ciphertext, and **#** pairing to denote

the number of pairing computation for decryption. The sizes are in terms of group elements and λ denotes the security parameter.

	BGK [19]	LV [66]	Ours	
size of GPK	5	$\mathcal{O}(\lambda)$	19	55
size of SK	2	3	6	9
size of KU	2	3	6	9
size of CT	4	5	6	9
# pairings	4	3	12	18
security	selective	adaptive	adaptive	adaptive
anonymity	No	No	Yes	Yes
assumption	DBDH	mDBDH	SXDH	DLIN

Table 6.1: Comparisons between existing and our RIBE schemes.

We compare our schemes against Boldyreva et al.’s scheme [19], which is under the DBDH assumption, and Libert and Vergnaud’s scheme [66], which is under the modified DBDH (mDBDH) assumption. Overall, our schemes are anonymous, adaptively secure, and have constant-size global public key, at the expense of bigger (but still seems acceptable) sizes in terms of the master secret key, private key, and key update.

6.3 Revocable IBE from Lattices

In this section, we give our RIBE scheme from lattices. If not explicitly specified, we assume that all vectors are column vectors in this section.

Motivation We note that all the previous RIBE schemes [19, 66] are constructed from bilinear pairings. In recent years, however, lattice-based IBE [52, 39, 2, 3] has received considerable attention from the cryptographic research community. Lattices have becoming an attractive and powerful tool to build a broad

range of cryptographic primitives [7, 68, 9, 83, 84, 50]. This is so as many lattice-based constructions are quite efficient and typically simple to implement. Moreover they are all believed to be secure against attacks using quantum computers, a property not achievable by cryptographic primitives based on factoring or discrete logarithm. In the spirit of expanding the study of lattice-based IBE, we also show, in this paper, how to construct an RIBE scheme in the lattice setting.

The Agrawal et al. IBE Scheme We use the Agrawal et al. lattice IBE scheme [2] as a building block for our construction. Briefly, their IBE scheme can be described as follows.

The global public key in the scheme of [2] consist of three random $n \times m$ matrices over $\mathbb{Z}_q$ denoted by $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as well as a vector $\mathbf{u} \in \mathbb{Z}_q^n$. The master secret is a trapdoor $\mathbf{T}_{\mathbf{A}}$ for the lattice $\Lambda_q^\perp(\mathbf{A})$. The secret key for an identity id is a short vector $\mathbf{e} \in \mathbb{Z}^{2m}$, which is generated using the **SampleLeft** algorithm of Theorem 4.2 and satisfies $\mathbf{F}_{\text{id}}\mathbf{e} = \mathbf{u}$ in $\mathbb{Z}_q$ where $\mathbf{F}_{\text{id}} := (\mathbf{A}|\mathbf{B} + \text{H}(\text{id})\mathbf{C}) \in \mathbb{Z}_q^{n \times 2m}$. In the security proof for a selective IBE security game, the adversary announces an identity id^* that it plans to attack. Instead of using a trapdoor for $\Lambda_q^\perp(\mathbf{A})$, it samples $\mathbf{C}$ at random and obtains a trapdoor $\mathbf{T}_{\mathbf{C}}$ for $\Lambda_q^\perp(\mathbf{C})$. It also chooses the global public key $\mathbf{A}$ at random and sets $\mathbf{B} := \mathbf{A}\mathbf{R} - \text{H}(\text{id}^*)\mathbf{C}$, where $\mathbf{R}$ is a random matrix in $\{1, -1\}^{m \times m}$. Since $\mathbf{A}\mathbf{R}$ is uniform and independent in $\mathbb{Z}_q^{n \times m}$, $\mathbf{B}$ is uniformly distributed as required. We then have

$$\mathbf{F}_{\text{id}} := (\mathbf{A}|\mathbf{A} \cdot \mathbf{R} + \mathbf{C}') \in \mathbb{Z}_q^{n \times 2m},$$

where $\mathbf{C}' := (\text{H}(\text{id}) - \text{H}(\text{id}^*))\mathbf{C}$. To respond to a private key query for an identity $\text{id} \neq \text{id}^*$, the simulator could produce a short vector $\mathbf{e}$ satisfying $\mathbf{F}_{\text{id}}\mathbf{e} = \mathbf{u}$ in $\mathbb{Z}_q$ by using the **SampleRight** algorithm of Theorem 4.3 and the basis $\mathbf{T}_{\mathbf{C}}$. This is so since $\text{id} \neq \text{id}^*$ is full rank by the definition of FRD in Section 4.6 and therefore $\mathbf{T}_{\mathbf{C}}$ is also a trapdoor for the lattice $\Lambda_q^\perp(\mathbf{C}')$. When $\text{id} = \text{id}^*$, the matrix $\mathbf{F}_{\text{id}}$ no longer depends on $\mathbf{C}$ and the simulator's trapdoor is removed. The simulator can then produce a challenge ciphertext that helps to solve the given LWE challenge.

Intuition Our construction of RIBE from lattices makes use of the following building blocks: (i) lattice IBE proposed by Agrawal, Boneh, and Boyen [2]; (ii) trapdoors for lattice IBE proposed by Gentry, Peikerty, and Vaikuntanathan [52]; and (iii) the binary-tree data structure for key update used in [6, 73, 19, 66]. More specifically, we extend the lattice IBE scheme of [2] with trapdoors from [52] to enable non-interactive key revocation. As with prior work, the binary-tree data structure is used to improve the efficiency of secret key update, allowing us to achieve key update with logarithmic complexity in the maximal number of users and linear in the number of revoked users for the key authority.

We note that our RIBE scheme is not a straightforward combination of the aforementioned building blocks because we require that our user public key comprises two components: identity and time, in order to obtain the “non-interactive” property. We first consider how to create a link between an identity and a time for each node. In our construction, we use two instances of Agrawal et al.’s IBE scheme and its techniques to deal with users’ identities and times respectively, but require only a single random vector $\mathbf{u} \in \mathbb{Z}_q^n$ in the public parameters. We split it into two random vectors $\mathbf{u}_1, \mathbf{u}_2$ for each node corresponding to identity and time, respectively. The randomly split $\mathbf{u}$ links identity to time for each node. Moreover, our technique does not require information about $\mathbf{u}_1, \mathbf{u}_2$ to be included in ciphertexts, and hence does not increase the size of ciphertexts.

Clearly, the simulator can answer all private key queries for all identities $\text{id} \neq \text{id}^*$, key update queries for all time $t \neq t^*$ by two trapdoors $\mathbf{T}_{\mathbf{C}_1}, \mathbf{T}_{\mathbf{C}_2}$. The main difficulty in the simulation is as follows. The simulator may be required to answer either a key update query at time t^* with node in $\text{Path}(\nu^*)$ or a private key query for identity id^* and a key update query at time t^* without any node in $\text{Path}(\nu^*)$, where id^* is assigned in ν^* (id^* must be revoked before or at time t^*). In other words, the simulator should answer either a query for identity id^* or a query for time t^* for each node. To overcome this difficulty, we use the **SampleGaussian** algorithm of Lemma 4.1 to sample a short vector and generate either $\mathbf{u}_1$ or $\mathbf{u}_2$

(instead of generating one of them randomly). Such $\mathbf{u}_1$ or $\mathbf{u}_2$ is indistinguishable from the uniform distribution, which is guaranteed by Lemma 4.2. More precisely, there are two possibilities for those nodes in $\text{Path}(\nu^*)$ (we can pick a node ν^* beforehand and assign id^* to it if necessary) depending on whether or not identity id^* will be queried:

- If identity id^* is queried, then it must be revoked before or at time t^* . In this case, we set $\mathbf{u}_1$ to be the product of $\mathbf{F}_{\text{id}^*}$ and a short vector $\mathbf{e}$ sampled by $\text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$.
- If identity id^* is not queried. In this case, we set $\mathbf{u}_2$ to be the product of $\mathbf{F}_{t^*}$ and a short vector $\mathbf{e}$ sampled by $\text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$.

For those nodes that are not in $\text{Path}(\nu^*)$, we set $\mathbf{u}_2$ to be the product of $\mathbf{F}_{t^*}$ and a short vector $\mathbf{e}$ sampled by $\text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$. We have probability $1/2$ to simulate the correct game and the adversary cannot distinguish which one is simulated.

Construction We now describe our RIBE scheme from lattices. At the end of each algorithm, we provide some intuition and/or remark (marked by the symbol “//”) about the algorithm.

- **Setup**($1^\lambda, N_{\max}$) This algorithm takes in a security parameter λ and a maximal number $N_{\max}$ of users, set the parameters q, n, m, σ, α as specified in Section 6.3 below. Next perform the following steps:
 1. Use the **TrapGen**(q, n) algorithm to select a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with a basis $\mathbf{T}_\mathbf{A}$ for $\Lambda_q^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{T}_\mathbf{A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.
 2. Select four uniformly random matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1$, and $\mathbf{C}_2$ in $\mathbb{Z}_q^{n \times m}$.
 3. Select a uniformly random vector $\mathbf{u} \leftarrow_{\mathbf{R}} \mathbb{Z}_q^n$.
 4. Let **RL** be an empty set and **BT** be a binary-tree with at least $N_{\max}$ leaf nodes, set $\mathbf{ST} := \mathbf{BT}$. Select an FRD map $\mathbf{H}$ as defined in Section 4.6.

5. Output RL , ST , the global public key, and the master secret key MSK ,

$$GPK := \{H, \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2, \mathbf{u}\}, \quad MSK := \{\mathbf{T}_A\}.$$

- **PriKeyGen**(GPK, MSK, id, RL, ST) This algorithm takes in the global public key GPK , the master secret key MSK , an identity $id \in \mathbb{Z}_q^n$, the revocation list RL , and the state ST , it picks an unassigned leaf node ν from BT and stores id in that node. It then performs the following steps:

1. For any $\theta \in \text{Path}(\nu)$, if $\mathbf{u}_{\theta,1}, \mathbf{u}_{\theta,2}$ are undefined, then pick $\mathbf{u}_{\theta,1} \leftarrow_{\mathbf{r}} \mathbb{Z}_q^n$, set $\mathbf{u}_{\theta,2} := \mathbf{u} - \mathbf{u}_{\theta,1}$, and store them in node θ . Sample $\mathbf{e}_{\theta,1} \in \mathbb{Z}^{2m}$ as $\mathbf{e}_{\theta,1} \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B}_1 + H(id)\mathbf{C}_1, \mathbf{T}_A, \mathbf{u}_{\theta,1}, \sigma)$.
2. Output $SK_{id} := \{(\theta, \mathbf{e}_{\theta,1})\}_{\theta \in \text{Path}(\nu)}, ST$.

//The algorithm computes the id -component of the decryption key for all the nodes on the path from ν to **root**.

- **KeyUpd**(GPK, MSK, t, RL, ST) This algorithm takes in the global public key GPK , the master secret key MSK , a time $t \in \mathbb{Z}_q^n$, the revocation list RL , and the state ST , it performs the following steps:

1. $\forall \theta \in \text{KUNodes}(BT, RL, t)$, if $\mathbf{u}_{\theta,1}, \mathbf{u}_{\theta,2}$ are undefined, then pick $\mathbf{u}_{\theta,1} \leftarrow_{\mathbf{r}} \mathbb{Z}_q^n$, set $\mathbf{u}_{\theta,2} := \mathbf{u} - \mathbf{u}_{\theta,1}$, and store them in node θ . Sample $\mathbf{e}_{\theta,2} \in \mathbb{Z}^{2m}$ as $\mathbf{e}_{\theta,2} \leftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B}_2 + H(t)\mathbf{C}_2, \mathbf{T}_A, \mathbf{u}_{\theta,2}, \sigma)$.
2. Output $KU_t := \{(\theta, \mathbf{e}_{\theta,2})\}_{\theta \in \text{KUNodes}(BT, RL, t)}$.

