Advances in Functional Encryption

Hoeteck Wee

École normale supérieure
LIENS – École Doctorale de Sciences Mathématiques de Paris Centre

Advances in Functional Encryption

THÈSE D'HABILITATION

présentée pour l'obtention du

Diplôme d'Habilitation à Diriger des Recherches
de l'École normale supérieure

(Spécialité Informatique)

par

Hoeteck Wee

Soutenue publiquement le 1 Jul 2016 devant le jury composé de

Dario Catalano ..... *Rapporteur*
Pierre-Alain Fouque ..... *Rapporteur*
Leonid Reyzin ..... *Rapporteur*
Michel Abdalla ..... *Examinateur*
Iordanis Kerenidis ..... *Examinateur*
Eike Kiltz ..... *Examinateur*
David Pointcheval ..... *Examinateur*
Damien Stehlé ..... *Examinateur*

Travaux effectués au sein de l'Équipe de Cryptographie
du Laboratoire d'Informatique de l'École normale supérieure

Abstract

Functional encryption is a novel paradigm for public-key encryption that enables both fine-grained access control and selective computation on encrypted data, as is necessary to protect big, complex data in the cloud. In this thesis, I provide a brief introduction to functional encryption, and an overview of my contributions to the area.

Contents

I Advances in Functional Encryption 5

1 Introduction 5

2 Functional Encryption 5

3 My Contributions 7

4 Computational Assumptions 9

5 Attribute-Based Encryption for Circuits 10

6 Déjà Q: Encore! Un Petit IBE 14

7 Quasi-Adaptive NIZK for Linear Subspaces Revisited 18

8 Tightly CCA-secure Encryption without Pairings 20

9 Additional Contributions 26

10 Conclusion 29

II Curriculum Vitae 30

III Attribute-Based Encryption for Circuits 36

1 Introduction 36

2 Our Contributions 37

3 Preliminaries 42

4 Two-to-One Recoding Schemes 44

5 TOR from LWE 48

6 Attribute-Based Encryption for Circuits 50

7 Attribute-Based Encryption for Branching Programs 55

8 Extensions 63

IV Predicate Encryption for Circuits from LWE 67

1 Introduction 67

2 Preliminaries 73

3 Partially Hiding Predicate Encryption 75

4 Predicate Encryption for Circuits 79

V Fully, (Almost) Tightly Secure IBE and Dual System Groups 84

1 Introduction 84

2 Preliminaries 92

3 Nested Dual System Groups 93

4 (Almost) Tight IBE from Nested Dual System Groups 97

5 Concrete (almost) tight IBE scheme from d-LIN in prime-order groups 105

6 Dual System Groups 106

7 Compact HIBE scheme from d-LIN in prime-order groups 108

VI Tightly CCA-Secure Encryption without Pairings 110

1 Introduction 110

2 Preliminaries 117

3 Multi-ciphertext PCA-secure KEM 121

4 Multi-ciphertext CCA-secure Public Key Encryption scheme 131

Part I

Advances in Functional Encryption

1 Introduction

Recent computing and technological advances such as the ubiquity of high-speed network access and the proliferation of mobile devices have had a profound impact on our society, our lives and our behavior. In the past decade, we have seen a substantial shift towards a digital and paperless society, along with a migration of data and computation to the cloud. Storing data in the cloud offers tremendous benefits: easy and convenient access to the data and reliable data storage for individuals, as well as scalability and financial savings for organizations. On the flip side, storing data remotely poses an acute security threat as these data – government, financial, medical records as well as personal information exchanged over email and social networks – are outside our control and could potentially be accessed by untrusted parties. Without taking measures to protect our data, we are at risk of devastating privacy breaches and living under digital surveillance in an Orwellian future.

However, traditional public-key encryption lacks the expressiveness needed to protect big, complex data:

(i) First, traditional encryption only provides coarse-grained access to encrypted data, namely, only a single secret key can decrypt the data. Corporate entities want to share data with groups of users based on their credentials. Similarly, individuals want to selectively grant access to their personal data on social networks and Google Docs.

(ii) Second, access to encrypted data is "all or nothing": one either decrypts the entire plaintext or learns nothing about the plaintext. In applications such as data-mining on encrypted medical records or social networks, we want to provide only partial access and selective computation on the encrypted data, for instance, restricted classes of statistical or database queries.

Ideally, we want to encrypt data while enabling fine-grained access control and selective computation; that is, we want control over *who* has access to the encrypted data and *what* they can compute. Such a mechanism would reconcile the conflict between our desire to outsource and compute on data and the need to protect the data.

2 Functional Encryption

Over the past decade, cryptographers have put forth a novel paradigm for public-key encryption [140, 89, 33, 131] that addresses the above goal: **(i)** *attribute-based encryption* (ABE), which enables fine-grain access control, and **(ii)** its generalization to *functional encryption*, which enables selective computation.

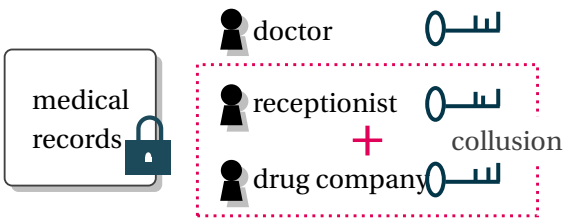
- In **attribute-based encryption (ABE)**, encrypted data are associated with a set of attributes and secret keys with policies that control which ciphertexts the key can decrypt. For instance, a digital

content provider can issue keys that decrypt basic and premium channel contents on weekdays and only basic ones on weekends.

- In **functional encryption**, a secret key enables a user to learn a specific function of the encrypted data and nothing else. For example, decrypting an encrypted image with a cropping key will reveal a cropped version of the image and nothing else about the image.

A salient feature of both attribute-based and functional encryption is that there are many possible secret keys with different decryption capabilities. Moreover, the keys are resilient to collusion attacks, namely any group of users holding different secret keys learns nothing about the plaintext beyond what each of them could individually learn. Together, attribute-based and functional encryption constitute a crisp generalization of several advanced notions of encryption, such as broadcast and identity-based encryption as well as searching on encrypted data; indeed, many advances in public-key encryption over the past decade can be viewed as special cases of attribute-based and functional encryption.

As a concrete application, consider an encrypted database of medical records. With functional encryption, we can create customized keys for a doctor to obtain medical records for her patients, for a receptionist to retrieve appointment history, and for drug companies to collect anonymized aggregate statistics. On the other hand, even a collusion of a receptionist and a drug company should not able to compromise any individual medical record.



State of the art

The fundamental goals in the study of attribute-based and functional encryption are two-fold: **(i)** to build expressive schemes that support a large class of policies and functions; and **(ii)** to obtain efficient instantiations based on widely-believed intractability of basic computational problems.

The simplest example of attribute-based encryption (ABE) is that of identity-based encryption (IBE), where both the ciphertext and secret key are associated with identities i.e. bit strings, and decryption is possible exactly when the identities are equal. Starting with identity-based encryption (IBE), substantial advances in ABE were made over the past decade showing how to support fairly expressive but nonetheless limited subset of policies, culminating most recently in schemes supporting any policy computable by general circuits [86, 34].

In addition, we have a wide spectrum of techniques for efficient IBE and ABE that yields various trade-offs between efficiency, expressiveness, security and intractability assumptions. The specific assumptions in use may be broadly classified into two categories: (i) pairing-based, such as variants of the Diffie-Hellman problem over bilinear groups, and (ii) lattice-based, notably the learning with errors (LWE) assumption.

Beyond ABE, our understanding of functional encryption is much more limited. The only efficient schemes we have are for very simple functionalities related to computing an inner product [106]. In

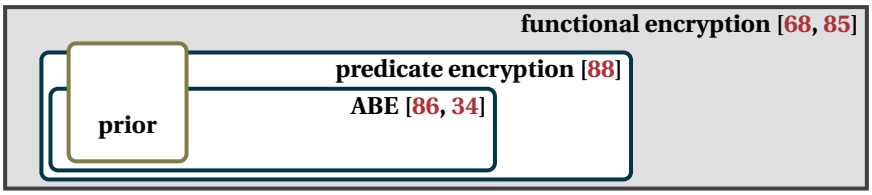


Figure 1: Advances in attribute-based and functional encryption since 2012. The white region refers to ABE and functionalities for which we have efficient instantiations under standard assumptions; the grey region refers to functionalities beyond ABE for which our understanding is much more limited.

a recent break-through work, Garg et al. [68] gave a beautiful construction of functional encryption for general circuits; however, the construction relies on "multi-linear maps", for which we have few candidates, along with complex intractability assumptions which are presently poorly understood. In contrast, if we consider collusions of a priori *bounded* size, a weaker guarantee that is still meaningful for many applications, then it is possible to obtain functional encryption for general circuits under a large class of standard assumptions.

Along with these cryptographic advances, the community has also made a greater push towards implementation, prototypes and deployment of attribute-based and functional encryption: several IBE schemes are now standardized in RFC 5091; the CHARM project provides a Python framework for rapidly prototyping cryptosystems and includes implementations of several IBE and ABE schemes; the SHARPS project explores the use of ABE for protecting health-care data; the Mylar project presents a web application platform that uses ABE to provide fine-grained access to encrypted data.

3 My Contributions

My research over the past five years addresses the fundamental goals of building **(i)** expressive schemes that support a large class of policies and functions, along with **(ii)** efficient instantiations, both based on widely-believed intractability of basic computational problems. In particular, my research has significantly advanced the state-of-the-art vis-a-viz expressiveness (c.f. Figure 6).

Expressiveness. In joint work with Gorbunov and Vaikuntanathan [86], I presented the first attribute-based encryption scheme that supports the class of *all* circuits (i.e. polynomial-time computable policies), resolving a central open problem in the area. Our scheme resists collusion attacks, as long as a well-studied problem in lattices remains intractable —which is widely believed to be true, even against quantum computers— and provides further evidence of the power of lattice-based cryptography. Our work also provided the critical building block towards a resolution of the 25-year-old open problem of constructing reusable garbled circuits [82]. In addition, our construction yields a so-called publicly verifiable delegation scheme: that is, a computationally weak client can delegate any arbitrarily expensive computation to the cloud, with the assurance that the computation is done correctly. In a subsequent work [88], we built ABE for circuits that achieve the additional security guarantee of attribute privacy, providing the first construction of so-called predicate encryption for all circuits. This result can

also be viewed as realizing a weaker one-sided variant of functional encryption. Together, these two works constitute the frontier of attribute-based and functional encryption under standard intractability assumptions.

In [85], Gorbunov, Vaikuntanathan and I put forth and investigated functional encryption secure against collusions of an a priori *bounded* size, a weaker guarantee that is still meaningful for many applications. We presented a construction that supports the class of all (polynomial-time computable) *functions*, which is more general than policies as it supports revealing partial information about the plaintext. Our construction relies on a novel connection to secure multi-party computation, a well-studied area in cryptography, and also presents a powerful technique for bootstrapping from shallow circuits of small depth to arbitrary circuits via fully homomorphic encryption, which has been used in several subsequent works [82, 68, 88]. Moreover, our construction achieves strong simulation-based security, which we later demonstrated in [11] to be impossible to achieve for unbounded collusions.

Efficiency. Together with several collaborators and my PhD students, I have also been working on efficient ABE schemes based on pairing groups. This is motivated in part by the fact that pairing groups are in use in many cryptographic standards and implementations, including applications beyond ABE. The use of pairing groups does come at a cost: we seem limited in expressiveness to ABE for shallow circuits, which are nonetheless sufficient for many applications. An example of such an ABE is identity-based encryption (IBE), where both the ciphertext and secret key are associated with identities e.g. email addresses, and decryption is possible only when the identities match.

In [148, 53], we put forth a new conceptual framework for building efficient ABE in pairing groups, by showing how to compile certain private-key primitive —which are much easier to design and to analyze— into a public-key one via Waters' "dual system" methodology [146]. This provides a simple and unifying approach for constructing efficient ABE in pairing groups that achieve a very strong guarantee of adaptive security; in addition, we obtained concrete efficiency improvements for several ABE schemes. In yet another work [71], we used the framework to explain why further efficiency improvements seem unlikely: we showed that the trade-offs between ciphertext and key sizes in existing ABE schemes are in some sense almost optimal; we obtained our result via a new connection to communication complexity, a well-studied area in theoretical computer science.

Our framework also uncovered a new connection between IBE schemes and pseudo-random functions; this connection in turn inspired several new IBE schemes. In [48], we constructed the first IBE whose performance does not deteriorate with the number of secret keys the adversary sees, thereby resolving an open problem posed in several prior works; along the way, we showed how to overcome seemingly inherent limitations of prior proof techniques. The ideas and techniques developed in this work have already been used in a number of follow-up works, e.g. [25, 117, 95, 73]. In [150], I built an IBE of essentially optimal efficiency, albeit in the less efficient composite-order pairing groups. Nonetheless, the construction suggests a path towards a 50% improvement in the state-of-the-art IBE schemes in prime-order groups.

In two other recent works [108, 109], we leveraged insights from the framework for applications beyond IBE and ABE. We obtained improved constructions of powerful building blocks for advanced pairing-based cryptographic primitives such as anonymous credentials. Our approach is conceptually

different from those of prior works, and yields simpler schemes that admit a modular and intuitive proof of security. In joint work with Gay et al. [73] building upon our earlier IBE scheme [48], we constructed a CCA-secure encryption scheme under the standard Diffie-Hellman assumption whose performance does not deteriorate with the number of challenge ciphertexts or decryption queries.

Organization. In the rest of this thesis, I provide a more detailed exposition of my works. In Section 4, I describe the assumptions used for efficient IBE and ABE. In Sections 5 and 6, I present two results on efficient IBE and ABE schemes. In Sections 7 and 8, I present two results on applications of IBE-inspired techniques. In Section 9, I describe additional results and contributions in the field of functional encryption. Finally, in Parts III, IV, V and VI, I attached several articles corresponding to my most significant contributions in functional encryption, of which two were joint works with my PhD students.

4 Computational Assumptions

As described in Section 2, there are two main specific assumptions used for efficient IBE and ABE, notably the decisional k-Linear (k-Lin) assumption —of which the standard Diffie-Hellman assumption is a special case with $k = 1$— used in pairing-based schemes, and the learning with errors (LWE) assumption used in lattice-based schemes. Both assumptions essentially stipulate that random linear equations of a random secret unknown vector $\mathbf{s}$ are computationally indistinguishable from random, namely:

$$\left(\boxed{\mathbf{A}}, \boxed{\mathbf{A}}\boxed{\mathbf{s}}\right) \approx_c \left(\boxed{\mathbf{A}}, \boxed{\mathbf{z}}\right)$$

where $\mathbf{A}, \mathbf{s}, \mathbf{z}$ denote uniformly random matrices and vectors over $\mathbb{Z}_q$. Of course, such a statement is blatantly false thanks to Gaussian elimination, and the two specific assumptions refers to two computational settings for which we do not know how to carry out Gaussian elimination efficiently:

- The k-Lin assumption refers to random linear equations *in the exponent* of a cyclic group of prime order q, that is, both $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times k}, \mathbf{As} \in \mathbb{Z}_q^{k+1}$ are computed in the exponent component-wise. Concretely, we will typically work with a pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We adopt the implicit representation notation for group elements: for fixed generators g_1 and g_2 of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and for a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times t}$, we define $[\mathbf{M}]_1 := g_1^{\mathbf{M}}$ and $[\mathbf{M}]_2 := g_2^{\mathbf{M}}$ (component-wise).
- The LWE assumption refers to *noisy* random linear equations $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ where $\mathbf{e}$ is a small "noise" vector whose entries are bounded by $B \ll q$. LWE hardness depends on the modulus-to-noise ratio q/B as a function of the length n of the secret $\mathbf{s}$: the smaller the ratio, the harder the problem. LWE is believed to be hard even for q/B as large as 2^{n^ϵ} for some $\epsilon < 1$.

5 Attribute-Based Encryption for Circuits

— "*Breakthrough result on a mainstream problem in cryptography*" (STOC 13 reviewer)

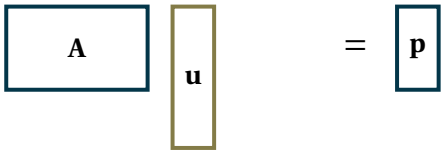
In an attribute-based encryption (ABE) scheme, a ciphertext is associated with an ℓ-bit *public index* ind and a message m, and a secret key is associated with a Boolean predicate P. The secret key allows to decrypt the ciphertext and learn m iff $P(\mathsf{ind}) = 1$. Moreover, the scheme should be secure against collusions of users, namely, given secret keys for polynomially many predicates, an adversary learns nothing about the message if none of the secret keys can individually decrypt the ciphertext.

We present attribute-based encryption schemes for circuits of any arbitrary polynomial size, where the public parameters and the ciphertext grow linearly with the depth of the circuit. Our construction is secure under the standard learning with errors (LWE) assumption. Previous constructions of attribute-based encryption were for Boolean formulas, captured by the complexity class NC^1.

In the course of our construction, we present a new framework for constructing ABE schemes. As a by-product of our framework, we obtain ABE schemes for polynomial-size branching programs, corresponding to the complexity class $\mathsf{LOGSPACE}$, under quantitatively better assumptions.

5.1 Trapdoor functions

Informally, a trapdoor function is a function that is easy to evaluate and hard to invert on its own, but which can be generated together with some extra "trapdoor" information that makes inversion easy. The prototypical candidate trapdoor function is the RSA function $f_{N,e}(x) = x^e \pmod N$, where N is the product of distinct primes p, q, and $\gcd(e, \psi(N)) = 1$. Under the LWE assumption, we may also derive a candidate trapdoor function given by $f_{\mathbf{A}}(\mathbf{u}) = \mathbf{A}\mathbf{u}$ [77], which may be represented pictorially as:



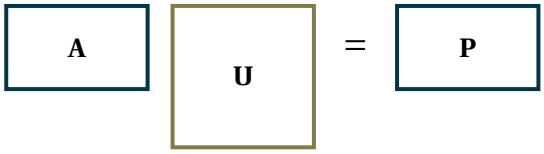
We should think of $\mathbf{A}$ and $\mathbf{p}$ as matrices and vectors over $\mathbb{Z}_q$ with uniformly random entries, whereas we restrict $\mathbf{u}$ to be a "short" vector with small entries; without the restriction, we may trivially solve for $\mathbf{u}$ satisfying $\mathbf{A}\mathbf{u} = \mathbf{p}$ via Gaussian elimination. It follows from the LWE assumption that given a random $\mathbf{A}, \mathbf{p}$, finding a short $\mathbf{u}$ satisfying $\mathbf{A}\mathbf{u} = \mathbf{p}$ is hard; on the other hand, it is possible to sample $\mathbf{A}$ along with a trapdoor for which finding such a $\mathbf{u}$ is easy.

Note that $\mathbf{u}$ allows us to "recode" a noisy version of $\mathbf{A}^\top\mathbf{s}$ to that of $\mathbf{p}^\top\mathbf{s}$ via the relation:

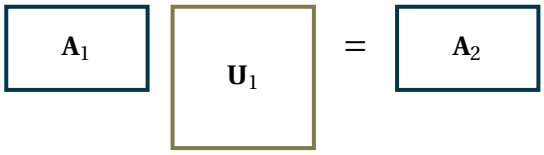
$$\mathbf{u}^\top(\mathbf{A}^\top\mathbf{s} + \overbrace{\mathbf{e}}^{\text{small}}) = \mathbf{p}^\top\mathbf{s} + \overbrace{\mathbf{u}^\top\mathbf{e}}^{\text{small}}$$

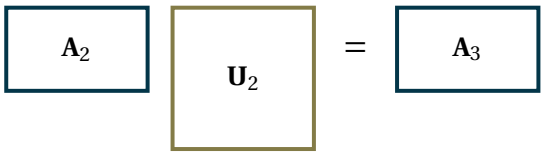
where the noise blows up slightly from $\mathbf{e}$ to $\mathbf{u}^\top\mathbf{e}$.

In our ABE for circuits based on LWE [86], we essentially considered a variant of the above trapdoor function in which we replaced the vectors $\mathbf{u}, \mathbf{p}$ with matrices $\mathbf{U}, \mathbf{P}$, where $\mathbf{P}$ has the same dimensions as $\mathbf{A}$ and $\mathbf{U}$ is short; this may be represented pictorially as:



We may then associate $\mathbf{P}$ with another trapdoor function $f_{\mathbf{P}}$. More generally, our construction uses a sequence of trapdoor functions $f_{\mathbf{A}_1}, f_{\mathbf{A}_2}, f_{\mathbf{A}_3}, \ldots$ along with short matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots$ such that





where the matrix $\mathbf{A}_i$ is associated with wires at level i of a circuit. Observe that given $\mathbf{U}_1, \mathbf{U}_2$, we may "recode" a noisy version of $\mathbf{A}_1^\top \mathbf{s}$ to that of $\mathbf{A}_3^\top \mathbf{s}$ via the relation:

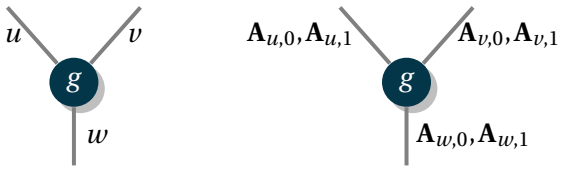
$$\mathbf{U}_2^\top \mathbf{U}_1^\top (\mathbf{A}_1^\top \mathbf{s} + \overbrace{\mathbf{e}}^{\text{small}}) = \mathbf{A}_3^\top \mathbf{s} + \overbrace{\mathbf{U}_2^\top \mathbf{U}_1^\top \mathbf{e}}^{\text{small}}$$

where the noise blows up slightly from $\mathbf{e}$ to $\mathbf{U}_2^\top \mathbf{U}_1^\top \mathbf{e}$. In the setting of bilinear groups, the quantities corresponding to $\mathbf{A}, \mathbf{u}, \mathbf{p}$ lie in different groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ respectively whereas matrix-vector multiplication correspond to a pairing. As such, defining an analogous sequence of of trapdoor functions would require the use of multi-linear maps.

5.2 Handling circuits

Our ABE for circuits essentially proceeds by replacing labels in Yao's garbled circuits with trapdoor functions. The underlying intuition is that labels are single-use and therefore susceptible to collusion attacks, whereas functions are reusable and resists collusions.

Handling a single gate. Consider a two-input boolean gate with incoming wires u, v and outgoing wire w computing a function $g : \{0,1\} \times \{0,1\} \to \{0,1\}$:



In garbled circuits, we associate each wire w with a pair of strings $L_{w,0}, L_{w,1}$ (called "labels") and we provide a translation table comprising of four values $\mu_{0,0}, \mu_{0,1}, \mu_{1,0}, \mu_{1,1}$ where $\mu_{b,c}$ allows us to perform

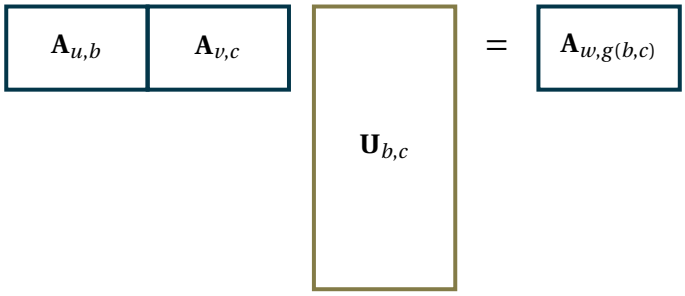
the transformation:

$$L_{u,b}, L_{v,c} \mapsto L_{w,g(b,c)}$$

In our construction, we associate each wire w in a circuit with a pair of matrices $\mathbf{A}_{w,0}, \mathbf{A}_{w,1}$. For each gate, we publish four "short" matrices $\mathbf{U}_{0,0}, \mathbf{U}_{0,1}, \mathbf{U}_{1,0}, \mathbf{U}_{1,1}$ (analogous to the translation table in garbled circuits) satisfying

$$\begin{aligned}
(\mathbf{A}_{u,0} \mid \mathbf{A}_{v,0})\mathbf{U}_{0,0} &= \mathbf{A}_{w,g(0,0)} \\
(\mathbf{A}_{u,0} \mid \mathbf{A}_{v,1})\mathbf{U}_{0,1} &= \mathbf{A}_{w,g(0,1)} \\
(\mathbf{A}_{u,1} \mid \mathbf{A}_{v,0})\mathbf{U}_{1,0} &= \mathbf{A}_{w,g(1,0)} \\
(\mathbf{A}_{u,1} \mid \mathbf{A}_{v,1})\mathbf{U}_{1,1} &= \mathbf{A}_{w,g(1,1)}
\end{aligned}$$

That is, for all $b, c \in \{0,1\}$, we have



Observe that given $\mathbf{U}_{b,c}$, we may recode noisy versions of $\mathbf{A}_{u,b}^\top \mathbf{s}$ and $\mathbf{A}_{v,c}^\top \mathbf{s}$ to that of $\mathbf{A}_{w,g(b,c)}^\top \mathbf{s}$ via the relation:

$$\mathbf{U}_{b,c}^\top \begin{pmatrix} \mathbf{A}_{u,b}^\top \mathbf{s} + \mathbf{e}_{u,b} \\ \mathbf{A}_{v,c}^\top \mathbf{s} + \mathbf{e}_{v,c} \end{pmatrix} = \mathbf{A}_{w,g(b,c)}^\top \mathbf{s} + \mathbf{U}_{b,c}^\top \begin{pmatrix} \mathbf{e}_{u,b} \\ \mathbf{e}_{v,c} \end{pmatrix}$$

For circuits of depth d, the noise grows from B to $n^{\Omega(d)} \cdot B$ so we need to set $q/B > n^{\Omega(d)}$. We also provided an ABE for branching programs where the noise grow is polynomial in the length of the branching program.

ABE for circuits. Recall that in an ABE scheme for circuits, a ciphertext is associated with an ℓ-bit *public index* ind and a message $m \in \{0,1\}$, and a secret key is associated with a circuit $C : \{0,1\}^\ell \to \{0,1\}$. The secret key decrypts the ciphertext iff $C(\mathsf{ind}) = 1$.

- The public parameters comprises of a matrix $\mathbf{A}$ together with ℓ pairs of matrices $(\mathbf{A}_{1,0}, \mathbf{A}_{1,1}), \ldots, (\mathbf{A}_{\ell,0}, \mathbf{A}_{\ell,1})$ for the ℓ input wires, and a pair of matrices $(\mathbf{A}_{\mathsf{out},0}, \mathbf{A}_{\mathsf{out},1})$ for the output wire.
- The ciphertext contains noisy versions of $\mathbf{A}^\top \mathbf{s}, \mathbf{A}_{1,\mathsf{ind}_1}^\top \mathbf{s}, \ldots, \mathbf{A}_{\ell,\mathsf{ind}_\ell}^\top \mathbf{s}$ along with a noisy version of $\mathbf{A}_{\mathsf{out},1}^\top \mathbf{s} + m \cdot q/2$.
- The secret key for a circuit C is generated as follows: (i) associate the wires going out of the i'th input bit with $(\mathbf{A}_{i,0}, \mathbf{A}_{i,1})$ for $i \in [\ell]$ and the output wire with $(\mathbf{A}_{\mathsf{out},0}, \mathbf{A}_{\mathsf{out},1})$, (ii) pick fresh random matrices $\mathbf{A}_{w,0}, \mathbf{A}_{w,1}$ for each internal wire w, (iii) for each internal gate g, compute and output the

four matrices $(\mathbf{U}^g_{0,0}, \mathbf{U}^g_{0,1}, \mathbf{U}^g_{1,0}, \mathbf{U}^g_{1,1})$.

Correctness follows from the fact that if wire w in C on input ind carries the bit b, then we can compute a noisy version of $\mathbf{A}^\top_{w,b}\mathbf{s}$. This holds trivially for the input wires by definition of the ciphertext, and the general case follows from the recoding properties for each gate. In particular, this means that if $C(\mathsf{ind}) = 1$, then we can compute a noisy version of $\mathbf{A}^\top_{\mathsf{out},1}\mathbf{s}$ and recover m. Security follows roughly by showing that the noisy version of $\mathbf{A}^\top_{w,1-b}\mathbf{s}$ is pseudorandom. This means that if $C(\mathsf{ind}) = 0$, then the noisy version of $\mathbf{A}^\top_{\mathsf{out},1}\mathbf{s}$ is pseudorandom and masks m.

5.3 Applications

Tools and techniques developed in the context of functional encryption have often found numerous applications beyond functional encryption. We outline two examples here.

Verifiable computation. In verifiable computation, a computationally weak client with input x wishes to delegate a complex computation f to an untrusted server, with the assurance that the server cannot convince the client to accept an incorrect computation [81, 74]. We focus on the online/offline setting, where the protocol proceeds in two phases. In the offline phase, the client sends to the server a possibly long message that may be expensive to compute. Later on, in the online phase (when the input x arrives), the client sends a short message to the server, and receives the result of the computation $f(x)$ together with a certificate for correctness. Applying an existing transformation [132] to our ABE for general circuits [86], we obtain a protocol for verifiable computation on general circuits f with a number of highly desirable properties: (i) the client's communication and computational complexity in the online phase depends only on the input/output lengths and depth of the circuit computing f but not the circuit size; (ii) anyone can check the server's work given a "verification" key published by the client; (iii) we may securely reuse the computation of the offline phase across multiple inputs in the online phase (in particular, our construction is immune to the "rejection problem" from [74]).

Fully homomorphic encryption. In 2009, Gentry [76] presented the first candidate fully homomorphic encryption (FHE) for all circuits, and substantial progress have since been made towards improving the efficiency and the underlying assumptions [37, 78]. We note that while both FHE and functional encryption support some form of computation on encrypted data, it is not known how to construct functional encryption from FHE or vice versa. Nonetheless, our lattice-based ABE for branching programs [86] has recently inspired the first FHE schemes based on the LWE assumption with a polynomial modulus-to-noise ratio [39, 14]. Roughly speaking, we propagate LWE samples across computation during decryption in ABE, and during homomorphic evaluation in FHE. If we compute on circuits, the noise accumulated in the LWE samples grows exponentially with the depth D of the circuit (the noise grows as n^D where n is the length of the LWE secret). On the other hand, by exploiting an asymmetry in computation on branching programs, it is possible to achieve noise growth that is linear in the length of the branching program. The latest FHE schemes in [39, 14] then use a branching program instead of a log-depth circuit to compute the decryption function during bootstrapping, thus incurring a polynomial as opposed to a quasi-polynomial noise growth.

6 Déjà Q: Encore! Un Petit IBE

— "*masterfully written*", "*extremely elegant*" (TCC 16A reviewers)

We present an identity-based encryption (IBE) scheme in composite-order bilinear groups with essentially optimal parameters: the ciphertext overhead and the secret key are *one* group element each and decryption requires only *one* pairing. Our scheme achieves adaptive security and anonymity under standard decisional subgroup assumptions as used in Lewko and Waters (TCC '10). Our construction relies on a novel extension to the Déjà Q framework of Chase and Meiklejohn (Eurocrypt '14).

6.1 Introduction

In identity-based encryption (IBE) [142, 29], ciphertexts and secret keys are associated with identities, and decryption is possible only when the identities match. IBE has been studied extensively over the last decade, with a major focus on obtaining constructions that simultanously achieve short parameters and full adaptive security under static assumptions in the standard model. This was first achieved in the works of Lewko and Waters [146, 113], which also introduced the powerful dual system encryption methodology. The design of the Lewko-Waters IBE and the underlying proof techniques have since had a profound impact on both attribute-based encryption and pairing-based cryptography.

6.2 Our Contributions

In this work, we obtain the first efficiency improvement to the Lewko-Waters IBE in composite-order bilinear groups. We present an adaptively secure and anonymous identity-based encryption (IBE) scheme with essentially optimal parameters: the ciphertext overhead and the secret key are *one* group element each, and decryption only requires *one* pairing; this improves upon the Lewko-Waters IBE [113] in three ways: shorter parameters, faster decryption, and anonymity. Via Naor's transformation, we obtain a fully secure signature scheme where the signature is again only *one* group element. We stress that we achieve all of these improvements while relying on the same computational subgroup assumptions as in the Lewko-Waters IBE, notably in composite-order groups whose order is the product of three primes. We refer to Fig 10 for a comparison with prior works.

The Lewko-Waters IBE has played a foundational role in recent developments of IBE and more generally attribute-based encryption (ABE). Indeed, virtually all of the state-of-the-art prime-order IBE schemes in [111, 25] —along with the subsequent extensions to ABE [115, 148, 17, 53]— follow the basic design and proof strategy introduced in the Lewko-Waters IBE. For this reason, we are optimistic that our improvement to the Lewko-Waters IBE will lead to further advances in IBE and ABE. In fact, our improved composite-order IBE already hints at the potential of a more efficient prime-order IBE that subsumes all known schemes.

Scheme	|mpk|	|sk|	|ct|	decryption	anonymous	number of primes
TCC:LewWat10 [113]	$3\|G_N\|+\|G_T\|$	$2\|G_N\|$	$2\|G_N\|+\|G_T\|$	2 pairings	no	3
DIP10 [44]	$3\|G_N\|+\|G_T\|$	$2\|G_N\|$	$2\|G_N\|+\|G_T\|$	2 pairings	✓	4
YCZY14 [151]	$3\|G_N\|+\|G_T\|$	$2\|G_N\|$	$2\|G_N\|+\|G_T\|$	2 pairings	✓	4
this work (Fig 3)	$2\|G_N\|+\|G_T\|$	$\|G_N\|$	$\|G_N\|+\|G_T\|$	1 pairing	✓	3

Figure 2: Comparison amongst adaptively secure IBEs in composite-order bilinear groups $e : G_N \times G_N \to G_T$.

6.3 Our Techniques

The starting point of our constructions is the Déjà Q framework introduced by Chase and Meiklejohn [47]; this is an extension of Waters' dual system techniques to eliminate the use of q-type assumptions in settings beyond the reach of previous techniques. These settings include deterministic primitives such as pseudo-random functions (PRF) and —quite remarkably— schemes based on the inversion framework [141, 27, 35]. However, the Déjà Q framework is also limited in that it cannot be applied to advanced encryption systems such as identity-based and broadcast encryption, where certain secret exponents appear in both ciphertexts and secret keys on both sides of the pairing. We show how to overcome this limitation using several simple ideas.

IBE Overview. We describe our IBE scheme and the security proof next. We present a simplified variant of the constructions, suppressing many details pertaining to randomization and subgroups. Following the Lewko-Waters IBE [113], we rely on composite-order bilinear groups whose order N is the product of three primes p_1, p_2, p_3. We will use the subgroup G_{p_1} of order p_1 for functionality, and the subgroup G_{p_2} of order p_2 in the proof of security. The third subgroup corresponding to p_3 is used for additional randomization.

Recall that the Lewko-Waters IBE has the following form:

$$\mathsf{mpk} := (g, g^{\beta}, g^{\gamma}, e(g,u)),\ \mathsf{ct}_{\mathsf{id}} := (g^{s}, g^{(\beta+\gamma \mathsf{id})s}, e(g,u)^{s} \cdot m),\ \mathsf{sk}_{\mathsf{id}} := (u \cdot g^{(\beta+\gamma \mathsf{id})r}, g^{r}))$$

Our IBE scheme has the following form:

$$\mathsf{mpk} := (g, g^{\alpha}, e(g,u)),\ \mathsf{ct}_{\mathsf{id}} := (g^{(\alpha+\mathsf{id})s}, e(g,u)^{s} \cdot m),\ \mathsf{sk}_{\mathsf{id}} := (u^{\frac{1}{\alpha+\mathsf{id}}})$$

Note that our scheme uses the "exponent inversion" framework [35], which has traditionally eluded a proof of security under static assumptions. In both schemes, g, u are random group elements of order p_1, and α, β, γ are random exponents over $\mathbb{Z}_N$. It is easy to see that decryption in our scheme only requires a single pairing to compute $e(g^{(\alpha+\mathsf{id})s}, u^{\frac{1}{\alpha+\mathsf{id}}}) = e(g,u)^{s}$.

IBE security proof. We rely on the same assumption as the Lewko-Waters IBE in [113], namely the $(p_1 \mapsto p_1 p_2)$-subgroup assumption, which asserts that random elements of order p_1 and those of order $p_1 p_2$ are computationally indistinguishable. In the proof of security, we rely on the assumption to introduce random G_{p_2}-components to the ciphertext and the secret keys.

We begin with the secret keys. We introduce a random G_{p_2}-component to the secret key $\mathsf{sk}_{\mathsf{id}}$ following

$\underline{\mathsf{Setup}(\mathbb{G})}$: $\mathsf{msk} := (\alpha, u, g_3) \leftarrow_{\mathrm{R}} \mathbb{Z}_N \times G_{p_1} \times G_{p_3}^*$; $\mathsf{mpk} := (g_1, g_1^\alpha, e(g_1,u), \mathsf{H})$; return $(\mathsf{mpk}, \mathsf{msk})$ $\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id} \in \mathbb{Z}_N)}$: pick $R_3 \leftarrow_{\mathrm{R}} G_{p_3}$; return $\mathsf{sk}_{\mathsf{id}} := u^{\frac{1}{\alpha+\mathsf{id}}} R_3$	$\underline{\mathsf{Enc}(\mathsf{mpk}, \mathsf{id} \in \mathbb{Z}_N)}$: pick $s \leftarrow_{\mathrm{R}} \mathbb{Z}_N$; return $(\mathsf{ct}, \kappa) := (g_1^{(\alpha+\mathsf{id})s}, \mathsf{H}(e(g_1,u)^s))$ $\underline{\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct})}$: return $\mathsf{H}(e(\mathsf{ct}, \mathsf{sk}_{\mathsf{id}}))$

Figure 3: Adaptively secure anonymous IBE w.r.t. a composite-order bilinear group $\mathbb{G}$. Here, $\mathsf{H} : G_T \to \{0,1\}^\lambda$ is drawn from a family of pairwise-independent hash functions. In asymmetric groups, randomization with R_3 in KeyGen is not necessary (i.e., KeyGen is deterministic).

the Déjà Q framework [47] as follows:

$$\mathsf{sk}_{\mathsf{id}} = u^{\frac{1}{\alpha+\mathsf{id}}} \xrightarrow{\text{subgroup}} u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_1}{\alpha+\mathsf{id}}} \xrightarrow{\text{CRT}} u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_1}{\alpha_1+\mathsf{id}}}, \tag{1}$$

where $\alpha_1 \leftarrow \mathbb{Z}_N$. In the first transition, we use the $(p_1 \mapsto p_1 p_2)$-subgroup assumption which says that $u \approx_c u g_2^{r_1}, r_1 \leftarrow_{\mathrm{R}} \mathbb{Z}_N$, where g_2 is a generator of order p_2. In the second transition, we use the Chinese Reminder Theorem (CRT), which tell us α mod p_1 and α mod p_2 are independently random values, so we may replace α mod p_2 with α_1 mod p_2 for a fresh $\alpha_1 \leftarrow_{\mathrm{R}} \mathbb{Z}_N$; this is fine as long as the challenge ciphertext and mpk reveal no information about α mod p_2, as is the case here. We may then repeat this transition q more times:

$$\begin{array}{lllll} u^{\frac{1}{\alpha+\mathsf{id}}} & \xrightarrow{\text{subgroup}} & u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_1}{\alpha+\mathsf{id}}} & \xrightarrow{\text{CRT}} & u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_1}{\alpha_1+\mathsf{id}}} \\ & \xrightarrow{\text{subgroup}} & u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_2}{\alpha+\mathsf{id}}} g_2^{\frac{r_1}{\alpha_1+\mathsf{id}}} & \xrightarrow{\text{CRT}} & u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_2}{\alpha_2+\mathsf{id}}+\frac{r_1}{\alpha_1+\mathsf{id}}} \\ & \longrightarrow & \cdots & \xrightarrow{\text{CRT}} & u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\frac{r_{q+1}}{\alpha_{q+1}+\mathsf{id}}+\cdots+\frac{r_2}{\alpha_2+\mathsf{id}}+\frac{r_1}{\alpha_1+\mathsf{id}}} \end{array}$$

where $r_1, \ldots, r_{q+1}, \alpha_1, \ldots, \alpha_{q+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_N$, and q is an upper bound on the number of key queries made by the adversary.[1]

Next, we show that for distinct $x_1, \ldots, x_q$, the following matrix

$$\begin{pmatrix} \frac{1}{\alpha_1+x_1} & \frac{1}{\alpha_1+x_2} & \cdots & \frac{1}{\alpha_1+x_q} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_q+x_1} & \frac{1}{\alpha_q+x_2} & \cdots & \frac{1}{\alpha_q+x_q} \end{pmatrix} \tag{2}$$

is invertible with overwhelming probability over $\alpha_1, \ldots, \alpha_q \leftarrow_{\mathrm{R}} \mathbb{Z}_p$. As it turns out, we can write the determinant of this matrix explicitly as:

$$\frac{\Pi_{1 \le i < j \le q}(x_i - x_j)(\alpha_i - \alpha_j)}{\Pi_{1 \le i,j \le q}(\alpha_i + x_j)};$$

[1] We use $q+1$ values to account for the q key queries plus the challenge identity.

this is the only place in the proof where we crucially exploit the "exponent inversion" structure. We can then replace

$$\mathsf{id} \mapsto \frac{r_{q+1}}{\alpha_{q+1}+\mathsf{id}} + \cdots + \frac{r_2}{\alpha_2+\mathsf{id}} + \frac{r_1}{\alpha_1+\mathsf{id}}$$

by a truly random function $\mathsf{RF}(\cdot)$. Indeed, $\mathsf{sk}_{\mathsf{id}}$ can now be written as $u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\mathsf{RF}(\mathsf{id})}$, which have independently random G_{p_2}-components.

So far, what we have done is the same as the use of Déjà Q framework for showing that $x \mapsto u^{\frac{1}{x+\alpha}}$ yields a PRF [47] (the explicit formula for the matrix determinant is new), and this is where the similarity ends. At this point, we still need to hide the message m in the ciphertext $(g^{(\alpha+\mathsf{id})s}, e(g,u)^s \cdot m)$. Towards this goal, we want to introduce a G_{p_2}-component into the ciphertext, which will then interact with newly random G_{p_2}-component in the keys to generate extra statistical entropy to hide m. At the same time, we need to ensure that the ciphertext still hides $\alpha \bmod p_2$ so that we may carry out the transition of the secret keys in (1). Indeed, naively applying the $(p_1 \mapsto p_1 p_2)$-subgroup assumption to g^s in the ciphertext would leak $\alpha \bmod p_2$.

To circumvent this difficulty, note that we can rewrite the ciphertext in terms of $\mathsf{sk}_{\mathsf{id}}$ as

$$\mathsf{ct}_{\mathsf{id}} = (g^{(\alpha+\mathsf{id})s}, e(g^{(\alpha+\mathsf{id})s}, \mathsf{sk}_{\mathsf{id}}) \cdot m)$$

Moreover, as long as $\alpha + \mathsf{id} \neq 0$, we can replace $(\alpha+\mathsf{id})s$ with s without changing the distribution, which allows us to rewrite the challenge ciphertext as

$$\mathsf{ct}_{\mathsf{id}} = (g^s, e(g^s, \mathsf{sk}_{\mathsf{id}}) \cdot m).$$

This means that the challenge ciphertext leaks no information about α except through $\mathsf{sk}_{\mathsf{id}}$. In addition, the challenge ciphertext also leaks no information about id, which allows us to prove anonymity. In contrast, the Lewko-Waters IBE is not anonymous, and anonymous variants there-of in [44, 151] requires the use of 4 primes and additional assumptions.

We can now apply the $(p_1 \mapsto p_1 p_2)$-subgroup assumption to the ciphertext to replace g^s with $g^s g_2^{r'}$. Now, the ciphertext distribution is completely independent of α except what is leaked through $\mathsf{sk}_{\mathsf{id}}$, so we can apply the secret key transitions as before, at the end of which the challenge ciphertext is given by:

$$(g^s g_2^{r'}, e(g^s g_2^{r'}, u^{\frac{1}{\alpha+\mathsf{id}}} g_2^{\mathsf{RF}(\mathsf{id})}) \cdot m) = (g^s g_2^{r'}, e(g^s, u^{\frac{1}{\alpha+\mathsf{id}}}) \cdot \boxed{e(g_2^{r'}, g_2^{\mathsf{RF}(\mathsf{id})})} \cdot m)$$

Recall that we only allow the adversary to request for secret keys corresponding to identities different from id, which means those keys leak no information about $\mathsf{RF}(\mathsf{id})$. We can then use the $\log p_2$ bits of entropy from $\mathsf{RF}(\mathsf{id})$ over G_{p_2} to hide m; this requires modifying the original scheme so that an encryption of m is given by $(g^{(\alpha+\mathsf{id})s}, \mathsf{H}(e(g,u)^s) \cdot m)$, where H denotes a strong randomness extractor whose seed is specified in mpk.

7 Quasi-Adaptive NIZK for Linear Subspaces Revisited

— "*very refreshing ... core lemma is fantastic*" (EUROCRYPT 15 reviewer)

Non-interactive zero-knowledge (NIZK) proofs for algebraic relations in a group, such as the Groth-Sahai proofs, are an extremely powerful tool in pairing-based cryptography. A series of recent works focused on obtaining very efficient NIZK proofs for linear spaces in a weaker quasi-adaptive model. We revisit recent quasi-adaptive NIZK constructions, providing clean, simple, and improved constructions via a conceptually different approach inspired by recent developments in identity-based encryption. We then extend our techniques also to linearly homomorphic structure-preserving signatures, an object both of independent interest and with many applications.

7.1 Introduction

Non-interactive zero-knowledge (NIZK) proofs for efficiently proving algebraic relations in a group [93, 94, 91, 26] have had a profound impact on pairing-based cryptography, notably in (i) improving the concrete efficiency of non-interactive cryptography schemes like group signatures [92], (ii) realizing stronger security guarantees in applications like anonymous credentials [19, 20, 65], and (iii) minimizing interaction in secure computation and two-party protocols [105, 63].

A recent fruitful line of works has focused in obtaining very efficient NIZK proofs for proving membership in a linear subspace over a group, which is an important subset of the algebraic relations supported by the Groth-Sahai NIZK [93]. For linear subspaces, the Groth-Sahai proofs were linear in the dimensions of the (sub)space. The first substantial improvement was obtained by Jutla and Roy [102] in a weaker *quasi-adaptive* model, where the CRS may depend on the linear subspace, and the soundness guarantee is computational but adaptive. In addition, they used quasi-adaptive NIZK (QANIZK) for linear subspaces to obtain improved KDM-CCA2-secure encryption as well as CCA2-secure IBE scheme with short, publicly verifiable ciphertexts [41, 43]. Further efficiency improvements were subsequently obtained in [118, 103, 2], leading to constant-size proofs, independent of the dimensions of space and subspace; several of these constructions also realized stronger notions of soundness like one-time simulation soundness and unbounded simulation soundness [138, 60], which in turn enable new applications.

7.2 Our Results and Techniques: QANIZK

We present clean, simple, and improved constructions of QANIZK protocols via a conceptually novel approach. Previous constructions use fairly distinct techniques, resulting in a large family of schemes with incomparable efficiency and security guarantees. We obtain a family of schemes that simultaneously match – and in many settings, improve upon – the efficiency, assumptions, and security guarantees of all of the previous constructions. Figure 4 summarizes the efficiency of our constructions. Like the earliest Jutla-Roy scheme [102], our schemes are fully explicit and simple to describe: the prover and verifier carry out simple matrix-vector products in the exponent, and both correctness and zero-knowledge follow readily from one simple equation. Furthermore, our schemes have a natural derivation from a

symmetric-key setting, and the derivation even extends to a modular and intuitive proof of security. Finally, in all but the settings with unbounded security, we obtain a qualitative improvement in the underlying assumptions from decisional to computational (search) assumptions; specifically, security relies on a natural computational analogue of the decisional k-Lin assumption.

Our constructions and techniques are inspired by recent developments in obtaining adaptively secure identity-based encryption schemes, notably the use of pairing groups to "compile" a symmetric-key primitive into an asymmetric-key primitive [25, 148, 53], and the dual system encryption methodology for achieving adaptive security against unbounded collusions [146, 113]. We then extend our techniques to linearly homomorphic structure-preserving signatures [116, 118], an object both of independent interest and with many applications.

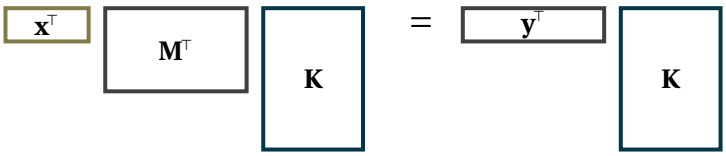
Overview of our constructions.

Fix a pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We present a very simple non-interactive argument system for linear subspaces over $\mathbb{G}_1$ as defined by a matrix $[\mathbf{M}]_1 := g_1^{\mathbf{M}} \in \mathbb{G}_1^{n \times t}$ ($n > t$) and captured by the language:

$$\mathcal{L}_{\mathbf{M}} = \left\{ [\mathbf{y}]_1 \in \mathbb{G}_1^n : \exists\, \mathbf{x} \in \mathbb{Z}_q^t \text{ s.t. } \mathbf{y} = \mathbf{M}\mathbf{x} \right\}.$$

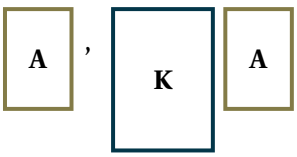
The starting point of our construction is a hash proof system [58] for the language, which is essentially a symmetric-key analogue of NIZK with a designated verifier. Namely, we pick a secret hash key $\mathbf{K} \leftarrow_R \mathbb{Z}_q^{n \times (k+1)}$ known to the verifier ($k \geq 1$ is a parameter of the security assumption) and publish the projection $[\mathbf{P}]_1 := [\mathbf{M}^\top \mathbf{K}]_1$ in the CRS. The proof is given by $[\pi]_1 := [\mathbf{x}^\top \mathbf{P}]_1$, and verification works by checking whether $\pi \stackrel{?}{=} \mathbf{y}^\top \mathbf{K}$. Completeness and perfect zero-knowledge follow readily from the fact that for all $\mathbf{y} = \mathbf{M}\mathbf{x}$ and $\mathbf{P} = \mathbf{M}^\top \mathbf{K}$:

$$\mathbf{x}^\top \mathbf{P} = \mathbf{x}^\top (\mathbf{M}^\top \mathbf{K}) = \mathbf{y}^\top \mathbf{K}.$$



Next, observe that if $\mathbf{y}$ is outside the span of $\mathbf{M}$, then $\mathbf{y}^\top \mathbf{K}$ is completely random given $\mathbf{M}^\top \mathbf{K}$; this is the case even if such a $\mathbf{y}$ is adaptively chosen after seeing $\mathbf{M}^\top \mathbf{K}$. Thus, the construction achieves statistical adaptive soundness: namely, a computationally unbounded cheating prover, upon seeing $\mathbf{P}$, still cannot produce a vector outside $\mathcal{L}_{\mathbf{M}}$ along with an accepting proof.

To achieve public verifiability, we carry out the hash proof system in $\mathbb{G}_1$ and publish a "partial commitment" to $\mathbf{K}$ in $\mathbb{G}_2$ as given by $[\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2$,



where the choice of $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times k}$ is defined by the security assumption. Instead of checking whether $\pi \stackrel{?}{=} \mathbf{y}^\top \mathbf{K}$ as before, anyone can now publicly check whether $\pi\mathbf{A} \stackrel{?}{=} \mathbf{y}^\top \mathbf{K}\mathbf{A}$ via a pairing. As $[\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2$ leaks additional information about the secret hash key $\mathbf{K}$, we can only prove computational adaptive soundness. In particular, we rely on the k-KerLin Assumption [123], which stipulates that given a random $[\mathbf{A}]_2$, it is hard to find a non-zero $[\mathbf{s}]_1 \in \mathbb{G}_1^{k+1}$ such that $\mathbf{s}^\top \mathbf{A} = \mathbf{0}$; this is implied by the k-Lin Assumption (c.f. Section 4). Therefore, for any $([\mathbf{y}]_1, [\pi]_1)$ produced by an efficient adversary,

$$\pi\mathbf{A} = \mathbf{y}^\top \mathbf{K}\mathbf{A} \Longrightarrow (\pi - \mathbf{y}^\top \mathbf{K})\mathbf{A} = \mathbf{0} \stackrel{\text{using assumption}}{\Longrightarrow} \pi - \mathbf{y}^\top \mathbf{K} = \mathbf{0} \Longrightarrow \pi = \mathbf{y}^\top \mathbf{K},$$

upon which we are back in the symmetric-key setting, with a little more work to account for the leakage about $\mathbf{K}$ from $\mathbf{K}\mathbf{A}$. Moreover, adaptive security in the symmetric-key setting (which is easy to analyze via a purely information-theoretic argument) carries over to adaptive security in the public-key setting.

Two simple extensions.

We extend this simple construction in two simple ways:

- First, we show that we can use $\mathbf{A}$ with the bottom row deleted, which saves one element to obtain proofs of size k, albeit at the cost of a more intricate security reduction and a restriction to witness-sampleable (WS) distributions for $[\mathbf{M}]_1$ [102]. The latter means that we are given an explicit description of $\mathbf{M}$ in the security reduction, which we need to program the CRS as with prior works [103, 2] that achieve the same proof size. In the case $k = 1$, the proof consists of 1 element and the CRS only contains $n + t$ group elements, which seems optimal.
- Second, we show how to achieve one-time simulation soundness, by replacing $\mathbf{K}$ with 2-wise independent hash function $\mathbf{K}_0 + \tau\mathbf{K}_1$ where τ is a tag, and we publish $[\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}_1\mathbf{A}]_2$ for public verification. A single simulated proof reveals only an evaluation of the hash function at a single point, while its evaluation at every other point remains hidden, upon which we are back in the setting of standard adaptive soundness.

8 Tightly CCA-secure Encryption without Pairings

— "*this year's best paper award for your work*" (EUROCRYPT 16 PC Chair)

We present the first CCA-secure public-key encryption scheme based on DDH where the security loss is independent of the number of challenge ciphertexts and the number of decryption queries. Our construction extends also to the standard k-Lin assumption in pairing-free groups, whereas all prior constructions starting with Hofheinz and Jager (Crypto '12) rely on the use of pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half (to only 3 group elements under DDH), in addition to eliminating the use of pairings.

	Soundness	WS?	Assumption	Proof	CRS	#pairings
GS08 [93]	AS		2-Lin ($\mathbb{G}_2$)	$2n+3t$	6	$3n(t+3)$
LPJY14 [118]	AS		2-KerLin ($\mathbb{G}_2$)	3	$2n+3t+3$	$2n+4$
ABP15 [2]	AS		k-Lin ($\mathbb{G}_2$)	$k+1$	$kn+(k+1)t+k$	$kn+k+1$
KW15 (Fig 4)	AS		k-KerLin ($\mathbb{G}_2$) ✓	$k+1$	$kn+(k+1)t+k$	$kn+k$ ✓
JR13 [102]	AS	yes	k-KerLin ($\mathbb{G}_2$)	$k(n-t)$	$2kt(n-t)+k+1$	$k(n-t)(t+2)$
JR14 [103]	AS	yes	k-Lin ($\mathbb{G}_2$)	k	$kn+kt+k^2$	$kn+k^2$
ABP15 [2]	AS	yes	k-Lin ($\mathbb{G}_2$)	k	$kn+kt+k$	$kn+k$
KW15 (Fig 5)	AS	yes	k-KerLin ($\mathbb{G}_2$) ✓	k	$kn+kt+k-1$ ✓	$kn+k-1$ ✓
ABP15 [2]	OTSS		k-Lin ($\mathbb{G}_2$)	$k+1$	$2kn+2(k+1)t+k$	$kn+k+1$
KW15 (Fig 6)	OTSS		k-KerLin ($\mathbb{G}_2$) ✓	$k+1$	$2kn+2(k+1)t+k$	$kn+k$ ✓
ABP15 [2]	OTSS	yes	k-Lin ($\mathbb{G}_2$)	k	$2\lambda(kn+(k+1)t)+k$	$\lambda kn+k$
KW15 (Fig 9)	OTSS	yes	k-KerLin ($\mathbb{G}_2$) ✓	k	$2\lambda(kn+(k+1)t)+k-1$ ✓	$\lambda kn+k-1$ ✓
CCS09 [41]	USS		2-Lin ($\mathbb{G}_1,\mathbb{G}_2$)	$2n+6t+52$	18	$O(tn)$
LPJY14 [118]	USS	yes	2-Lin ($\mathbb{G}_1,\mathbb{G}_2$)	20	$2n+3t+3\lambda+10$	$2n+30$
KW15 (Fig 7)	USS	yes	k-Lin ($\mathbb{G}_1,\mathbb{G}_2$) ✓	$2k+2$ ✓	$kn+4(k+t+1)k+2k$ ✓	$k(n+k+1)+k$ ✓

Figure 4: QANIZK for linear subspaces of $\mathbb{Z}_q^n$ of dimension t and tag-space $\{0,1\}^\lambda$. For the soundness column we use AS for adaptive soundness, OTSS for one-time simulation soundness, and USS for unbounded simulation soundness. WS stands for witness sampleability [102] and slightly restricts the class of languages. We omit the generators for the group when computing the CRS size. In all settings, we improve upon either the assumption, the CRS size, or # pairings used in verification (which can be further reduced using randomized verification), as indicated by a ✓.

We also show how to use our techniques in the NIZK setting. Specifically, we construct the first tightly simulation-sound designated-verifier NIZK for linear languages without pairings. Using pairings, we can turn our construction into a highly optimized publicly verifiable NIZK with tight simulation-soundness.

8.1 Introduction

The most basic security guarantee we require of a public key encryption scheme is that of semantic security against chosen-plaintext attacks (CPA) [80]: it is infeasible to learn anything about the plaintext from the ciphertext. On the other hand, there is a general consensus within the cryptographic research community that in virtually every practical application, we require semantic security against adaptive chosen-ciphertext attacks (CCA) [134, 61], wherein an adversary is given access to decryptions of ciphertexts of her choice.

In this work, we focus on the issue of security reduction and security loss in the construction of CPA and CCA-secure public-key encryption from the DDH assumption. Suppose we have such a scheme along with a security reduction showing that attacking the scheme in time t with success probability ϵ implies breaking the DDH assumption in time roughly t with success probability ϵ/L; we refer to L as the security loss. In general, L would depend on the security parameter λ as well as the number of challenge ciphertexts Q_{enc} and the number decryption queries Q_{dec}, and we say that we have a *tight security reduction* if L depends only on the security parameter and is independent of both Q_{enc} and Q_{dec}. Note that for typical settings of parameters (e.g., $\lambda = 80$ and $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{20}$, or even $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{30}$ in truly large settings), λ is much smaller than Q_{enc} and Q_{dec}.

In the simpler setting of CPA-secure encryption, the ElGamal encryption scheme already has a tight security reduction to the DDH assumption [124, 22], thanks to random self-reducibility of DDH with

a tight security reduction. In the case of CCA-secure encryption, the best result is still the seminal Cramer-Shoup encryption scheme [59], which achieves security loss Q_{enc}.[2] This raises the following open problem:

> Does there exist a CCA-secure encryption scheme with a tight security reduction to the DDH assumption?

Hofheinz and Jager [96] gave an affirmative answer to this problem under stronger (and pairing-related) assumptions, notably the 2-Lin assumptions in bilinear groups, albeit with large ciphertexts and secret keys; a series of follow-up works [117, 119, 18, 83] leveraged techniques introduced in the context of tightly-secure IBE [48, 25, 99] to reduce the size of ciphertext and secret keys to a relatively small constant. However, all of these works rely crucially on the use of pairings, and seem to shed little insight on constructions under the standard DDH assumption; in fact, a pessimist may interpret the recent works as strong indication that the use of pairings is likely to be necessary for tightly CCA-secure encryption.

We may then restate the open problem as eliminating the use of pairings in these prior CCA-secure encryption schemes while still preserving a tight security reduction. From a theoretical standpoint, this is important because an affirmative answer would yield tightly CCA-secure encryption under qualitatively weaker assumptions, and in addition, shed insight into the broader question of whether tight security comes at the cost of qualitative stronger assumptions.

Eliminating the use of pairings is also important in practice as it allows us to instantiate the underlying assumption over a much larger class of groups that admit more efficient group operations and more compact representations, and also avoid the use of expensive pairing operations. Similarly, tight reductions matter in practice because as L increases, we should increase the size of the underlying groups in order to compensate for the security loss, which in turn increases the running time of the implementation. Note that the impact on performance is quite substantial, as exponentiation in a r-bit group takes time roughly $\mathcal{O}(r^3)$.

8.2 Our Results

We settle the main open problem affirmatively: we construct a tightly CCA-secure encryption scheme from the DDH assumption without pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half, in addition to eliminating the use of pairings. We refer to Figure 14 for a comparison with prior works.

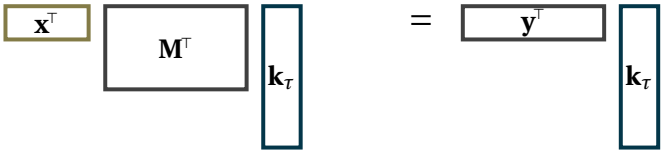
Overview of our construction. In this overview, we will consider a weaker notion of security, namely tag-based KEM security against plaintext check attacks (PCA) [126]. In the PCA security experiment, the adversary gets no decryption oracle (as with CCA security), but a PCA oracle that takes as input a tag and a ciphertext/plaintext pair and checks whether the ciphertext decrypts to the plaintext. Furthermore, we restrict the adversary to only query the PCA oracle on tags different from those used in the challenge ciphertexts. PCA security is strictly weaker than the CCA security we actually strive for, but allows us to present our solution in a clean and simple way. (We show how to obtain full CCA security separately.)

[2] We ignore contributions to the security loss that depend only on a statistical security parameter.

The starting point of our construction is the Cramer-Shoup KEM. The public key is given by $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_0], [\mathbf{M}^\top \mathbf{k}_1])$ for $\mathbf{M} \leftarrow_R \mathbb{Z}_q^{(k+1)\times k}$ corresponding to the matrix in the k-Lin assumption (c.f. Section 4). On input pk and a tag τ, the encryption algorithm outputs the ciphertext/plaintext pair

$$([\mathbf{y}], [z]) = ([\mathbf{M}\mathbf{x}], [\mathbf{x}^\top \mathbf{M}^\top \mathbf{k}_\tau]), \tag{3}$$

where $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ and $\mathbf{x} \leftarrow_R \mathbb{Z}_q^k$. Decryption relies on the fact that



The KEM is PCA-secure under k-Lin, with a security loss that depends on the number of ciphertexts Q (via a hybrid argument) but independent of the number of PCA queries [59, 3].

Following the "randomized Naor-Reingold" paradigm introduced by Chen and Wee on tightly secure IBE [48], our starting point is (4), where we replace $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ with

$$\mathbf{k}_\tau = \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$$

and $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_{j,b}]_{j=1,\ldots,\lambda,b=0,1})$, where $(\tau_1, \ldots, \tau_\lambda)$ denotes the binary representation of the tag $\tau \in \{0,1\}^\lambda$.

Following [48], we want to analyze this construction by a sequence of games in which we first replace $[\mathbf{y}]$ in the challenge ciphertexts by uniformly random group elements via random self-reducibility of MDDH (k-Lin), and then incrementally replace $\mathbf{k}_\tau$ in both the challenge ciphertexts and in the PCA oracle by $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, where RF is a truly random function and $\mathbf{m}^\perp$ is a random element from the kernel of $\mathbf{M}$, i.e., $\mathbf{M}^\top \mathbf{m}^\perp = 0$. Concretely, in Game i, we will replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)$ where RF_i is a random function on $\{0,1\}^i$ applied to the i-bit prefix of τ. We proceed to outline the two main ideas needed to carry out this transition. Looking ahead, note that once we reach Game λ, we would have replaced $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, upon which security follows from a straight-forward information-theoretic argument (and the fact that ciphertexts and decryption queries carry pairwise different τ).

First idea. First, we show how to transition from Game i to Game $i+1$, under the restriction that the adversary is only allowed to query the encryption oracle on tags whose $i+1$-st bit is 0; we show how to remove this unreasonable restriction later. Here, we rely on an *information-theoretic* argument similar to that of Cramer and Shoup to increase the entropy from RF_i to RF_{i+1}. This is in contrast to prior works which rely on a computational argument; note that the latter requires encoding secret keys as group elements and thus a pairing to carry out decryption.