//The algorithm first finds a minimal set of nodes which contains an ancestor (or, the node itself) of all the non-revoked nodes. It then computes the t -component of the decryption key for all the nodes in that set.

- **DecKeyGen**(SK_{id}, KU_t) This algorithm takes in a private secret key $SK_{id} := \{(i, \mathbf{e}_{i,1})\}_{i \in I}$, $KU_t := \{(j, \mathbf{e}_{j,2})\}_{j \in J}$ for some set of nodes I, J , it runs the following steps:

1. $\forall (i, \mathbf{e}_{i,1}) \in \mathbf{SK}_{\text{id}}, (j, \mathbf{e}_{j,2}) \in \mathbf{KU}_t$, if $\exists (i, j)$ s.t. $i = j$ then $\mathbf{DK}_{\text{id},t} \leftarrow (\mathbf{e}_{i,1}, \mathbf{e}_{j,2})$; else (if $\mathbf{SK}_{\text{id}}$ and $\mathbf{KU}_t$ do not have any node in common) $\mathbf{DK}_{\text{id},t} \leftarrow \perp$.
2. Output $\mathbf{DK}_{\text{id},t}$.

// We can drop the subscripts i, j since they are equal, i.e., $\mathbf{DK}_{\text{id},t} := (\mathbf{e}_1, \mathbf{e}_2)$. The algorithm finds components of $\mathbf{SK}_{\text{id}}$ and $\mathbf{KU}_t$ such that $\mathbf{F}_{\text{id}}\mathbf{e}_1 + \mathbf{F}_t\mathbf{e}_2 = \mathbf{u}$ since they are in the same node.

- **Enc(GPK, id, t, $\widehat{\text{msg}}$)** This algorithm takes in the global public key **GPK**, an identity $\text{id} \in \mathbb{Z}_q^n$, a time $t \in \mathbb{Z}_q^n$, and a message $\widehat{\text{msg}} \in \{0, 1\}$, it runs the following steps:

1. Set $\mathbf{F}_{\text{id},t} \leftarrow (\mathbf{A}|\mathbf{B}_1 + \mathbf{H}(\text{id})\mathbf{C}_1|\mathbf{B}_2 + \mathbf{H}(t)\mathbf{C}_2) \in \mathbb{Z}_q^{n \times 3m}$.
2. Choose a uniformly random $\mathbf{s} \leftarrow_{\text{R}} \mathbb{Z}_q^n$.
3. For $i = 1, 2$, choose a uniformly random matrix $\mathbf{R}_i \leftarrow_{\text{R}} \{-1, 1\}^{m \times m}$.
4. Choose noise $x \leftarrow_{\text{R}} \overline{\Psi}_\alpha$ and noise vectors $\mathbf{y} \leftarrow_{\text{R}} \overline{\Psi}_\alpha^m$ and for $i = 1, 2$ set $\mathbf{z}_i \leftarrow \mathbf{R}_i^\top \mathbf{y} \in \mathbb{Z}_q^m$. (The distribution $\overline{\Psi}_\alpha$ is as defined by Definition 4.3)
5. Set $c_0 \leftarrow \mathbf{u}^\top \mathbf{s} + x + \widehat{\text{msg}} \cdot \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$, $\mathbf{c}_1 \leftarrow \mathbf{F}_{\text{id},t}^\top \mathbf{s} + \begin{bmatrix} \mathbf{y} \\ \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
6. Output the ciphertext $\mathbf{CT}_{\text{id},t} := (c_0, \mathbf{c}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m}$.

- **Dec(GPK, $\mathbf{DK}_{\text{id},t}$, $\mathbf{CT}_{\text{id},t}$)** This algorithm takes in the global public key **GPK**, a decryption key $\mathbf{DK}_{\text{id},t} := (\mathbf{e}_1, \mathbf{e}_2)$, and a ciphertext $\mathbf{CT}_{\text{id},t} := (c_0, \mathbf{c}_1)$, it runs the following steps:

1. Parse $\mathbf{c}_1$ as $\begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \\ \mathbf{c}_{1,2} \end{bmatrix}$, where $\mathbf{c}_{1,i} \in \mathbb{Z}_q^m$.

2. Compute $w \leftarrow c_0 - \mathbf{e}_1^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \end{bmatrix} - \mathbf{e}_2^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,2} \end{bmatrix} \in \mathbb{Z}_q$.
 3. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $|w - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$, output 1, otherwise output 0.
- **KeyRev(id, t, RL, ST)** This algorithm takes in an identity id , a time t , the revocation list RL , and the state ST , the algorithm adds (id, t) to RL for all nodes ν associated with identity id and returns RL .

Parameters and Correctness As in [2], the following error term is bounded by $[q\sigma m\alpha\omega(\sqrt{\log m}) + \mathcal{O}(\sigma m^{3/2})]$, that is

$$w = c_0 - \mathbf{e}_1^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \end{bmatrix} - \mathbf{e}_2^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,2} \end{bmatrix} = \widehat{\text{msg}}\lfloor \frac{q}{2} \rfloor + \underbrace{x - \mathbf{e}_1^\top \begin{bmatrix} \mathbf{y} \\ \mathbf{z}_1 \end{bmatrix} - \mathbf{e}_2^\top \begin{bmatrix} \mathbf{y} \\ \mathbf{z}_2 \end{bmatrix}}_{\text{error term}}.$$

We can similarly set the parameters (q, m, σ, α) to ensure that the error term is less than $q/5$ and the system works:

$$\begin{aligned} m &= 2n^{1+\delta}, & q &= m^2\sqrt{n} \cdot \omega(\log n), \\ \sigma &= m \cdot \omega(\log n), & \alpha &= [m^2 \cdot \omega(\log n)]^{-1}, \end{aligned}$$

and round up m to the nearest larger integer and q to the nearest larger prime. We choose δ such that $n^\delta > \lceil \log q \rceil = \mathcal{O}(\log n)$.

Proof of Security We show that our RIBE construction is secure in the following theorem:

Theorem 6.3. *The RIBE system is selective-ID secure under the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$ -LWE assumption.*

Proof. Our proof proceeds in a sequence of games where the first game is identical to the selective-ID secure game from Definition 2.9 and the adversary has no

advantage in the last game. We show that a PPT adversary cannot distinguish between the games. This proves that the adversary has negligible advantage in winning the original selective-ID secure game.

To simplify reduction, we divide adversaries into two types:

- Type I adversaries choose to be challenged on the target identity id^* but which is revoked before or on time t^* .
- Type II adversaries do not challenge the target identity id^* at any time.

We pick a random bit $rev \leftarrow_{\mathcal{R}} \{0, 1\}$ as a guess for the type of adversarial behavior that it will be faced with. All the sequence of games for these two types of adversaries are similar except for **Game**₂. We will show that the adversary cannot distinguish which type of challenger simulated and thus we have probability $1/2$ to simulate the correct game.

- **Game**₀: This is the original selective-ID secure game from Section 2.3.
- **Game**₁: In **Game**₁ we change the way that the challenger generates $\mathbf{B}_1, \mathbf{B}_2$ in the global public key. Let id^* be the identity and t^* be the time that $\mathcal{A}$ intends to attack. The **Game**₁ challenger chooses $\mathbf{R}_1^*, \mathbf{R}_2^*$ at the setup phase and constructs $\mathbf{B}_1, \mathbf{B}_2$ as

$$\mathbf{B}_1 \leftarrow \mathbf{A}\mathbf{R}_1^* - \text{H}(\text{id}^*)\mathbf{C}_1 \quad (6.11)$$

$$\mathbf{B}_2 \leftarrow \mathbf{A}\mathbf{R}_2^* - \text{H}(t^*)\mathbf{C}_2 \quad (6.12)$$

The remainder of the game is unchanged.

We now show that **Game**₀ is statistically indistinguishable from **Game**₁. Observe that in **Game**₁ the matrices $\mathbf{R}_1^*, \mathbf{R}_2^*$ are used only in the construction of $\mathbf{B}_1, \mathbf{B}_2$ and in the construction of the challenge ciphertext where $\mathbf{z}_1 \leftarrow (\mathbf{R}_1^*)^\top \mathbf{y}$, $\mathbf{z}_2 \leftarrow (\mathbf{R}_2^*)^\top \mathbf{y}$. By Lemma 4.3 the distribution $(\mathbf{A}, \mathbf{A}\mathbf{R}_1^*, \mathbf{z}_1)$ and $(\mathbf{A}, \mathbf{A}\mathbf{R}_2^*, \mathbf{z}_2)$ are statistically close to the distribution $(\mathbf{A}, \mathbf{B}'_1, \mathbf{z}_1)$ and $(\mathbf{A}, \mathbf{B}'_2, \mathbf{z}_2)$ respectively, where $\mathbf{B}'_1, \mathbf{B}'_2$ are uniform $\mathbb{Z}_q^{n \times m}$ matrices. It follows that in the adversary's view, the matrices $\mathbf{A}\mathbf{R}_1^*$ and $\mathbf{A}\mathbf{R}_2^*$ are statistically

close to uniform and therefore $\mathbf{B}_1$ and $\mathbf{B}_2$ as defined in the above equation (6.11)(6.12) are close to uniform. Hence, $\mathbf{B}_1, \mathbf{B}_2$ in Game_0 and Game_1 are indistinguishable.

- **Game₂**: In Game_2 we change the way that $\mathbf{u}_{\theta,1}, \mathbf{u}_{\theta,2}$ for each node are chosen and the challenger generates short vectors for private key query of id^* and update key query of t^* as follows (WLOG, we can pick a node ν^* from BT beforehand, to which id^* may be assigned):

- If $\text{rev} = 0$, we simulate a game to face Type I adversaries. We generate $\mathbf{u}_{\theta,1}$ and $\mathbf{u}_{\theta,2}$ for every node θ in BT as follows:

- * if $\theta \in \text{Path}(\nu^*)$, then $\mathbf{e}_{\theta,1} \leftarrow \text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$. Set $\mathbf{u}_{\theta,1} \leftarrow \mathbf{F}_{\text{id}} \cdot \mathbf{e}_{\theta,1}$ and $\mathbf{u}_{\theta,2} \leftarrow \mathbf{u} - \mathbf{u}_{\theta,1}$. Store them in the node θ .
- * if $\theta \notin \text{Path}(\nu^*)$, then $\mathbf{e}_{\theta,2} \leftarrow \text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$. Set $\mathbf{u}_{\theta,2} \leftarrow \mathbf{F}_{\text{t}} \cdot \mathbf{e}_{\theta,2}$ and $\mathbf{u}_{\theta,1} \leftarrow \mathbf{u} - \mathbf{u}_{\theta,2}$. Store them in the node θ .

Since id^* has been revoked before the update key query for t^* is queried, we have $\text{KUNodes}(\text{BT}, \text{RL}, \text{t}^*) \cap \text{Path}(\nu^*) = \emptyset$. Then the challenger responds a private key query for id^* by using $\{(\theta, \mathbf{e}_{\theta,1})\}_{\theta \in \text{Path}(\nu^*)}$ and a update key query for t^* by using $\{(\theta, \mathbf{e}_{\theta,2})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}^*)}$.

- If $\text{rev} = 1$, we simulate a game to face Type II adversaries. We generate $\mathbf{u}_{\theta,1}$ and $\mathbf{u}_{\theta,2}$ for every node θ in BT as follows:

- * $\mathbf{e}_{\theta,2} \leftarrow \text{SampleGaussian}(\mathbf{B}_z, \sigma, 0)$. Set $\mathbf{u}_{\theta,2} \leftarrow \mathbf{F}_{\text{t}} \mathbf{e}_{\theta,2}$ and $\mathbf{u}_{\theta,1} \leftarrow \mathbf{u} - \mathbf{u}_{\theta,2}$. Store them in the node θ .