More precisely, we pick a random function RF'_i on $\{0,1\}^i$, and implicitly define RF_{i+1} as follows:

$$\mathsf{RF}_{i+1}(\tau) = \begin{cases} \mathsf{RF}_i(\tau) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}'_i(\tau) & \text{if } \tau_{i+1} = 1 \end{cases}$$

Observe all of the challenge ciphertexts leak no information about RF'_i or $\mathbf{k}_{i+1,1}$ since they all correspond to tags whose $i+1$-st bit is 0. To handle a PCA query $(\tau, [\mathbf{y}], [z])$, we proceed via a case analysis:

- if $\tau_{i+1} = 0$, then $\mathbf{k}_\tau + \mathsf{RF}_{i+1}(\tau) = \mathbf{k}_\tau + \mathsf{RF}_i(\tau)$ and the PCA oracle returns the same value in both Games i and $i+1$.
- if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies in the span of $\mathbf{M}$, we have

  $$\mathbf{y}^\top \mathbf{m}^\perp = 0 \Longrightarrow \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)) = \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_{i+1}(\tau)),$$

  and again the PCA oracle returns the same value in both Games i and $i+1$.
- if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies outside the span of $\mathbf{M}$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random given $\mathbf{M}, \mathbf{M}^\top \mathbf{k}_{i+1,1}$. (Here, we crucially use that the adversary does not query encryptions with $\tau_{i+1} = 1$, which ensures that the challenge ciphertexts do not leak additional information about $\mathbf{k}_{i+1,1}$.) This means that $\mathbf{y}^\top \mathbf{k}_\tau$ is uniformly random from the adversary's view-point, and therefore the PCA oracle will reject with high probability in both Games i and $i+1$. (At this point, we crucially rely on the fact that the PCA oracle only outputs a *single* check bit and not all of $\mathbf{k}_\tau + \mathsf{RF}(\tau)$.)

Via a hybrid argument, we may deduce that the distinguishing advantage between Games i and $i+1$ is at most Q/q where Q is the number of PCA queries.

Second idea. Next, we remove the restriction on the encryption queries using an idea of Hofheinz, Koch and Striecks [99] for tightly-secure IBE in the multi-ciphertext setting, and its instantiation in prime-order groups [83]. The idea is to create two "independent copies" of $(\mathbf{m}^\perp, \mathsf{RF}_i)$; we use one to handle encryption queries on tags whose $i+1$-st bit is 0, and the other to handle those whose $i+1$-st bit is 1. We call these two copies $(\mathbf{M}_0^*, \mathsf{RF}_i^{(0)})$ and $(\mathbf{M}_1^*, \mathsf{RF}_i^{(1)})$, where $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{0}$.

Concretely, we replace $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1)\times k}$ with $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k\times k}$. We decompose $\mathbb{Z}_q^{3k}$ into the span of the respective matrices $\mathbf{M}, \mathbf{M}_0, \mathbf{M}_1$, and we will also decompose the span of $\mathbf{M}^\perp \in \mathbb{Z}_q^{3k\times 2k}$ into that of $\mathbf{M}_0^*, \mathbf{M}_1^*$. Similarly, we decompose $\mathbf{M}^\perp \mathsf{RF}_i(\tau)$ into $\mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$. We then refine the prior transition

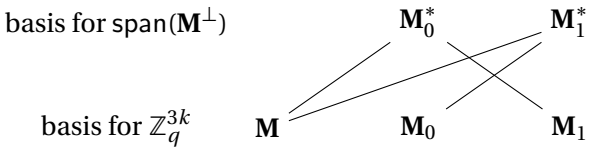


Figure 5: Solid lines mean orthogonal, that is: $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0} = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{M}_0^\top \mathbf{M}_1^*$.

from Games i to $i+1$ as follows:

- Game $i.0$ (= Game i): pick $\mathbf{y} \leftarrow \mathbb{Z}_q^{3k}$ for ciphertexts, and replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$;
- Game $i.1$: replace $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ with $\mathbf{y} \leftarrow_{\mathrm{R}} \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$;
- Game $i.2$: replace $\mathsf{RF}_i^{(0)}(\tau)$ with $\mathsf{RF}_{i+1}^{(0)}(\tau)$;

- Game $i.3$: replace $\mathsf{RF}_i^{(1)}(\tau)$ with $\mathsf{RF}_{i+1}^{(1)}(\tau)$;
- Game $i.4$ (= Game $i+1$): replace $\mathbf{y} \leftarrow_R \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ with $\mathbf{y} \leftarrow_R \mathbb{Z}_q^{3k}$.

For the transition from Game $i.0$ to Game $i.1$, we rely on the fact that the uniform distributions over $\mathbb{Z}_q^{3k}$ and $\mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ encoded in the group are computationally indistinguishable, even given a random basis for $\mathsf{span}(\mathbf{M}^\perp)$ (in the clear). This extends to the setting with multiple samples, with a tight reduction to the k-Lin Assumption independent of the number of samples.

For the transition from Game $i.1$ to $i.2$, we rely on an information-theoretic argument like the one we just outlined, replacing $\mathsf{span}(\mathbf{M})$ with $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ and $\mathbf{M}^\perp$ with $\mathbf{M}_0^*$ in the case analysis. In particular, we will exploit the fact that if $\mathbf{y}$ lies outside $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random even given $\mathbf{M}, \mathbf{M}\mathbf{k}_{i+1,1}, \mathbf{M}_1, \mathbf{M}_1\mathbf{k}_{i+1,1}$. The transition from Game $i.2$ to $i.3$ is completely analogous.

From PCA to CCA. Using standard techniques from [59, 110, 107, 32, 4], we could transform our basic tag-based PCA-secure scheme into a "full-fledged" CCA-secure encryption scheme by adding another hash proof system (or an authenticated symmetric encryption scheme) and a one-time signature scheme. However, this would incur an additional overhead of several group elements in the ciphertext. Instead, we show how to directly modify our tag-based PCA-secure scheme to obtain a more efficient CCA-secure scheme with the minimal additional overhead of a single symmetric-key authenticated encryption. In particular, the overall ciphertext overhead in our tightly CCA-secure encryption scheme is merely *one* group element more than that for the best known non-tight schemes [110, 97].

To encrypt a message M in the CCA-secure encryption scheme, we will (i) pick a random $\mathbf{y}$ as in the tag-based PCA scheme, (ii) derive a tag τ from $\mathbf{y}$, (iii) encrypt M using a one-time authenticated encryption under the KEM key $[\mathbf{y}^\top \mathbf{k}_\tau]$. The naive approach is to derive the tag τ by hashing $[\mathbf{y}] \in \mathbb{G}^{3k}$, as in [110]. However, this creates a circularity in Game $i.1$ where the distribution of $[\mathbf{y}]$ depends on the tag. Instead, we will derive the tag τ by hashing $[\overline{\mathbf{y}}] \in \mathbb{G}^k$, where $\overline{\mathbf{y}} \in \mathbb{Z}_q^k$ are the top k entries of $\mathbf{y} \in \mathbb{Z}_q^{3k}$. We then modify $\mathbf{M}_0, \mathbf{M}_1$ so that the top k rows of both matrices are zero, which avoids the circularity issue. In the proof of security, we will also rely on the fact that for any $\mathbf{y}_0, \mathbf{y}_1 \in \mathbb{Z}_q^{3k}$, if $\overline{\mathbf{y}}_0 = \overline{\mathbf{y}}_1$ and $\mathbf{y}_0 \in \mathsf{span}(\mathbf{M})$, then either $\mathbf{y}_0 = \mathbf{y}_1$ or $\mathbf{y}_1 \notin \mathsf{span}(\mathbf{M})$. This allows us to deduce that if the adversary queries the CCA oracle on a ciphertext which shares the same tag as some challenge ciphertext, then the CCA oracle will reject with overwhelming probability.

9 Additional Contributions

We describe our additional results and contributions in the field of functional encryption, beyond those covered in Sections 5 through 8 [87, 149, 108, 73].

Functional Encryption with Bounded Collusions via Multi-Party Computation [85, CRYPTO 12]

We construct a functional encryption scheme secure against an a-priori bounded polynomial number of collusions for the class of all polynomial-size circuits. Our constructions require only semantically secure public-key encryption schemes and pseudorandom generators computable by small-depth circuits (known to be implied by most concrete intractability assumptions). For certain special cases such as predicate encryption schemes with public index, the construction requires only semantically secure encryption schemes, which is clearly the minimal necessary assumption.

Functional Encryption: New Perspectives and Lower Bounds [11, CRYPTO 13]

Functional encryption is an emerging paradigm for public-key encryption that enables fine-grained control of access to encrypted data. In this work, we present new lower bounds and impossibility results on functional encryption, as well as new perspectives on security definitions. Our main contributions are as follows:

- We show that functional encryption schemes that satisfy even a weak (non-adaptive) simulation-based security notion are impossible to construct in general. This is the *first* impossibility result that exploits *unbounded* collusions in an essential way. In particular, we show that there are no such functional encryption schemes for the class of weak pseudo-random functions (and more generally, for any class of incompressible functions).

  More quantitatively, our technique also gives us a lower bound for functional encryption schemes secure against *bounded* collusions. To be secure against q collusions, we show that the ciphertext in any such scheme must have size $\Omega(q)$.

- We put forth and discuss a simulation-based notion of security for functional encryption, with an unbounded simulator (called USIM). We show that this notion interpolates indistinguishability and simulation-based security notions, and is inspired by results and barriers in the zero-knowledge and multi-party computation literature.

Fully, (Almost) Tightly Secure IBE and Dual System Groups [48, CRYPTO 13]

We present the first fully secure Identity-Based Encryption scheme (IBE) from the standard assumptions where the security loss depends only on the security parameter and is independent of the number of secret key queries. This partially answers an open problem posed by Waters (EUROCRYPT 2005). Our construction combines Waters' dual system encryption methodology (CRYPTO 2009) with the Naor-Reingold pseudo-random function (J. ACM, 2004) in a novel way. The security of our scheme relies on

the DLIN assumption in prime-order groups. Along the way, we introduce a novel notion of *dual system groups* and a new randomization and parameter-hiding technique for prime-order bilinear groups.

Dual System Encryption via Predicate Encodings [148, TCC 14]

We introduce the notion of *predicate encodings*, an information-theoretic primitive reminiscent of linear secret-sharing that in addition, satisfies a novel notion of reusability. Using this notion, we obtain a unifying framework for adaptively-secure public-index predicate encryption schemes for a large class of predicates. Our framework relies on Waters' dual system encryption methodology (CRYPTO '09), and encompass the identity-based encryption scheme of Lewko and Waters (TCC '10), and the attribute-based encryption scheme of Lewko et al. (Eurocrypt '10). In addition, we obtain several concrete improvements over prior works. Our work offers a novel interpretation of dual system encryption as a methodology for amplifying a one-time private-key primitive (i.e. predicate encodings) into a many-time public-key primitive (i.e. predicate encryption).

Partial Garbling Schemes and Their Applications [101, ICALP 14]

Garbling schemes (aka randomized encodings of functions) represent a function F by a "simpler" randomized function $\hat{F}$ such that $\hat{F}(x)$ reveals $F(x)$ and no additional information about x. Garbling schemes have found applications in many areas of cryptography. Motivated by the goal of improving the efficiency of garbling schemes, we make the following contributions:

- We suggest a general new notion of *partial garbling* which unifies several previous notions from the literature, including standard garbling schemes, secret sharing schemes, and "conditional disclosure of secrets". This notion considers garbling schemes in which part of the input is public, in the sense that it can be leaked by $\hat{F}$.
- We present constructions of partial garbling schemes for (boolean and arithmetic) formulas and branching programs which take advantage of the public input to gain better efficiency.
- We demonstrate the usefulness of the new notion by presenting applications to efficient attribute-based encryption, delegation, and secure computation. In each of these applications, we obtain either new schemes for larger classes of functions or efficiency improvements from quadratic to linear. In particular, we obtain the first ABE scheme in bilinear groups for arithmetic formulas, as well as more efficient delegation schemes for boolean and arithmetic branching programs.

Semi-Adaptive Attribute-Based Encryption and Improved Delegation for Boolean Formula [50, SCN 14]

We consider *semi-adaptive* security for attribute-based encryption, where the adversary specifies the challenge attribute vector after it sees the public parameters but before it makes any secret key queries. We present two constructions of semi-adaptive attribute-based encryption under static assumptions with *short* ciphertexts. Previous constructions with short ciphertexts either achieve the weaker notion of selective security, or require parameterized assumptions.

As an application, we obtain improved delegation schemes for Boolean formula with *semi-adaptive* soundness, where correctness of the computation is guaranteed even if the client's input is chosen adaptively depending on its public key. Previous delegation schemes for formula achieve one of adaptive soundness, constant communication complexity, or security under static assumptions; we show how to achieve semi-adaptive soundness and the last two simultaneously.

Predicate Encryption for Multi-Dimensional Range Queries from Lattices [72, PKC 15]

We construct a lattice-based predicate encryption scheme for multi-dimensional range and multi-dimensional subset queries. Our scheme is selectively secure and weakly attribute-hiding, and its security is based on the standard learning with errors (LWE) assumption. Multi-dimensional range and subset queries capture many interesting applications pertaining to searching on encrypted data. To the best of our knowledge, these are the first lattice-based predicate encryption schemes for functionalities beyond IBE and inner product.

Improved Dual System ABE in Prime-Order Groups via Predicate Encodings [53, EUROCRYPT 15]

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

Predicate Encryption for Circuits from LWE [88, CRYPTO 15]

In predicate encryption, a ciphertext is associated with descriptive attribute values x in addition to a plaintext μ, and a secret key is associated with a predicate f. Decryption returns plaintext μ if and only if $f(x) = 1$. Moreover, security of predicate encryption guarantees that an adversary learns nothing about the attribute x or the plaintext μ from a ciphertext, given arbitrary many secret keys that are not authorized to decrypt the ciphertext individually.

We construct a leveled predicate encryption scheme for all circuits, assuming the hardness of the subexponential learning with errors (LWE) problem. That is, for any polynomial function $d = d(\lambda)$, we construct a predicate encryption scheme for the class of all circuits with depth bounded by $d(\lambda)$, where λ is the security parameter.

Structure-Preserving Signatures from Standard Assumptions, Revisited [109, CRYPTO 15]

Structure-preserving signatures (SPS) are pairing-based signatures where all the messages, signatures and public keys are group elements, with numerous applications in public-key cryptography. We present new, simple and improved SPS constructions under standard assumptions via a conceptually different approach. Our constructions significantly narrow the gap between existing constructions from standard assumptions and optimal schemes in the generic group model.

Communication Complexity of Conditional Disclosure of Secrets and Attribute-Based Encryption [71, CRYPTO 15]

We initiate a systematic treatment of the communication complexity of conditional disclosure of secrets (CDS), where two parties want to disclose a secret to a third party if and only if their respective inputs satisfy some predicate. We present a general upper bound and the first non-trivial lower bounds for conditional disclosure of secrets. Moreover, we achieve tight lower bounds for many interesting setting of parameters for CDS with linear reconstruction, the latter being a requirement in the application to attribute-based encryption. In particular, our lower bounds explain the trade-off between ciphertext and secret key sizes of several existing attribute-based encryption schemes based on the dual system methodology.

10 Conclusion

New developments in cryptography tend to go hand-in-hand with the emergence of new computing technologies like smart phones, cloud computing and social networks. After all, the success of new computing technologies and paradigms hinges crucially on our ability to ensure security; indeed, e-commerce would not have thrived without secure online payments and public key cryptography. It is this synergistic and symbiotic relationship between cryptography and computing technologies that makes cryptography such an exciting area to work in: new computing technologies pose new cryptographic challenges, and solutions to these challenges facilitate adoption and deployment of these technologies.

My long-term vision is the ubiquitous use of functional encryption to secure our data and our computation, just as public-key encryption is widely used today to secure our communications. The use of functional encryption could help eliminate devastating privacy breaches and the prospect of massive digital surveillance, threats borne out by the Snowden leaks and several high-profile security breaches. Furthermore, functional encryption enables searches on encrypted travel records and surveillance video as well as medical studies on encrypted medical records in a privacy-preserving manner we can give law enforcement authorities and drug companies restricted secret keys that allow them to only learn the outcome of specific searches and tests. These mechanisms ensure that we can maintain public safety without compromising on civil liberties, and facilitate medical break-throughs without compromising on individual privacy. Realizing this vision requires further advances in the foundations of functional encryption, which is the goal of my future research.

Part II

Curriculum Vitae

Education

University of California, Berkeley
PHD IN COMPUTER SCIENCE, DEC 2007

Massachusetts Institute of Technology
BSC IN COMPUTER SCIENCE & IN MATHEMATICS, JUN 2002

Current Positions

CNRS - DI ENS, École Normale Supérieure
Chargé de recherche.

Columbia University
Adjunct Research Scientist at the Data Science Institute

Awards and Honors

- ERC Starting Grant, 2015 [ABSTRACT]
- ANR JCJC, 2014.
- Google Faculty Research Award, 2014 [LINK]
- Humboldt Research Fellowship for Experienced Researchers, 2012
- US NSF Faculty Early Career Development (CAREER) Award, 2010 [ABSTRACT]
  "*the National Science Foundation's most prestigious awards in support of junior faculty*"
- Tong Leong Lim Pre-Doctoral Prize, 2004.
- UC Berkeley Regents Fellowship, 2002-2003.
- William Lowell Putnam Examination: $1,000 award, 1999.

Publications

1. *Tightly CCA-Secure Encryption without Pairings,*
   R. Gay*, D. Hofheinz, E. Kiltz, and H. Wee.
   EUROCRYPT 2016: To appear. **BEST PAPER AWARD**
2. *The OPTLS Protocol and TLS 1.3,*
   H. Krawczyk, and H. Wee.
   IEEE European Symposium on Security and Privacy (**EuroS&P**) 2016: To appear.

3. *Déjà Q: Encore! Un Petit IBE,*
   H. Wee.
   Theory of Cryptography Conference (**TCC**) 2016-A: pp. 237–258

4. *Obfuscating Conjunctions under Entropic Ring LWE,*
   Z. Brakerski, V. Vaikuntanathan, D. Wichs, and H. Wee.
   Innovations in Theoretical Computer Science (**ITCS**) 2016: pp. 147–156.

5. *KDM-Security via Homomorphic Smooth Projective Hashing,*
   H. Wee.
   Public Key Cryptography (**PKC**) 2016: pp. 159–179.

6. *Predicate Encryption for Circuits from LWE,*
   S. Gorbunov*, V. Vaikuntanathan, and H. Wee.
   CRYPTO (2) 2015: pp. 503–523. Invited to CRYPTO **SPECIAL ISSUE** in Journal of Cryptology.

7. *Communication Complexity of Conditional Disclosure of Secrets and Attribute-Based Encryption,*
   R. Gay*, I. Kerenidis, and H. Wee.
   CRYPTO (2) 2015: pp. 485–502.

8. *Structure-Preserving Signatures from Standard Assumptions, Revisited,*
   E. Kiltz, J. Pan*, and H. Wee.
   CRYPTO (2) 2015: pp. 275–295.

9. *Implicit Zero-Knowledge Arguments and Applications to the Malicious Setting,*
   F. Benhamouda*, G. Couteau*, D. Pointcheval, and H. Wee.
   CRYPTO (2) 2015: pp. 107–129.

10. *Improved Dual System ABE in Prime-Order Groups via Predicate Encodings,*
    J. Chen, R. Gay*, and H. Wee.
    EUROCRYPT (2) 2015: pp. 595–624.

11. *Quasi-Adaptive NIZK for Linear Subspaces Revisited,*
    E. Kiltz, and H. Wee.
    EUROCRYPT (2) 2015: pp. 101–128.

12. *Security Against Related Randomness Attacks via Reconstructive Extractors,*
    K. G. Paterson, J. C. N. Schuldt, D. L. Sibborn*, and H. Wee.
    IMA International Conference in Cryptography and Coding (**IMACC**) 2015: pp. 23–40. **BEST PAPER AWARD**

13. *Predicate Encryption for Multi-Dimensional Range Queries from Lattices,*
    R. Gay*, P. Méaux*, and H. Wee.
    Public Key Cryptography (**PKC**) 2015: pp. 752–776.

14. *Semi-Adaptive Attribute-Based Encryption and Improved Delegation for Boolean Formula,*
J. Chen* and H. Wee.
Conference on Security and Cryptography for Networks (**SCN**) 2014: pp. 277–297.

15. *Partial Garbling Schemes and Their Applications,*
Y. Ishai and H. Wee.
ICALP (1) 2014: pp. 650–662.

16. *On the Complexity of UC Commitments,*
J. Garay, Y. Ishai, R. Kumaresan, and H. Wee.
EUROCRYPT 2014: pp. 677–694.

17. *Dual System Encryption via Predicate Encodings,*
H. Wee.
Theory of Cryptography Conference (**TCC**) 2014: pp. 262–279.

18. *Doubly Spatial Encryption from DBDH,*
J. Chen* and H. Wee.
Theoretical Computer Science 543: 79-89 (2014)

19. *On the Security of the TLS Protocol: A Systematic Analysis,*
H. Krawczyk, K. G. Paterson, and H. Wee.
CRYPTO (1) 2013: pp. 429–448.

20. *Fully, (Almost) Tightly Secure IBE and Dual Systems Groups,*
J. Chen* and H. Wee.
CRYPTO (2) 2013: pp. 435–460.

21. *Functional Encryption: New Perspectives and Lower Bounds,*
S. Agrawal, S. Gorbunov*, V. Vaikuntanathan, and H. Wee.
CRYPTO (2) 2013: pp. 500–518.

22. *Attribute-Based Encryption for Circuits,*
S. Gorbunov*, V. Vaikuntanathan, and H. Wee.
ACM Symposium on the Theory of Computation (**STOC**) 2013: pp. 545–554.
Invited to STOC **SPECIAL ISSUE** in SIAM Journal of Computing.
Journal of ACM (**JACM**): 62(6): 45: 1–33.

23. *Multi-Party Computation for Polynomials and Branching Programs without Simultaneous Interaction,*
S. Gordon, T. Malkin, M. Rosulek, and H. Wee.
EUROCRYPT 2013: pp. 575–591.

24. *Leakage-Resilient Cryptography from Minimal Assumptions,*
C. Hazay, A. López-Alt*, H. Wee, and D. Wichs.
EUROCRYPT 2013: pp. 160–176.

25. *Efficient, Adaptively Secure, and Composable Oblivious Transfer with a Single, Global CRS,*
S. G. Choi, J. Katz, H. Wee, and H. Zhou.
Public Key Cryptography (**PKC**) 2013: pp. 73–88.

26. *Functional Encryption with Bounded Collusions via Multi-Party Computation,*
S. Gorbunov*, V. Vaikuntanathan, and H. Wee.
CRYPTO 2012: pp. 162–179.

27. *Functional Encryption for Threshold Functions (or, Fuzzy IBE) from Lattices,*
S. Agrawal*, X. Boyen, V. Vaikuntanathan, P. Voulgaris*, and H. Wee.
Public Key Cryptography (**PKC**) 2012: pp. 262–279.

28. *Efficient Password Authenticated Key Exchange via Oblivious Transfer,*
R. Canetti, D. Dachman-Soled, V. Vaikuntanathan and H. Wee.
Public Key Cryptography (**PKC**) 2012: pp. 449–466.

29. *Public Key Encryption Against Related Key Attacks,*
H. Wee.
Public Key Cryptography (**PKC**) 2012: pp. 262–279.

30. *Shorter IBE and Signatures via Asymmetric Pairings,*
J. Chen*, H.W. Lim, S. Ling, H. Wang and H. Wee.
Pairing 2012.: pp. 122–140.

31. *Dual Projective Hashing and its Applications — Lossy Trapdoor Functions and More,*
H. Wee.
EUROCRYPT 2012: pp. 246–262.

32. *Lossy Trapdoor Functions from Homomorphic Reproducible Encryption,*
S. G. Choi and H. Wee.
Information Processing Letters 112:20 (2012) 794–798

33. *Threshold and Revocation Cryptosystems via Extractable Hash Proofs,*
H. Wee.
EUROCRYPT 2011: pp. 589–609.

34. *Black-Box, Round-Efficient Secure Computation via Non-Malleability Amplification,*
H. Wee.
IEEE Foundations of Computer Science (**FOCS**) 2010: pp. 531-540.

35. *Efficient Chosen-Ciphertext Security via Extractable Hash Proofs,*
H. Wee.
CRYPTO 2010: pp. 314–332.

36. *Constant-Round Non-Malleable Commitments from Sub-Exponential One-Way Functions,*
R. Pass and H. Wee.
EUROCRYPT 2010: pp. 638–655.

37. *Universal One-Way Hash Functions via Inaccessible Entropy,*
I. Haitner, T. Holenstein, O. Reingold, S. Vadhan, and H. Wee.
EUROCRYPT 2010: pp. 616–637.

38. *Encryption Schemes Secure Against Chosen-Ciphertext Selective Opening Attacks,*
S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee.
EUROCRYPT 2010: pp. 381–402.

39. *On the Round Complexity of Zero-Knowledge Proofs Based on One-Way Permutations,*
S. D. Gordon*, H. Wee, D. Xiao, and A. Yerukhimovich*.
Latincrypt 2010: pp. 189–204.

40. *Inaccessible Entropy,*
I. Haitner, O. Reingold, S. Vadhan, and H. Wee.
ACM Symposium on the Theory of Computation (**STOC**) 2009: pp. 611–620.

41. *Simple, Black-Box Constructions of Adaptively Secure Protocols,*
S. G. Choi*, D. Dachman-Soled*, T. Malkin, and H. Wee.
Theory of Cryptography Conference (**TCC**) 2009: pp. 387–402.

42. *Black-Box Constructions of Two-Party Protocols from One-Way Functions,*
R. Pass and H. Wee.
Theory of Cryptography Conference (**TCC**) 2009: pp. 403–418.

43. *Improved Non-Committing Encryption with Applications to Adaptively Secure Protocols,*
S. G. Choi*, D. Dachman-Soled*, T. Malkin, and H. Wee.
Asiacrypt 2009: pp.287–302.

44. *Zero Knowledge in the Random Oracle Model, Revisited,*
H. Wee.
Asiacrypt 2009: pp 417–434.

45. *Black-Box Construction of a Non-Malleable Encryption Scheme from Any Semantically Secure One,*
S. G. Choi*, D. Dachman-Soled*, T. Malkin, and H. Wee.
Theory of Cryptography Conference (**TCC**) 2008: pp. 427–444.

46. *Optimal Cryptographic Hardness of Learning Monotone Functions,*
D. Dachman-Soled*, H. Lee*, T. Malkin, R. Servedio, A. Wan*, and H. Wee.
ICALP (1) 2008: pp. 36–47.

47. *Black-Box Construction of a Non-Malleable Encryption Scheme from Any Semantically Secure One,*
S. G. Choi*, D. Dachman-Soled*, T. Malkin, and H. Wee.
Theory of Cryptography Conference (**TCC**) 2008: pp. 427–444.

48. *Amplifying Collision Resistance: A Complexity-Theoretic Treatment,*
R. Canetti, R. Rivest, M. Sudan, L. Trevisan, S. Vadhan, and H. Wee.
CRYPTO 2007: pp. 264–283.

49. *One-Way Permutations, Interactive Hashing and Statistically Hiding Commitments,*
H. Wee.
Theory of Cryptography Conference (**TCC**) 2007: pp. 419–433.

50. *Lower Bounds for Non-Interactive Zero Knowledge,*
H. Wee.
Theory of Cryptography Conference (**TCC**) 2007: pp. 103–117.

51. *Finding Pessiland,*
H. Wee.
Theory of Cryptography Conference (**TCC**) 2006: pp. 429–442.

52. *On Obfuscating Point Functions,*
H. Wee.
ACM Symposium on the Theory of Computation (**STOC**) 2005: pp. 523–532.

53. *Pebbling and Proofs of Work,*
C. Dwork, M. Naor, and H. Wee.
CRYPTO 2005: pp. 37–54.

54. *On Hardness Amplification of One-Way Functions,*
H. Lin, L. Trevisan, and H. Wee.
Theory of Cryptography Conference (**TCC**) 2005: pp. 34–49.

55. *Towards Privacy in Public Databases,*
S. Chawla, C. Dwork, F. McSherry, A. Smith and H. Wee.
Theory of Cryptography Conference (**TCC**) 2005: pp. 363–385.

56. *On Round-Efficient Argument Systems,*
H. Wee.
ICALP (3) 2005: pp. 140-52.

57. *More on Non-commutative Polynomial Identity Testing,*
A. Bogdanov and H. Wee.
IEEE Conference on Computational Complexity (**CCC**) 2005: pp. 92–99.

58. *A Stateful Implementation of a Random Function Supporting Parity Queries over Hypercubes,*
A. Bogdanov and H. Wee.
RANDOM 2004: pp. 298–309

59. *Selfish Caching in Distributed Systems: A Game Theoretic Analysis,*
B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou and J. Kubiatowicz.
ACM Symposium on Principles of Distributed Computing (**PODC**) 2004: pp. 21–30

60. *On Compressibility vs Pseudoentropy,*
H. Wee.
IEEE Conference on Computational Complexity (**CCC**) 2004: pp. 29–41

Part III
Attribute-Based Encryption for Circuits

Sergey Gorbunov and Vinod Vaikuntanathan and Hoeteck Wee
STOC 2013, invited to SPECIAL ISSUE, **JACM** 2015

Abstract. In an attribute-based encryption (ABE) scheme, a ciphertext is associated with an ℓ-bit *public index* ind and a message m, and a secret key is associated with a Boolean predicate P. The secret key allows to decrypt the ciphertext and learn m iff $P(\mathsf{ind}) = 1$. Moreover, the scheme should be secure against collusions of users, namely, given secret keys for polynomially many predicates, an adversary learns nothing about the message if none of the secret keys can individually decrypt the ciphertext.

We present attribute-based encryption schemes for circuits of any arbitrary polynomial size, where the public parameters and the ciphertext grow linearly with the depth of the circuit. Our construction is secure under the standard learning with errors (LWE) assumption. Previous constructions of attribute-based encryption were for Boolean formulas, captured by the complexity class NC^1.

In the course of our construction, we present a new framework for constructing ABE schemes. As a by-product of our framework, we obtain ABE schemes for polynomial-size branching programs, corresponding to the complexity class *LOGSPACE*, under quantitatively better assumptions.

1 Introduction

Attribute-based encryption [140, 89] is an emerging paradigm for public-key encryption which enables fine-grained control of access to encrypted data. In traditional public-key encryption, access to the encrypted data is all or nothing: given the secret key, one can decrypt and read the entire message, but without it, nothing about the message is revealed (other than its length). In attribute-based encryption, an encryption of a message m is labeled with a *public* attribute vector ind (also called the "index"), and secret keys are associated with predicates P. A secret key sk_P decrypts the ciphertext and recovers the message m if and only if ind satisfies the predicate, namely if and only if $P(\mathsf{ind}) = 1$.

Attribute-based encryption captures as a special case previous cryptographic notions such as identity-based encryption (IBE) [142, 29, 56] and fuzzy IBE [140]. It has also found applications in scenarios that demand complex policies to control access to encrypted data, as well as in designing cryptographic protocols for verifiably outsourcing computations [132].

The crucial component in the security requirement for attribute-based encryption stipulates that it resists *collusion attacks*, namely any group of users collectively learns nothing about the message m if none of them is individually authorized to decrypt the ciphertext.

In the past few years, there has been significant progress in attribute-based encryption in terms of efficiency, security guarantees, and diversifying security assumptions [89, 146, 113, 115, 46, 7, 128]. On the other hand, little progress has been made in terms of supporting larger classes of predicates. The state of the art is Boolean formulas [89, 115, 128], which is a subclass of log-space computations. Constructing a secure attribute-based encryption for all polynomial-time predicates was posed as a central challenge by Boneh, Sahai and Waters [33]. We resolve this problem affirmatively in this work.

2 Our Contributions

We construct attribute-based encryption schemes for circuits of every a-priori bounded depth, based on the learning with errors (LWE) assumption. In the course of our construction, we present a new framework for constructing attribute-based encryption schemes, based on a primitive that we call "two-to-one recoding" (TOR). Our methodology departs significantly from the current line of work on attribute-based encryption [89, 115] and instead, builds upon the connection to garbled circuits developed in the context of *bounded* collusions [144, 85]. Along the way, we make the first substantial progress towards the 25-year-old open problem of constructing (fully) reusable garbled circuits. In a follow-up work, Goldwasser et al. [82] completely resolved this open problem; moreover, their construction relies crucially on our ABE scheme as an intermediate building block. More details follow.

2.1 Attribute-based encryption

For every class of predicate circuits with depth bounded by a polynomial function $d = d(\lambda)$ (where λ is the security parameter), we construct an ABE scheme that supports this class of circuits, under the learning with errors (LWE) assumption. Informally, the (decisional) LWE problem [136] asks to distinguish between "noisy" random linear combinations of n numbers $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{Z}_q^n$ from uniformly random numbers over $\mathbb{Z}_q$.

Regev [136] showed that solving the LWE problem *on the average* is as hard as (quantumly) solving several notoriously difficult lattice problems *in the worst case*. Since then, the LWE assumption has become a central fixture in cryptography. We now have a large body of work building cryptographic schemes under the LWE assumption, culminating in the construction of a fully homomorphic encryption scheme [37].

The key parameter that determines the hardness of LWE is the ratio between the modulus q and the maximum absolute value of the noise B; as such, we refer to q/B as the hardness factor of LWE. The problem becomes easier as this ratio grows, but is believed to be hard for 2^{n^ϵ}-time algorithms when $q/B = 2^{O(n^\epsilon)}$, where $0 < \epsilon < 1/2$. Our results will hold as long as the latter holds for *some constant* ϵ.

In particular, we show:

Theorem 2.1 (informal)**.** *Assume that there is a constant* $0 < \epsilon < 1$ *for which the LWE problem is hard for a* $\exp(n^\epsilon)$ *factor in dimension* n*, for all large enough* n*. Then, for any polynomial* d*, there is a* selectively secure *attribute encryption scheme for general circuits of depth* d*.*

Moreover, our scheme has succinct ciphertexts, in the sense that the ciphertext size depends polynomially on the depth d and the length ℓ of the attribute vector ind, but not on the size of the circuits in the class. The construction as stated achieves the weaker notion of selective security, but we can easily obtain a fully secure scheme following [27] (but using sub-exponential hardness in a crucial way):

Corollary 2.2. *Assume that there is a constant* $0 < \epsilon < 1/2$ *such that the LWE problem with a factor of* $\exp(n^\epsilon)$ *is hard in dimension* n *for* $\exp(n^\epsilon)$*-time algorithms. Then, for any polynomial* d*, there is a* fully secure *attribute-based encryption scheme for general circuits of depth* d*.*

We also obtain a new ABE scheme for branching programs (which correspond to the complexity class *LOGSPACE*) under the weaker quasi-polynomial hardness of LWE:

Theorem 2.3 (informal)**.** *There exist attribute-based encryption schemes for the class of branching programs under either (1) the hardness of the LWE problem with an $n^{\omega(1)}$ factor, or (2) the bilinear decisional Diffie-Hellman assumption.*

Here, there is no a-prori bound on the size or the depth of the branching program. In addition, we achieve succinct ciphertexts of size $O(\ell)$ where ℓ is the number of bits in the index. Prior to this work, we only knew how to realize IBE and inner product encryption under $n^{\omega(1)}$-hardness of LWE [46, 7, 9], whereas our bilinear construction is a different way to achieve the results of Goyal et al. [89] which uses secret-sharing for general access structures. Our construction exploits a combinatorial property of branching programs to overcome limitations of previous approaches based on secret sharing for monotone formulas (c.f. [10]). The construction is inspired by a pairings-based scheme for regular languages in [147].

We now move on to provide a technical roadmap of our construction: first, we define a new primitive that we call a *two-to-one recoding* (TOR) scheme; we then show how TOR gives us an attribute-based encryption scheme for circuits, and how to construct a TOR scheme from the LWE assumption.

2.2 New Framework: TOR

A Two-to-One Recoding (TOR) scheme is a family of (probabilistic) functions $\{\mathsf{Encode}(\mathsf{pk},\cdot)\}$ indexed by pk, together with a "two-to-one" recoding mechanism. The basic computational security guarantee for $\mathsf{Encode}(\mathsf{pk},\cdot)$ is that of (correlated) pseudorandomness [137]: $\mathsf{Encode}(\mathsf{pk},s)$ should be pseudorandom given $\mathsf{Encode}(\mathsf{pk}_i,s)$ for polynomially many pk_i's, where s is a uniformly random "seed".

The recoding mechanism guarantees that given any triple of public keys $(\mathsf{pk}_0,\mathsf{pk}_1,\mathsf{pk}_{\mathrm{tgt}})$, there is a recoding key rk that allows us to perform the transformation

$$(\mathsf{Encode}(\mathsf{pk}_0,s),\mathsf{Encode}(\mathsf{pk}_1,s)) \mapsto \mathsf{Encode}(\mathsf{pk}_{\mathrm{tgt}},s).$$

Such a recoding key rk can be generated using either of the two secret keys sk_0 or sk_1. Furthermore, the recoding mechanism must satisfy a natural simulation requirement: namely, we can generate rk given just $\mathsf{pk}_0,\mathsf{pk}_1$ (and neither of the two secret keys), if we are allowed to "program" $\mathsf{pk}_{\mathrm{tgt}}$. That is, there are three ways of generating the pair $(\mathsf{pk}_{\mathrm{tgt}},\mathsf{rk})$ that are (statistically) indistinguishable: (1) given $\mathsf{pk}_{\mathrm{tgt}}$, generate rk using the secret key sk_0; (2) given $\mathsf{pk}_{\mathrm{tgt}}$, generate rk using the secret key sk_1; and (3) generate rk without either secret key, by "programming" the output public key $\mathsf{pk}_{\mathrm{tgt}}$.

This requirement demonstrates the *intuitive guarantee* that we expect from a two-to-one recoding mechanism: namely, the recoding key is "useless" given only one encoding, but not both encodings. For example, it is easy to see that given $\mathsf{Encode}(\mathsf{pk}_0,s)$ and rk (but not $\mathsf{Encode}(\mathsf{pk}_1,s)$), the output $\mathsf{Encode}(\mathsf{pk}_{\mathrm{tgt}},s)$ is pseudorandom. Indeed, this is because rk could as well have been "simulated" using sk_1, in which case it is of no help in the distinguishing task.

The simulation requirement also rules out the trivial construction from trapdoor functions where rk is a trapdoor for inverting $\mathsf{Encode}(\mathsf{pk}_0,\cdot)$ or $\mathsf{Encode}(\mathsf{pk}_1,\cdot)$.

From TOR to Garbled Circuits. We start from the observation that our TOR primitive implies a form of *reusable* garbled circuits *with no input or circuit privacy*, but instead, with a form of *authenticity guarantee*. As we will see, this leads directly into our attribute-based encryption scheme.

Consider a two-input boolean gate with input wires u, v and output wire w, computing a function $G : \{0,1\} \times \{0,1\} \to \{0,1\}$. In Yao's garbled circuit construction, we associate each wire with a pair of strings (called "labels"), and we provide a translation table comprising of four values $v_{b,c}$ where $v_{b,c}$ allows us to perform the transformation:

$$L_{u,b}, L_{v,c} \mapsto L_{w,G(b,c)}$$

The garbled circuits construction guarantees that given the translation table and labels L_{u,b^*} and L_{v,c^*} for specific input bits b^* and c^*, we can obtain $L_{w,G(b^*,c^*)}$; however, the other label at the output, namely $L_{w,1-G(b^*,c^*)}$ remains hidden.

In our setting, we replace labels with public keys, so that each wire is associated with a pair of public keys. As before, we also provide a translation table comprising four values $\mathsf{rk}_{b,c}$ where the recoding key $\mathsf{rk}_{b,c}$ allows us to perform the transformation

$$\mathsf{Encode}(\mathsf{pk}_{u,b}, s), \mathsf{Encode}(\mathsf{pk}_{v,c}, s) \mapsto \mathsf{Encode}(\mathsf{pk}_{w,G(b,c)}, s)$$

The security properties of the TOR scheme then give us the following guarantee: Given the translation table and encodings of s corresponding to b^*, c^*, we clearly compute the encoding of s corresponding to $G(b^*, c^*)$. However, the encoding corresponding to $1 - G(b^*, c^*)$ remains pseudorandom.

Moreover, crucially, the translation table is independent of s, so we can now "reuse" the translation table by providing fresh encodings with different choices of s. In a sentence, replacing strings by functions gives us the power of reusability.

In the garbled circuits construction, the four entries of the table are permuted and thus, one can perform the translation even without knowing what the input bits b^* and c^* are. This is possible because there is an efficient way to verify when the "correct" translation key is being used. In contrast, in the reusable construction above, one has to know exactly which of the recoding keys to use. This is part of the reason why we are *unable to provide circuit or input privacy, but instead, only guarantee authenticity*, namely that an adversary can obtain only one of the two possible encodings at the output wire.

This construction forms the cornerstone of the subsequent work of Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [82] who construct reusable garbled circuits with input and circuit privacy, by additionally leveraging the power of fully homomorphic encryption [76, 37].

From TOR to Attribute-Based Encryption. How is all this related to attribute-based encryption? In our attribute-based encryption scheme for circuits, the encodings of s are provided in the ciphertext, and the translation tables are provided in the secret key. More precisely, each wire is associated with two TOR public keys, and the encryption of a message m under an index ind is obtained by computing $\mathsf{Encode}(\mathsf{pk}_{i,\mathsf{ind}_i}, s)$ for every input wire i. The output encoding $\mathsf{Encode}(\mathsf{pk}_{\mathsf{out}}, s)$ is then used to mask the message. We obtain the secret key corresponding to a circuit C by "stitching" multiple translation tables together, where the public keys for the input and output wires are provided in the public parameters, and we pick fresh public keys for the internal wires during key generation. In a nutshell, this gives us the guarantee that given a secret key sk_C and an encryption $\mathsf{Enc}(\mathsf{ind}, m)$ such that $C(\mathsf{ind}) = 1$, we can compute $\mathsf{Encode}(\mathsf{pk}_{\mathsf{out}}, s)$ and thus recover the message. On the other hand, this value looks pseudorandom if $C(\mathsf{ind}) = 0$.

In our outline of reusable garbled circuits with authenticity, we wanted to reuse the garbled circuit

$G(C)$ across multiple encryptions with indices $\mathsf{ind}_1, \mathsf{ind}_2, \ldots$ on which C always evaluates to 0. In attribute-based encryption, we also want reusability across multiple circuits $C_1, C_2, \ldots$ all of which evaluate to 0 on a fixed index ind (in addition to multiple indices). Fortunately, the strong security properties of the TOR primitive provide us with this guarantee.

To obtain attribute-based encryption for branching programs, we are able to support a different notion of translation tables, which we can realize using a slightly weaker notion of TOR. In branching programs, the transition function depends on an input variable and the current state. The fact that one of these two values is always an input variable makes things simpler; in circuits, both of the input values to a gate could be internal wires.

TOR from LWE. We show how to instantiate TOR from LWE, building upon previous lattice-based IBE techniques in [77, 46, 7, 8]. The public key is given by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and

$$\mathsf{Encode}(\mathbf{A}, \mathbf{s}) = \mathbf{A}^T \mathbf{s} + \mathbf{e}$$

where $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathbb{Z}_q^m$ is an error vector, and $\mathbf{A}^T$ denotes the transpose of the matrix $\mathbf{A}$. (Correlated) pseudorandomness follows directly from the LWE assumption. Given $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_{\mathrm{tgt}} \in \mathbb{Z}_q^{n \times m}$, the recoding key rk is given by a low-norm matrix $\mathbf{R} \in \mathbb{Z}_q^{2m \times m}$ such that

$$[\, \mathbf{A}_0 \,\|\, \mathbf{A}_1 \,]\, \mathbf{R} = \mathbf{A}_{\mathrm{tgt}}$$

Note that

$$\mathbf{R}^T \begin{bmatrix} \mathbf{A}_0^T \mathbf{s} + \mathbf{e}_0 \\ \mathbf{A}_1^T \mathbf{s} + \mathbf{e}_1 \end{bmatrix} \approx \mathbf{A}_{\mathrm{tgt}}^T \mathbf{s}$$

which gives us the recoding mechanism. There are three ways of generating the public key $\mathbf{A}_{\mathrm{tgt}}$ together with the recoding key $\mathbf{R}$: (1) using the trapdoor for $\mathbf{A}_0$, (2) using the trapdoor for $\mathbf{A}_1$, or (3) first generating $\mathbf{R}$ and then "programming" $\mathbf{A}_{\mathrm{tgt}} := [\mathbf{A}_0 || \mathbf{A}_1]\, \mathbf{R}$. These three ways are statistically indistinguishable by the "bonsai trick" of [46]. In fact, our recoding mechanism is very similar to the lattice delegation mechanism introduced in [8], which also uses random low norm matrices to move from one lattice to another.

The multiplicative mechanism for recoding means that the noise grows exponentially with the number of sequential recodings. This, in turn, limits the depth of the circuits we can handle. In particular, the noise grows by a multiplicative $\mathsf{poly}(n)$ factor on each recoding, which means that after depth d, it becomes $n^{O(d)}$. Since $n^{O(d)} < q/4 < 2^{n^\epsilon}$, we can handle circuits of depth $\tilde{O}(n^\epsilon)$ (here, the first inequality is for correctness and the second for security). Viewed differently, setting the LWE dimension $n = d^{1/\epsilon}$ lets us handle circuits of maximum depth $d = d(\ell)$.

Our weak TOR for branching programs uses an additive mechanism, namely the recoding key is given by a low-norm matrix $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$ such that $\mathbf{A}_0 \mathbf{R} = \mathbf{A}_{\mathrm{tgt}} - \mathbf{A}_1$. Note that $\mathbf{R}^T (\mathbf{A}_0^T \mathbf{s} + \mathbf{e}_0) + (\mathbf{A}_1^T \mathbf{s} + \mathbf{e}_1) \approx \mathbf{A}_{\mathrm{tgt}}^T \mathbf{s}$ which gives us our recoding mechanism. Since in our branching program construction, $\mathbf{A}_0^T \mathbf{s} + \mathbf{e}_0$ will always be a fresh encoding provided in the ciphertext, the noise accumulation is additive rather than multiplicative.

2.3 Applications

Let us now explain the application of our result to the problem of publicly verifiable delegation of computation without input privacy.

A verifiable delegation scheme allows a computationally weak client to delegate expensive computations to the cloud, with the assurance that a malicious cloud cannot convince the client to accept an incorrect computation [120, 81, 74, 55, 15]. Recent work of Parno, Raykova and Vaikuntanathan [132] showed that any attribute-based encryption scheme for a class of circuits with encryption time at most linear in the length of the index immediately yields a two-message delegation scheme for the class in the pre-processing model. Namely, there is an initial pre-processing phase which fixes the circuit C the client wishes to compute, produces a circuit key and sends it to the server. Afterwards, to delegate computation on an input x, the client only needs to send a single message. Moreover, the ensuing delegation scheme satisfies public delegatability, namely anyone can delegate computations to the cloud; as well as public verifiability, namely anyone can check the cloud's work (given a "verification" key published by the client). The previous delegation schemes that satisfy both these properties (secure in the standard model) supported the class NC^1 [132, 89, 114]. Our attribute-based encryption schemes for circuits gives us a verifiable delegation scheme for all circuits, where the computation time of the client in the online phase is polynomial in the length of its input and the depth of the circuit, but is otherwise independent of the circuit size. We note that this scheme does not guarantee privacy of the input. Building on this work, Goldwasser et al. [82] show how to achieve a publicly verifiable delegation scheme with input privacy.

2.4 Related Work

Prior to this work, the state-of-art for lattice-based predicate encryption was threshold and inner product predicates [10, 9]; realizing Boolean formula was itself an open problem. A different line of work considers definitional issues in the more general realm of functional encryption [33, 131], for which general feasibility results are known for the restricted setting of a-priori bounded collusions developed from classical "one-time" garbled circuits [139, 85] (the ciphertext size grows with both the circuit size and the collusion bound). Our methodology takes a fresh perspective on how to achieve reusability of garbled circuits with respect to authenticity. Our primitive (TOR) can be thought of as a generalization of the notion of proxy re-encryption [24, 16, 100] which can be thought of as a one-to-one re-encryption mechanism.

Independent work. Boyen [36] gave a construction of an ABE scheme for Boolean formulas based on LWE; our result for LWE-based branching program subsumes the result since Boolean formulas are a subclass of branching programs. Garg, Gentry, Halevi, Sahai and Waters [69] gave a construction of attribute-based encryption for general circuits under a DBDH-like assumption in multi-linear groups; the construction extends to so-called graded encodings, for which we have candidates under non-standard assumptions in ideal lattices [66, 57]. The public parameters in the construction also grow with the depth of the circuit.

Subsequent Work. Our attribute-based encryption scheme has been used as the crucial component in the subsequent work of [82] to construct a (private index) functional encryption scheme with succinct ciphertexts. They also show a number of applications of their construction, including reusable garbled circuits *with input and circuit privacy*. Also subsequently, Boneh et al. [34] gave asymptotic improvements on the sizes of secret keys and ciphertexts in two different constructions respectively. Their main construction is built from a new *fully key-homomorphic encryption* reduces the size of the secret key for a predicate P from $|P| \times \mathrm{poly}(\lambda, d)$, shown in this work, to $|P| + \mathrm{poly}(\lambda, d)$ where λ is the security parameter and d is the circuit depth.

Organization. We present our TOR framework and its instantiation in Sections 4 and 5. We present our ABE scheme in Section 6. We present the scheme for branching programs in Section 7.

3 Preliminaries

Notation. Let PPT denote probabilistic polynomial-time. For any integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q and we represent $\mathbb{Z}_q$ as integers in $(-q/2, q/2]$. We let $\mathbb{Z}_q^{n \times m}$ denote the set of $n \times m$ matrices with entries in $\mathbb{Z}_q$. We use bold capital letters (e.g. $\mathbf{A}$) to denote matrices, bold lowercase letters (e.g. $\mathbf{x}$) to denote vectors. The notation $\mathbf{A}^\mathsf{T}$ denotes the transpose of the matrix $\mathbf{A}$.

If $\mathbf{A}_1$ is an $n \times m$ matrix and $\mathbf{A}_2$ is an $n \times m'$ matrix, then $[\mathbf{A}_1 \| \mathbf{A}_2]$ denotes the $n \times (m + m')$ matrix formed by concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$. A similar notation applies to vectors. When doing matrix-vector multiplication we always view vectors as column vectors.

We say a function $f(n)$ is *negligible* if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathrm{negl}(n)$ to denote a negligible function of n. We say $f(n)$ is *polynomial* if it is $O(n^c)$ for some $c > 0$, and we use $\mathrm{poly}(n)$ to denote a polynomial function of n. We say an event occurs with *overwhelming probability* if its probability is $1 - \mathrm{negl}(n)$. The function $\lg x$ is the base 2 logarithm of x. The notation $\lfloor x \rceil$ denotes the nearest integer to x, rounding towards 0 for half-integers.

3.1 Attribute-Based Encryption

We define attribute-based encryption (ABE), following [89].An ABE scheme for a class of predicate circuits $\mathcal{C}$ (namely, circuits with a single bit output) consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda, 1^\ell) \to (\mathsf{pp}, \mathsf{mpk}, \mathsf{msk})$: The setup algorithm gets as input the security parameter λ, the length ℓ of the index, and outputs the public parameter $(\mathsf{pp}, \mathsf{mpk})$, and the master key msk. All the other algorithms get pp as part of its input.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m) \to \mathsf{ct}_{\mathsf{ind}}$: The encryption algorithm gets as input mpk, an index $\mathsf{ind} \in \{0,1\}^\ell$ and a message $m \in \mathcal{M}$. It outputs a ciphertext $\mathsf{ct}_{\mathsf{ind}}$. Note that ind is public given $\mathsf{ct}_{\mathsf{ind}}$.

$\mathsf{KeyGen}(\mathsf{msk}, C) \to \mathsf{sk}_C$: The key generation algorithm gets as input msk and a predicate specified by $C \in \mathcal{C}$. It outputs a secret key sk_C (where C is also public).

$\mathsf{Dec}(\mathsf{sk}_C, \mathsf{ct}_{\mathsf{ind}}) \to m$: The decryption algorithm gets as input sk_C and $\mathsf{ct}_{\mathsf{ind}}$, and outputs either $\perp$ or a message $m \in \mathcal{M}$.

We require that for all (ind, C) such that $C(\mathsf{ind}) = 1$, all $m \in \mathcal{M}$ and $\mathsf{ct}_{\mathsf{ind}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m)$, $\mathsf{Dec}(\mathsf{sk}_C, \mathsf{ct}_{\mathsf{ind}}) = m$.

Security Definition. For a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda)$ to be

$$\Pr\left[b = b' : \begin{array}{l} \mathsf{ind} \leftarrow \mathcal{A}(1^\lambda, 1^\ell); \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}), |m_0| = |m_1|; \\ b \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}_{\mathsf{ind}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{ct}_{\mathsf{ind}}) \end{array}\right] - \frac{1}{2}$$

with the restriction that all queries C that $\mathcal{A}$ makes to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ satisfies $C(\mathsf{ind}) = 0$ (that is, sk_C does not decrypt $\mathsf{ct}_{\mathsf{ind}}$). an attribute-based encryption scheme is *selectively secure* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ. We call an attribute-based encryption scheme *fully secure* if the adversary $\mathcal{A}$ is allowed to choose the challenge index ind after seeing secret keys, namely, along with choosing (m_0, m_1).

3.2 Learning With Errors (LWE) Assumption

The LWE problem was introduced by Regev [136], who showed that solving it *on the average* is as hard as (quantumly) solving several standard lattice problems *in the worst case*.

Definition 3.1 (LWE)**.** *For an integer $q = q(n) \geq 2$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the learning with errors problem $\mathsf{dLWE}_{n,m,q,\chi}$ is to distinguish between the following pairs of distributions:*

$$\{\mathbf{A}, \mathbf{As} + \mathbf{x}\} \quad \textit{and} \quad \{\mathbf{A}, \mathbf{u}\}$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{x} \xleftarrow{\$} \chi^m, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$.

Connection to lattices. Let $B = B(n) \in \mathbb{N}$. A family of distributions $\chi = \{\chi_n\}_{n \in \mathbb{N}}$ is called B-bounded if

$$\Pr[\chi \in \{-B, \ldots, B-1, B\}] = 1.$$

There are known quantum [136] and classical [133] reductions between $\mathsf{dLWE}_{n,m,q,\chi}$ and approximating short vector problems in lattices in the worst case, where χ is a B-bounded (truncated) discretized Gaussian for some appropriate B. The state-of-the-art algorithms for these lattice problems run in time nearly exponential in the dimension n [13, 122]; more generally, we can get a 2^k-approximation in time $2^{\tilde{O}(n/k)}$. Combined with the connection to LWE, this means that the $\mathsf{dLWE}_{n,m,q,\chi}$ assumption is quite plausible for a poly(n)-bounded distribution χ and q as large as 2^{n^ϵ} (for any constant $0 < \epsilon < 1$). Throughout this paper, the parameter $m = \mathrm{poly}(n)$, in which case we will shorten the notation slightly to $\mathsf{LWE}_{n,q,\chi}$.

3.3 Trapdoors for Lattices and LWE

Gaussian distributions. Let $D_{\mathbb{Z}^m,\sigma}$ be the truncated discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter σ, that is, we replace the output by $\mathbf{0}$ whenever the $||\cdot||_\infty$ norm exceeds $\sqrt{m}\cdot\sigma$. Note that $D_{\mathbb{Z}^m,\sigma}$ is $\sqrt{m}\cdot\sigma$-bounded.

Lemma 3.1 (Lattice Trapdoors [12, 77, 121])**.** *There is an efficient randomized algorithm* $\mathsf{TrapSamp}(1^n, 1^m, q)$ *that, given any integers* $n \geq 1$, $q \geq 2$, *and sufficiently large* $m = \Omega(n \log q)$, *outputs a parity check matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a 'trapdoor' matrix* $\mathbf{T} \in \mathbb{Z}^{m\times m}$ *such that the distribution of* $\mathbf{A}$ *is* $\mathrm{negl}(n)$*-close to uniform.*

Moreover, there is an efficient algorithm $\mathsf{SampleD}$ *that with overwhelming probability over all random choices, does the following: For any* $\mathbf{u} \in \mathbb{Z}_q^n$, *and large enough* $s = \Omega(\sqrt{n \log q})$, *the randomized algorithm* $\mathsf{SampleD}(\mathbf{A}, \mathbf{T}, \mathbf{u}, s)$ *outputs a vector* $\mathbf{r} \in \mathbb{Z}^m$ *with norm* $||\mathbf{r}||_\infty \leq ||\mathbf{r}||_2 \leq s\sqrt{n}$ *(with probability* 1*). Furthermore, the following distributions of the tuple* $(\mathbf{A}, \mathbf{T}, \mathbf{U}, \mathbf{R})$ *are within* $\mathrm{negl}(n)$ *statistical distance of each other for any polynomial* $k \in \mathbb{N}$*:*

- $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$; $\mathbf{U} \leftarrow \mathbb{Z}_q^{n\times k}$; $\mathbf{R} \leftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{T}, \mathbf{U}, s)$.
- $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$; $\mathbf{R} \leftarrow (D_{\mathbb{Z}^m,s})^k$; $\mathbf{U} := \mathbf{AR} \pmod q$.

4 Two-to-One Recoding Schemes

An overview is provided in Section 2.2.

Symmetric encryption. In our construction, we will use $\mathsf{Encode}(\mathsf{pk}, s)$ as a one-time key for a symmetric-key encryption scheme (E, D). If Encode is deterministic, then we could simply use a one-time pad. However, since Encode is probabilistic, the one-time pad will not guarantee correctness. Instead, we require (E, D) to satisfy a stronger correctness guarantee, namely for all messages m and for all ψ, ψ' in the support $\mathsf{Encode}(\mathsf{pk}, s)$, $\mathsf{D}(\psi', \mathsf{E}(\psi, m)) = m$.

Allowing degradation. With each recoding operation, the "quality" of encoding potentially degrades. In order to formalize this, we allow the initial global public parameters to depend on $d_{\max}$, an a-prior upper bound on the number of nested recoding operations. We then require that given any encodings ψ and ψ' that are a result of at most $d_{\max}$ nested recodings, $\mathsf{D}(\psi', \mathsf{E}(\psi, m)) = m$. We stress that we allow $d_{\max}$ to be super-polynomial, and in fact, provide such instantiations for a relaxed notion of TOR.

4.1 Definition of TOR

Formally, a TOR scheme over the input space $\mathcal{S} = \{\mathcal{S}_\lambda\}$ consists of six polynomial-time algorithms $(\mathsf{Params}, \mathsf{Keygen}, \mathsf{Encode}, \mathsf{ReKeyGen}, \mathsf{SimReKeyGen}, \mathsf{Recode})$ and a symmetric-key encryption scheme (E, D) with the following properties:

- $\mathsf{Params}(1^\lambda, d_{\max})$ is a probabilistic algorithm that takes as input the security parameter λ and an upper bound $d_{\max}$ on the number of nested recoding operations (written in binary), outputs "global" public parameters pp.

- $\mathsf{Keygen}(\mathsf{pp})$ is a probabilistic algorithm that outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Encode}(\mathsf{pk}, s)$ is a probabilistic algorithm that takes pk and an input $s \in \mathcal{S}$, and outputs an encoding ψ.

In addition, there is a recoding mechanism together with two ways to generate recoding keys: given one of the two secret keys, or by programming the output public key.

- $\mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{pk}_{\mathrm{tgt}})$ is a probabilistic algorithm that takes a key pair $(\mathsf{pk}_0, \mathsf{sk}_0)$, another public key pk_1, a "target" public key $\mathsf{pk}_{\mathrm{tgt}}$, and outputs a recoding key rk.
- $\mathsf{SimReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1)$ is a probabilistic algorithm that takes two public keys $\mathsf{pk}_0, \mathsf{pk}_1$ and outputs a recoding key rk together with a "target" public key $\mathsf{pk}_{\mathrm{tgt}}$.
- $\mathsf{Recode}(\mathsf{rk}, \psi_0, \psi_1)$ is a deterministic algorithm that takes the recoding key rk, two encodings ψ_0 and ψ_1, and outputs an encoding ψ_{tgt}.

Remark 4.1. *For our instantiation from lattices, we can in fact invert* $\mathsf{Encode}(\mathsf{pk}, s)$ *to recover* s *using the corresponding* sk. *However, we will not require this property in our generic constructions from TOR. Indeed, realizing this property over bilinear groups would be hard, since* s *is typically encoded in the exponent.*

Correctness. Correctness of a TOR scheme requires two things. First, for every pk and $s \in \mathcal{S}$, there exists a family of sets $\Psi_{\mathsf{pk},s,j}$, $j = 0, 1, \ldots, d_{\max}$:

- $\Pr[\mathsf{Encode}(\mathsf{pk}, s) \in \Psi_{\mathsf{pk},s,0}] = 1$, where the probability is taken over the coin tosses of Encode;
- $\Psi_{\mathsf{pk},s,0} \subseteq \Psi_{\mathsf{pk},s,1} \subseteq \cdots \subseteq \Psi_{\mathsf{pk},s,d_{\max}}$.
- for all $\psi, \psi' \in \Psi_{\mathsf{pk},s,d_{\max}}$ and all $m \in \mathcal{M}$, $\mathsf{D}(\psi', \mathsf{E}(\psi, m)) = m$.

Note that these properties hold trivially if Encode is deterministic and (E, D) is the one-time pad. Secondly, the correctness of recoding requires that for any triple of key pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}})$, and any encodings $\psi_0 \in \Psi_{\mathsf{pk}_0,s,j_0}$ and $\psi_1 \in \Psi_{\mathsf{pk}_1,s,j_1}$,

$$\mathsf{Recode}(\mathsf{rk}, \psi_0, \psi_1) \in \Psi_{\mathsf{pk}_{\mathrm{tgt}},s,\max(j_0,j_1)+1}$$

Statistical Security Properties. Note that we have three ways of sampling recoding keys: using $\mathsf{ReKeyGen}$ along with one of two secret keys sk_0 or sk_1; using $\mathsf{SimReKeyGen}$ while programming $\mathsf{pk}_{\mathrm{tgt}}$. We require that all three ways lead to the same distribution of recoding keys, up to some statistical error.

(Key Indistinguishability) : Let $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Keygen}(\mathsf{pp})$ for $b = 0, 1$ and $(\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$.

The following two ensembles must be statistically indistinguishable:

$$\left[\mathsf{Aux}, \mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \boxed{\mathsf{sk}_0}, \mathsf{pk}_{\mathrm{tgt}})\right] \overset{s}{\approx} \left[\mathsf{Aux}, \mathsf{ReKeyGen}(\mathsf{pk}_1, \mathsf{pk}_0, \boxed{\mathsf{sk}_1}, \mathsf{pk}_{\mathrm{tgt}})\right]$$

where $\mathsf{Aux} = ((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}}))$. Informally, this says that sampling recoding keys using sk_0 or sk_1 yields the same distribution.

(Recoding Simulation) : Let $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Keygen}(\mathsf{pp})$ for $b = 0, 1$. Then, the following two ways of sampling the tuple $\left[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{\mathrm{tgt}}, \mathsf{rk}\right]$ must be statistically indistinguishable:

$$\left[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{\mathrm{tgt}}, \mathsf{rk} : (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}}) \leftarrow \mathsf{Keygen}(\mathsf{pp}); \mathsf{rk} \leftarrow \mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{pk}_{\mathrm{tgt}})\right] \overset{s}{\approx}$$
$$\left[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{\mathrm{tgt}}, \mathsf{rk} : (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{rk}) \leftarrow \mathsf{SimReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1)\right]$$

In addition, we require one-time semantic security for (E, D):

(One-time Semantic Security) : For all $m_0, m_1 \in \mathcal{M}$, the following two distributions must be statistically indistinguishable:

$$\left[\, \mathsf{E}(\psi, m_0) : \psi \overset{\$}{\leftarrow} \mathcal{K} \,\right] \overset{s}{\approx} \left[\, \mathsf{E}(\psi, m_1) : \psi \overset{\$}{\leftarrow} \mathcal{K} \,\right]$$

For all three properies, computational indistinguishability is sufficient for our applications, but we will achieve the stronger statistical indistinguishability in our instantiations.

Computational Security Property. We require that given the encoding of a random s on $\ell = \mathrm{poly}(\lambda)$ keys, the evaluation at a fresh key is pseudorandom.