Since id^* is never queried, the challenger responds a update key query for t^* by using $\{(\theta, \mathbf{e}_{\theta,2})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}^*)}$.

From Lemma 4.1, all above $\mathbf{e}_{\theta,i}$ ($i = 1, 2$) are sampled from $\mathcal{D}_{\mathbb{Z}^m, \sigma}$, which is statistically close, given $\mathbf{u}_{\theta,i}$, to the distribution of $\mathcal{D}_{\Lambda_q^{\mathbf{u}_{\theta,i}}}$ ($i = 1, 2$) in the real scheme. In particular, we have $\mathbf{e}_{\theta,1} \in \mathcal{D}_{\Lambda_q^{\mathbf{u}_{\theta,1}}(\mathbf{F}_{\text{id}^*}), \sigma}$ and $\mathbf{e}_{\theta,2} \in \mathcal{D}_{\Lambda_q^{\mathbf{u}_{\theta,2}}(\mathbf{F}_{\text{t}^*}), \sigma}$.

Since $\mathbf{F}_{\text{id}^*}$ and $\mathbf{F}_{\text{t}^*}$ could be viewed as random matrices from $\mathbb{Z}_q^{n \times 2m}$, we argue that $\mathbf{u}_{\theta,1}$ or $\mathbf{u}_{\theta,2}$ is statistically close to uniform over $\mathbb{Z}_q^n$ from Lemma 4.2. Hence, the adversary cannot distinguish which type of challenger simulated and we have probability $1/2$ to simulate the correct game. Therefore, Game_1 and Game_2 are indistinguishable if the correct game is simulated.

- **Game₃**: We now change how $\mathbf{A}$, $\mathbf{C}_1$, $\mathbf{C}_2$ in GPK are chosen. In **Game₃** we generate $\mathbf{A}$ as random matrix in $\mathbb{Z}_q^{n \times m}$, but generate $\mathbf{C}_1, \mathbf{C}_2$ using algorithm **TrapGen** so that $\mathbf{C}_1, \mathbf{C}_2$ is a random matrix in $\mathbb{Z}_q^{n \times m}$, but the challenger has a trapdoor $\mathbf{T}_{\mathbf{C}_1}, \mathbf{T}_{\mathbf{C}_2}$ for $\Lambda_q^\perp(\mathbf{C}_1), \Lambda_q^\perp(\mathbf{C}_2)$. The choice of $\mathbf{B}_1, \mathbf{B}_2$ remains as in **Game₁** and **Game₂**, i.e. $\mathbf{B}_1 = \mathbf{A} \cdot \mathbf{R}_1^* - \text{H}(\text{id}^*)\mathbf{C}_1$, $\mathbf{B}_2 = \mathbf{A} \cdot \mathbf{R}_2^* - \text{H}(\text{t}^*)\mathbf{C}_2$. To respond to a private key query for $\text{id} \neq \text{id}^*$ and key update query $\text{t} \neq \text{t}^*$, we will use the trapdoor $\mathbf{T}_{\mathbf{C}_1}$ and $\mathbf{T}_{\mathbf{C}_2}$ instead of $\mathbf{T}_{\mathbf{A}}$, respectively. Let

$$\mathbf{F}_{\text{id}} := (\mathbf{A} | \mathbf{B}_1 + \text{H}(\text{id})\mathbf{C}_1) = (\mathbf{A} | \mathbf{A}\mathbf{R}_1^* + (\text{H}(\text{id}) - \text{H}(\text{id}^*))\mathbf{C}_1),$$

$$\mathbf{F}_{\text{t}} := (\mathbf{A} | \mathbf{B}_2 + \text{H}(\text{t})\mathbf{C}_2) = (\mathbf{A} | \mathbf{A}\mathbf{R}_2^* + (\text{H}(\text{t}) - \text{H}(\text{t}^*))\mathbf{C}_2),$$

by the definition of FRD in Section 4.6, $[\mathbf{H}(\text{id}) - \mathbf{H}(\text{id}^*)]$ and $[\mathbf{H}(\text{t}) - \mathbf{H}(\text{t}^*)]$ are non-singular. Therefore, $\mathbf{T}_{\mathbf{C}_1}$ and $\mathbf{T}_{\mathbf{C}_2}$ are also trapdoors for $\Lambda_q^\perp(\mathbf{C}'_1)$ and $\Lambda_q^\perp(\mathbf{C}'_2)$ respectively, where $\mathbf{C}'_1 := (\text{H}(\text{id}) - \text{H}(\text{id}^*))\mathbf{C}_1$ and $\mathbf{C}'_2 := (\text{H}(\text{t}) - \text{H}(\text{t}^*))\mathbf{C}_2$. Moreover, since $\mathbf{C}_1, \mathbf{C}_2$ are rank n , so are $\mathbf{C}'_1$ and $\mathbf{C}'_2$. The challenger can now respond to all private key queries for $\text{id} \neq \text{id}^*$ by running

$$\mathbf{e}_{\theta,1} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{C}'_1, \mathbf{R}_1^*, \mathbf{T}_{\mathbf{C}_1}, \mathbf{u}_{\theta,1}, \sigma) \in \mathbb{Z}_q^{2m},$$

and key update queries for $\text{t} \neq \text{t}^*$ by running

$$\mathbf{e}_{\theta,2} \leftarrow \text{SampleRight}(\mathbf{A}, \mathbf{C}'_2, \mathbf{R}_2^*, \mathbf{T}_{\mathbf{C}_2}, \mathbf{u}_{\theta,2}, \sigma) \in \mathbb{Z}_q^{2m}.$$

Since the σ used in the system is sufficiently large, these $\mathbf{e}_{\theta,1}$ and $\mathbf{e}_{\theta,2}$ are distributed close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}_{\theta,1}}(\mathbf{F}_{\text{id}}), \sigma}$ and $\mathcal{D}_{\Lambda_q^{\mathbf{u}_{\theta,2}}(\mathbf{F}_{\text{t}}), \sigma}$ as in **Game₂** by Theorem 4.3.

The challenger responds to private key query for id^* and key update query for t^* the same as in Game_2 rather than being sampled with SampleRight , where these $\mathbf{e}_{\theta,1}$ and $\mathbf{e}_{\theta,2}$ have been already defined in Game_2 .

Game_3 is otherwise the same as Game_2 . Since $\mathbf{A}$, $\mathbf{C}_1$, $\mathbf{C}_2$ and responses to private key and key update queries are statistically close to those in Game_2 , the adversary's advantage in Game_3 is at most negligibly different from its advantage in Game_2 .

- **Game₄:** This game is identical to Game_3 except that the challenge ciphertext $(c_0^*, \mathbf{c}_1^*)$ is always chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{3m}$. Since the challenge ciphertext is always a fresh random element in the ciphertext space, $\mathcal{A}$'s advantage in this game is zero.

It remains to show that Game_3 and Game_4 are computationally indistinguishable for a PPT adversary, which we do by giving a reduction from the LWE problem.

Reduction from LWE. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing Game_3 and Game_4 . We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$. Recall from Definition 4.5 that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_s$ or a noisy pseudo-random $\mathcal{O}_s$ for some secret $\mathbf{s} \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between the two, and proceeds as follows:

Instance: $\mathcal{B}$ requests from $\mathcal{O}$ and receives, for each $i = 0, \dots, m$, a fresh pair $(\mathbf{u}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Initial: $\mathcal{A}$ announces to $\mathcal{B}$ the identity id^* and time t^* that it intends to attack.

Setup: $\mathcal{B}$ constructs the system's global public key **GPK** as follows:

1. Assemble the random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ from m of the previously given LWE samples by letting the j -th column of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be the

n -vector $\mathbf{u}_j$ for all $j = 1, \dots, m$.

2. Assign the zeroth LWE sample (so far unused) to become the public random n -vector $\mathbf{u}_0 \in \mathbb{Z}_q^n$.
3. The remainder of the global public key, namely $\mathbf{B}_1$, $\mathbf{C}_1$, $\mathbf{B}_2$, and $\mathbf{C}_2$, are constructed as in **Game₃** using id^* , $\mathbf{R}_1^*$, $\mathbf{t}^*$, and $\mathbf{R}_2^*$.

Query: $\mathcal{B}$ answers each private key and key update queries as in **Game₃**.

Challenge: $\mathcal{B}$ prepares, when prompted by $\mathcal{A}$ with a message bit $\widehat{\text{msg}}^* \in \{0, 1\}$, a challenge ciphertext for the target identity id^* , as follows:

1. Let $v_0, \dots, v_m$ be entries from the LWE instance.

$$\text{Set } \mathbf{v}^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m.$$

2. Blind the message bit by letting $c_0^* = v_0 + \widehat{\text{msg}}^* \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$

$$3. \text{ Set } \mathbf{c}_1^* = \begin{bmatrix} \mathbf{v}^* \\ (\mathbf{R}_1^*)^\top \mathbf{v}^* \\ (\mathbf{R}_2^*)^\top \mathbf{v}^* \end{bmatrix} \in \mathbb{Z}_q^{3m}.$$

4. Choose a random bit $r \leftarrow_{\text{R}} \{0, 1\}$. If $r = 0$ send $(c_0^*, \mathbf{c}_1^*)$ to the adversary. If $r = 1$ choose a random $(c_0, \mathbf{c}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m}$ and send $(c_0, \mathbf{c}_1)$ to the adversary.

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then $(c_0^*, \mathbf{c}_1^*)$ is distributed exactly as in **Game₃**. First, observe that $\mathbf{F}_{\text{id}^*} = (\mathbf{A} | \mathbf{A}\mathbf{R}_1^*)$ and $\mathbf{F}_{\mathbf{t}^*} = (\mathbf{A} | \mathbf{A}\mathbf{R}_2^*)$. Second, by definition of $\mathcal{O}_s$ we know that $\mathbf{v}^* = \mathbf{A}^\top \mathbf{s} + \mathbf{y}$ for some random noise vector $\mathbf{y} \in \mathbb{Z}_q^m$ distributed as $\bar{\Psi}_\alpha^m$.

Therefore, $\mathbf{c}_1^*$ defined in step (3) above satisfies

$$\begin{aligned} \mathbf{c}_1^* &= \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + \mathbf{y} \\ (\mathbf{R}_1^*)^\top \mathbf{A}^\top \mathbf{s} + (\mathbf{R}_1^*)^\top \mathbf{y} \\ (\mathbf{R}_2^*)^\top \mathbf{A}^\top \mathbf{s} + (\mathbf{R}_2^*)^\top \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + \mathbf{y} \\ (\mathbf{A}\mathbf{R}_1^*)^\top \mathbf{s} + (\mathbf{R}_1^*)^\top \mathbf{y} \\ (\mathbf{A}\mathbf{R}_2^*)^\top \mathbf{s} + (\mathbf{R}_2^*)^\top \mathbf{y} \end{bmatrix} \\ &= (\mathbf{F}_{\text{id}^*, t^*})^\top \mathbf{s} + \begin{bmatrix} \mathbf{y} \\ (\mathbf{R}_1^*)^\top \mathbf{y} \\ (\mathbf{R}_2^*)^\top \mathbf{y} \end{bmatrix} \end{aligned}$$

and the quantity on the right is precisely the id_1 part of a valid challenge ciphertext in **Game**₃. Also note that $v_0 = \mathbf{u}_0^\top \mathbf{s} + x$, just as the c_0 part of the challenge ciphertext in **Game**₃.