(Correlated Pseudorandomness) : For every polynomial $\ell = \ell(\lambda)$, let $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Keygen}(\mathsf{pp})$ for $i \in [\ell + 1]$. Let $s \overset{\$}{\leftarrow} \mathcal{S}$, and let $\psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_i, s)$ for $i \in [\ell + 1]$. Then, the following two ensembles must be computationally indistinguishable:

$$\left[\, (\mathsf{pk}_i, \psi_i)_{i \in [\ell]}, \mathsf{pk}_{\ell+1}, \boxed{\psi_{\ell+1}} \,\right] \overset{c}{\approx}$$
$$\left[\, (\mathsf{pk}_i, \psi_i)_{i \in [\ell]}, \mathsf{pk}_{\ell+1}, \boxed{\psi} : \psi \overset{\$}{\leftarrow} \mathcal{K} \,\right]$$

That is, we define the advantage function $\mathsf{Adv}^{\mathrm{CP}}_{\mathcal{A}}(\lambda)$ to be:

$$\Pr\left[b = b' : \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda); s \leftarrow \mathcal{S}; \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Keygen}(\mathsf{pp}), \\ \psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_i, s), i = 1, \ldots, \ell; \\ \psi'_0 \leftarrow \mathsf{Encode}(\mathsf{pk}_{\ell+1}, s); \\ b \overset{\$}{\leftarrow} \{0,1\}; \psi'_1 \overset{\$}{\leftarrow} \mathcal{K} \\ b' \leftarrow \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_{\ell+1}, \psi_1, \ldots, \psi_\ell, \psi'_b) \end{array} \right] - \frac{1}{2}$$

and we require that for all PPT $\mathcal{A}$, the advantage function $\mathsf{Adv}^{\mathrm{CP}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.

4.2 Simple Applications of TOR

First example. We revisit the example from Section 2.2. Consider a two-input boolean gate g with input wires u, v and output wire w, computing a function $G : \{0,1\} \times \{0,1\} \to \{0,1\}$. Analogous to Yao's garbled

circuit, we provide a translation table Γ comprising four values

$$\Gamma := (\, \mathsf{rk}_{b,c} \,:\, b, c \in \{0,1\}\,)$$

where $\mathsf{rk}_{b,c}$ allows us to perform the transformation

$$\mathsf{Encode}(\mathsf{pk}_{u,b}, s), \mathsf{Encode}(\mathsf{pk}_{v,c}, s) \mapsto \mathsf{Encode}(\mathsf{pk}_{w,G(b,c)}, s)$$

Now, fix b^*, c^* and $d^* := G(b^*, c^*)$. Given an encoding of s corresponding to b^* and c^*, we can compute that under for d^* using the recoding key rk_{b^*,c^*}; in addition, we claim that the encoding corresponding to $1 - d^*$ remains pseudorandom. To prove this, it suffices to simulate Γ given $\mathsf{pk}_{u,b^*}, \mathsf{pk}_{v,c^*}, \mathsf{pk}_{w,1-d^*}$ as follows:

- we sample $(\mathsf{pk}_{w,d^*}, \mathsf{rk}_{b^*,c^*})$ using $\mathsf{SimReKeyGen}$;
- we sample $\mathsf{pk}_{u,1-b^*}$ and $\mathsf{pk}_{v,1-c^*}$ along with the corresponding secret keys; using these secret keys, we can sample the other three recoding keys $\mathsf{rk}_{1-b^*,c^*}, \mathsf{rk}_{b^*,1-c^*}, \mathsf{rk}_{1-b^*,1-c^*}$.

IBE from TOR. As a warm-up, we show how to build a selectively secure IBE for identity space $\{0,1\}^\ell$.

$$\mathsf{mpk} := \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \dots & \mathsf{pk}_{\ell,0} & \mathsf{pk}_{\text{start}} \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \dots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\text{out}} \end{pmatrix}$$

The ciphertext for identity ind and message m is given by:

$$\Big(\mathsf{Encode}(\mathsf{pk}_{1,\mathsf{ind}_1}, s), \dots, \mathsf{Encode}(\mathsf{pk}_{\ell,\mathsf{ind}_\ell}, s), \mathsf{Encode}(\mathsf{pk}_{\text{start}}, s), \mathsf{E}(\mathsf{Encode}(\mathsf{pk}_{\text{out}}, s), m) \Big)$$

The secret key for identity ind is given by $(\mathsf{rk}_1, \dots, \mathsf{rk}_\ell)$ where we first sample

$$(\mathsf{pk}'_1, \mathsf{sk}'_1), \dots, (\mathsf{pk}'_{\ell-1}, \mathsf{sk}'_{\ell-1}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$$

and then sample

$$\begin{aligned} \mathsf{rk}_1 &\leftarrow \mathsf{ReKeyGen}(\mathsf{pk}_{\text{start}}, \mathsf{pk}_{1,\mathsf{ind}_1}, \mathsf{sk}_{\text{start}}, \mathsf{pk}'_1) \\ \mathsf{rk}_2 &\leftarrow \mathsf{ReKeyGen}(\mathsf{pk}'_1, \mathsf{pk}_{2,\mathsf{ind}_2}, \mathsf{sk}'_1, \mathsf{pk}'_2) \\ &\vdots \\ \mathsf{rk}_\ell &\leftarrow \mathsf{ReKeyGen}(\mathsf{pk}'_{\ell-1}, \mathsf{pk}_{\ell,\mathsf{ind}_\ell}, \mathsf{sk}'_{\ell-1}, \mathsf{pk}_{\text{out}}) \end{aligned}$$

To prove selective security, we need to generate secret keys for any $\mathsf{ind} \neq \mathsf{ind}^*$, given $\mathsf{sk}_{1,1-\mathsf{ind}^*_1}, \dots, \mathsf{sk}_{\ell,1-\mathsf{ind}^*_\ell}$ but not $\mathsf{sk}_{\text{start}}$ or sk_{out}. We can achieve this as follows: pick an i for which $\mathsf{ind}_i \neq \mathsf{ind}^*_i$;

- pick $(\mathsf{rk}_1, \mathsf{pk}'_1), \dots, (\mathsf{rk}_{i-1}, \mathsf{pk}'_{i-1})$ using $\mathsf{SimReKeyGen}$;
- pick $(\mathsf{pk}'_i, \mathsf{sk}'_i), \dots, (\mathsf{pk}'_{\ell-1}, \mathsf{sk}'_{\ell-1})$ using Keygen;
- pick $\mathsf{rk}_i, \mathsf{rk}_{i+1}, \dots, \mathsf{rk}_\ell$ using $\mathsf{ReKeyGen}$ with secret keys $\mathsf{sk}_{1-\mathsf{ind}^*_i}, \mathsf{sk}'_i, \dots, \mathsf{sk}'_{\ell-1}$ respectively.

5 TOR from LWE

In this section, we present an instantiation of TOR from LWE, building upon ideas previously introduced in [77, 46, 7, 8].

Lemma 5.1. *Assuming* $\mathsf{dLWE}_{n,q,\chi}$ *where* $q = n^{\Theta(d_{\max})}$*, there is a TOR scheme that is correct up to* $d_{\max}$ *levels.*

- $\mathsf{Params}(1^\lambda, d_{\max})$: First choose the LWE dimension $n = n(\lambda)$. Let the error distribution $\chi = \chi(n) = D_{\mathbb{Z},\sqrt{n}}$, the error bound $B = B(n) = O(n)$, the modulus $q = q(n) = \tilde{O}(n^2 d_{\max})^{d_{\max}} n$, the number of samples $m = m(n) = O(n \log q)$ and the Gaussian parameter $s = s(n) = O(\sqrt{n \log q})$. Output the global public parameters $\mathsf{pp} = (n, \chi, B, q, m, s)$.
- $\mathsf{Keygen}(\mathsf{pp})$: Run the trapdoor generation algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with the trapdoor matrix $\mathbf{T} \in \mathbb{Z}^{m \times m}$. Output $\mathsf{pk} := \mathbf{A}$ and $\mathsf{sk} := \mathbf{T}$.
- $\mathsf{Encode}(\mathsf{pk}, s)$: Sample an error vector $\mathbf{e} \leftarrow \chi^m$ and output the encoding $\boldsymbol{\psi} := \mathbf{A}^T \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^n$.

The recoding algorithms work as follows:

- $\mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_b, \mathsf{pk}_{\mathrm{tgt}})$: Let $\mathsf{pk}_0 = \mathbf{A}_0$, $\mathsf{pk}_1 = \mathbf{A}_1$, $\mathsf{sk}_b = \mathbf{T}_b$ and $\mathsf{pk}_{\mathrm{tgt}} = \mathbf{A}_{\mathrm{tgt}}$. Compute the matrix $\mathbf{R} \in \mathbb{Z}^{2m \times m}$ in the following way:
  - Choose a discrete Gaussian matrix $\mathbf{R}_{1-b} \leftarrow (D_{\mathbb{Z},s})^{m \times m}$. Namely, each entry of the matrix is an independent sample from the discrete Gaussian distribution $D_{\mathbb{Z},s}$.
  - Compute $\mathbf{U} := \mathbf{A}_{\mathrm{tgt}} - \mathbf{A}_{1-b}\mathbf{R}_{1-b} \in \mathbb{Z}_q^{n \times m}$.
  - Compute the matrix $\mathbf{R}_b$ by running the algorithm $\mathsf{SampleD}$ to compute a matrix $\mathbf{R}_b \in \mathbb{Z}^{m \times m}$ as follows:
$$\mathbf{R}_b \leftarrow \mathsf{SampleD}(\mathbf{A}_b, \mathbf{T}_b, \mathbf{U})$$

  Output
$$\mathsf{rk}_{0,1}^{\mathrm{tgt}} := \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix} \in \mathbb{Z}^{2m \times m}$$
  (We remark that $\mathbf{A}_b\mathbf{R}_b = \mathbf{U} = \mathbf{A}_{\mathrm{tgt}} - \mathbf{A}_{1-b}\mathbf{R}_{1-b}$, and thus, $\mathbf{A}_0\mathbf{R}_0 + \mathbf{A}_1\mathbf{R}_1 = \mathbf{A}_{\mathrm{tgt}}$).
- $\mathsf{SimReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1)$: Let $\mathsf{pk}_0 = \mathbf{A}_0$ and $\mathsf{pk}_1 = \mathbf{A}_1$.
  - Sample a matrix $\mathbf{R} \leftarrow (D_{\mathbb{Z},s})^{2m \times m}$ by sampling each entry from the discrete Gaussian distribution $D_{\mathbb{Z},s}$.
  - Define
$$\mathbf{A}_{\mathrm{tgt}} := [\mathbf{A}_0 \,||\, \mathbf{A}_1]\, \mathbf{R} \in \mathbb{Z}_q^{n \times m}$$
    Output the pair $(\mathsf{pk}_{\mathrm{tgt}} := \mathbf{A}_{\mathrm{tgt}}, \mathsf{rk}_{0,1}^{\mathrm{tgt}} := \mathbf{R})$.
- $\mathsf{Recode}(\mathsf{rk}_{0,1}^{\mathrm{tgt}}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1)$: Let $\mathsf{rk}_{0,1}^{\mathrm{tgt}} = \mathbf{R}$. Compute the recoded ciphertext
$$\boldsymbol{\psi}_{\mathrm{tgt}} = [\boldsymbol{\psi}_0^T \,||\, \boldsymbol{\psi}_1^T]\, \mathbf{R}$$

We also need a one-time symmetric encryption scheme (E,D) which we will instantiate as an *error-tolerant* version of the one-time pad with $\mathcal{K} = \mathbb{Z}_q^n, \mathcal{M} = \{0,1\}^n$, as follows:

- $\mathsf{E}(\boldsymbol{\psi}, \boldsymbol{\mu})$ takes as input a vector $\boldsymbol{\psi} \in \mathbb{Z}_q^n$ and a bit string $\boldsymbol{\mu} \in \mathcal{M}$ and outputs the encryption

$$\boldsymbol{\gamma} := \boldsymbol{\psi} + \lceil q/2 \rceil \, \boldsymbol{\mu} \pmod{q}$$

- $\mathsf{D}(\boldsymbol{\psi}', \boldsymbol{\gamma})$ takes as input a vector $\boldsymbol{\psi}' = (\psi'_1, \ldots, \psi'_n) \in \mathbb{Z}_q^n$, an encryption $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n) \in \mathbb{Z}_q^n$ and does the following. Define a function $\mathsf{Round}(x)$ where $x \in [-(q-1)/2, \ldots, (q-1)/2]$ as:

$$\mathsf{Round}(x) = \begin{cases} 0 & \text{if } |x| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

  The decryption algorithm outputs a vector $\boldsymbol{\mu} = (\mathsf{Round}(\gamma_1 - \psi'_1), \ldots, \mathsf{Round}(\gamma_n - \psi'_n))$.

We defer the analysis of (E,D) to the full version.

5.1 Analysis

Correctness. We define the sets $\Psi_{\mathbf{A},\mathbf{s},j}$ for $\mathsf{pk} := \mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and $j \in [1 \ldots d_{\max}]$ as follows:

$$\Psi_{\mathbf{A},\mathbf{s},j} = \left\{ \mathbf{A}^T\mathbf{s} + \mathbf{e} : \ ||\mathbf{e}||_\infty \leq B \cdot (2sm\sqrt{m})^j \right\}$$

Given this definition:

- Observe that when $\mathbf{e} \leftarrow \chi^m$, $||\mathbf{e}||_\infty \leq B$ by the definition of χ and B. $\Pr[\mathsf{Encode}(\mathbf{A},\mathbf{s}) \in \Psi_{\mathbf{A},\mathbf{s},0}] = 1$.
- $\Psi_{\mathbf{A},\mathbf{s},0} \subseteq \Psi_{\mathbf{A},\mathbf{s},1} \subseteq \ldots \subseteq \Psi_{\mathbf{A},\mathbf{s},d_{\max}}$, by definition of the sets above.
- For any two encodings $\boldsymbol{\psi} = \mathbf{A}^T\mathbf{s} + \mathbf{e}, \boldsymbol{\psi}' = \mathbf{A}^T\mathbf{s} + \mathbf{e}' \in \Psi_{\mathbf{A},\mathbf{s},d_{\max}}$,

$$||\boldsymbol{\psi} - \boldsymbol{\psi}'||_\infty = ||\mathbf{e} - \mathbf{e}'||_\infty \leq 2 \cdot B \cdot (2sm\sqrt{m})^{d_{\max}} < q/4,$$

  which holds as long as $n \cdot O(n^2 \log q)^{d_{\max}} < q/4$. Thus, $\boldsymbol{\psi}$ and $\boldsymbol{\psi}'$ are "close", and by the correctness property of the symmetric encryption scheme (E,D) described above, $\mathsf{D}(\boldsymbol{\psi}', \mathsf{E}(\boldsymbol{\psi}, \boldsymbol{\mu})) = \boldsymbol{\mu}$ for any $\boldsymbol{\mu} \in \{0,1\}^n$.

- Consider two encodings $\boldsymbol{\psi}_0 \in \Psi_{\mathbf{A}_0,\mathbf{s},j_0}$ and $\boldsymbol{\psi}_1 \in \Psi_{\mathbf{A}_1,\mathbf{s},j_1}$ for any $j_0, j_1 \in \mathbb{N}$, any $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{s} \in \mathbb{Z}_q^n$. Then, $\boldsymbol{\psi}_0 = \mathbf{A}_0^T\mathbf{s} + \mathbf{e}_0$ and $\boldsymbol{\psi}_1 := \mathbf{A}_1^T\mathbf{s} + \mathbf{e}_1$ where $||\mathbf{e}_0||_\infty \leq B \cdot (2sm\sqrt{m})^{j_0}$ and $||\mathbf{e}_1||_\infty \leq B \cdot (2sm\sqrt{m})^{j_1}$.

  Then, the recoded ciphertext $\boldsymbol{\psi}_{\mathrm{tgt}}$ is computed as follows:

$$\begin{aligned} \boldsymbol{\psi}_{\mathrm{tgt}}^T &:= \left[\boldsymbol{\psi}_0^T \,||\, \boldsymbol{\psi}_1^T\right] \mathbf{R}_{0,1}^{\mathrm{tgt}} \\ &= \left[\mathbf{s}^T\mathbf{A}_0 + \mathbf{e}_0^T \,||\, \mathbf{s}^T\mathbf{A}_1 + \mathbf{e}_1^T\right] \mathbf{R}_{0,1}^{\mathrm{tgt}} \\ &= \mathbf{s}^T \left[\mathbf{A}_0 \,||\, \mathbf{A}_1\right] \mathbf{R}_{0,1}^{\mathrm{tgt}} + \left[\mathbf{e}_0^T \,||\, \mathbf{e}_1^T\right] \mathbf{R}_{0,1}^{\mathrm{tgt}} \\ &= \mathbf{s}^T\mathbf{A}_{\mathrm{tgt}} + \mathbf{e}_{\mathrm{tgt}} \end{aligned}$$

where the last equation is because $\mathbf{A}_{\mathrm{tgt}} = [\mathbf{A}_0 \,||\, \mathbf{A}_1]\, \mathbf{R}_{0,1}^{\mathrm{tgt}}$ and we define $\mathbf{e}_{\mathrm{tgt}} := [\mathbf{e}_0^T \,||\, \mathbf{e}_1^T]\, \mathbf{R}_{0,1}^{\mathrm{tgt}}$. Thus,

$$\begin{aligned}||\mathbf{e}_{\mathrm{tgt}}||_\infty &\le m \cdot ||\mathbf{R}_{0,1}^{\mathrm{tgt}}||_\infty \cdot (||\mathbf{e}_0||_\infty + ||\mathbf{e}_1||_\infty) \\ &\le m \cdot s\sqrt{m} \cdot (B \cdot (2sm\sqrt{m})^{j_0} + B \cdot (2sm\sqrt{m})^{j_1}) \\ &\le B \cdot (2sm\sqrt{m})^{\max(j_0, j_1)+1}\end{aligned}$$

exactly as required. Here, the second inequality is because $||\mathbf{R}_{0,1}^{\mathrm{tgt}}||_\infty \le s\sqrt{m}$ by Lemma 3.1. This finishes our proof of correctness.

Key Indistinguishability. Recall that in ReKeyGen, we given sampling $(\mathbf{R}_0, \mathbf{R}_1)$ satisfying $\mathbf{A}_0\mathbf{R}_0 + \mathbf{A}_1\mathbf{R}_1 = \mathbf{A}_{\mathrm{tgt}}$. Key indistinguishability basically says that we obtain the same distribution whether we use a trapdoor for $\mathbf{A}_0$ or that for $\mathbf{A}_1$. Indeed, this follows directly from the following statement in [46, 77] (see also [45, Theorem 3.4]): for every $(\mathbf{A}_0, \mathbf{T}_0)$, $(\mathbf{A}_1, \mathbf{T}_1)$ generated by $\mathsf{TrapSamp}(1^n, 1^m, q)$, every matrix $\mathbf{V} \in \mathbb{Z}_q^{n \times m}$, and any $s = \Omega(\sqrt{n \log q})$, the following two experiments generate distributions with $\mathrm{negl}(n)$ statistical distance:

- Sample $\mathbf{R}_0 \leftarrow (D_{\mathbb{Z}^m, s})^m$, compute $\mathbf{U} := \mathbf{V} - \mathbf{A}_0\mathbf{R}_0 \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{R}_1 \leftarrow \mathsf{SampleD}(\mathbf{A}_1, \mathbf{T}_1, \mathbf{U}, s)$. Output $(\mathbf{R}_0, \mathbf{R}_1)$.
- Sample $\mathbf{R}_1 \leftarrow (D_{\mathbb{Z}^m, s})^m$, compute $\mathbf{U} := \mathbf{V} - \mathbf{A}_1\mathbf{R}_1 \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{R}_0 \leftarrow \mathsf{SampleD}(\mathbf{A}_0, \mathbf{T}_0, \mathbf{U}, s)$. Output $(\mathbf{R}_0, \mathbf{R}_1)$.

The recoding simulation property follows readily from Lemma 3.1, as is done in [46]. Correlated pseudorandomness directly from the decisional LWE assumption $\mathsf{dLWE}_{n,(\ell+1)\cdot m, q, \chi}$ where $q = n^{\Theta(d_{\max})}$.

6 Attribute-Based Encryption for Circuits

In this section, we show how to construct attribute-based encryption for circuits from any TOR scheme. Let TOR be the scheme consisting of algorithms $(\mathsf{Params}, \mathsf{Keygen}, \mathsf{Encode})$ with the "two-to-one" recoding mechanism $(\mathsf{Recode}, \mathsf{ReKeyGen}, \mathsf{SimReKeyGen})$ with input space $\mathcal{S}$. For every $d_{\max}$, let $d_{\max}$-TOR denote a secure "two-to-one" recoding scheme that is correct for $d_{\max}$ recoding levels.

Theorem 6.1. *For every ℓ and polynomial $d_{\max} = d_{\max}(\lambda)$, let $\mathcal{C}_{\ell, d_{\max}}$ denote a family of polynomial-size circuits of depth at most $d_{\max}$ that take ℓ bits of input. Assuming the existence of a $d_{\max}$-TOR scheme, there exists a selectively secure attribute-based encryption scheme $\mathcal{ABE}$ for $\mathcal{C}$.*

Combining Theorem 1.1 and Lemma 5.1, we obtain a selectively secure attribute-based encryption scheme from LWE. Furthermore, invoking an argument from [27, Theorem 7.1] and using subexponential hardness of LWE, we obtain a fully secure scheme:

Corollary 6.2. *For all ℓ and polynomial $d_{\max} = d_{\max}(\ell)$, there exists a selectively secure attribute-based encryption scheme $\mathcal{ABE}$ for any family of polynomial-size circuits with ℓ inputs and depth at most $d_{\max}$, assuming the hardness of $\mathsf{dLWE}_{n,q,\chi}$ for sufficiently large $n = \mathrm{poly}(\lambda, d_{\max})$, $q = n^{O(d_{\max})}$ and some $\mathrm{poly}(n)$-bounded error distribution χ.*

Moreover, assuming $2^{O(\ell)}$-hardness of $\mathsf{dLWE}_{n,q,\chi}$ for parameters $n = \mathrm{poly}(\lambda, d_{\max}, \ell)$, and q and χ as above, the attribute-based encryption scheme $\mathcal{ABE}$ is fully secure.

The reader is referred to the text after the construction for further explanation of how to choose the LWE parameters.

Observe that if we start with a TOR scheme that supports $d_{\max} = \ell^{\omega(1)}$, then our construction immediately yields an attribute-based encryption scheme for arbitrary polynomial-size circuit families (without any restriction on the depth). This can be achieved if, for example, we had an LWE-based TOR scheme where q grows polynomially instead of exponentially in $d_{\max}$ as in our LWE-based weak TOR.

We now prove Theorem 1.1.

Circuit Representation. Let $\mathcal{C}_\lambda$ be a collection of circuits each having $\ell = \ell(\lambda)$ input wires and one output wire. Define a collection $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$. For each $C \in \mathcal{C}_\lambda$, we index the wires of C in the following way. The input wires are indexed 1 to ℓ, the internal wires have indices $\ell+1, \ell+2, \ldots, |C|-1$ and the output wire has index $|C|$, which also denotes the size of the circuit. We assume that the circuit is composed of arbitrary two-to-one gates. Each gate g is indexed as a tuple (u, v, w) where u and v are the incoming wire indices, and $w > \max\{u, v\}$ is the outgoing wire index. The gate computes the function $g_w : \{0,1\} \times \{0,1\} \to \{0,1\}$. The "fan-out wires" in the circuit are given a single number. That is, if the outgoing wire of a gate feeds into the input of multiple gates, then all these wires are indexed the same. (See e.g. [23, Fig 4].)

6.1 Construction from TOR

The ABE scheme $\mathcal{ABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ is defined as follows.

$\mathsf{Setup}(1^\lambda, 1^\ell, d_{\max})$: For each of the ℓ input wires, generate two public/secret key pairs. Also, generate an additional public/secret key pair:

$$(\mathsf{pk}_{i,b}, \mathsf{sk}_{i,b}) \leftarrow \mathsf{Keygen}(\mathsf{pp}) \qquad \text{for } i \in [\ell], b \in \{0,1\}$$
$$(\mathsf{pk}_{\text{out}}, \mathsf{sk}_{\text{out}}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$$

Output

$$\mathsf{mpk} := \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \ldots & \mathsf{pk}_{\ell,0} & \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \ldots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\text{out}} \end{pmatrix} \qquad \mathsf{msk} := \begin{pmatrix} \mathsf{sk}_{1,0} & \mathsf{sk}_{2,0} & \ldots & \mathsf{sk}_{\ell,0} \\ \mathsf{sk}_{1,1} & \mathsf{sk}_{2,1} & \ldots & \mathsf{sk}_{\ell,1} \end{pmatrix}$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m)$: For $\mathsf{ind} \in \{0,1\}^\ell$, choose a uniformly random $s \xleftarrow{\$} \mathcal{S}$ and encode it under the public keys specified by the index bits:

$$\psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_{i,\mathsf{ind}_i}, s) \text{ for all } i \in [\ell]$$

Encrypt the message m:

$$\tau \leftarrow \mathsf{E}(\mathsf{Encode}(\mathsf{pk}_{\text{out}}, s), m)$$

Output the ciphertext

$$\mathsf{ct}_{\mathsf{ind}} := \begin{pmatrix} \psi_1, & \psi_2, & \ldots, & \psi_\ell, & \tau \end{pmatrix}$$

$\mathsf{KeyGen}(\mathsf{msk}, C)$:

1. For every non-input wire $w = \ell+1,\ldots,|C|$ of the circuit C, and every $b \in \{0,1\}$, generate public/secret key pairs:

$$(\mathsf{pk}_{w,b}, \mathsf{sk}_{w,b}) \leftarrow \mathsf{Keygen}(\mathsf{pp}) \text{ if } w < |C| \text{ or } b = 0$$

and set $\mathsf{pk}_{|C|,1} := \mathsf{pk}_{\text{out}}$.

2. For the gate $g = (u, v, w)$ with outgoing wire w, compute the four recoding keys $\mathsf{rk}^w_{b,c}$ (for $b, c \in \{0,1\}$):

$$\mathsf{rk}^w_{b,c} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{u,b}, \mathsf{pk}_{v,c}, \mathsf{sk}_{u,b}, \mathsf{pk}_{w,g_w(b,c)}\Big)$$

Output the secret key which is a collection of $4(|C|-\ell)$ recoding keys

$$\mathsf{sk}_C := \Big(\mathsf{rk}^w_{b,c} \,:\, w \in \big[\,\ell+1, |C|\,\big],\, b,c \in \{0,1\} \Big)$$

$\mathsf{Dec}(\mathsf{sk}_C, \mathsf{ct}_{\mathsf{ind}})$: We tacitly assume that $\mathsf{ct}_{\mathsf{ind}}$ contains the index ind. For $w = \ell+1,\ldots,|C|$, let $g = (u, v, w)$ denote the gate with outgoing wire w. Suppose wires u and v carry the values b^* and c^*, so that wire w carries the value $d^* := g_w(b^*, c^*)$. Compute

$$\psi_{w,d^*} \leftarrow \mathsf{Recode}\Big(\mathsf{rk}^w_{b^*,c^*}, \psi_{u,b^*}, \psi_{v,c^*}\Big)$$

If $C(\mathsf{ind}) = 1$, then we would have computed $\psi_{|C|,1}$. Output the message

$$m \leftarrow \mathsf{D}\big(\,\psi_{|C|,1}, \tau\,\big)$$

If $C(\mathsf{ind}) = 0$, output $\perp$.

LWE Parameters. Fix $\ell = \ell(\lambda)$ and $d_{\max} = d_{\max}(\ell)$, and suppose the $\mathsf{dLWE}_{n,m,q,\chi}$ assumption holds for $q = 2^{n^\epsilon}$ for some $0 < \epsilon < 1$. Then, in our LWE-based TOR, we will set:

$$n = \tilde{\Theta}(d_{\max}^{1/\epsilon}) \quad \text{and} \quad q = n^{\Theta(d_{\max})}$$

By Corollary 6.2, we get security under 2^{n^ϵ}-LWE.

6.2 Correctness

Lemma 6.3 (correctness)**.** *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ be family where each $\mathcal{C}_\lambda$ is a finite collection of polynomial-size circuits each of depth at most $d_{\max}$. Let* TOR *be a correct "two-to-one" recoding scheme for $d_{\max}$ levels. Then, the construction presented above is a correct attribute-based encryption scheme.*

Proof. Fix a circuit C of depth at most $d_{\max}$ and an input ind such that $C(\mathsf{ind}) = 1$. Informally, we rely on recoding correctness for $d_{\max}$ recodings to show that $w = 1,\ldots,|C|$, we have

$$\psi_{w,d^*} = \mathsf{Encode}(\mathsf{pk}_{w,d^*}, s),$$

where d^* is the value carried by the wire w and ψ_{w,d^*} is computed as in Dec. Formally, we proceed via induction on w to show that

$$\psi_{w,d^*} \in \Psi_{\mathsf{pk}_{w,d^*},s,j}.$$

where j is the depth of wire w. The base case $w = 1,\ldots,\ell$ follows immediately from correctness of Encode. For the inductive step, consider a wire w at depth j for some gate $g = (u, v, w)$ where $u, v < w$. By the induction hypothesis,

$$\psi_{u,b^*} \in \Psi_{\mathsf{pk}_{u,b^*},s,j_0}, \quad \psi_{u,c^*} \in \Psi_{\mathsf{pk}_{v,c^*},s,j_1}$$

where $j_0, j_1 < j$ denote the depths of wires u and v respectively. It follows immediately from the correctness of Recode that

$$\psi_{w,d^*} \in \Psi_{\mathsf{pk}_{w,d^*},s,\max(i_0,i_1)+1} \subseteq \Psi_{\mathsf{pk}_{w,d^*},s,j}$$

which completes the inductive proof. Since $C(\mathsf{ind}) = 1$ and $\mathsf{pk}_{|C|,1} = \mathsf{pk}_{\mathrm{out}}$, we have $\psi_{|C|,1} \in \Psi_{\mathsf{pk}_{\mathrm{out}},s,d_{\max}}$. Finally, by the correctness of (E,D), $\mathsf{D}(\psi_{|C|,1},\tau) = m$. □

6.3 Security

Lemma 6.4 (selective security)**.** *For any adversary $\mathcal{A}$ against selective security of the attribute-based encryption scheme, there exist an adversary $\mathcal{B}$ against correlated pseudorandomness of TOR whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathrm{CP}}_{\mathcal{B}}(\lambda) + \mathrm{negl}(\lambda)$$

where $\mathrm{negl}(\lambda)$ *captures the statistical security terms in TOR.*

We begin by describing alternative algorithms, which would be useful later for constructing the adversary $\mathcal{B}$ for the correlated pseudorandomness security game.

Alternative algorithms. Fix the selective challenge ind. We get from the "outside" the challenge $\mathsf{pp}, (\mathsf{pk}_i, \psi_i)_{i\in[\ell+1]}$ for correlated pseudorandomness, The main challenge is to design an alternative algorithm KeyGen^* for answering secret key queries without knowing $\mathsf{sk}_{1,\mathsf{ind}_1},\ldots,\mathsf{sk}_{\ell,\mathsf{ind}_\ell}$ or $\mathsf{sk}_{\mathrm{out}}$. The algorithm KeyGen^* will maintain the following invariant: on input C with $C(\mathsf{ind}) = 0$,

- for every non-output wire $w = 1,\ldots,|C|-1$ carrying the value b^*, we will know $\mathsf{sk}_{w,1-b^*}$ but not sk_{w,b^*}.

Moreover, we do not know $\mathsf{sk}_{|C|,0}$ or $\mathsf{sk}_{|C|,1} = \mathsf{sk}_{\mathrm{out}}$.

$\mathsf{Setup}^*(\mathsf{ind}, 1^\lambda, 1^\ell, d_{\max})$: Let

$$\begin{aligned} (\mathsf{pk}_{i,1-\mathsf{ind}_i}, \mathsf{sk}_{i,1-\mathsf{ind}_i}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}) \text{ for } i \in [\ell] \\ \mathsf{pk}_{\mathrm{out}} &:= \mathsf{pk}_{\ell+1} \\ \mathsf{pk}_{i,\mathsf{ind}_i} &:= \mathsf{pk}_i \text{ for } i \in [\ell] \end{aligned}$$

$$\text{Output } \mathsf{mpk} = \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \ldots & \mathsf{pk}_{\ell,0} & \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \ldots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\mathrm{out}} \end{pmatrix}$$

$\mathsf{Enc}^*(\mathsf{mpk}, \mathsf{ind}, m)$: Set $\tau \leftarrow \mathsf{E}(\psi_{\ell+1}, m)$ and output the ciphertext

$$\mathsf{ct}_{\mathsf{ind}} = \begin{pmatrix} \psi_1, & \psi_2, & \ldots, & \psi_\ell, & \tau \end{pmatrix}$$

where $\psi_1, \ldots, \psi_{\ell+1}$ are provided in the challenge.