When $\mathcal{O} = \mathcal{O}_s$ we have that v_0 is uniform in $\mathbb{Z}_q$ and $\mathbf{v}^*$ are uniform in $\mathbb{Z}_q^m$. Therefore $\mathbf{c}_1^*$ as defined in step (3) above is uniform and independent in $\mathbb{Z}_q^{3m}$ by the standard left over hash lemma (e.g. Theorem 8.38 of [89]). Hence, the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{3m}$, as in **Game**₄.

Guess: After being allowed to make additional queries, $\mathcal{A}$ guesses if it is interacting with a **Game**₃ or **Game**₄ challenger. Our simulator outputs $\mathcal{A}$'s guess as the answer to the LWE challenge it is trying to solve.

We already argued that when $\mathcal{O} = \mathcal{O}_s$ the adversary's view is as in **Game**₃. When $\mathcal{O} = \mathcal{O}_\S$ the adversary's view is as in **Game**₄. Hence, $\mathcal{B}$'s advantage in solving LWE is the same as $\mathcal{A}$'s advantage in distinguishing **Game**₃ and **Game**₄, as required.

This completes the description of algorithm $\mathcal{B}$ and completes the proof. $\square$

Chapter 7

The Relation and Transformation between HIPE and SE

Contents

7.1	Motivation	132
7.2	Concepts from Linear Algebra	135
7.3	Transformation from HIPE to SE	136
7.4	Transformation from SE to HIPE	141
7.5	Working in $\mathbb{Z}_N$	146
7.6	Discussion	147

In this chapter, we study the relation and transformation between HIPE and SE. We first derive some lemmas, which are essential for our forthcoming property-preserving transformation. Subsequently, we discover the techniques for constructing an SE scheme from an HIPE scheme, and vice versa that work in $\mathbb{Z}_q$ for some prime q (e.g. prime order bilinear groups). We also explain how an SE or HIPE scheme is defined and why our transformation techniques also work in $\mathbb{Z}_N$ for some hard factoring composite number N (e.g. composite order bilinear groups).

Figure 7.1: Partial Relations in FE.

7.1 Motivation

As illustrated in [58, 23], HIPE and SE are two important classes of FE that have numerous and different applications.

From an IPE scheme, one can obtain an IBE scheme and a Hidden-Vector Encryption (HVE) scheme. Moreover, we get other variants of Public-Key Encryption schemes supporting Polynomial Evaluation (PKE-PE), Equality Queries (PKE-EQ, which is also called Keyword Search [28]), Subset Queries (PKE-SQ), Comparison Queries (PKE-CQ), Conjunctive Normal Form (PKE-CNF), and Disjunctive Normal Form (PKE-DNF) [25, 58]. By making use of the delegation mechanism in HIPE, the applications of IPE can be broaden and generalized to cover other cryptographic primitives such as Hierarchical IBE (HIBE), Generalized Keyword Search (GPKE-EQ) [1, 75], and delegatable HVE [88]. Indeed, (H)IPE is seen as a very powerful tool or building block for constructing a broad

range of PKE schemes. This is summarized in Figure 7.1.

On the other hand, SE is a very generalized encryption system in the sense that many other types of IBE [23], such as HIBE, Inclusive IBE (IIBE), Co-Inclusive IBE (CIIBE), Broadcast HIBE (BHIBE) and Forward-Secure IBE (FSIBE), are embedded in it. Moreover, we can construct a “product” scheme by embedding multiple instances of SE to obtain additional system properties. For example, an SE scheme can be extended to support multiple authorities by taking the product of the SE scheme and a broadcast scheme whose identities are the names of authorities. See [23, 55] for more embeddings of other cryptosystems into SE.

Although HIPE and SE are related in the sense that they both are instantiations of FE, the former works in vectors while the latter is based on spaces. We notice that existing HIPE schemes [63, 76, 77] possess some properties that are complementary to the SE scheme [23], and vice versa. For example, the HIPE schemes are fully secure and attribute-hiding (anonymous) but do not have short ciphertexts, while the SE scheme has short ciphertexts but is only payload-hiding (non-anonymous) and proven secured in the selective security model. Note that the main tool for constructing an HIPE scheme is DPVS [74, 75], which has very natural orthogonal structure by randomly generating a pair of orthonormal bases. However, one limitation of randomly generated a pair of orthonormal bases is that the size of each element of DPVS is $\mathcal{O}(n)$, where n is the dimension of the relevant vector space. Hence, the resulting ciphertexts, such as those in previous works [63, 76, 77], are at least $\mathcal{O}(n)$. Okamoto and Takashima [78] recently showed how to reduce the size of ciphertexts by randomly generating a pair of orthonormal bases with special form, such that the resulting HIPE scheme has constant-size ciphertexts. However, the size of ciphertexts is still relatively large in comparison with that of the Boneh-Hamburg SE scheme. Meanwhile, the security of the Boneh-Hamburg SE scheme is analyzed based on the conventional partitioning techniques [29, 30]. It seems difficult to construct an attribute-hiding SE scheme that enjoys full security proven using the partitioning strategy [92].

The main motivation for this work is to understand the relation between HIPE and SE. Particularly, we are looking for an efficient “black-box” or generic construction of SE from HIPE and vice versa. Here, we first observe that there is a simple connection between SE and (non-hierarchical) IPE. It is indeed not difficult to transform an SE scheme to an IPE scheme. We note that the condition $\mathbf{x} \cdot \mathbf{v} = 0$ in IPE holds iff $\mathbf{x} \in S^\perp(\mathbf{v})$, which can be viewed as $\hat{F}(S^\perp(\mathbf{v}), (\mathbf{x}, \widehat{\text{msg}})) = \widehat{\text{msg}}$ in the SE setting. In other words, to generate a secret key associated with a vector $\mathbf{v} \in \mathcal{K}$ in an IPE scheme, we can run the key generation algorithm of an SE scheme for the space $S^\perp(\mathbf{v})$. Attrapadung and Libert [12] showed that an IPE scheme can be constructed from an SE scheme in a similar way. They obtained a selective secure, payload-hiding IPE scheme with short ciphertexts from the SE scheme of [23]. This, in turn, implies that given an SE scheme, we can derive an IPE scheme and a broad range of public-key encryption schemes with properties similar to those of the SE scheme.

Although HIPE and SE are conceptually similar, namely they both involve some notions of linear algebra, their relation from a technical view point is less obvious. It is not at all trivial and clear how one could construct an HIPE scheme from an SE scheme, and vice versa. This is so because of the following reason. In HIPE, $\mathbf{x}_i \cdot \mathbf{v}_i = 0 \Leftrightarrow \mathbf{x}_i \in S^\perp(\mathbf{v}_i)$ for all $i \in [\ell_s]$. We require that these orthogonal (Euclidean) spaces be independent from each other. However, in the SE setting, we make use of relations between (affine) spaces and their (affine) subspaces to allow key delegation. This seems to be a contradicting requirement from that for HIPE. Hence, there is no obvious way to directly construct an SE scheme from an HIPE scheme. Nevertheless, it seems more feasible to construct the latter from the former. One possible and natural way to do that is by making use of multiple SE schemes to construct an HIPE scheme such that each SE scheme generates a secret key for an orthogonal space associated with a specific level of a hierarchy. However, such transformation is complicated, inefficient and the security of key delegation is not guaranteed. Furthermore, it is likely that such transformation

will not lead to generic construction of HIPE.

In our work, we discover some tricks from linear algebra on which our transformation techniques are based to overcome the aforementioned challenges.

7.2 Concepts from Linear Algebra

We first derive some lemmas associated with vectors, matrices, and linear spaces, and which are essential for our property-preserving transformation between HIPE and SE.

Lemma 7.1. *Given a space S , then $\dim(S) + \dim(S^\perp) = n$ and $(S^\perp)^\perp = S$.*

The proof for the above lemma is obvious and is omitted from here.

Lemma 7.2. *Given a space S , there exists a polynomial-time algorithm taking as input S and outputting a basis $B^\perp$ of $S^\perp$.*

Proof. Let the rows of a matrix $\mathbf{M}$ be formed by the row vectors which span S . Then we can use an algorithm (such as Algorithm 2.3.1 of [41]) that takes as input $\mathbf{M}$ and outputs a basis of the kernel of $\mathbf{M}$. This basis is also a basis of $S^\perp$. $\square$

Henceforth, we refer to the algorithm described in Lemma 7.2 as the BasisGen algorithm. We will later use it to transform an element in $\mathcal{K}_{\text{HIPE}}$ to an element in $\mathcal{K}_{\text{SE}}$, and vice versa.

Lemma 7.3. *Given a space S , a subspace S' and a basis $B^\perp$ of $S^\perp$, there exists a polynomial-time algorithm taking as input $S, S', B^\perp$ and outputting a basis $B'^\perp$ of $S'^\perp$, which contains all the vectors of $B^\perp$.*

Proof. We first run the BasisGen(S') algorithm to output a basis $B''^\perp$ of $S'^\perp$. We then want to transform this basis to a basis of $S'^\perp$ which contains $B^\perp$ (we note that if S' is a subspace of S , then $S^\perp$ is a subspace of $S'^\perp$). Let the rows of matrices

$\mathbf{T}$ and $\mathbf{T}'$ be formed by $B^\perp$ and $B''^\perp$, respectively. We can then use an algorithm (such as Algorithm 2.3.7 of [41]) that takes as input $\mathbf{T}^\top$, $\mathbf{T}'^\top$ (the transpose of matrices $\mathbf{T}$ and $\mathbf{T}'$, respectively) and outputs a basis for a supplement of $S^\perp$ in $S'^\perp$. We add the output basis to $B^\perp$, which, in turn, is a basis of $S'^\perp$. $\square$

Henceforth, we refer to the algorithm described in Lemma 7.3 as the **BasisDel** algorithm. It is used during key delegation to transform a partial order associated with $\mathcal{K}_{\text{SE}}$, such as $S' \preceq S$, to a partial order associated with $\mathcal{K}_{\text{HIPE}}$, such as $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$.

7.3 Transformation from HIPE to SE

Our main idea of transformation from HIPE to SE is transforming the vector inclusion by space relation of SE into the hierarchical inner product relation of HIPE, meanwhile, it preserves the partial order relation in the key space of SE. Our technique involves two key steps: we first describe the transformation from HIPE to SE that works in a subset of affine spaces—Euclidean spaces; we then sketch in Section 7.3 how to derive an n -dimensional SE scheme in affine spaces from an $(n + 1)$ -dimensional SE scheme in Euclidean spaces.

Idea for Transforming HIPE to SE The idea of transformation from HIPE to n -dimensional SE in Euclidean spaces is as follows.

We consider an HIPE scheme with a hierarchy such that each level is n dimensional and which produces ciphertexts that are associated with only hierarchical vectors of the form $(\mathbf{x}, \dots, \mathbf{x})$ (that is all levels with the same vector $\mathbf{x}$). In such a scheme, a secret key associated with a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{\text{HIPE}}$ can also be viewed as being associated with the space $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$. In other words, we can interpret the relation between a ciphertext and a secret key as $\hat{F}(S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), (\mathbf{x}, \widehat{\text{msg}})) = \widehat{\text{msg}}$ in the SE setting.

With the above observation and the property $(S^\perp)^\perp = S$ from Lemma 7.1,

instead of generating a secret key for an $(n - \ell_s)$ -dimensional space $S \in \mathcal{K}_{SE}$, we generate a secret key for a basis $B^\perp(S) = \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\} \in \mathcal{K}_{HIPE}$ of $S^\perp$ by using the **KeyGen** algorithm of the HIPE scheme. Here, $B^\perp(S)$ can be generated by running the **BasisGen** algorithm described in Lemma 7.2 and we can arrange the basis as a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ in the HIPE setting. The initial order of the arranged basis is arbitrary, but the order should be fixed after key generation or delegation.

Furthermore, we note that when more linearly independent vectors are added in the hierarchy, the dimension of the orthogonal space gets smaller. Namely, we have that $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s})$ is a subspace of $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$. This property is crucial for key delegation. Given a space S , a subspace S' of S , and a “fixed” basis $B^\perp = \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$ of $S^\perp$, we can derive a basis in the form $B'^\perp = \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s}\}$ of $S'^\perp$ by running the **BasisDel** algorithm described in Lemma 7.3. Then we can arrange the basis $B^\perp, B'^\perp$ such that $B'^\perp \preceq B^\perp$ in the HIPE setting for key delegation.

Construction of SE from HIPE We now describe the construction of an SE scheme from an HIPE scheme using the idea described before. From the property $\dim(S) + \dim(S^\perp) = n$ of Lemma 7.1, to construct an n -dimensional SE scheme, we require an HIPE scheme with hierarchy $\vec{\mu} := ((n-1)n, n-1; n, 2n, \dots, (n-1)n)$. Given an HIPE scheme with five algorithms: **HIPE.Setup**, **HIPE.KeyGen**, **HIPE.Enc**, **HIPE.Dec**, and **HIPE.KeyDel**, we construct an SE scheme with the corresponding five algorithms: **SE.Setup**, **SE.KeyGen**, **SE.Enc**, **SE.Dec**, and **SE.KeyDel**, as follows:

- **SE.Setup** $(1^\lambda, n)$ runs

$$(\text{GPK}, \text{MSK}) \leftarrow \text{HIPE.Setup}(1^\lambda, \vec{\mu}).$$

It outputs the global public key **GPK** and master secret key **MSK**.

- **SE.KeyGen** $(\text{GPK}, \text{MSK}, S)$ generates a secret key for an $(n - \ell_s)$ -dimensional

space S . It first runs

$$B^\perp(S) := \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\} \leftarrow \text{BasisGen}(S),$$

and then runs

$$SK_S \leftarrow \text{HIPE.KeyGen}(\text{GPK}, \text{MSK}, B^\perp(S)).$$

It outputs the secret key SK_S with $B^\perp(S)$.

- $\text{SE.Enc}(\text{GPK}, \mathbf{x}, \widehat{\text{msg}})$ runs

$$CT_{\mathbf{x}} \leftarrow \text{HIPE.Enc}(\text{GPK}, (\mathbf{x}, \dots, \mathbf{x}), \widehat{\text{msg}}).$$

It outputs the ciphertext $CT_{\mathbf{x}}$.

- $\text{SE.Dec}(\text{GPK}, SK_S, CT_{\mathbf{x}})$ runs

$$\widehat{\text{msg}} \leftarrow \text{HIPE.Dec}(\text{GPK}, SK_S, CT_{\mathbf{x}}).$$

It outputs the message $\widehat{\text{msg}}$.

- $\text{SE.KeyDel}(\text{GPK}, S, SK_S, S')$ delegates a secret key to an $(n - \ell'_s)$ -dimensional subspace S' of S , where $B^\perp(S) = \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$. It first runs

$$B^\perp(S') := \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s}\} \leftarrow \text{BasisDel}(S, S', B^\perp(S))$$

and then runs

$$SK_{S'} \leftarrow \text{HIPE.KeyDel}(\text{GPK}, B^\perp(S), SK_S, B^\perp(S')).$$

It outputs the secret key $SK_{S'}$ with $B^\perp(S')$.

We now show that the resulting SE scheme works correctly and is indeed secure. Since our generic construction of SE scheme requires only a single instance of an HIPE scheme (along with some additional efficient algorithms), we use techniques similar to those for the Embedding Lemma of [23, 55].

Theorem 7.1. *The SE scheme transformed from the HIPE scheme works correctly.*

Proof. From our construction, the only algorithm that needs to be considered is delegation. Given a partial order pair $S' \preceq S$ of $\mathcal{K}_{SE}$, we transform it into a pair $(B^\perp(S'), B^\perp(S))$ such that $B^\perp(S') \preceq B^\perp(S)$ in the HIPE setting. Thus, the key delegation algorithm also works.

Given a plaintext $(\mathbf{x}, \widehat{\text{msg}}) \in \mathcal{X}_{SE}$ and an $(n - \ell_s)$ -dimensional space $S \in \mathcal{K}_{SE}$, we transform them into

$$((\mathbf{x}, \dots, \mathbf{x}), \widehat{\text{msg}}) \in \mathcal{X}_{HIPE} \quad \text{and} \quad (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{HIPE}$$

respectively, where $B^\perp(S) = \{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$ is a basis of $S^\perp$. Then we have

$$\begin{aligned} \hat{F}(S, (\mathbf{x}, \widehat{\text{msg}})) &= \widehat{\text{msg}} \Leftrightarrow \mathbf{x} \in S \\ &\Leftrightarrow \mathbf{x} \cdot \mathbf{v}_i = 0 \text{ for all } \mathbf{v}_i \in B^\perp(S) \\ &\Leftrightarrow \hat{F}((\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), ((\mathbf{x}, \dots, \mathbf{x}), \widehat{\text{msg}})) = \widehat{\text{msg}}. \end{aligned}$$

This implies that the resulting SE scheme inherits the decryptability

$$\hat{F}(S, (\mathbf{x}, \widehat{\text{msg}})) = \widehat{\text{msg}} \text{ iff } \mathbf{x} \in S \text{ (as defined in Section 2.2)}$$

from the original HIPE scheme. □

Theorem 7.2. *For any adversary $\mathcal{A}$ against the SE scheme in the same security model for the original HIPE scheme, there is an adversary $\mathcal{B}$ against the HIPE scheme, running in about the same time as $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{SE}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{HIPE}(\lambda).$$

Moreover, the SE scheme preserves properties from the original HIPE scheme.

Proof. Given any adversary $\mathcal{A}$ against the SE scheme in the same security model for the original HIPE scheme (fully/selective secure, attribute/payload-hiding), we simulate an adversary $\mathcal{B}$ with advantage $\text{Adv}_{\mathcal{A}}^{SE}(\lambda)$ against the HIPE scheme as follows:

- **Setup:** It runs a real game $\mathcal{RG}$ of HIPE and forwards GPK to adversary $\mathcal{A}$.
- **Query:** It answers $\mathcal{A}$'s queries by querying $\mathcal{RG}$'s key generation and delegation oracles.
- **Challenge:** It forwards $\mathcal{A}$'s challenge to $\mathcal{RG}$ and then returns $\mathcal{RG}$'s output to $\mathcal{A}$.
- **Guess:** It answers $\mathcal{A}$'s queries as Query phase and forwards $\mathcal{A}$'s guess to $\mathcal{RG}$.

In the above security game, we can efficiently transform the elements in $\mathcal{K}_{\text{SE}}$ and $\mathcal{X}_{\text{SE}}$, as required by the HIPE setting. From Theorem 7.1 and its proof, all the oracles of the SE scheme can be simulated correctly. Moreover, the plaintexts $x_{(0)}^* = (\mathbf{x}_{(0)}^*, \widehat{\text{msg}}_{(0)}^*)$, $x_{(1)}^* = (\mathbf{x}_{(1)}^*, \widehat{\text{msg}}_{(1)}^*) \in \mathcal{X}_{\text{SE}}$, any space $S \in \mathcal{K}_{\text{SE}}$ of $\mathcal{A}$'s choices and those in the HIPE setting satisfy requirement (2.1) (and other restrictions of the security model) simultaneously. Thus, the simulation is perfect and we conclude that $\text{Adv}_{\mathcal{B}}^{\text{HIPE}}(\lambda)$ is at least $\text{Adv}_{\mathcal{A}}^{\text{SE}}(\lambda)$.

It is clear that properties such as full/selective security (under simple assumptions) and attribute/payload-hiding are preserved in the transformation since the model of security game we simulate for the SE scheme is identical to that for the original HIPE scheme. Moreover, should the original HIPE scheme work in prime order bilinear groups and have short ciphertexts (and other properties), the derived SE scheme would also inherit such properties since the SE scheme can be viewed as a “restricted” form or an embedding of the HIPE scheme. $\square$

Remark: We sometimes may require that the key space $\mathcal{K}_{\text{SE}}$ to be containing all vectors in $\mathbb{Z}_q^n$. To achieve this, we can use n extra levels each with two dimensions for the hierarchy $\vec{\mu}$ in the HIPE scheme to “fix” a vector in the transformation. More precisely, we require a hierarchy $\vec{\mu} := ((n+1)n, 2n-1; n, 2n, \dots, (n-1)n, (n-1)n+2, \dots, (n+1)n)$ and encode $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{K}_{\text{SE}}$ as $((1, z_1), \dots, (1, z_n))$, $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{I}_{\text{SE}}$ as $((x_1, -1), \dots, (x_n, -1))$ in the last n levels. Nevertheless, we will shortly explain why this is unnecessary for the construction of SE in affine spaces.

Construction of SE in Affine Spaces Now, we briefly show how to construct an n -dimensional SE scheme in affine spaces from an $(n + 1)$ -dimensional SE scheme in Euclidean spaces.

Given an affine space $S(\mathbf{M}, \mathbf{z}) = \{\mathbf{a}\mathbf{M} + \mathbf{z} : \mathbf{a} \in \mathbb{Z}_q^{\ell_s \times n}\}$, where $\mathbf{M} \in \mathbb{Z}_q^{\ell_s \times n}$, we embed it in $S((\mathbf{M}^{[1]}, 0), \dots, (\mathbf{M}^{[\ell_s]}, 0), (\mathbf{z}, 1)) \in \mathcal{X}_{\text{SE}}$. Given a vector $\mathbf{x} \in \mathbb{Z}_q^n$, we embed it in $(\mathbf{x}, 1) \in \mathcal{I}_{\text{SE}}$. Then it is not difficult to check that $\mathbf{x} \in S(\mathbf{M}, \mathbf{z})$ iff $(\mathbf{x}, 1) \in S((\mathbf{M}^{[1]}, 0), \dots, (\mathbf{M}^{[\ell_s]}, 0), (\mathbf{z}, 1))$. Moreover, if $S(\mathbf{M}', \mathbf{z}')$ is a subspace of $S(\mathbf{M}, \mathbf{z})$, namely there is some matrix $\mathbf{T} \in \mathbb{Z}_q^{\ell'_s \times \ell_s}$ and vector $\mathbf{b} \in \mathbb{Z}_q^{\ell_s}$ such that $\mathbf{M}' = \mathbf{T}\mathbf{M}$ and $\mathbf{z}' = \mathbf{z} + \mathbf{b}\mathbf{M}$, then $S((\mathbf{M}'^{[1]}, 0), \dots, (\mathbf{M}'^{[\ell'_s]}, 0), (\mathbf{z}', 1))$ is a subspace of $S((\mathbf{M}^{[1]}, 0), \dots, (\mathbf{M}^{[\ell_s]}, 0), (\mathbf{z}, 1))$, and vice versa.

Remark: We note that a vector $\mathbf{z}$ in the key space can be viewed as a special affine space $S(\mathbf{O}, \mathbf{z})$ (where $\mathbf{O}$ is the zero matrix) and $\mathbf{x} = \mathbf{z}$ is equivalent to $\mathbf{x} \in S(\mathbf{O}, \mathbf{z})$. Therefore, no additional structure is required to add vectors in the key space $\mathcal{K}_{\text{SE}}$ in our transformation.

7.4 Transformation from SE to HIPE

We now turn our attention to the transformation from SE to HIPE. Here, we only require an SE scheme in Euclidean spaces.