$\mathsf{KeyGen}^*(\mathsf{ind}, \mathsf{msk}, C)$: where $C(\mathsf{ind}) = 0$,

1. For each internal wire $w \in [\ell+1, |C|-1]$ of the circuit C carrying the value b^* for input ind, generate public/secret key pairs:

$$(\mathsf{pk}_{w,1-b^*}, \mathsf{sk}_{w,1-b^*}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$$

   We will generate pk_{w,b^*} using $\mathsf{SimReKeyGen}$ as described next.

2. For $w = \ell+1, \ldots, |C|$, let $g = (u, v, w)$ denote the gate for which w is the outgoing wire. Suppose wires u and v carry the values b^* and c^*, so that wire w carries the value $d^* := g_w(b^*, c^*)$. By the invariant above, we know $\mathsf{sk}_{u,1-b^*}$ and $\mathsf{sk}_{v,1-c^*}$ but not sk_{u,b^*} and sk_{v,c^*}. We start by generating

$$(\mathsf{pk}_{w,d^*}, \mathsf{rk}^w_{b^*,c^*}) \leftarrow \mathsf{SimReKeyGen}(\mathsf{pk}_{u,b^*}, \mathsf{pk}_{v,c^*})$$

   We generate the other three recoding keys using $\mathsf{ReKeyGen}$ as follows:

$$\begin{aligned}
\mathsf{rk}^w_{1-b^*,c^*} &\leftarrow \mathsf{ReKeyGen}\big(\mathsf{pk}_{u,1-b^*}, \mathsf{pk}_{v,c^*}, \quad \mathsf{sk}_{u,1-b^*}, \mathsf{pk}_{w,g_w(1-b^*,c^*)}\big)\\
\mathsf{rk}^w_{b^*,1-c^*} &\leftarrow \mathsf{ReKeyGen}\big(\mathsf{pk}_{v,1-c^*}, \mathsf{pk}_{u,b^*}, \quad \mathsf{sk}_{v,1-c^*}, \mathsf{pk}_{w,g_w(b^*,1-c^*)}\big)\\
\mathsf{rk}^w_{1-b^*,1-c^*} &\leftarrow \mathsf{ReKeyGen}\big(\mathsf{pk}_{u,1-b^*}, \mathsf{pk}_{v,1-c^*}, \mathsf{sk}_{u,1-b^*}, \mathsf{pk}_{w,g_w(1-b^*,1-c^*)}\big)
\end{aligned}$$

   Note that $\mathsf{rk}^w_{1-b^*,c*}, \mathsf{rk}^w_{1-b^*,1-c*}$ are generated the same way in both KeyGen and KeyGen^* using $\mathsf{sk}_{u,1-b^*}$.

Output the secret key

$$\mathsf{sk}_C := \Big(\mathsf{rk}^w_{b,c} \,:\, w \in \big[\, \ell+1, |C| \,\big],\, b, c \in \{0,1\} \Big)$$

Informally, the recoding key $\mathsf{rk}^w_{b^*,1-c^*}$ looks the same as in Keygen because of key indistinguishability, and $\mathsf{rk}^w_{b^*,c^*}$ (together with the simulated pk_{w,d^*}) looks the same as in Keygen because of the recoding simulation property.

Game sequence. Next, consider the following sequence of games. We use $\mathsf{Adv}_0, \mathsf{Adv}_1, \ldots$ to denote the advantage of the adversary $\mathcal{A}$ in Games 0, 1, etc. Game 0 is the real experiment.

Game i for $i = 1, 2, \ldots, q$. As in Game 0, except the challenger answers the first $i-1$ key queries using KeyGen^* and the remaining $q-i$ key queries using KeyGen. For the i'th key query C_i, we consider sub-Games $i.w$ as follows:

Game $i.w$, for $w = \ell+1, \ldots, |C_i|$. The challenger switches $(\mathsf{rk}^w_{b,c} : b, c \in \{0,1\})$ from KeyGen to KeyGen^*. More precisely:

- First, we switch $(\mathsf{pk}_{w,d^*}, \mathsf{rk}^w_{b^*,c*})$ from KeyGen to KeyGen^*. This relies on recoding simulation.
- Next, we switch $\mathsf{rk}^w_{b^*,1-c*}$ from KeyGen to KeyGen^*. This relies on key indistinguishability, w.r.t. sk_{b^*} and sk_{1-c^*}.

- The other two keys $\mathsf{rk}^{w}_{1-b^*,c*}, \mathsf{rk}^{w}_{1-b^*,1-c*}$ are generated the same way in both KeyGen and KeyGen^*.

By key indistinguishability and recoding simulation, we have

$$|\mathsf{Adv}_{i,w} - \mathsf{Adv}_{i,w+1}| \leq \mathrm{negl}(\lambda) \text{ for all } i, w$$

Note that in Game q, the challenger runs Setup^* and answers all key queries using KeyGen^* with the selective challenge ind and generates the challenge ciphertext using Enc.

Game $q+1$**.** Same as Game q, except the challenger generates the challenge ciphertext using Enc^* with $\psi_{\ell+1} = \mathsf{Encode}(\mathsf{pk}_{\ell+1}, s)$. Clearly,

$$\mathsf{Adv}_{q+1} = \mathsf{Adv}_q$$

Game $q+2$**.** Same as Game $q+1$, except $\psi_{\ell+1} \xleftarrow{\$} \mathcal{K}$. It is straight-forward to construct an adversary $\mathcal{B}$ such that

$$|\mathsf{Adv}_{q+1} - \mathsf{Adv}_{q+2}| \leq \mathsf{Adv}^{\mathrm{CP}}_{\mathcal{B}}(\lambda)$$

Finally, $\mathsf{Adv}_{q+2} \leq \mathrm{negl}(\lambda)$ by the one-time semantic security of (E, D). The lemma then follows readily.

7 Attribute-Based Encryption for Branching Programs

In this section, we present weak TOR and attribute-based encryption for branching programs, which capture the complexity class log-space. As noted in Section 2.2, we exploit the fact that in branching programs, the transition function depends on an input variable and the current state; this means that one of the two input encodings during recoding is always a "depth 0" encoding.

Branching programs. Recall that a branching program Γ is a directed acyclic graph in which every nonterminal node has exactly two outgoing edges labeled $(i,0)$ and $(i,1)$ for some $i \in [\ell]$. Moreover, there is a distinguished terminal accept node. Every input $x \in \{0,1\}^{\ell}$ naturally induces a subgraph Γ_x containing exactly those edges labeled (i, x_i). We say that Γ accepts x iff there is a path from the start node to the accept node in Γ_x. At the cost of possibly doubling the number of edges and vertices, we may assume that there is at most one edge connecting any two nodes in Γ.

7.1 Weak TOR

A weak "two-to-one" encoding (wTOR) scheme consists of the same algorithms as TOR, except that $\mathsf{Keygen}(\mathsf{pp}, j)$ takes an additional input $j \in \{0,1\}$. That is, Keygen may produce different distribution of public/secret key pairs depending on j. Moreover, in $\mathsf{ReKeyGen}$, the first public key is always generated using $\mathsf{Keygen}(\mathsf{pp}, 0)$ and the second using $\mathsf{Keygen}(\mathsf{pp}, 1)$; similarly, in Recode, the first encoding is always generated with respect to a public key from $\mathsf{Keygen}(\mathsf{pp}, 0)$ and the second from $\mathsf{Keygen}(\mathsf{pp}, 1)$. Similarly, the correctness and statistical security properties are relaxed.

Correctness. First, for every pk and $s \in \mathcal{S}$, there exists a family of sets $\Psi_{\mathsf{pk},s,j}$, $j = 0, 1, \ldots, d_{\max}$:

- $\Psi_{\mathsf{pk},s,1} \subseteq \cdots \subseteq \Psi_{\mathsf{pk},s,d_{\max}}$.
- for all $\psi, \psi' \in \Psi_{\mathsf{pk},s,d_{\max}}$ and all $m \in \mathcal{M}$,

$$\mathsf{D}(\psi', \mathsf{E}(\psi, m)) = m$$

Secondly, the correctness of recoding requires that for any triple of key pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}})$ respectively in the support of $\mathsf{Keygen}(\mathsf{pp}, 0), \mathsf{Keygen}(\mathsf{pp}, 1), \mathsf{Keygen}(\mathsf{pp}, 1)$ and any encodings $\psi_0 \in \mathsf{Encode}(\mathsf{pk}_0, s)$ and $\psi_1 \in \Psi_{\mathsf{pk}_1,s,j_1}$ where $0 < j_1$,

$$\mathsf{Recode}(\mathsf{rk}, \psi_0, \psi_1) \in \Psi_{\mathsf{pk}_{\mathrm{tgt}},s,j_1+1}$$

Statistical Security Properties. We require recoding simulation as before, but not key indistinguishability. However, we require the following additional property:

(Back-tracking) : For all $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 0)$ and all $(\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_{\mathrm{tgt}}) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 1)$, the following distributions are identical:

$$\mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{pk}_{\mathrm{tgt}}) \equiv -\mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_{\mathrm{tgt}}, \mathsf{sk}_0, \mathsf{pk}_1)$$

Informally, this says that switching the order of pk_1 and $\mathsf{pk}_{\mathrm{tgt}}$ as inputs to $\mathsf{ReKeyGen}$ is the same as switching the "sign" of the output. In our instantiations, the output of $\mathsf{ReKeyGen}$ lies in a group, so negating the output simply refers to applying the group inverse operation.

Remark 7.1. *Due to the additional back-tracking property, it is not the case that a TOR implies a weak TOR. However, we are able to instantiate weak TOR under weaker and larger classes of assumptions than TOR.*

Computational Security Property. We define the advantage function $\mathsf{Adv}^{\mathrm{CP}}_{\mathcal{A}}(\lambda)$ (modified to account for the additional input to Keygen) to be the absolute value of:

$$\Pr\left[\, b = b' : \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda); s \leftarrow \mathcal{S}; \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 0), \\ \psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_i, s), i = 1, \ldots, \ell; \\ (\mathsf{pk}_{\ell+1}, \mathsf{sk}_{\ell+1}) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 1); \\ \psi'_0 \leftarrow \mathsf{Encode}(\mathsf{pk}_{\ell+1}, s); \\ b \xleftarrow{\$} \{0,1\}; \psi'_1 \xleftarrow{\$} \mathcal{K} \\ b' \leftarrow \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_{\ell+1}, \psi_1, \ldots, \psi_\ell, \psi'_b) \end{array} \right] - \frac{1}{2}$$

and we require that for all PPT $\mathcal{A}$, the advantage function $\mathsf{Adv}^{\mathrm{CP}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.

7.2 Weak TOR from LWE

We provide an instantiation of weak TOR from LWE. The main advantage over our construction of TOR in Section 5 is that the dependency of q on $d_{\max}$ is linear in $d_{\max}$ instead of exponential. Therefore, if q

is quasi-polynomial, we can handle any polynomial $d_{\max}$, as opposed to an a-prior bounded $d_{\max}$.

Lemma 7.1. *Assuming* $\mathsf{dLWE}_{n,(\ell+2)m,q,\chi}$ *where* $q = O(d_{\max} n^3 \log n)$*, there is a weak TOR scheme that is correct up to* $d_{\max}$ *levels.*

Note that the parameters here are better than in Lemma 5.1. The construction of weak TOR from learning with errors follows:

- $\mathsf{Params}(1^\lambda, d_{\max})$: First choose the LWE dimension $n = n(\lambda)$. Let the error distribution $\chi = \chi(n) = D_{\mathbb{Z},\sqrt{n}}$, the error bound $B = B(n) = O(n)$, the modulus $q = q(n) = d_{\max} \cdot O(n^3 \log n)$, the number of samples $m = m(n) = O(n \log q)$ and the Gaussian parameter $s = s(n) = O(\sqrt{n \log q})$. Output the global public parameters $\mathsf{pp} = (n, \chi, B, q, m, s)$. Define the domain $\mathcal{S}$ of the encoding scheme to be $\mathbb{Z}_q^n$.
- $\mathsf{Keygen}(\mathsf{pp}, j)$: Run the trapdoor generation algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with the trapdoor $\mathbf{T}$. Output

$$\mathsf{pk} = \mathbf{A}; \quad \mathsf{sk} = \mathbf{T}.$$

- $\mathsf{Encode}(\mathbf{A}, \mathbf{s})$: Sample an error vector $\mathbf{e} \leftarrow \chi^m$ and output the encoding $\boldsymbol{\psi} := \mathbf{A}^T \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^n$.
- $\mathsf{ReKeyGen}(\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_{\mathrm{tgt}}, \mathbf{T})$: Outputs a low-norm matrix $\mathbf{R}$ such that $\mathbf{A}_0 \mathbf{R} = \mathbf{A}_{\mathrm{tgt}} - \mathbf{A}_1$. In particular,

$$\mathbf{R} \leftarrow \mathsf{SampleD}(\mathbf{A}_0, \mathbf{T}_0, \mathbf{A}_{\mathrm{tgt}} - \mathbf{A}_1, s)$$

- $\mathsf{SimReKeyGen}(\mathbf{A}_0, \mathbf{A}_1)$: Sample a matrix $\mathbf{R} \leftarrow (D_{\mathbb{Z},s})^{m \times m}$ by sampling each entry from the discrete Gaussian distribution $D_{\mathbb{Z},s}$. Output

$$\mathsf{rk} := \mathbf{R}; \quad \mathbf{A}_{\mathrm{tgt}} := \mathbf{A}_0 \mathbf{R} + \mathbf{A}_1$$

- $\mathsf{Recode}(\mathsf{rk}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1)$: Outputs $\mathsf{rk}^T \boldsymbol{\psi}_0 + \boldsymbol{\psi}_1$.

Correctness. We define the sets $\Psi_{\mathbf{A},\mathbf{s},j}$ for $\mathsf{pk} := \mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and $j \in [1 \ldots d_{\max}]$ as follows:

$$\Psi_{\mathbf{A},\mathbf{s},j} = \left\{ \mathbf{A}^T \mathbf{s} + \mathbf{e} : \ ||\mathbf{e}||_\infty \leq B \cdot j \cdot (sm\sqrt{m}) \right\}$$

The analysis is similar to that in the previous section. In particular, we observe right away that

- $\Psi_{\mathbf{A},\mathbf{s},1} \subseteq \Psi_{\mathbf{A},\mathbf{s},1} \subseteq \ldots \subseteq \Psi_{\mathbf{A},\mathbf{s},d_{\max}}$.
- For any two encodings $\boldsymbol{\psi}, \boldsymbol{\psi}' \in \Psi_{\mathbf{A},\mathbf{s},d_{\max}}$ and $\boldsymbol{\mu} \in \{0,1\}^n$, $\mathsf{D}(\boldsymbol{\psi}', \mathsf{E}(\boldsymbol{\psi}, \boldsymbol{\mu})) = \boldsymbol{\mu}$, as long as

$$B \cdot d_{\max} \cdot (sm\sqrt{m}) \leq q/4.$$

- Consider two encodings $\mathbf{A}^T \mathbf{s} + \mathbf{e} \in \mathsf{Encode}(\mathbf{A}, \mathbf{s})$ and $\boldsymbol{\psi}_1 \in \Psi_{\mathbf{A}_1,\mathbf{s},j_1}$ for any $j_1 \in \mathbb{N}$. Then, $\boldsymbol{\psi}_0 = \mathbf{A}_0^T \mathbf{s} + \mathbf{e}_0$ and $\boldsymbol{\psi}_1 := \mathbf{A}_1^T \mathbf{s} + \mathbf{e}_1$ where $||\mathbf{e}_0||_\infty \leq B$ and $||\mathbf{e}_1||_\infty \leq j_1 \cdot B \cdot (sm\sqrt{m})$.

Then, the recoded ciphertext $\boldsymbol{\psi}_{\text{tgt}}$ is computed as follows:

$$\begin{aligned}\boldsymbol{\psi}_{\text{tgt}} &:= \mathbf{R}^T\boldsymbol{\psi}_0 + \boldsymbol{\psi}_1 \\ &= \mathbf{R}^T(\mathbf{A}_0^T\mathbf{s} + \mathbf{e}_0) + (\mathbf{A}_1^T\mathbf{s} + \mathbf{e}_1) \\ &= \mathbf{A}_{\text{tgt}}^T\mathbf{s} + \mathbf{e}_{\text{tgt}}\end{aligned}$$

where the last equation is because $\mathbf{A}_{\text{tgt}} = \mathbf{A}_0\mathbf{R} + \mathbf{A}_1$ and we define $\mathbf{e}_{\text{tgt}} := \mathbf{R}^T\mathbf{e}_0 + \mathbf{e}_1$. Thus,

$$\begin{aligned}||\mathbf{e}_{\text{tgt}}||_\infty &\le m \cdot ||\mathbf{R}||_\infty ||\mathbf{e}_0||_\infty + ||\mathbf{e}_1||_\infty \\ &\le m \cdot s\sqrt{m} \cdot B + B \cdot j_1 \cdot (sm\sqrt{m}) \\ &= (j_1 + 1) \cdot B \cdot (sm\sqrt{m})\end{aligned}$$

exactly as required. Here, the second inequality is because $||\mathbf{R}||_\infty \le s\sqrt{m}$ by Lemma 3.1. This finishes our proof of correctness.

Security. Correlated pseudorandomness follows from $\mathsf{dLWE}_{n,(\ell+2)m,q,\chi}$ where $q = n \cdot d_{\max}$. Recoding simulation follows from Lemma 3.1 by an argument identical to the one for the construction of TOR in Section 5. For back-tracking, negation is simply the additive inverse over $\mathbb{Z}_q^m$.

7.3 Weak TOR from Bilinear Maps

We use asymmetric groups for maximal generality and for conceptual clarity. We consider cyclic groups G_1, G_2, G_T of prime order q and $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map. We require that the group operations in G and G_T as well the bilinear map e are computable in deterministic polynomial time with respect to λ. Let g_1, g_2 denote random generators of G_1, G_2 respectively. The DBDH Assumption says that, given $g_1, g_2, g_1^a, g_2^a, g_2^b$ and g_1^s, $e(g_1, g_2)^{abs}$ is pseudorandom.

- $\mathsf{Params}(1^\lambda, d_{\max})$: Outputs $\mathsf{pp} := (g_1, g_2, g_1^a, g_2^a)$.
- $\mathsf{Keygen}(\mathsf{pp}, j)$:
  - If $j = 0$, then samples $t \xleftarrow{\$} \mathbb{Z}_q$ and outputs
  $$(\mathsf{pk}, \mathsf{sk}) := ((g_1^{a/t}, g_2^{a/t}), t)$$
  - If $j \ge 1$, output $\mathsf{pk} \xleftarrow{\$} G_2$.
- $\mathsf{Encode}(\mathsf{pk}, s)$:
  - If $\mathsf{pk} = (g_1^{a/t}, g_2^{a/t}) \in G_1 \times G_2$, output $(g_1^{a/t})^s$
  - If $\mathsf{pk} \in G_2$, output $e(g_1^a, \mathsf{pk})^s$
- $\mathsf{Recode}(\mathsf{rk}, c_0, , c_1)$: Outputs $e(c_0, \mathsf{rk}) \cdot c_1$.
- $\mathsf{ReKeyGen}((g_1^{a/t}, g_2^{a/t}), \mathsf{pk}_1, \mathsf{pk}_{\text{tgt}}, t)$: Outputs $\mathsf{rk} := (\mathsf{pk}_{\text{tgt}} \cdot \mathsf{pk}_1^{-1})^t \in G_2$.

- $\mathsf{SimReKeyGen}((g_1^{a/t}, g_2^{a/t}), \mathsf{pk}_1)$: Picks $z \xleftarrow{\$} Z_q$ and outputs

$$\mathsf{rk} := (g_2^{a/t})^z, \quad \mathsf{pk}_{\mathrm{tgt}} := \mathsf{pk}_1 \cdot (g_2^a)^z$$

Correctness. Define $\Psi_{\mathsf{pk},s,j} := \{\mathsf{Encode}(\mathsf{pk}, s)\}$. For recoding, observe that:

$$\begin{aligned}
&\mathsf{Recode}((\mathsf{pk}_{\mathrm{tgt}} \cdot \mathsf{pk}_1^{-1})^t, g_1^{as/t}, e(g_1^a, \mathsf{pk}_1)^s \\
&= e(g_1^{as/t}, (\mathsf{pk}_{\mathrm{tgt}} \cdot \mathsf{pk}_1^{-1})^t) \cdot e(g_1^a, \mathsf{pk}_1)^s \\
&= e(g_1^a, (\mathsf{pk}_{\mathrm{tgt}} \cdot \mathsf{pk}_1^{-1})^s) \cdot e(g_1^a, \mathsf{pk}_1)^s \\
&= e(g_1^a, \mathsf{pk}_{\mathrm{tgt}})^s = \mathsf{Encode}(\mathsf{pk}_{\mathrm{tgt}}, s)
\end{aligned}$$

For back-tracking, negation is simply the multiplicative inverse over G_q.

Security. Correlation pseudorandomness follows readily from the DBDH assumption and its random self-reducibility.

7.4 Attribute-Based Encryption from weak TOR

$\mathsf{Setup}(1^\lambda, 1^\ell, d_{\max})$: For each one of ℓ input bits, generate two public/secret key pairs. Also, generate a public/secret key pair for the start and accept states:

$$\begin{aligned}
(\mathsf{pk}_{i,b}, \mathsf{sk}_{i,b}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}, 0) \quad \text{for } i \in [\ell], b \in \{0,1\} \\
(\mathsf{pk}_{\mathrm{start}}, \mathsf{sk}_{\mathrm{start}}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}, 1) \\
(\mathsf{pk}_{\mathrm{accept}}, \mathsf{sk}_{\mathrm{accept}}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}, 1)
\end{aligned}$$

Output

$$\mathsf{mpk} := \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \dots & \mathsf{pk}_{\ell,0} & \mathsf{pk}_{\mathrm{start}} \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \dots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\mathrm{accept}} \end{pmatrix}$$

$$\mathsf{msk} := \begin{pmatrix} \mathsf{sk}_{1,0} & \mathsf{sk}_{2,0} & \dots & \mathsf{sk}_{\ell,0} & \mathsf{sk}_{\mathrm{start}} \\ \mathsf{sk}_{1,1} & \mathsf{sk}_{2,1} & \dots & \mathsf{sk}_{\ell,1} & \mathsf{sk}_{\mathrm{accept}} \end{pmatrix}$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m)$: For $\mathsf{ind} \in \{0,1\}^\ell$, choose a uniformly random $s \xleftarrow{\$} \mathcal{S}$ and encode it under the public keys specified by the index bits and the start state:

$$\begin{aligned}
\psi_i &\leftarrow \mathsf{Encode}(\mathsf{pk}_{i,\mathsf{ind}_i}, s) \text{ for all } i \in [\ell] \\
\psi_{\mathrm{start}} &\leftarrow \mathsf{Encode}(\mathsf{pk}_{\mathrm{start}}, s)
\end{aligned}$$

Encrypt the message:

$$\tau \leftarrow \mathsf{E}(\mathsf{Encode}(\mathsf{pk}_{\mathrm{accept}}, s), m)$$

Output the ciphertext:

$$\mathsf{ct}_{\mathsf{ind}} = \begin{pmatrix} \psi_1, & \psi_2, & \dots, & \psi_\ell, & \psi_{\mathrm{start}}, & \tau \end{pmatrix}$$

$\mathsf{KeyGen}(\mathsf{msk}, \Gamma)$: $\Gamma : \{0,1\}^\ell \to \{0,1\}$ is a branching program that takes a ℓ-bit input and outputs a single bit.

- For every node u, except the start and accept nodes, sample public/secret key pair:

$$(\mathsf{pk}_u, \mathsf{sk}_u) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 1)$$

- For every edge (u, v) labeled (i, b) in Γ, sample a recoding key $\mathsf{rk}_{u,v}$ as follows:

$$\mathsf{rk}_{u,v} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{i,b}, \mathsf{pk}_u, \mathsf{sk}_{i,b}, \mathsf{pk}_v\Big)$$

The secret key sk_Γ is the collection of all the recoding keys $\mathsf{rk}_{u,v}$ for every edge (u, v) in Γ.

$\mathsf{Dec}(\mathsf{sk}_\Gamma, \mathsf{ct}_{\mathsf{ind}})$: Suppose $\Gamma(\mathsf{ind}) = 1$; output $\perp$ otherwise. Let Π denote the (directed) path from the start node to the accept node in Γ_{ind}. For every edge (u, v) labeled (i, ind_i) in Π, apply the recoding algorithm on the two encodings ψ_i, ψ_u and the recoding key $\mathsf{rk}_{u,v}$:

$$\psi_v \leftarrow \mathsf{Recode}\Big(\mathsf{rk}_{u,v}, \psi_i, \psi_u\Big)$$

If $\Gamma(\mathsf{ind}) = 1$, we obtain ψ_{accept}. Decrypt and output the message:

$$m \leftarrow \mathsf{D}(\psi_{\text{accept}}, \tau)$$

7.4.1 Correctness

Lemma 7.2 (correctness)**.** *Let $\mathcal{G} = \{\Gamma\}_\lambda$ be a collection of polynomial-size branching programs of depth at most $d_{\max}$ and let* wTOR *be a weak "two-to-one" recoding scheme for $d_{\max}$ levels. Then, the construction presented above is a correct attribute-based encryption scheme for $\mathcal{G}$.*

Proof. Let Π denote the directed path from the start to the accept nodes in Γ_{ind}. We show via induction on nodes v along the path Π that

$$\psi_v \in \Psi_{\mathsf{pk}_v, s, j}$$

where j is the depth of node v along the path. The base case for $v :=$ start node follows immediately from correctness of Encode. For the inductive step, consider a node v along the path Π at depth j for some edge (u, v) labeled (i, ind_i). By the induction hypothesis,

$$\psi_u \in \Psi_{\mathsf{pk}_u, s, j_0}$$

where $j_0 < j$ denote the depths of node u. Also by the correctness of the Encode algorithm, for all $i \in [\ell]$

$$\psi_i \in \Psi_{\mathsf{pk}_{i,\mathsf{ind}_i}, s, 0}$$

It follows immediately from the correctness of Recode that

$$\psi_v \in \Psi_{\mathsf{pk}_v, s, j_0+1} \subseteq \Psi_{\mathsf{pk}_v, s, j}$$

which completes the inductive proof. Since $C(\mathsf{ind}) = 1$, we have

$$\psi_{\text{accept}} \in \Psi_{\mathsf{pk}_{\text{accept}},s,d_{\max}}$$

Recall that $\tau \leftarrow \mathsf{E}(\mathsf{Encode}(\mathsf{pk}_{\text{accept}}, s), m)$. Finally, by the correctness of (E, D),

$$\mathsf{D}(\psi_{\text{accept}}, \tau) = m$$ □

7.4.2 Selective Security

Lemma 7.3 (selective security)**.** *For any adversary $\mathcal{A}$ against selective security of the attribute-based encryption scheme for branching programs, there exist an adversary $\mathcal{B}$ against correlated pseudorandomness of wTOR whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$\mathsf{Adv}^{\text{PE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{CP}}_{\mathcal{B}}(\lambda) + \text{negl}(\lambda)$$

where $\text{negl}(\lambda)$ *captures the statistical security terms in TOR.*

In the proof of security, we will rely crucially on the following combinatorial property of branching programs: for any input x, the graph Γ_x does not contain any cycles as an *undirected* graph.

Alternative algorithms. Fix the selective challenge ind. We also get a collection of public keys, corresponding encodings from the "outside": $(\mathsf{pk}_i, \psi_i)_{i\in[\ell+2]}$, where the challenge is to decide whether $\psi_{\ell+1}$ is $\mathsf{Encode}(\mathsf{pk}_{\ell+2}, s)$ or random. The main challenge is design an alternative algorithm KeyGen^* for answering secret key queries without knowing $\mathsf{sk}_{1,\mathsf{ind}_1}, \ldots, \mathsf{sk}_{\ell,\mathsf{ind}_\ell}$ or $\mathsf{sk}_{\text{start}}, \mathsf{sk}_{\text{accept}}$. We consider the following "alternative" algorithms.

$\mathsf{Setup}^*(1^\lambda, 1^\ell, d_{\max})$: Let

$$\begin{aligned}
(\mathsf{pk}_{i,1-\mathsf{ind}_i}, \mathsf{sk}_{i,1-\mathsf{ind}_i}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}, 0) \text{ for } i \in [\ell] \\
\mathsf{pk}_{i,\mathsf{ind}_i} &:= \mathsf{pk}_i \text{ for } i \in [\ell] \\
\mathsf{pk}_{\text{start}} &:= \mathsf{pk}_{\ell+1} \\
\mathsf{pk}_{\text{accept}} &:= \mathsf{pk}_{\ell+2}
\end{aligned}$$

Define and output the master public key as follows:

$$\mathsf{mpk} = \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \ldots & \mathsf{pk}_{\ell,0} & \mathsf{pk}_{\text{start}} \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \ldots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\text{accept}} \end{pmatrix}$$

$\mathsf{Enc}^*(\mathsf{mpk}, \mathsf{ind}, m)$: Define

$$\begin{aligned}
\psi_{i,\mathsf{ind}_i} &:= \psi_i \quad \text{for all } i \in [\ell] \\
\psi_{\text{start}} &:= \psi_{\ell+1} \\
\psi_{\text{accept}} &:= \psi_{\ell+2}
\end{aligned}$$

Encrypt the message m:

$$\tau \leftarrow \mathsf{E}(\psi_{\text{accept}}, b)$$

Output the simulated ciphertext

$$\mathsf{ct}_{\mathsf{ind}} = \left(\ \psi_1, \quad \psi_2, \quad \ldots, \quad \psi_\ell, \quad \psi_{\text{start}}, \quad \tau \ \right)$$

$\mathsf{KeyGen}^*(\mathsf{msk}, \Gamma)$: Let Γ'_{ind} denote the *undirected* graph obtained from Γ_{ind} by treating every directed edge as an undirected edge (while keeping the edge label). Observe that Γ'_{ind} satisfies the following properties:

- Γ'_{ind} contains no cycles. This is because Γ_{ind} is acyclic and every nonterminal node contains exactly one outgoing edge.
- The start node and the accept node lie in different connected components in Γ'_{ind}, since $\Gamma(\mathsf{ind}) = 0$.

Simulation invariant: for each "active" edge labeled (i, ind_i) from node u to node v, simulate the recoding key. Choose our own public/secret key pair for each "inactive" edges $(i, 1-\mathsf{ind}_i)$ and generate the recoding key honestly.

- Run a DFS in Γ'_{ind} starting from the start node. Whenever we visit a new node v from a node u along an edge labeled (i, ind_i), we set:

$$\begin{array}{lcll} (\mathsf{pk}_v, \mathsf{rk}_{u,v}) & \leftarrow & \mathsf{SimReKeyGen}\big(\mathsf{pk}_{i,\mathsf{ind}}, \mathsf{pk}_u\big) & \text{if } (u,v) \text{ is a directed edge in } \Gamma \\ (\mathsf{pk}_v, -\mathsf{rk}_{v,u}) & \leftarrow & \mathsf{SimReKeyGen}\big(\mathsf{pk}_{i,\mathsf{ind}}, \mathsf{pk}_u\big) & \text{if } (v,u) \text{ is a directed edge in } \Gamma \end{array}$$

  Here, we exploit the back-tracking property in wTOR.

  Note that since $\Gamma(\mathsf{ind}) = 0$, then the accept node is not assigned a public key by this process.
- For all nodes u without an assignment, run $(\mathsf{pk}_u, \mathsf{sk}_u) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 1)$.
- For every remaining edge (u, v) labeled $(i, 1-\mathsf{ind}_i)$ in Γ, sample a recoding key $\mathsf{rk}_{u,v}$ as in KeyGen using $\mathsf{sk}_{i,1-\mathsf{ind}}$ as follows:

$$\mathsf{rk}_{u,v} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{i,1-\mathsf{ind}}, \mathsf{pk}_u, \mathsf{sk}_{i,1-\mathsf{ind}}, \mathsf{pk}_v\Big)$$

The secret key sk_Γ is simply the collection of all the recoding keys $\mathsf{rk}_{u,v}$ for every edge (u, v) in Γ.

Game sequence. Next, consider the following sequence of games. We use $\mathsf{Adv}_0, \mathsf{Adv}_1, \ldots$ to denote the advantage of the adversary $\mathcal{A}$ in Games 0, 1, etc. Let n denote the number of edges in a branching program Γ labeled (i, ind_i) for some i, and for all $j \in [n]$ let e_j denote the actual edge.

Game 0**.** Real experiment.

Game i **for** $i = 1, 2, \ldots, q$**.** As in Game 0, except the challenger answers the first $i-1$ key queries using KeyGen^* and the remaining $q-i$ key queries using KeyGen. For the i'th key query Γ_i, we consider sub-Games $i.e$ as follows:

Game $i.j$, for $j = 1,\ldots,n$. For edge $e_j = (u,v)$ labeled (i,ind_i), the challenger switches the simulated recoding key $\mathsf{rk}_{u,v}$ from KeyGen to KeyGen^*. We rely on recoding simulation and back-tracking properties simultaneously.

By recoding simulation and back-tracking, we have:

$$|\mathsf{Adv}_{i,e} - \mathsf{Adv}_{i,e+1}| \leq \mathrm{negl}(\lambda) \text{ for all } i,e$$

Note that in Game q, the challenger runs Setup^* and answers all key queries using KeyGen^* with the selective challenge ind and generates the challenge ciphertext using Enc.

Game $q+1$. Same as Game q, except the challenger generates the challenge ciphertext using Enc^* with $\psi_{\ell+2} \leftarrow \mathsf{Encode}(\mathsf{pk}_{\ell+2}, s)$.

$$\mathsf{Adv}_{q+1} = \mathsf{Adv}_q$$

Game $q+2$. Same as Game $q+1$, except $\psi_{\ell+2} \xleftarrow{\$} \mathcal{K}$. It is straight-forward to construct an adversary $\mathcal{B}$ such that

$$|\mathsf{Adv}_{q+1} - \mathsf{Adv}_{q+2}| \leq \mathsf{Adv}^{\mathrm{CP}}_{\mathcal{B}}(\lambda)$$

Finally, $\mathsf{Adv}_{q+2} \leq \mathrm{negl}(\lambda)$ by the one-time semantic security of (E,D). The lemma then follows readily.

8 Extensions

8.1 Outsourcing Decryption

In this section we show how to modify our main construction of attribute-based encryption to support outsourcing of decryption circuits, similar to [90]. We require that the Keygen algorithm returns two keys:

- the evaluation key ek_C, that is given to a computationally powerful proxy,
- and a decryption key dk, given to the client.

Given a ciphertext $\mathsf{ct}_{\mathsf{ind}}$, the proxy must perform the "bulk" of the computation and return a new ciphertext $\mathsf{ct}'_{\mathsf{ind}}$ that is forwarded to the client. Using the decryption key dk, the client can decrypt and learn the message m iff the predicate $C(\mathsf{ind})$ is satisfied. We emphasize that that amount of computation the client needs to perform to decrypt the message must be independent on the circuit size. Intuitively, the security ensures that an adversary should learn nothing about the message, conditioned on that it queries for decryption keys dk's for predicates that are not satisfied by the challenge index (note, the adversary *can* query for evaluation keys separately for predicates that are satisfied).

Intuitively, we modify the main construction as follows. As before, the key-generation algorithm assigns two keys for each circuit wire. The evaluation key consists of all the recoding keys for the circuit. In addition, the output wire has another key $\mathsf{pk}_{\mathrm{out}}$ which now plays a special role. The recoding key from $\mathsf{pk}_{|C|,1}$ to $\mathsf{pk}_{\mathrm{out}}$ is only given to the client as the decryption key. If $C(\mathsf{ind}) = 1$, the the proxy computes

an encoding under the $\mathsf{pk}_{|C|,1}$ and forwards it to the client. The client applies the transformation, and decrypts the message. For technical reasons, since we are using "two-to-one" recoding mechanism, we need to introduce an auxiliary public key pk_{in} and a corresponding encoding.

$\mathsf{Setup}(1^\lambda, 1^\ell, d_{\max})$: For each of the ℓ input wires, generate two public/secret key pairs. Also, generate an additional public/secret key pair:

$$\begin{aligned}(\mathsf{pk}_{i,b}, \mathsf{sk}_{i,b}) &\leftarrow \mathsf{Keygen}(\mathsf{pp}) \qquad \text{for } i \in [\ell], b \in \{0,1\}\\ (\mathsf{pk}_{\text{out}}, \mathsf{sk}_{\text{out}}) &\leftarrow \mathsf{Keygen}(\mathsf{pp})\\ (\mathsf{pk}_{\text{in}}, \mathsf{sk}_{\text{in}}) &\leftarrow \mathsf{Keygen}(\mathsf{pp})\end{aligned}$$

Output

$$\mathsf{mpk} := \begin{pmatrix} \mathsf{pk}_{1,0} & \mathsf{pk}_{2,0} & \dots & \mathsf{pk}_{\ell,0} & \mathsf{pk}_{\text{in}} \\ \mathsf{pk}_{1,1} & \mathsf{pk}_{2,1} & \dots & \mathsf{pk}_{\ell,1} & \mathsf{pk}_{\text{out}} \end{pmatrix} \qquad \mathsf{msk} := \begin{pmatrix} \mathsf{sk}_{1,0} & \mathsf{sk}_{2,0} & \dots & \mathsf{sk}_{\ell,0} & \mathsf{sk}_{\text{in}} \\ \mathsf{sk}_{1,1} & \mathsf{sk}_{2,1} & \dots & \mathsf{sk}_{\ell,1} & \mathsf{sk}_{\text{out}} \end{pmatrix}$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ind}, m)$: For $\mathsf{ind} \in \{0,1\}^\ell$, choose a uniformly random $s \xleftarrow{\$} \mathcal{S}$ and encode it under the public keys specified by the index bits:

$$\psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_{i,\mathsf{ind}_i}, s) \text{ for all } i \in [\ell]$$

Encode s under the input public key:

$$\psi_{\text{in}} \leftarrow \mathsf{Encode}(\mathsf{pk}_{\text{in}}, s)$$

Encrypt the message m:

$$\tau \leftarrow \mathsf{E}(\mathsf{Encode}(\mathsf{pk}_{\text{out}}, s), m)$$

Output the ciphertext

$$\mathsf{ct}_{\mathsf{ind}} := \begin{pmatrix} \psi_1, & \psi_2, & \dots, & \psi_\ell, & \psi_{\text{in}}, & \tau \end{pmatrix}$$

$\mathsf{KeyGen}(\mathsf{msk}, C)$:

1. For every non-input wire $w = \ell+1, \dots, |C|$ of the circuit C, and every $b \in \{0,1\}$, generate public/secret key pairs:
$$(\mathsf{pk}_{w,b}, \mathsf{sk}_{w,b}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$$
2. For the gate $g = (u, v, w)$ with output wire w, compute the four recoding keys $\mathsf{rk}^w_{b,c}$ (for $b, c \in \{0,1\}$):
$$\mathsf{rk}^w_{b,c} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{u,b}, \mathsf{pk}_{v,c}, \mathsf{sk}_{u,b}, \mathsf{pk}_{w,g_w(b,c)}\Big)$$
3. Also, compute the recoding key
$$\mathsf{rk}^{\text{out}} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{|C|,1}, \mathsf{pk}_{\text{in}}, \mathsf{sk}_{|C|,1}, \mathsf{pk}_{\text{out}}\Big)$$

Output the evaluation key which is a collection of $4(|C|-\ell)$ recoding keys

$$\mathsf{ek}_C := \left(\mathsf{rk}^{w}_{b,c} \ : \ w \in \left[\ell+1, |C| \right], b, c \in \{0,1\} \right)$$

and the decryption key $\mathsf{dk} := \mathsf{rk}^{\text{out}}$.

$\mathsf{Eval}(\mathsf{ek}_C, \mathsf{ct}_{\mathsf{ind}})$: We tacitly assume that $\mathsf{ct}_{\mathsf{ind}}$ contains the index ind. For $w = \ell+1, \ldots, |C|$, let $g = (u, v, w)$ denote the gate with output wire w. Suppose wires u and v carry the values b^* and c^*, so that wire w carries the value $d^* := g_w(b^*, c^*)$. Compute

$$\psi_{w,d^*} \leftarrow \mathsf{Recode}\left(\mathsf{rk}^{w}_{b^*,c^*}, \psi_{u,b^*}, \psi_{v,c^*}\right)$$

If $C(\mathsf{ind}) = 1$, then we would have computed $\psi_{|C|,1}$. Output

$$\mathsf{ct}'_{\mathsf{ind}} := (\psi_{|C|,1}, \psi_{\text{in}}, \tau)$$

If $C(\mathsf{ind}) = 0$, output $\perp$.

$\mathsf{Dec}(\mathsf{dk}, \mathsf{ct}'_{\mathsf{ind}})$: Apply the transformation

$$\psi_{\text{out}} \leftarrow \mathsf{Recode}\left(\mathsf{rk}^{\text{out}}, \psi_{\text{in}}, \psi_{|C|,1}\right)$$

and output the message

$$m \leftarrow \mathsf{D}\left(\psi_{\text{out}}, \tau\right)$$

Security. We informally state how to modify the simulator in the proof of security in Section-6.4. The simulator gets $\{\mathsf{pk}_i, \psi_i\}_{i\in[\ell+2]}$ from the "outside". It assigns $\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell$ as the public keys specified by the bits of ind and $\mathsf{pk}_{\text{in}} := \mathsf{pk}_{\ell+1}, \mathsf{pk}_{\text{out}} := \mathsf{pk}_{\ell+2}$. It is easy to see how to simulate the ciphertext: all the input encodings become a part of it, as well as an encryption of the message using $\psi_{\text{out}} := \psi_{\ell+2}$. Now, the evaluation key ek is simulated by applying the TOR simulator.

- For query C such that $C(\mathsf{ind}) = 0$, the simulator can choose $(\mathsf{pk}_{|C|,1}, \mathsf{sk}_{|C|,1})$ by itself (the public key $\mathsf{pk}_{|C|,0}$ is "fixed" by the TOR simulator). Hence, the decryption key dk can be computed using $\mathsf{sk}_{|C|,1}$.
- On the other hand, for query C such that $C(\mathsf{ind}) = 1$, the adversary is not allowed to obtain the decryption key dk, hence there is not need to simulate it.

8.2 Extending Secret Keys

Consider the following problem: a users holds two (or more) secret keys sk_{C_1} and sk_{C_2}. C_1 allows to decrypt all ciphertexts addressed to *human resources* department and C_2 allows to decrypt ciphertexts addressed to *share holders*. The user wishes to create (and delegate) another secret key sk_{C^*} that allows to decrypt ciphertexts addressed to *human resources* and *share holders*. The question that we study is whether it is possible to allow the user to compute sk_{C^*} without calling the authority holding the master secret key msk.[3] More formally, given $\{\mathsf{sk}_{C_i}\}_{i\in[q]}$ a users should be able to compute a secret key

[3] In a subsequent work, Boneh et al. [34] showed how to construct a more general notion of delegatable ABE. In their scheme, given *only the secret key for* C_1, users can delegate a secret key for C_1 AND C_2 for any circuit C_2.

sk_{C^*} for any circuit C^* that is an black-box monotone composition of C_i's. Note that only monotone compositions are realizable, since otherwise a users holding a secret keys sk_{C_1} where $C_1(\mathsf{ind}) = 0$ could come up with a secret key for $\overline{C_1}$ and hence break any notion of security.

To suppose monotone extensions, it is enough to show how to obtain (1) $\mathsf{sk}_{C_1 \text{ AND } C_2}$ given $\mathsf{sk}_{C_1}, \mathsf{sk}_{C_2}$, and (2) $\mathsf{sk}_{C_1 \text{ OR } C_2}$ given $\mathsf{sk}_{C_1}, \mathsf{sk}_{C_2}$. We start from the construction presented in Section-8.1. We note that the security of that construction does not break if we give the secret key associated with the output value 0 ($\mathsf{sk}_{|C_i|,1}$) as a part of the secret key sk_{C_i}. This is because our simulation proceeds by sampling $(\mathsf{pk}_{|C_i|,1}, \mathsf{sk}_{|C_i|,1})$ honestly using Keygen algorithm and the fact the adversary is restricted to quires C_i such that $C_i(\mathsf{ind}) = 0$. Hence, given $\mathsf{sk}_{|C_1|,1}$ and $\mathsf{sk}_{|C_2|,1}$, let $C^* = C_1 \text{ AND } C_2$. The user computes sk_{C^*} as $(\mathsf{ek}_{C_1}, \mathsf{ek}_{C_2})$ plus four recoding keys $\mathsf{rk}^{C^*}_{b,c}$ (for $b, c \in \{0,1\}$):

$$
\begin{aligned}
&(\mathsf{pk}_{|C^*|,0}, \mathsf{rk}^{C^*}_{0,0}) \leftarrow \mathsf{SimReKeyGen}(\mathsf{pk}_{|C_1|,0}, \mathsf{pk}_{|C_2|,0}) \\
&\mathsf{rk}^{C^*}_{0,1} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{|C_1|,0}, \mathsf{pk}_{|C_2|,1}, \mathsf{sk}_{|C_2|,1}, \mathsf{pk}_{|C^*|,0}\Big) \\
&\mathsf{rk}^{C^*}_{1,0} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{|C_1|,1}, \mathsf{pk}_{|C_2|,0}, \mathsf{sk}_{|C_1|,1}, \mathsf{pk}_{|C^*|,0}\Big) \\
&\mathsf{rk}^{C^*}_{1,1} \leftarrow \mathsf{ReKeyGen}\Big(\mathsf{pk}_{|C_1|,1}, \mathsf{pk}_{|C_2|,1}, \mathsf{sk}_{|C_1|,1}, \mathsf{pk}_{\text{out}}\Big)
\end{aligned}
$$

As before, the message is encrypted under the encoding $\psi_{\text{out}} \leftarrow \mathsf{Encode}(\mathsf{pk}_{\text{out}}, s)$. The construction extends similarly to support OR compositions. Furthermore, arbitrary monotone structures can be realized by sampling keys associated with value 1 $(\mathsf{pk}_1, \mathsf{sk}_1)$ honestly and computing the recoding keys as above, until the final wire is assigned to pk_{out}.

Part IV

Predicate Encryption for Circuits from LWE

Sergey Gorbunov and Vinod Vaikuntanathan and Hoeteck Wee
CRYPTO 2015, invited to SPECIAL ISSUE

Abstract. In predicate encryption, a ciphertext is associated with descriptive attribute values x in addition to a plaintext μ, and a secret key is associated with a predicate f. Decryption returns plaintext μ if and only if $f(x) = 1$. Moreover, security of predicate encryption guarantees that an adversary learns nothing about the attribute x or the plaintext μ from a ciphertext, given arbitrary many secret keys that are not authorized to decrypt the ciphertext individually.

We construct a leveled predicate encryption scheme for all circuits, assuming the hardness of the subexponential learning with errors (LWE) problem. That is, for any polynomial function $d = d(\lambda)$, we construct a predicate encryption scheme for the class of all circuits with depth bounded by $d(\lambda)$, where λ is the security parameter.

1 Introduction

Predicate encryption [30, 143, 106] is a new paradigm for public-key encryption that supports searching on encrypted data. In predicate encryption, ciphertexts are associated with descriptive attribute values x in addition to plaintexts μ, secret keys are associated with a predicate f, and a secret key decrypts the ciphertext to recover μ if and only if $f(x) = 1$. The security requirement for predicate encryption enforces privacy of x and the plaintext even amidst multiple secret key queries: an adversary holding secret keys for different query predicates learns nothing about the attribute x and the plaintext (apart from the fact that x does not satisfy any of the query predicates) if none of them is individually authorized to decrypt the ciphertext.

Motivating applications. We begin with several motivating applications for predicate encryption [30, 143]:

- For inspecting recorded log files for network intrusions, we would encrypt network flows labeled with a set of attributes from the network header, such as the source and destination addresses, port numbers, time-stamp, and protocol numbers. We could then issue auditors with restricted secret keys that can only decrypt the network flows that fall within a particular range of IP addresses and some specific time period.

- For credit card fraud investigation, we would encrypt credit card transactions labeled with a set of attributes such as time, costs and zipcodes. We could then issue investigators with restricted secret keys that decrypt transactions over $1,000 which took place in the last month and originated from a particular range of zipcodes.

- For anti-terrorism investigation, we would encrypt travel records labeled with a set of attributes such as travel destination and basic traveller data. We could then issue investigators with restricted secret keys that match certain suspicious travel patterns.
- For online dating, we would encrypt personal profiles labeled with dating preferences pertaining to age, height, weight, salary and hobbies. Secret keys are associated with specific attributes and can only decrypt profiles for which the attributes match the dating preferences.

In all of these examples, it is important that unauthorized parties do not learn the contents of the ciphertexts, nor of the meta-data associated with the ciphertexts, such as the network header or dating preferences. On the other hand, it is often okay to leak the meta-data to authorized parties. We stress that privacy of the meta-data is an additional security requirement provided by predicate encryption but not by the related and weaker notion of attribute-based encryption (ABE) [140, 89]; the latter only guarantees the privacy of the plaintext μ and not the attribute x.

Utility and expressiveness. The utility of predicate encryption is intimately related to the class of predicates for which we could create secret keys. Ideally, we would like to support the class of all circuits. Over the past decade, substantial advances were made for the weaker primitive of ABE, culminating most recently in schemes supporting any policy computable by general circuits [86, 34] under the standard LWE assumption [136]. However, the state-of-the-art for predicate encryption is largely limited to very simple functionalities related to computing an inner product [30, 143, 106, 9, 72].

1.1 Our Contributions

In this work, we substantially advance the state of the art to obtain predicate encryption for all circuits (c.f. Figure 6):

> **Theorem (informal).** Under the LWE assumption, there exists a predicate encryption scheme for all circuits, with succint ciphertexts and secret keys independent of the size of the circuit.

As with prior LWE-based ABE for circuits [86, 34], to support circuits of depth d, the parameters of the scheme grow with poly(d), and we require sub-exponential $n^{\Omega(d)}$ hardness of the LWE assumption. In addition, the security guarantee is selective, but can be extended to adaptive security via complexity leveraging [27].

Privacy guarantees. The privacy notion we achieve is a *simulation-based* variant of "attribute-hiding" from the literature [143, 128, 9]. That is, we guarantee privacy of the attribute x and the plaintext μ against collusions holding secret keys for functions f such that $f(x) = 0$. An even stronger requirement would be to require privacy of x even against authorized keys corresponding to functions f where $f(x) = 1$; in the literature, this stronger notion is referred to as "full attribute-hiding" [30, 106]. This stronger requirement is equivalent to "full-fledged" functional encryption [33], for which we cannot hope to achieve simulation-based security for all circuits as achieved in this work [33, 11].

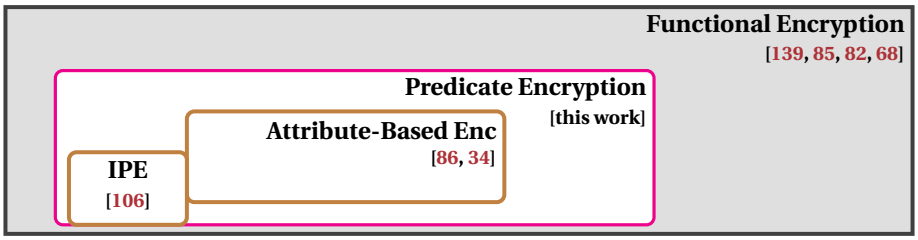


Figure 6: State of the art in functional encryption. The white region refers to functionalities for which we have constructions under standard cryptographic assumptions like LWE or decisional problems in bilinear groups: these functionalities include inner product encryption (IPE), attribute-based encryption for general circuits (ABE) and predicate encryption for general circuits. The grey region refers to functionalities beyond predicate encryption for which we only have constructions for weaker security notions like bounded collusions, or under non-standard cryptographic assumptions like obfuscation or multilinear maps.

Relation to prior works. Our result subsumes all prior works on predicate encryption under standard cryptographic assumptions, apart from a few exceptions pertaining to the inner product predicate [30, 106, 130]. These results achieve a stronger security notion for predicate encryption, known as full (or strong) security (please refer to Section 3.1, and the full version for definitions).

In a recent break-through work, Garg et al. [68] gave a beautiful candidate construction of functional encryption (more general primitive than predicate encryption) for arbitrary circuits. However, the construction relies on "multi-linear maps" [67, 57, 79], for which we have few candidates and which rely on complex intractability assumptions that are presently poorly understood and not extensively studied in the literature. It remains an intriguing open problem to construct a functional encryption scheme from a standard assumption, such as LWE.

In contrast, if we consider functional encryption with *a-priori bounded collusions size* (that is, the number of secret keys any collusion of adversaries may obtain is fixed by the scheme at the setup phase), then it is possible to obtain functional encryption for general circuits under a large class of standard assumptions [139, 85, 82]. This notion is *weaker* than standard notion of functional encryption, yet remains very meaningful for many applications.

1.2 Overview of Our Construction

Our starting point is the work of Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [82] who show how to convert an attribute-based encryption (ABE) scheme into a *single key secure* functional encryption (FE) scheme. Recall that in an attribute-based encryption scheme [89], a ciphertext is associated with a descriptive value (a public "attribute") x and plaintext μ, and it hides μ, but not x. The observation of Goldwasser et al. [82] is to hide x by encrypting it using a fully homomorphic encryption (FHE) scheme [76, 37], and then using the resulting FHE ciphertext as the public "attribute" in an ABE scheme for general circuits [86, 34]. This has the dual benefit of guaranteeing privacy of x, while at the same time allowing homomorphic computation of predicates f on the encryption of x.

This initial idea quickly runs into trouble. The decryptor who is given the predicate secret key for f and a predicate encryption of (x,μ) can indeed compute an *FHE encryption* of $f(x)$. However, the

	Interface	Security Guarantee given sk_f
ABE	$\mathsf{Enc}(x,\mu)$	μ is secret iff $f(x)=0$ x is always public
PE	$\mathsf{Enc}(x,\mu)$	(x,μ) is secret iff $f(x)=0$
FE	$\mathsf{Enc}(x)$	user learns only $f(x)$

Figure 7: Comparison of the security guarantees provided by attribute-based (ABE), predicate (PE) and functional encryption (FE), where secret keys are associated with a Boolean function f; the main distinction lies in how much information about x is potentially leaked to the adversary. The main distinction between ABE and PE is that x is always public in ABE, but remains secret in PE when the user is not authorized to decrypt. The main distinction between PE and FE is that x always remains hidden (even when $f(x)=1$) and hence the user only learns the output of the computation of f on x.

decryption process is confronted with a decision, namely whether to release the message μ or not, and this decision depends on whether the *plaintext* $f(x)$ is 0 or 1.[4] Clearly, resolving this conundrum requires obtaining $f(x)$, which requires knowledge of the FHE secret key. Goldwasser et al. [82] solved this by employing a (single use) Yao garbling of the FHE decryption circuit, however this limited them to obtaining *single key secure* predicate/functional encryption schemes.[5]

Our first key idea is to embed the FHE secret key as part of the attributes in the ABE ciphertext. That is, in order to encrypt a plaintext μ with attributes x in the predicate encryption scheme, we first choose a symmetric key $\mathsf{fhe.sk}$ for the FHE scheme, encrypt x into a FHE ciphertext $\hat{x}$, and encrypt μ using the ABE scheme with $(\mathsf{fhe.sk},\hat{x})$ as the attributes to obtain an ABE ciphertext ct. Our predicate encryption ciphertext is then given by

$$(\hat{x},\mathsf{ct})$$

To generate the predicate secret key for a function f, one simply generates the ABE secret key for the function g that takes as input $(\mathsf{fhe.sk},\hat{x})$ and computes

$$g(\mathsf{fhe.sk},\hat{x}) = \mathsf{FHE.Dec}(\mathsf{fhe.sk};\mathsf{FHE.Eval}(f,\hat{x}))$$

That is, g first homomorphically computes a FHE encryption of $f(x)$, and then decrypts it using the FHE secret key to output $f(x)$.

At first glance, this idea evokes strong and conflicting emotions as it raises two problems. The first pertains to correctness: we can no longer decrypt the ciphertext since the ABE decryption algorithm needs to know all of the attributes ($\hat{x}$ and $\mathsf{fhe.sk}$), but $\mathsf{fhe.sk}$ is missing. The second pertains to security: the ABE ciphertext ct is not guaranteed to protect the privacy of the attributes, and could leak all of $\mathsf{fhe.sk}$ which together with $\hat{x}$ would leak all of x. Solving both of these problems seems to require designing a

[4] In fact, there is a syntactic mismatch since $\hat{f}(\cdot)$ is not a predicate, as it outputs an FHE ciphertext.

[5] A reader familiar with [82] might wonder whether replacing single-use garbled circuits in their construction with reusable garbled circuits (also from [82]) might remove this limitation. We remark that this does not seem possible, essentially because the construction in [82] relies crucially on the simplicity of computing garbled inputs from the "garbling key". In particular, in Yao's garbled circuit scheme, the garbling key is (many) pairs of "strings" L_0 and L_1, and a garbling of an input bit b is simply L_b. This fits perfectly with the semantics of ABE (rather, a variant termed two-input ABE in [82]) that releases one of two possible "messages" L_0 or L_1 depending on the outcome of a computation. In contrast, computing a garbled input in the reusable garbling scheme is a more complex and randomized function of the garbling key, and does not seem to align well with the semantics of ABE.

predicate encryption scheme from scratch!

Our next key observation is that the bulk of the computation in g, namely the homomorphic evaluation of the function f, is performed on the *public* attribute $\hat{x}$. The only computation performed on the secret value fhe.sk is FHE decryption which is a fairly lightweight computation. In particular, with all known FHE schemes [76, 37, 38, 40, 78, 39, 14], decryption corresponds to computing an inner product followed by a threshold function. Furthermore, we do know how to construct lattice-based predicate encryption schemes for threshold of inner product [9, 72]. We stress that the latter do not correspond to FHE decryption since the inner product is computed over a vector in the ciphertext and one in the key, whereas FHE decryption requires computing an inner product over two vectors in the ciphertext; nonetheless, we will build upon the proof techniques in achieving attribute-hiding in [9, 72] in the proof of security.

In other words, if we could enhance ABE with a modicum of secrecy so that it can perform a heavy-weight computation on public attributes followed by a lightweight privacy-preserving computation on *secret* attributes, we are back in business. Our first contribution is to define such an object, that we call *partially hiding predicate encryption*.

Partially Hiding Predicate Encryption. We introduce the notion of partially hiding predicate encryption (PHPE), an object that interpolates between attribute-based encryption and predicate encryption (analogously to partial garbling in [101]). In PHPE, the ciphertext, encrypting message μ, is associated with an attribute (x, y) where x is private but y is always public. The secret key is associated with a function f, and decryption succeeds iff $f(x, y) = 1$. On the one extreme, considering a dummy x or functions f that ignore x and compute on y, we recover attribute-based encryption. On the other end, considering a dummy y or functions f that ignore y and compute on x, we recover predicate encryption.

We will be interested in realizing PHPE for functions ϕ of the form $\phi(x, y) = g(x, h(y))$ for some functions g and h where h may perform arbitrary heavy-weight computation on the public y and g only performs light-weight computation on the private x. Mapping back to our discussion, we would like to achieve PHPE for the "evaluate-then-decrypt" class of functions, namely where g is the FHE decryption function, h is the FHE evaluation function, x is the FHE secret key, and y is the FHE ciphertext. In general, we would like g to be simple and will allow h to be complex. It turns out that we can formalize the observation above, namely that PHPE for this class of functions gives us a predicate encryption scheme. The question now becomes: can we construct PHPE schemes for the "evaluate-then-decrypt" class of functions?

Assuming the subexponential hardness of learning with errors (LWE), we show how to construct a partially hiding predicate encryption for the class of functions $f : \mathbb{Z}_q^t \times \{0,1\}^\ell \to \{0,1\}$ of the form

$$f_\gamma(\mathbf{x}, \mathbf{y}) = \mathsf{IP}_\gamma(\mathbf{x}, h(\mathbf{y})),$$

where $h : \{0,1\}^\ell \to \{0,1\}^t$, $\gamma \in \mathbb{Z}_q$, and $\mathsf{IP}_\gamma(\mathbf{x}, \mathbf{z}) = 1$ iff $\langle \mathbf{x}, \mathbf{z} \rangle = \left(\sum_{i \in [t]} \mathbf{x}[i] \cdot \mathbf{z}[i] \right) = \gamma \mod q$.

This is almost what we want, but not quite. Recall that FHE decryption in many recent schemes [37, 40, 78, 39, 14] is a function that checks whether an inner product of two vectors in $\mathbb{Z}_q^t$ (one of which could be over $\{0,1\}^t$) lies in a certain range. Indeed, if $\mathbf{z} \in \{0,1\}^t$ is an encryption of 1 and $\mathbf{x} \in \mathbb{Z}_q^t$ is the secret key, we know that $\langle \mathbf{x}, \mathbf{z} \rangle \in [q/2 - B, q/2 + B] \pmod q$, where B is the noise range. Applying the

so-called "modulus reduction" [37] transformation to all these schemes, we can assume that this range is polynomial in size.

In other words, we will manage to construct a partially hiding PE scheme for the function

$$f_\gamma(\mathbf{x},\mathbf{y}) : \langle \mathbf{x}, h(\mathbf{y}) \rangle \stackrel{?}{=} \gamma \pmod{q}$$

whereas we need a partially hiding PE scheme for the FHE decryption function which is

$$f'_R(\mathbf{x},\mathbf{y}) : \langle \mathbf{x}, h(\mathbf{y}) \rangle \stackrel{?}{\in} R \pmod{q}$$

where R is the polynomial size range $[q/2 - B, q/2 + B]$ from above. How do we reconcile this disparity?

The "Lazy OR" Trick. The solution, called the "lazy OR trick" [143, 72] is to publish secret keys for all functions f_γ for $\gamma \in R := [q/2 - B, q/2 + B]$. This will indeed allow us to test if the FHE decryption of the evaluated ciphertext is 1 (and reveal the message μ if it is), but it is also worrying. Publishing these predicate secret keys for the predicates f_γ reveals more information than whether $\langle \mathbf{x}, h(\mathbf{y}) \rangle \stackrel{?}{\in} R$. In particular, it reveals what $\langle \mathbf{x}, h(\mathbf{y}) \rangle$ is. This means that an authorized key would leak partial information about the attribute, which we do allow for predicate encryption. On the other hand, for an unauthorized key where the FHE decryption is 0, each of these $f_\gamma, \gamma \in R$ is also an unauthorized key in the PHPE and therefore leaks no information about the attribute. This extends to the collection of keys in R since the PHPE is secure against collusions. For simplicity, we assume in the rest of this overview that FHE decryption corresponds to exactly to inner product.

Asymmetry to the Rescue: Constructing Partially Hiding PE. Our final contribution is the construction of a partially hiding PE for the function class $f_\gamma(\mathbf{x},\mathbf{y})$ above. We will crucially exploit the fact that f_γ computes an inner product on the private attribute $\mathbf{x}$. There are two challenges here: first, we need to design a decryption algorithm that knows f_γ and $\mathbf{y}$ but not $\mathbf{x}$ (this is different from decryption in ABE where the algorithm also knows $\mathbf{x}$); second, show that the ciphertext does not leak too much information about $\mathbf{x}$. We use the fully key-homomorphic encryption techniques developed by Boneh et al [34] in the context of constructing an "arithmetic" ABE scheme. The crucial observation about the ABE scheme of [34] is that while it was not designed to hide the attributes, it can be made to partially hide them in exactly the way we want. In particular, the scheme allows us to carry out an inner product of a public attribute vector (corresponding to the evaluated FHE ciphertext) and a private attribute vector (corresponding to the FHE secret key $\mathsf{fhe.sk}$), thanks to an inherent asymmetry in homomorphic evaluation of a multiplication gate on ABE ciphertexts. More concretely, in the homomorphic evaluation of a ciphertext for a multiplication gate in [34], the decryption algorithm works even if one of the attribute remains private, and for addition gates, the decryption algorithms works even if both attributes remain private. This addresses the first challenge of a decryption algorithm that is oblivious to $\mathbf{x}$. For the second challenge of security, we rely on techniques from inner product predicate encryption [9] to prove the privacy of $\mathbf{x}$ Note that in the latter, the inner product is computed over a vector in the ciphertext and one in the key, whereas in our scheme, the inner product is computed over two vectors in the ciphertext. Interestingly, the proof still goes through since the ciphertext in the ABE [34] has the same structure as the ciphertext in [9]. We refer the reader to Section 3.2 for a detailed overview of the partial hiding PE,

and to Section 4 for an overview of how we combine the partial hiding PE with FHE to obtain our main result.