Idea for Transforming SE to HIPE We first give the idea of constructing an HIPE scheme with hierarchy $\vec{\mu} := (n, d; \mu_1, \dots, \mu_d)$ from an SE scheme. Given a vector $\mathbf{v} \in \Phi_i$ (resp. $\mathbf{x} \in \Phi_i$), we use

$$\mathbf{w}^{(i)} := (0, \dots, 0, \mathbf{v}, 0, \dots, 0) \quad (\text{resp } \mathbf{y}^{(i)} := (0, \dots, 0, \mathbf{x}, 0, \dots, 0))$$

to denote an n -dimensional vector, where $\mathbf{v}$ (resp. $\mathbf{x}$) is embedded in the $(\mu_{i-1} + 1)$ -th to the μ_i -th scalars.

The key idea of our transformation is adopting some “space partition” technique. We use an n -dimensional SE scheme to embed the i -th level of $\vec{\mu}$ into the $(\mu_{i-1} + 1)$ -th up to the μ_i -th scalars. Namely, given a hierarchical vector

$(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{\text{HIPE}}$ (resp. $(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}) \in \mathcal{I}_{\text{HIPE}}$), we embed $\mathbf{v}_i$ (resp. $\mathbf{x}_i$) in $\mathbf{w}_i^{(i)}$ (resp. $\mathbf{y}_i^{(i)}$), which makes each level “independent”. More precisely, we generate secret keys and ciphertexts as follows:

1. A secret key generated using the **KeyGen** algorithm of the SE scheme for the space $S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s)}) \in \mathcal{K}_{\text{SE}}$ is set to be a secret key for a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{\text{HIPE}}$;
2. A message to be encrypted is now associated with the vector $\mathbf{y} = \sum_{i=1}^{\ell_c} \mathbf{y}_i^{(i)} \in \mathcal{I}_{\text{SE}}$ instead of a hierarchical vector $(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}) \in \mathcal{I}_{\text{HIPE}}$.

If $\ell_s \leq \ell_c$, then we have

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{v}_i = 0 \text{ for } i \in [\ell_s] &\Leftrightarrow \mathbf{y} \cdot \mathbf{w}_i^{(i)} = 0 \text{ for } i \in [\ell_s] \\ &\Leftrightarrow \mathbf{y} \in S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_c}^{(\ell_s)}). \end{aligned}$$

This indicates that the functionality $\hat{F}$ of the resulting HIPE scheme correctly outputs the payload message (as defined in Section 2.2) if the original SE scheme does.

For a partial order pair $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}, \mathbf{v}_{\ell_s+1}, \dots, \mathbf{v}_{\ell'_s}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ of HIPE, we have $S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s)})$ being a subspace of $S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s)})$. We can then run the delegation algorithm of the SE scheme for $S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s)})$ and the resulting key is set to be the secret key associated with $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell'_s})$.

Up to now, we have considered the case $\ell_s \leq \ell_c$. However, there is an issue when $\ell_c < \ell_s$. If we encrypt a message associated with a vector $\mathbf{y} = \sum_{i=1}^{\ell_c} \mathbf{y}_i^{(i)} \in \mathcal{I}_{\text{SE}}$ and generate a secret key associated with a space $S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s)}) \in \mathcal{K}_{\text{SE}}$ where $\mathbf{y} \cdot \mathbf{w}_i^{(i)} = 0$ for $i \in [h]$, then we still have $\mathbf{y} \in S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s)})$. This does not meet the correctness required for HIPE. To address this problem, we use one additional dimension for each level of hierarchy $\vec{\mu}$ of HIPE in SE, namely, we require an $(n + d)$ -dimensional SE scheme. It will always make $\mathbf{y} \notin S^\perp(\mathbf{w}_1^{(1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s)})$ if $\ell_c < \ell_s$.¹

¹We note that the technique of appending random vectors to the intended hierarchical vectors

Construction of HIPE From SE To construct an HIPE scheme with hierarchy $\vec{\mu} := (n, d; \mu_1, \dots, \mu_d)$, as explained before, we require an SE scheme with dimension $n' = n + d$. Given a vector $\mathbf{v} \in \Phi_i$ (resp. $\mathbf{x} \in \Phi_i$) and $\hat{b} \in \{0, 1\}$, we use

$$\mathbf{w}^{(i, \hat{b})} := (0, \dots, 0, (\mathbf{v}, \hat{b}), 0, \dots, 0) \quad (\text{resp } \mathbf{y}^{(i, \hat{b})} := (0, \dots, 0, (\mathbf{x}, \hat{b}), 0, \dots, 0))$$

to denote an n' -dimensional vector, where $(\mathbf{v}, \hat{b})$ (resp. $(\mathbf{x}, \hat{b})$) is embedded in the $(\mu_{i-1} + i)$ -th up to the $(\mu_i + i)$ -th scalars. Let

$$\mathbf{1}^{(i)} := (0, \dots, 0, 1, 0, \dots, 0)$$

be an n' -dimensional vector, where the $(\mu_i + i)$ -th scalar is 1.

Given an SE scheme with five algorithms: SE.Setup, SE.KeyGen, SE.Enc, SE.Dec, and SE.KeyDel, we construct an HIPE scheme with the corresponding five algorithms: HIPE.Setup, HIPE.KeyGen, HIPE.Enc, HIPE.Dec, and HIPE.KeyDel, as follows:

- HIPE.Setup($1^\lambda, \vec{\mu}$) runs

$$(\text{GPK}, \text{MSK}) \leftarrow \text{SE.Setup}(1^\lambda, n').$$

It outputs the global public key GPK and master secret key MSK.

- HIPE.KeyGen(GPK, MSK, $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$) first runs

$$\mathbf{B}^\perp := \mathbf{B}^\perp(\mathbf{S}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)})) \leftarrow \text{BasisGen}(\mathbf{S}(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)}))$$

and then runs

$$\text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})} \leftarrow \text{SE.KeyGen}(\text{GPK}, \text{MSK}, \mathbf{S}(\mathbf{B}^\perp)).$$

It outputs the secret key $\text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}$ with $\mathbf{S}(\mathbf{B}^\perp)$.

during encryption [75] may be applicable to some specific instantiations of SE to achieve more efficient transformation.

- $\text{HIPE.Enc}(\text{GPK}, (\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}), \widehat{\text{msg}})$ runs

$$\text{CT}_{(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c})} \leftarrow \text{SE.Enc}(\text{GPK}, \sum_{i=1}^d \mathbf{y}_i^{(i,0)}, \widehat{\text{msg}}),$$

where $\mathbf{y}_i^{(i,0)} = \mathbf{1}^{(i)}$ for $i \in [\ell_c + 1, d]$. It outputs the ciphertext $\text{CT}_{(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c})}$.

- $\text{HIPE.Dec}(\text{GPK}, \text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}, \text{CT}_{(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c})})$ runs

$$\widehat{\text{msg}} \leftarrow \text{SE.Dec}(\text{GPK}, \text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}, \text{CT}_{(\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c})}).$$

It outputs the message $\widehat{\text{msg}}$.

- $\text{HIPE.KeyDel}(\text{GPK}, (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), \text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}, (\mathbf{v}_1, \dots, \mathbf{v}_{\ell'_s}))$ where the secret key $\text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}$ is associated with a space $S(B^\perp)$. It first runs

$$B'^\perp := B^\perp(S(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s,1)})) \leftarrow \text{BasisGen}(S(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s,1)}))$$

and then runs

$$\text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell'_s})} \leftarrow \text{SE.KeyDel}(\text{GPK}, S(B^\perp), \text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})}, S(B'^\perp)).$$

It outputs the secret key $\text{SK}_{(\mathbf{v}_1, \dots, \mathbf{v}_{\ell'_s})}$ with $S(B'^\perp)$.

We show the resulting HIPE scheme works correctly and is indeed secure. The proof techniques are similar to those for Theorems 7.1 and 7.2.

Theorem 7.3. *The HIPE scheme transformed from the SE scheme works correctly.*

Proof. From our construction, the only algorithm that needs to be considered is key delegation. Given a partial order pair $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell'_s}) \preceq (\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ of $\mathcal{K}_{\text{HIPE}}$, we transform it into a pair $(S^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s,1)}), S^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)}))$ such that $S^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell'_s}^{(\ell'_s,1)}) \preceq S^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)})$ in the SE setting. Thus, the key delegation algorithm also works.

Given a plaintext $((\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}), \widehat{\text{msg}}) \in \mathcal{X}_{\text{HIPE}}$ and a hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{\text{HIPE}}$, we transform them into

$$\mathbf{y} = \sum_{i=1}^d \mathbf{y}_i^{(i,0)} \in \mathcal{X}_{\text{SE}} \quad \text{and} \quad \mathbf{S}^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)}) \in \mathcal{K}_{\text{SE}}$$

respectively. Then we have

- if $\ell_s > \ell_c$, we always let $\mathbf{y}_i^{(i,0)} = \mathbf{1}^{(i)}$ for $i \in [\ell_c + 1, d]$ in our transformation, then $\mathbf{y} \cdot \mathbf{w}_i^{(i,1)} = 1$ for $i \in [\ell_c + 1, \ell_s]$, implying that $\mathbf{y} \notin \mathbf{S}^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)})$;
- if $\ell_s \leq \ell_c$, then

$$\begin{aligned} \hat{F}((\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), ((\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}), \widehat{\text{msg}})) &= \widehat{\text{msg}} \\ \Leftrightarrow \mathbf{x}_i \cdot \mathbf{v}_i &= 0 \text{ for } i \in [\ell_s] \\ \Leftrightarrow \mathbf{y} \cdot \mathbf{w}_i^{(i,1)} &= 0 \text{ for } i \in [\ell_s] \\ \Leftrightarrow \mathbf{y} &\in \mathbf{S}^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)}) \\ \Leftrightarrow \hat{F}(\mathbf{S}^\perp(\mathbf{w}_1^{(1,1)}, \dots, \mathbf{w}_{\ell_s}^{(\ell_s,1)}), (\mathbf{y}, \widehat{\text{msg}})) &= \widehat{\text{msg}}. \end{aligned}$$

This shows that the resulting HIPE scheme inherits the decryptability from the original SE scheme, i.e., we have $\hat{F}((\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}), ((\mathbf{x}_1, \dots, \mathbf{x}_{\ell_c}), \widehat{\text{msg}})) = \widehat{\text{msg}}$ iff $\ell_s \leq \ell_c$ and $\mathbf{x}_i \cdot \mathbf{v}_i = 0$ for all $i \in [\ell_s]$ as defined in Section 2.2. $\square$

Theorem 7.4. *For any adversary $\mathcal{A}$ against the HIPE scheme in the same security model for the original SE scheme, there is an adversary $\mathcal{B}$ against the SE scheme, running in about the same time as $\mathcal{A}$, such that*

$$\text{Adv}_{\mathcal{A}}^{\text{HIPE}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{\text{SE}}(\lambda).$$

Moreover, the HIPE scheme preserves properties from the original SE scheme.

Proof. Given any adversary $\mathcal{A}$ against the HIPE scheme in the same security model for the original SE scheme (fully/selective secure, attribute/payload-hiding), we simulate an adversary $\mathcal{B}$ with advantage $\text{Adv}_{\mathcal{A}}^{\text{HIPE}}(\lambda)$ against the SE scheme as with the proof for Theorem 7.2. From Theorem 7.3 and its proof, we can conclude

that all the oracles of the HIPE scheme can be simulated correctly. Furthermore, the plaintexts $x_{(0)}^* = ((\mathbf{x}_{1,0}^*, \dots, \mathbf{x}_{\ell_{c,0},0}^*), \widehat{\text{msg}}_{(0)}^*)$, $x_{(1)}^* = ((\mathbf{x}_{1,1}^*, \dots, \mathbf{x}_{\ell_{c,1},1}^*), \widehat{\text{msg}}_{(1)}^*) \in \mathcal{X}_{\text{HIPE}}$, any hierarchical vector $(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}) \in \mathcal{K}_{\text{HIPE}}$ of $\mathcal{A}$'s choices and those in the SE setting satisfy requirement (2.1) (and other restrictions of the security model) simultaneously.

Using the same argument, the simulation is perfect and we conclude that $\text{Adv}_B^{\text{SE}}(\lambda)$ is at least $\text{Adv}_A^{\text{HIPE}}(\lambda)$. Preservation of properties could also be considered in a similar way. $\square$

7.5 Working in $\mathbb{Z}_N$

Up to this point, we have worked in $\mathbb{Z}_q$, where q is prime (e.g. prime order bilinear groups). Here, we explain how an SE or HIPE scheme is defined and why our transformation techniques also work in $\mathbb{Z}_N$, where N is composite and hard to factor (e.g. composite order bilinear groups).

Given a hard-factoring composite integer N and let $\mathbb{Z}_N = \mathbb{Z}_N^* \cup \{0\}$, operations in $\mathbb{Z}_N$ are then closed (i.e., no element in $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$) except for a negligible probability. In what follows, we assume that all operations are in $\mathbb{Z}_N^*$ without mentioning the negligible probability, since if some elements in $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$ are involved in any operation in $\mathbb{Z}_N$ with this negligible probability, then factoring N becomes trivial.

We first explain how to define HIPE and SE in $\mathbb{Z}_N$. We note that HIPE can be defined in $\mathbb{Z}_N$ in a similar way as before (Section 2.2), since it does not involve any notion of field. On the other hand, to define SE, we first define an “affine space” by considering the combination of a set of vectors that is in the form $\{\mathbf{a}\mathbf{M} + \mathbf{z} : \mathbf{a} \in \mathbb{Z}_N^{\ell_s}\}$, where $\mathbf{M} \in \mathbb{Z}_N^{\ell_s \times n}$. We still use $S(\mathbf{M}, \mathbf{z})$ to denote such a “space”. A “space” $S(\mathbf{M}', \mathbf{z}')$ is called a “subspace” of $S(\mathbf{M}, \mathbf{z})$ if there is some (efficiently computable) matrix $\mathbf{T} \in \mathbb{Z}_N^{\ell'_s \times \ell_s}$ and vector $\mathbf{b} \in \mathbb{Z}_N^{\ell_s}$ such that $\mathbf{M}' = \mathbf{T}\mathbf{M}$ and $\mathbf{z}' = \mathbf{z} + \mathbf{b}\mathbf{M}$. With these in mind, we can then define SE in $\mathbb{Z}_N$ as before.

Next, we consider transformation between HIPE and SE in $\mathbb{Z}_N$. Here, orthogonal spaces are essential elements required by the transformation and they can be defined in $\mathbb{Z}_N$ in a similar way.

Definition 7.1. *Given a set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$ in $\mathbb{Z}_N^n$, the “orthogonal space” $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ is formed by a set of vectors $\mathbf{x}$ that satisfy the condition that $\mathbf{x} \cdot \mathbf{v}_i = 0$ for all $i \in [\ell_s]$.*

Let $\mathbf{M}$ be a matrix whose rows are formed by a set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$. We can use row transformation of matrix to obtain a reduced echelon form of $\mathbf{M}$, represented by $\mathbf{E}_\mathbf{M}$. It is not difficult to see that $S(\mathbf{M}, \mathbf{0})$ and $S(\mathbf{E}_\mathbf{M}, \mathbf{0})$ are identical. Here, we still use $S(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$ to denote the “space” $S(\mathbf{M}, \mathbf{0})$ and let $\dim(S(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}))$ (“dimension” of $S(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$) denote the number of non-zero rows in $\mathbf{E}_\mathbf{M}$. From $\mathbf{E}_\mathbf{M}$, i.e. the reduced echelon form of $\mathbf{M}$, we can compute a “basis” of $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$, which can also form a matrix in a reduced echelon form if we only switch some columns. Conversely if we consider the “basis” of $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$, then $\mathbf{E}_\mathbf{M}$ (or $\{\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s}\}$) forms the “orthogonal space” of $S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})$. Moreover, $\dim(S(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})) + \dim(S^\perp(\mathbf{v}_1, \dots, \mathbf{v}_{\ell_s})) = n$.

From the above, it is easy to see that we obtain the similar properties as in Lemma 7.1. Moreover, it is also not difficult to check that algorithms 2.3.1, 2.3.7 of [41] still work in $\mathbb{Z}_N$ and the proofs for Lemmas 7.2 and 7.3 still hold. Therefore, our transformation techniques also work in $\mathbb{Z}_N$.

7.6 Discussion

We have seen how the property-preserving transformation is performed between an SE and an HIPE scheme, namely, we have shown the following relations:

$$\text{SE (in affine spaces)} \Longleftarrow \text{SE (in Euclidean spaces)} \Longleftrightarrow \text{HIPE}.$$

Since the inverse direction of the first arrow is trivial, we obtain:

$$\text{SE} \Longleftrightarrow \text{HIPE}.$$

Let us now compare some selected existing HIPE and SE schemes with schemes derived using our property-preserving transformation techniques. Let $\text{HIPE}_{\text{OT10}}$ denote the HIPE scheme of [76], $\text{HIPE}_{\text{OT11}}$ denote the HIPE scheme of [78], and SE_{BH08} denote the SE scheme of [23]. We also let $\text{SE}_{\text{OT10}}^*$, $\text{SE}_{\text{OT11}}^*$, and $\text{HIPE}_{\text{BH08}}^*$ be the corresponding transformed schemes, for example $\text{SE}_{\text{OT10}}^*$ is derived from $\text{HIPE}_{\text{OT10}}$. We compare them in terms of the size of global public key **GPK**, the size of secret keys **SK**, the size of ciphertexts **CT**, the number $\#$ of pairing computation for decryption, and their properties. Here, the sizes are in the number of group elements. Moreover, all the SE schemes are n -dimensional and the HIPE schemes are with hierarchy $\vec{\mu} := (n, d; \mu_1, \dots, \mu_d)$. A summary of the comparisons is presented in Table 7.1, where n_μ denotes the maximal value of $\{\mu_i - \mu_{i-1} : i \in [d]\}$.²

	$\text{HIPE}_{\text{OT10}}$	$\text{SE}_{\text{OT10}}^*$	$\text{HIPE}_{\text{OT11}}$	$\text{SE}_{\text{OT11}}^*$	SE_{BH08}	$\text{HIPE}_{\text{BH08}}^*$
$\#$ GPK	$\mathcal{O}(n_\mu^2 d)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
$\#$ SK	$\mathcal{O}(n_\mu^2 d^2)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
$\#$ CT	$\mathcal{O}(n_\mu d)$	$\mathcal{O}(n^2)$	14	14	3	3
$\#$ pairings	$\mathcal{O}(n_\mu d)$	$\mathcal{O}(n^2)$	13	13	2	2
fully secure	Yes	Yes	Yes	Yes	No	No
anonymous	Yes	Yes	No	No	No	No
short CT	No	No	Yes	Yes	Yes	Yes
prime order	Yes	Yes	Yes	Yes	Yes	Yes
assumptions	DLIN	DLIN	DLIN	DLIN	DBDHE	DBDHE

Table 7.1: Comparisons between existing and derived HIPE/SE schemes.

²We have not listed all the existing schemes such as the selective secure HIPE scheme of [75] and the SE scheme in composite order bilinear groups of [72]. Particularly, we note that it is possible to derive an efficient fully secure and (adaptively) attribute-hiding HIPE scheme by applying the technique of [78] to the HIPE scheme of [77], in which the resulting SE scheme may be more efficient than $\text{SE}_{\text{OT10}}^*$.

Table 7.1 shows that $\text{SE}_{\text{OT10}}^*$ achieves attribute-hiding and full security, in comparison with $\text{SE}_{\text{OT11}}^*$ and SE_{BH08} , at the expense of increases in the sizes of the global public key, secret keys and ciphertexts, and the number of pairing evaluations, approximately in the factor of between n and n^3 . Moreover, $\text{SE}_{\text{OT11}}^*$ has constant-size ciphertexts. On the other hand, $\text{HIPE}_{\text{BH08}}^*$ requires roughly about a factor of n elements less than $\text{HIPE}_{\text{OT10}}$ and $\text{HIPE}_{\text{OT11}}$ in terms of the sizes of the global public key and secret keys. This shows that $\text{HIPE}_{\text{BH08}}^*$ achieves significant improvement in terms of the size of ciphertexts and the number of pairing computations during decryption, with only 3 group elements and 2 pairings, respectively.

Conclusions

Contents

8.1 Concluding Remarks	151
8.2 Suggestions for Future Work	152

In this chapter, we summarize the thesis and gives some concluding remarks which reflect the problems that we have studied and the results that we have achieved. We also give suggestions for future work in this area.

8.1 Concluding Remarks

The root of FE trace back to IBE, which was first conceived by Shamir [87] and then first constructed by Boneh and Franklin [22] and Cocks [40]. Subsequently, IBE was extended/generated to HIBE [57, 51], query on encrypted data [27], (H)IPE [58, 75], SE [23], ABE [86] etc. To the best of our knowledge, (H)IPE and SE are the widest class of encryption systems supporting the attribute-hiding security. They were proposed for expressive access control and searching on encrypted data. On the other hand, ABE is the widest class of encryption systems supporting (non-)monotone functions, which can be expressed using AND, OR, Threshold, and NOT gates. However, existing ABE schemes have the limitation

that parameter $\widehat{\text{ind}}$ of ciphertext would be revealed to the user during decryption, namely they do not satisfy the attribute-hiding security. All of these encryption systems take a step towards the generalized, flexible, and powerful notion of FE. Hence, FE has gained considerable attention from the cryptographic research community and many works, such as proposing new instantiations, improving the efficiency, simplifying the required assumptions, and achieving stronger security, have emerged.

In this thesis, we studied some specific instantiations of FE and their relations, which is summarized as follows. In Chapter 5, we presented shorter IBE, IPE, KP-FE schemes via asymmetric pairings. By shifting from the DLIN assumption to the simpler SXDH assumption, our schemes are syntactically simpler and achieve shorter global public key, secret keys, and ciphertexts. Our results can be extended to other variants of FE, such as CP-FE [76], and applied in combination with Okamoto and Takashima’s techniques for achieving short ciphertexts or secret keys [78]. In Chapter 6, we presented some new, efficient, and non-interactive RIBE schemes that are from bilinear pairings and lattices respectively. More precisely, we obtained the first adaptive-ID secure and anonymous RIBE scheme with constant-size global public key and the first efficient RIBE scheme from lattices. In Chapter 7, we studied the relation and transformation between two important instantiations of FE—HIPE and SE. The “bridging” between HIPE and SE using our techniques converts the problem of constructing an HIPE scheme with certain properties into a problem of constructing an SE scheme with the required (similar) properties, and vice versa.

8.2 Suggestions for Future Work

FE is still in its infancy and many interesting open problems remain. One major challenge is to construct a secure FE scheme for other mathematical properties/concepts or a more general class of functionalities apart from inner product

and non-monotone access structures. Moreover, it is essential to improve the efficiency, simplify the required assumptions, and achieve stronger security for existing instantiations. While these improvement could be mainly considered based on pairings and lattices, we also hope for appearance of new powerful tools for constructing FE schemes.

Pairing-based cryptographic protocols have been extensively studied particularly over the last decade since Boneh and Franklin’s seminal work [22]. We believe that the topic is far from being exhausted. For example:

- Some FE schemes with specific primitives (for example, fully secure HIBE scheme with short ciphertexts [38]) were only realized in composite order bilinear groups, however, they are known to be less efficient in implementations [47]. Although there are some techniques for simulating specific features of composite order bilinear groups [47, 61], we still cannot obtain schemes with the corresponding properties in prime order bilinear groups. Discovering perfect simulation techniques for composite order bilinear groups in the prime order setting is still open.

Back to our work, we have obtained some new instances of HIPE and SE schemes with different properties from our transformation. We note that achieving short ciphertexts while preserving the adaptively attribute-hiding property seems impossible in the context of anonymous broadcast encryption (in the public-key setting) [59, 67]. Since (H)IPE/SE implies broadcast encryption [58, 23], the impossibility appears to carry over to the former. But weakly attribute-hiding HIPE/SE with short ciphertexts may be possible since it is not difficult to construct an SE scheme with these properties in composite order bilinear groups from the HIBE of [38]. However, how to construct a fully secure and weakly attribute-hiding HIPE/SE scheme with short ciphertexts in prime order bilinear groups, particularly those that work under simple assumptions, is still open. This is because when we use DPVS to simulate the parameter/entropy-hiding feature of composite

order bilinear groups, the required dimension for DPVS will be linear in the dimension of HIPE/SE.

- We note that the parameters of the derived schemes are longer than those of the original schemes by applying our transformation between SE and HIPE in Chapter 7. Especially, constructing SE from HIPE is quadratically expensive. Thus, how to reduce the complexity in our transformation is an interesting problem.
- Another interesting open problem is to construct a fully secure ABE scheme with short ciphertexts. Presently, there are only a few ABE schemes achieving short ciphertexts but were proved under selective security model [13]. The existing techniques for ABE [63, 76, 78] to achieve full security seem hard to derive an ABE scheme with short ciphertexts. When we modify existing ABE schemes achieving short ciphertexts, the main difficulty is to show that the distribution of a nominally semi-functional secret key and ciphertext pair remains the same as the distributions of semi-functional spaces from the adversary's view. We hope new techniques will be proposed to overcome this issue.

On the other hand, lattice-based cryptographic schemes have also received much attention in recent years because they are all believed to be secure against attacks using quantum computers, a property not achievable by cryptographic primitives based on factoring or discrete logarithm. However, due to lack of the “bi” in bilinear maps and operations over finite field or ring are very “limited”, lattices are not as flexible as we wish. For example:

- We note that the existing IPE, Fuzzy IBE schemes [4, 5] from lattices were only proven under the selective security model. How to build a dual system framework on lattices is still open, although the Dual System Encryption has become a powerful methodology to obtain full security for various classes of FE in bilinear groups.

- Another interesting open problem is to construct an ABE scheme from lattices (even under selective security model) which may require the discovery of new flexible trapdoors. We note that all the previous ABE constructions [53, 79, 18, 63, 76, 94] are based on bilinear pairings.

Finally, for key revocation in IBE, we note that although Boldyreva et al. [19] reduce the complexity of key updates from linear to logarithmic, this is achieved at the cost of increasing the size of private key from constant to logarithmic in the maximal number of users. Some possible directions are:

- The grand work is to reduce the complexity of all parameters such as workload in key updates, size of private keys, etc. However, this seems rather challenging.
- With regards to our work, a more achievable goal is to construct an adaptive-ID secure RIBE scheme from lattices. The main difficulty is that, in the security proof of our lattice-based RIBE scheme, the simulator required to prepare short vectors for t^* , id^* and $\{\mathbf{u}_{\theta,1}, \mathbf{u}_{\theta,2}\}$ before answering the adversary's queries. However, in adaptive-ID secure game, the adversary can adaptively choose (t^*, id^*) after issuing some key queries. As with other open problems from lattices, we may require new techniques to solve this problem.
- Another interesting open problem is to find a generic efficient transformation from IBE to RIBE, maybe we require some additional properties on the IBE schemes.

Bibliography

- [1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.
- [2] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010.
- [3] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *CRYPTO*, pages 98–115, 2010.
- [4] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, pages 21–40, 2011.
- [5] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In *Public Key Cryptography*, pages 280–297, 2012.

-
- [6] W. Aiello, S. Lodha, and R. Ostrovsky. Fast digital identity revocation (extended abstract). In *CRYPTO*, pages 137–152, 1998.
 - [7] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.
 - [8] M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, pages 1–9, 1999.
 - [9] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.
 - [10] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS*, pages 75–86, 2009.
 - [11] G. Ateniese, J. Kirsch, and M. Blanton. Secret handshakes with dynamic and fuzzy matching. In *NDSS*, 2007.
 - [12] 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.
 - [13] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pages 90–108, 2011.
 - [14] L. Ballard, M. Green, B. de Medeiros, and F. Monrose. Correlation-resistant storage via keyword-searchable encryption. IACR Cryptology ePrint Archive, Report 2005/417, 2005.
 - [15] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management—part 1: General (revised). *NIST Special Pub*, 800-57, 2007.

-
- [16] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331, 2005.
 - [17] A. Beimel. Secure schemes for secret sharing and key distribution. *PHD Thesis*, 1996.
 - [18] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
 - [19] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *ACM Conference on Computer and Communications Security*, pages 417–426, 2008.
 - [20] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.
 - [21] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.
 - [22] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
 - [23] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.
 - [24] D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *ACM Conference on Computer and Communications Security*, pages 211–220, 2006.
 - [25] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.
 - [26] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

- [27] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.
- [28] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.
- [29] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.
- [30] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.
- [31] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, pages 325–341, 2005.
- [32] D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*, pages 573–592, 2006.
- [33] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.
- [34] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, pages 501–510, 2010.
- [35] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000. ISBN 0262024918.
- [36] F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005.
- [37] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

-
- [38] A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts. In *Pairing*, pages 347–366, 2010.
 - [39] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.
 - [40] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.
 - [41] H. Cohen, editor. *A Course in Computational Algebraic Number Theory*. Springer, 1996. ISBN 0387556400.
 - [42] R. Cramer and I. Damgård. On the amortized complexity of zero-knowledge protocols. In *CRYPTO*, pages 177–191, 2009.
 - [43] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
 - [44] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
 - [45] L. Ducas. Anonymity from asymmetry: New constructions for anonymous HIBE. In *CT-RSA*, pages 148–164, 2010.
 - [46] D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010.
 - [47] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, pages 44–61, 2010.
 - [48] S. D. Galbraith and V. Rotger. Easy decision Diffie-Hellman groups. IACR Cryptology ePrint Archive, Report 2004/070, 2004.

-
- [49] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.
 - [50] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
 - [51] C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.
 - [52] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
 - [53] 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.
 - [54] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.
 - [55] M. Hamburg. Spatial encryption. *IACR Cryptology ePrint Archive*, 2011: 389, 2011.
 - [56] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai. Identity-based hierarchical strongly key-insulated encryption and its application. In *ASIACRYPT*, pages 495–514, 2005.
 - [57] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.
 - [58] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.
 - [59] A. Kiayias and K. Samari. Lower bounds for private broadcast encryption. In *Information Hiding*, 2012.

-
- [60] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48 (177):203–209, 2011.
 - [61] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.
 - [62] 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.
 - [63] 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.
 - [64] B. Libert and J.-J. Quisquater. Efficient revocation and threshold pairing based cryptosystems. In *PODC*, pages 163–171, 2003.
 - [65] B. Libert and D. Vergnaud. Towards black-box accountable authority ibe with short ciphertexts and private keys. In *Public Key Cryptography*, pages 235–255, 2009.
 - [66] B. Libert and D. Vergnaud. Adaptive-id secure revocable identity-based encryption. In *CT-RSA*, pages 1–15, 2009.
 - [67] B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *Public Key Cryptography*, pages 206–224, 2012.
 - [68] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *FOCS*, pages 356–365, 2002.
 - [69] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. *IACR Cryptology ePrint Archive*, 2011:501, 2011.

-
- [70] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, pages 417–426, 1985.
 - [71] A. Miyaji, M. Nakabayashi, and S. Takano. Characterization of elliptic curve traces under fr-reduction. In *ICISC*, pages 90–108, 2000.
 - [72] D. Moriyama and H. Doi. A fully secure spatial encryption scheme. *IEICE Transactions*, 94-A(1):28–35, 2011.
 - [73] M. Naor and K. Nissim. Certificate revocation and certificate update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–570, 2000.
 - [74] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.
 - [75] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.
 - [76] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.
 - [77] T. Okamoto and K. Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. *IACR Cryptology ePrint Archive*, 2011:543, 2011.
 - [78] 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.
 - [79] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.
 - [80] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.

-
- [81] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In *ACM Conference on Computer and Communications Security*, pages 99–112, 2006.
 - [82] S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of waters’ dual-system primitives using asymmetric pairings. IACR Cryptology ePrint Archive, Report 2012/024, 2012.
 - [83] O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6): 899–942, 2004.
 - [84] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
 - [85] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
 - [86] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.
 - [87] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
 - [88] E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, pages 560–578, 2008.
 - [89] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2008. ISBN 9780521516440.
 - [90] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

- [91] E. R. Verheul. Evidence that XTR is more secure than Supersingular Elliptic Curve cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004.
- [92] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
- [93] B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.
- [94] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, pages 53–70, 2011.

Estimated Bit Sizes of Elements in Bilinear Groups

The ordinary elliptic curves that give the best performance while providing discrete log security comparable to three commonly proposed levels of AES security are as follows. The group sizes follow the 2007 NIST recommendations [15], descriptions of the elliptic curves are in [46].

80-bit security : A 170-bit MNT curve [71] with embedding degree $k = 6$.

128-bit security : A 256-bit Barreto-Naehrig curve [16] with $k = 12$.

256-bit security : A 640-bit Brezing-Weng curve [36] with $k = 24$.

Pairings	80-bit AES			128-bit AES			256-bit AES		
	G_1	G_2	G_T	G_1	G_2	G_T	G_1	G_2	G_T
Asymmetric	170	340	1020	256	512	3072	640	2560	15360
Symmetric	176	176	1056	512	512	3072	2560	2560	15360

Table A.1: Estimated bit sizes of elements in bilinear groups.

Note that a symmetric pairing only exists on supersingular elliptic curves. The restriction to supersingular elliptic curves means that at high security levels the

group G_1 will be much larger than the group G_1 on an equivalent ordinary curve.

Nominally Semi-Functional Keys

In the proof of dual system encryption, the simulator can generate (semi-functional) ciphertexts and keys for any identities. When transforming the κ 's key into semi-functional and showing that two successive games are indistinguishable, this results in a potential problem. If the simulator is prepared to make a semi-functional ciphertext for an identity id and is also prepared to make the κ 'th key for the same identity id , it may seem like the simulator can determine whether the κ 's key is semi-functional for itself by test decrypting with a semi-functional ciphertext for the same identity. To resolve this paradox, Lewko and Waters introduced a variant of semi-functional keys—nominally semi-functional keys. These keys are semi-functional in name only, meaning that they are distributed like semi-functional keys, but are actually correlated with semi-functional ciphertexts so that when a nominally semi-functional key is used to decrypt a semi-functional ciphertext, the interaction of the two semi-functional components results in cancellation and decryption is successful. If the simulator attempts to answer its own question by creating the κ 's key and challenge ciphertext for the same identity, the created key will be nominally semi-functional and hence test decrypting will not distinguish this from a normal key. This nominally semi-functional key will appear to be distributed like a regular semi-functional key from the adversary's view, who cannot request a key that can decrypt the challenge ciphertext [62].