Finally, we remark that exploiting asymmetry in multiplication has been used in fairly different contexts in both FHE [78, 39] and in ABE [86, 84]. In [78] and in this work, the use of asymmetry was crucial for realizing the underlying cryptographic primitive; whereas in [86, 39, 84], asymmetry was used to reduce the noise growth during homomorphic evaluation, thereby leading to quantitative improvements in the underlying assumptions and hence improved efficiency.

1.3 Discussion

Comparison with other approaches. The two main alternative approaches for realizing predicate and functional encryption both rely on multi-linear maps either implicitly, or explicitly. The first is to use indistinguishability obfuscation as in [68], and the second is to extend the dual system encryption framework to multi-linear maps [146, 70]. A crucial theoretical limitation of these approaches is that they all rely on non-standard assumptions; we have few candidates for multi-linear maps [67, 57, 79] and the corresponding assumptions are presently poorly understood and not extensively studied in cryptanalysis, and in some cases, broken [54]. In particular, the latest attack in [54] highlight the importance of obtaining constructions and developing techniques that work under standard cryptographic assumptions, as is the focus of this work.

Barriers to functional encryption from LWE. We note the two main barriers to achieving full-fledged functional encryption from LWE using our framework. First, the lazy conjunction approach to handle threshold inner product for FHE decryption leaks the exact inner product and therefore cannot be used to achieve full attribute-hiding. Second, we do not currently know of a fully attribute-hiding inner product encryption scheme under the LWE assumption, although we do know how to obtain such schemes under standard assumptions in bilinear groups [130, 106].

2 Preliminaries

We refer the reader to the full version for the background on lattices.

2.1 Fully-Homomorphic Encryption

We present a fairly minimal definition of fully homomorphic encryption (FHE) which is sufficient for our constructions. A leveled homomorphic encryption scheme is a tuple of polynomial-time algorithms $(\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Eval}, \mathsf{HE.Dec})$:

- **Key generation.** $\mathsf{HE.KeyGen}(1^\lambda, 1^d, 1^k)$ is a probablistic algorithm that takes as input the security parameter λ, a depth bound d and message length k and outputs a secret key sk.
- **Encryption.** $\mathsf{HE.Enc}(\mathsf{sk}, \mu)$ is a probabilistic algorithm that takes as input sk and a message $\mu \in \{0,1\}^k$ and outputs a ciphertext ct.

- **Homomorphic evaluation.** $\mathsf{HE.Eval}(f, \mathsf{ct})$ is a deterministic algorithm that takes as input a boolean circuit $C : \{0,1\}^k \to \{0,1\}$ of depth at most d and a ciphertext ct and outputs another ciphertext ct'.
- **Decryption.** $\mathsf{HE.Dec}(\mathsf{sk}, \mathsf{ct}')$ is a deterministic algorithm that takes as input sk and ciphertext ct' and outputs a bit.

Correctness. We require perfect decryption correctness with respect to homomorphically evaluated ciphertexts: namely for all λ, d, k and all $\mathsf{sk} \leftarrow \mathsf{HE.KeyGen}(1^\lambda, 1^d, 1^k)$, all $\mu \in \{0,1\}^k$ and for all boolean circuits $C : \{0,1\}^k \to \{0,1\}$ of depth at most d:

$$\Pr\Big[\mathsf{HE.Dec}(\mathsf{sk}, \mathsf{HE.Eval}(C, \mathsf{HE.Enc}(\mathsf{sk}, \mu))) = C(\mu)\Big] = 1$$

where the probablity is taken over $\mathsf{HE.Enc}$ and $\mathsf{HE.KeyGen}$.

Security. We require semantic security for a single ciphertext: namely for every stateful p.p.t. adversary $\mathcal{A}$ and for all $d, k = \mathrm{poly}(\lambda)$, the following quantity

$$\Pr\left[b = b' : \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Setup}(1^\lambda, 1^d, 1^k); \\ (\mu_0, \mu_1) \leftarrow \mathcal{A}(1^\lambda, 1^d, 1^k); \\ b \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, \mu_b); \\ b' \leftarrow \mathcal{A}(\mathsf{ct}) \end{array}\right] - \frac{1}{2}$$

is negligible in λ.

2.1.1 FHE from LWE

We will rely on an instantiation of FHE from the LWE assumption:

Theorem 2.1 (FHE from LWE [37, 40, 78, 39, 14])**.** *There is a FHE scheme* $\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Eval}, \mathsf{HE.Dec}$ *that works for any* q *with* $q \geq O(\lambda^2)$ *with the following properties:*

- $\mathsf{HE.KeyGen}$ *outputs a secret key* $\mathsf{sk} \in \mathbb{Z}_q^t$ *where* $t = \mathrm{poly}(\lambda)$*;*
- $\mathsf{HE.Enc}$ *outputs a ciphertext* $\mathsf{ct} \in \{0,1\}^\ell$ *where* $\ell = \mathrm{poly}(k, d, \lambda, \log q)$*;*
- $\mathsf{HE.Eval}$ *outputs a ciphertext* $\mathsf{ct}' \in \{0,1\}^t$*;*
- *for any boolean circuit of depth* d*,* $\mathsf{HE.Eval}(C, \cdot)$ *is computed by a boolean circuit of depth* $\mathrm{poly}(d, \lambda, \log q)$*.*
- $\mathsf{HE.Dec}$ *on input* $\mathsf{sk}, \mathsf{ct}'$ *outputs a bit* $b \in \{0,1\}$*. If* ct' *is an encryption of* 1 *then*

$$\sum_{i=1}^{t} \mathsf{sk}[i] \cdot \mathsf{ct}'[i] \in [\lfloor q/2 \rfloor - B, \lfloor q/2 \rfloor + B]$$

for some fixed $B = \mathrm{poly}(\lambda)$. *Otherwise, if* ct' *is an encryption of* 0, *then*

$$\sum_{i=1}^{t} \mathsf{sk}[i] \cdot \mathsf{ct}'[i] \notin [\lfloor q/2 \rfloor - B, \lfloor q/2 \rfloor + B];$$

- *security relies on* $\mathsf{dLWE}_{\Theta(t),q,\chi}$.

We highlight several properties of the above scheme: (1) the ciphertext is a bit-string, (2) the bound B is a polynomial independent of q (here, we crucially exploit the new results in [39] together with the use of leveled bootstrapping)[6], (3) the size of normal ciphertexts is independent of the size of the circuit (this is the typical compactness requirement).

3 Partially Hiding Predicate Encryption

3.1 Definitions

We introduce the notation of partially hiding predicate encryption (PHPE), which interpolates attribute-based encryption and predicate encryption (analogously to partial garbling in [101]). In PHPE, the ciphertext, encrypting message μ, is associated with an attribute (x, y) where x is private but y is always public. The secret key is associated with a predicate C, and decryption succeeds iff $C(x, y) = 1$. The requirement is that a collusion learns nothing about (x, μ) if none of them is individually authorized to decrypt the ciphertext. Attribute-based encryption corresponds to the setting where x is empty, and predicate encryption corresponds to the setting where y is empty. We refer the reader to the full version for the standard notion of predicate encryption.

Looking ahead to our construction, we show how to:

- construct PHPE for a restricted class of circuits that is "low complexity" with respect to x and allows arbitrarily polynomial-time computation on y;
- bootstrap this PHPE using FHE to obtain PE for all circuits.

Syntax. A Partially-Hiding Predicate Encryption scheme $\mathcal{PHPE}$ for a pair of input-universes $\mathcal{X}, \mathcal{Y}$, a predicate universe $\mathcal{C}$, a message space $\mathcal{M}$, consists of four algorithms $(\mathsf{PH.Setup}, \mathsf{PH.Enc}, \mathsf{PH.Keygen}, \mathsf{PH.Dec})$:

$\mathsf{PH.Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M}) \to (\mathsf{ph.mpk}, \mathsf{ph.msk})$. The setup algorithm gets as input the security parameter λ and a description of $(\mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M})$ and outputs the public parameter $\mathsf{ph.mpk}$, and the master key $\mathsf{ph.msk}$.

$\mathsf{PH.Enc}(\mathsf{ph.mpk}, (x, y), \mu) \to \mathsf{ct}_y$. The encryption algorithm gets as input $\mathsf{ph.mpk}$, an attribute $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a message $\mu \in \mathcal{M}$. It outputs a ciphertext ct_y.

$\mathsf{PH.Keygen}(\mathsf{ph.msk}, C) \to \mathsf{sk}_C$. The key generation algorithm gets as input $\mathsf{ph.msk}$ and a predicate $C \in \mathcal{C}$. It outputs a secret key sk_C.

[6] Recall that no circular security assumption needs to be made for leveled bootstrapping.

$\mathsf{PH.Dec}((\mathsf{sk}_C, C), (\mathsf{ct}_y, y)) \to \mu$. The decryption algorithm gets as input the secret key sk_C, a predicate C, and a ciphertext ct_y and the public part of the attribute y. It outputs a message $\mu \in \mathcal{M}$ or $\perp$.

Correctness. We require that for all $\mathsf{PH.Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M}) \to (\mathsf{ph.mpk}, \mathsf{ph.msk})$, for all $(x, y, C) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{C}$, for all $\mu \in \mathcal{M}$,

- if $C(x, y) = 1$, $\Pr\left[\mathsf{PH.Dec}((\mathsf{sk}_C, C), (\mathsf{ct}_y, y)) = \mu\right] \geq 1 - \mathrm{negl}(\lambda)$,
- if $C(x, y) = 0$, $\Pr\left[\mathsf{PH.Dec}((\mathsf{sk}_C, C), (\mathsf{ct}_y, y)) = \perp\right] \geq 1 - \mathrm{negl}(\lambda)$,

where the probabilities are taken over $\mathsf{sk}_C \leftarrow \mathsf{PH.Keygen}(\mathsf{ph.msk}, C)$, $\mathsf{ct}_y \leftarrow \mathsf{PH.Enc}(\mathsf{ph.mpk}, (x, y), \mu)$ and coins of PH.Setup.

Definition 3.1 (PHPE Attribute-Hiding)**.** *Fix* (PH.Setup, PH.Enc, PH.Keygen, PH.Dec)*. For every stateful p.p.t. adversary* Adv*, and a p.p.t. simulator* Sim*, consider the following two experiments:*

$\underline{\exp^{\mathsf{real}}_{\mathcal{PHPE},\mathrm{Adv}}(1^\lambda)}$**:**	$\underline{\exp^{\mathsf{ideal}}_{\mathcal{PHPE},\mathrm{Sim}}(1^\lambda)}$**:**
1: $(x, y) \leftarrow \mathrm{Adv}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M})$	1: $(x, y) \leftarrow \mathrm{Adv}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M})$
2: $(\mathsf{ph.mpk}, \mathsf{ph.msk}) \leftarrow \mathsf{PH.Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M})$	2: $(\mathsf{ph.mpk}, \mathsf{ph.msk}) \leftarrow \mathsf{PH.Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{M})$
3: $\mu \leftarrow \mathrm{Adv}^{\mathsf{PH.Keygen}(\mathsf{msk},\cdot)}(\mathsf{ph.mpk})$	3: $\mu \leftarrow \mathrm{Adv}^{\mathsf{PH.Keygen}(\mathsf{ph.msk},\cdot)}(\mathsf{ph.mpk})$
4: $\mathsf{ct}_y \leftarrow \mathsf{PH.Enc}(\mathsf{ph.mpk}, (x, y), \mu)$	4: $\mathsf{ct}_y \leftarrow \mathrm{Sim}(\mathsf{mpk}, y, 1^{\lvert x\rvert}, 1^{\lvert\mu\rvert})$
5: $\alpha \leftarrow \mathrm{Adv}^{\mathsf{PH.Keygen}(\mathsf{ph.msk},\cdot)}(\mathsf{ct}_y)$	5: $\alpha \leftarrow \mathrm{Adv}^{\mathsf{PH.Keygen}(\mathsf{msk},\cdot)}(\mathsf{ct}_y)$
6: *Output* (x, y, μ, α)	6: *Output* (x, y, μ, α)

We say an adversary Adv *is admissible if all oracle queries that it makes $C \in \mathcal{C}$ satisfy $C(x, y) = 0$. The Partially-Hiding Predicate Encryption scheme $\mathcal{PHPE}$ is then said to be* attribute-hiding *if there is a p.p.t. simulator* Sim *such that for every stateful p.p.t. adversary* Adv*, the following two distributions are computationally indistinguishable:*

$$\left\{\exp^{\mathsf{real}}_{\mathcal{PHPE},\mathrm{Adv}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \left\{\exp^{\mathsf{ideal}}_{\mathcal{PHPE},\mathrm{Sim}}(1^\lambda)\right\}_{\lambda\in\mathbb{N}}$$

Remarks. We point out some remarks of our definition (SIM-AH) when treated as a regular predicate encryption (i.e. the setting where y is empty; see the full version for completeness) and how it compares to other definitions in the literature.

- We note the simulator for the challenge ciphertext gets y but not x; this captures the fact that y is public whereas x is private. In addition, the simulator is not allowed to program the public parameters or the secret keys. In the ideal experiment, the simulator does not explicitly learn any information about x (apart from its length); nonetheless, there is implicit leakage about x from the key queries made by an admissible adversary. Finally, we note that we can efficiently check whether an adversary is admissible.

- Our security notion is "selective", in that the adversary "commits" to (x, y) before it sees ph.mpk. It is possible to bootstrap selectively-secure scheme to full security using standard complexity leveraging arguments [27, 86], at the price of a $2^{|x|}$ loss in the security reduction.

- Our definition refers to a single challenge message, but the definition extends readily to a setting with multiple challenge messages. Moreover, our definition composes in that security for a single message implies security with multiple messages (see the full version). The following remarks refer to many messages setting.

- We distinguish between two notions of indistinguishability-based (IND) definitions used in the literature: attribute-hiding (IND-AH)[7] and strong attribute-hiding (IND-SAH)[8] [30, 143, 106, 9]. In the IND-AH, the adversary should not be able to distinguish between two pairs of attributes/messages given that it is restricted to queries which do not decrypt the challenge ciphertext (See the full version for details). It is easy to see that our SIM-AH definition is stronger than IND-AH. Furthermore, IND-SAH also ensures that adversary cannot distinguish between the attributes even when it is allowed to ask for queries that decrypt the messages (in this case, it must output $\mu_0 = \mu_1$). Our SIM-AH definition is weaker than IND-SAH, since we explicitly restrict the adversary to queries that do not decrypt the challenge ciphertext.

- In the context of arbitrary predicates, *strong* variants of definitions (that is, IND-SAH and SIM-SAH) are equivalent to security notions for functional encryption (the simulation definition must be adjusted to give the simulated the outputs of the queries). However, the strong variant of notion (SIM-SAH) is impossible to realize for many messages [33, 11]. We refer the reader to the full version for a sketch of the impossibility. Hence, SIM-AH is the best-possible simulation security for predicate encryption which we realize in this work. The only problem which we leave open is to realize IND-SAH from standard LWE.

3.2 Our Construction

We refer the reader to the full version for the complete description of our construction. Below, we provide an overview.

Overview. We construct a partially hiding predicate encryption for the class of predicate circuits $C : \mathbb{Z}_q^t \times \{0,1\}^\ell \to \{0,1\}$ of the form $\widehat{C} \circ \mathsf{IP}_\gamma$ where $\widehat{C} : \{0,1\}^\ell \to \{0,1\}^t$ is a boolean circuit of depth d, $\gamma \in \mathbb{Z}_q$, and

$$(\widehat{C} \circ \mathsf{IP}_\gamma)(\mathbf{x}, \mathbf{y}) = \mathsf{IP}_\gamma(\mathbf{x}, \widehat{C}(\mathbf{y})),$$

where $\mathsf{IP}_\gamma(\mathbf{x}, \mathbf{z}) = 1$ iff $\langle \mathbf{x}, \mathbf{z} \rangle = \left(\sum_{i \in [t]} \mathbf{x}[i] \cdot \mathbf{z}[i]\right) = \gamma \mod q$. We refer to circuit IP as the generic inner-product circuit of two vectors.

Looking ahead, $\widehat{C}$ corresponds to FHE evaluation of an arbitrary circuit C, whereas IP_γ corresponds to roughly to FHE decryption; in the language of the introduction in Section 1, $\widehat{C}$ corresponds to heavy-weight computation h, whereas IP_γ corresponds to light-weight computation g.

[7] Sometimes also referred as weak attribute-hiding.
[8] Sometimes also referred as full attribute-hiding.

The scheme. The public parameters are matrices

$$\left(\mathbf{A}, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{B}_1, \ldots, \mathbf{B}_t\right)$$

An encryption corresponding to the attribute $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_q^t \times \{0,1\}^\ell$ is a GPV ciphertext (an LWE sample) corresponding to the matrix

$$\left[\mathbf{A} \mid \mathbf{A}_1 + \mathbf{y}[1] \cdot \mathbf{G} \mid \cdots \mid \mathbf{A}_\ell + \mathbf{y}[\ell] \cdot \mathbf{G} \mid \mathbf{B}_1 + \mathbf{x}[1] \cdot \mathbf{G} \mid \cdots \mid \mathbf{B}_t + \mathbf{x}[t] \cdot \mathbf{G}\right]$$

To decrypt the ciphertext given $\mathbf{y}$ and a key for $\widehat{C} \circ \mathsf{IP}_\gamma$, we apply the BGGHNSVV algorithm to first transform the first part of the ciphertext into a GPV ciphertext corresponding to the matrix

$$\left[\mathbf{A} \mid \mathbf{A}_{\widehat{C}_1} + \mathbf{z}[1] \cdot \mathbf{G} \mid \cdots \mid \mathbf{A}_{\widehat{C}_t} + \mathbf{z}[t] \cdot \mathbf{G}\right]$$

where $\widehat{C}_i$ is the circuit computing the i'th bit of $\widehat{C}$ and $\mathbf{z} = \widehat{C}(\mathbf{y}) \in \{0,1\}^t$. Next, observe that

$$-\left((\mathbf{A}_{\widehat{C}_i} + \mathbf{z}[i] \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{B}_i)\right) + \mathbf{z}[i] \cdot \left(\mathbf{B}_i + \mathbf{x}[i] \cdot \mathbf{G}\right) = -\mathbf{A}_{\widehat{C}_i} \mathbf{G}^{-1}(\mathbf{B}_i) + \mathbf{x}[i] \cdot \mathbf{z}[i] \cdot \mathbf{G}.$$

Summing over i, we have

$$\sum_{i=1}^{\ell} -\left((\mathbf{A}_{\widehat{C}_i} + \mathbf{z}[i] \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{B}_i)\right) + \mathbf{z}[i] \cdot \left(\mathbf{B}_i + \mathbf{x}[i] \cdot \mathbf{G}\right) = \mathbf{A}_{\widehat{C} \circ \mathsf{IP}} + \langle \mathbf{x}, \mathbf{z} \rangle \cdot \mathbf{G}$$

where

$$\mathbf{A}_{\widehat{C} \circ \mathsf{IP}} := -\left(\mathbf{A}_{\widehat{C}_1} \mathbf{G}^{-1}(\mathbf{B}_1) + \cdots + \mathbf{A}_{\widehat{C}_t} \mathbf{G}^{-1}(\mathbf{B}_t)\right).$$

Therefore, given only the public matrices and $\mathbf{y}$ (but not $\mathbf{x}$), we may transform the ciphertext into a GPV ciphertext corresponding to the matrix

$$\left[\mathbf{A} \mid \mathbf{A}_{\widehat{C} \circ \mathsf{IP}} + \langle \mathbf{x}, \mathbf{z} \rangle \cdot \mathbf{G}\right].$$

The secret key corresponding to $\widehat{C} \circ \mathsf{IP}_\gamma$ is essentially a "short basis" for the matrix

$$\left[\mathbf{A} \mid \mathbf{A}_{\widehat{C} \circ \mathsf{IP}} + \gamma \cdot \mathbf{G}\right]$$

which can be sampled using a short trapdoor $\mathbf{T}$ of the matrix $\mathbf{A}$.

Proof strategy. There are two main components to the proof. Fix the selective challenge attribute $\mathbf{x}, \mathbf{y}$. First, we will simulate the secret keys without knowing the trapdoor for the matrix $\mathbf{A}$: here, we rely on the simulated key generation for the ABE [34]. Roughly speaking, we will need to generate a short basis for the matrix

$$\left[\mathbf{A} \mid \mathbf{A}\mathbf{R}_{\widehat{C} \circ \mathsf{IP}} + (\gamma - \widehat{C} \circ \mathsf{IP}(\mathbf{x}, \mathbf{y})) \cdot \mathbf{G}\right]$$

where $\mathbf{R}_{\widehat{C} \circ \mathsf{IP}}$ is a small-norm matrix known to the simulator. Now, whenever $\widehat{C} \circ \mathsf{IP}(\mathbf{x}, \mathbf{y}) \neq \gamma$ as is the case for admissible adversaries, we will be able to simulate secret keys using the puncturing techniques in [7, 9, 121]

Next, we will show that the attribute $\mathbf{x}$ is hidden in the challenge ciphertext. Here, we adopt the proof

strategy for attribute-hiding inner product encryption in [9, 72]. In the proof, we simulate the matrices $\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_t$ using

$$\mathbf{A}, \mathbf{A}\mathbf{R}'_1 - \mathbf{x}[1]\mathbf{G}, \ldots, \mathbf{A}\mathbf{R}'_t - \mathbf{x}[t]\mathbf{G}$$

where $\mathbf{R}'_1, \ldots, \mathbf{R}'_t \xleftarrow{\$} \{\pm 1\}^{m \times m}$. In addition, we simulate the corresponding terms in the challenge ciphertext by $\mathbf{c}, \mathbf{c}^\top \mathbf{R}'_1, \ldots, \mathbf{c}^\top \mathbf{R}'_t$, where $\mathbf{c}$ is a uniformly random vector, which we switched from $\mathbf{A}^\top \mathbf{s} + \mathbf{e}$ using the LWE assumption. Here we crucially rely on the fact that switched to simulation of secret keys without knowing the trapdoor of $\mathbf{A}$. Going further, once $\mathbf{c}$ is random, we can switch back to simulating secret keys using the trapdoor $\mathbf{T}$. Hence, the secret keys now do not leak any information about $\mathbf{R}'_1, \ldots, \mathbf{R}'_t$. Therefore, we may then invoke the left-over hash lemma to argue that $\mathbf{x}$ is information-theoretically hidden.

4 Predicate Encryption for Circuits

In this section, we present our main construction of predicate encryption for circuits by bootstrapping on top of the partially-hiding predicate encryption. That is,

- We construct a Predicate Encryption scheme $\mathcal{PE} = (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ for boolean predicate family $\mathcal{C}$ bounded by depth d over k bit inputs.

starting from

- an FHE scheme $\mathcal{FHE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ with properties as described in Section 2.1. Define ℓ as the size of the initial ciphertext encrypting k bit messages, and t as the size of the FHE secret key and evaluated ciphertext vectors;
- a partially-hiding predicate encryption scheme $\mathcal{PHPE} = (\mathsf{PH.Setup}, \mathsf{PH.Keygen}, \mathsf{PH.Enc}, \mathsf{PH.Dec})$ for the class $\mathcal{C}_{\mathsf{PHPE}}$ of predicates bounded by some depth parameter $d' = \mathrm{poly}(d, \lambda, \log q)$. Recall that

$$(\widehat{C} \circ \mathsf{IP}_\gamma)(\mathbf{x} \in \mathbb{Z}_q^t, \mathbf{y} \in \{0,1\}^t) = 1 \text{ iff } \Big(\sum_{i \in [t]} \mathbf{x}[i] \cdot \widehat{C}(\mathbf{y})[i] \Big) = \gamma \mod q$$

  where $\widehat{C} : \{0,1\}^\ell \to \{0,1\}^t$ is a circuit of depth at most d'.

Overview. At a high level, the construction proceeds as follows:

- the $\mathcal{PE}$ ciphertext corresponding to an attribute $\mathbf{a} \in \{0,1\}^k$ is a $\mathcal{PHPE}$ ciphertext corresponding to an attribute $(\mathsf{fhe.sk}, \mathsf{fhe.ct})$ where $\mathsf{fhe.sk} \xleftarrow{\$} \mathbb{Z}_q^t$ is private and $\mathsf{fhe.ct} := \mathsf{HE.Enc}(\mathbf{a}) \in \{0,1\}^\ell$ is public;
- the $\mathcal{PE}$ secret key for a predicate $C : \{0,1\}^k \to \{0,1\} \in \mathcal{C}$ is a collection of $2B+1$ $\mathcal{PHPE}$ secret keys for the predicates $\{\widehat{C} \circ \mathsf{IP}_\gamma : \mathbb{Z}_q^t \times \{0,1\}^\ell \to \{0,1\}\}_{\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B}$ where $\widehat{C} : \{0,1\}^\ell \to \{0,1\}$ is the circuit:

$$\widehat{C}(\mathsf{fhe.ct}) := \mathsf{HE.Eval}(\mathsf{fhe.ct}, C),$$

  so $\widehat{C}$ is a circuit of depth at most $d' = \mathrm{poly}(d, \lambda, \log q)$;
- decryption works by trying all possible $2B+1$ secret keys.

Note that the construction relies crucially on the fact that B (the bound on the noise in the FHE evaluated ciphertexts) is polynomial. For correctness, observe that for all $C, \mathbf{a}$:

$$\begin{aligned}
& C(\mathbf{a}) = 1 \\
& \quad \Leftrightarrow \quad \mathsf{HE.Dec}(\mathsf{fhe.sk}, \mathsf{HE.Eval}(C, \mathsf{fhe.ct})) = 1 \\
& \quad \Leftrightarrow \quad \exists\, \gamma \in [\lfloor q/2 \rfloor - B, \lfloor q/2 \rfloor + B] \text{ such that } \Big(\sum_{i \in [t]} \mathsf{fhe.sk}[i] \cdot \mathsf{fhe.ct}[i] \Big) = \gamma \mod q \\
& \quad \Leftrightarrow \quad \exists\, \gamma \in [\lfloor q/2 \rfloor - B, \lfloor q/2 \rfloor + B] \text{ such that } (\widehat{C} \circ \mathsf{IP}_\gamma)(\mathsf{fhe.sk}, \mathsf{fhe.ct}) = 1
\end{aligned}$$

where $\mathsf{fhe.sk}, \mathsf{fhe.ct}, \hat{C}$ are derived from $C, \mathbf{a}$ as in our construction.

4.1 Our Predicate Encryption scheme

Our construction proceeds as follows:

- $\mathsf{Setup}(1^\lambda, 1^k, 1^d)$: The setup algorithm takes the security parameter λ, the attribute length k and the predicate depth bound d.

  1. Run the partially-hiding PE scheme for family $\mathcal{C}_{\mathsf{PHPE}}$ to obtain a pair of master public and secret keys:
  $$(\mathsf{ph.mpk}, \mathsf{ph.msk}) \leftarrow \mathsf{PH.Setup}(1^\lambda, 1^t, 1^\ell, 1^{d'})$$
  where for k-bit messages and depth d circuits: t is the length of FHE secret key, ℓ is the bit-length of the initial FHE ciphertext and d' is the bound on FHE evaluation circuit (as described at the beginning of this section).
  2. Output $(\mathsf{mpk} := \mathsf{ph.mpk}, \mathsf{msk} := \mathsf{ph.msk})$.

- $\mathsf{Keygen}(\mathsf{msk}, C)$: The key-generation algorithms takes as input the master secret key msk and a predicate C. It outputs a secret key sk_C computed as follows.

  1. Let $\widehat{C}(\cdot) := \mathsf{HE.Eval}(\cdot, C)$ and let $(\widehat{C} \circ \mathsf{IP}_\gamma)$ be the predicates for $\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B$.
  2. For all $\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B$, compute
  $$\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}_\gamma} \leftarrow \mathsf{PH.Keygen}\big(\mathsf{ph.msk}, \widehat{C} \circ \mathsf{IP}_\gamma\big)$$
  3. Output the secret key as $\mathsf{sk}_C := \big(\{\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}}\}_{\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B}\big)$.

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{a}, \mu)$: The encryption algorithm takes as input the public key mpk, the input attribute vector $\mathbf{a} \in \{0,1\}^k$ and message $\mu \in \{0,1\}$. It proceeds as follow.

  1. Samples a fresh FHE secret key $\mathsf{fhe.sk} \in \mathbb{Z}_q^t$ by running $\mathsf{HE.KeyGen}(1^\lambda, 1^{d'}, 1^k)$.
  2. Encrypt the input to obtain
  $$\mathsf{fhe.ct} \leftarrow \mathsf{HE.Enc}(\mathsf{fhe.sk}, \mathbf{a}) \in \{0,1\}^\ell$$

3. Compute
$$\mathsf{ct}_{\mathsf{fhe.ct}} \leftarrow \mathsf{PH.Enc}\big(\mathsf{mpk}, (\mathsf{fhe.sk}, \mathsf{fhe.ct}), \mu\big)$$
Note that the fhe.sk corresponds to the hidden attribute and fhe.ct corresponds to the public attribute.
4. Output the ciphertext $\mathsf{ct} = (\mathsf{ct}_{\mathsf{fhe.ct}}, \mathsf{fhe.ct})$.

- $\mathsf{Dec}((\mathsf{sk}_C, C), \mathsf{ct})$: The decryption algorithm takes as input the secret key sk_C with corresponding predicate C and the ciphertext ct. If there exists $\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B$ such that
$$\mathsf{PH.Dec}((\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}_\gamma}, \widehat{C} \circ \mathsf{IP}_\gamma), (\mathsf{ct}_{\mathsf{fhe.ct}}, \mathsf{fhe.ct})) = \mu \neq \perp$$
then output μ. Otherwise, output $\perp$.

4.2 Correctness

Lemma 4.1. *Let $\mathcal{C}$ be a family of predicates bounded by depth d and let $\mathcal{PHPE}$ be the partially-hiding PE and $\mathcal{FHE}$ be a fully-homomorphic encryption as per scheme description. Then, our predicate encryption scheme $\mathcal{PE}$ is correct. Moreover, the size of each secret key is* $\mathrm{poly}(d, \lambda)$ *and the size of each ciphertext is* $\mathrm{poly}(d, \lambda, k)$.

We refer the reader to the full version for the proof.

4.3 Security

Theorem 4.2. *Let $\mathcal{C}$ be a family of predicates bounded by depth d and let $\mathcal{PHPE}$ be the secure partially-hiding PE and $\mathcal{FHE}$ be the secure fully-homomorphic encryption as per scheme description. Then, our predicate encryption scheme $\mathcal{PE}$ is secure.*

Proof. We define p.p.t. simulator algorithms $\mathsf{Enc}_{\mathrm{Sim}}$ and argue that its output is indistinguishable from the output of the real experiment. Let $\mathsf{PH.Enc}_{\mathrm{Sim}}$ be the p.p.t. simulator for partially-hiding predicate encryption scheme.

- $\mathsf{Enc}_{\mathrm{Sim}}(\mathsf{mpk}, 1^{|\mathbf{a}|}, 1^{|\mu|})$: To compute the encryption, the simulator does the following. It samples FHE secret key fhe.sk by running $\mathsf{HE.KeyGen}(1^\lambda, 1^{d'}, 1^k)$. It encrypts a zero-string $\mathsf{fhe.ct} \leftarrow \mathsf{HE.Enc}(\mathsf{fhe.sk}, \mathbf{0})$. It obtains the ciphertext as $\mathsf{ct}_{\mathsf{fhe.ct}} \leftarrow \mathsf{PH.Enc}_{\mathrm{Sim}}(\mathsf{mpk}, \mathsf{fhe.ct}, 1^{|\mathsf{fhe.sk}|}, 1^{|\mu|})$.

We now argue via a series of hybrids that the output of the ideal experiment.

- **Hybrid 0**: The real experiment.
- **Hybrid 1**: The real encryption algorithm is replaced with Enc^*, where Enc^* is an auxiliary algorithm defined below. On the high level, Enc^* computes the FHE ciphertext honestly by sampling a secret key and using the knowledge of $\mathbf{a}$. It then invokes $\mathsf{PH.Enc}_{\mathrm{Sim}}$ on the honestly generated ciphertext.
- **Hybrid 2**: The simulated experiment.

Auxiliary Algorithms. We define the auxiliary algorithm Enc^* used in Hybrid 1.

- $\mathsf{Enc}^*(\mathbf{a}, 1^{|\mu|})$: The auxiliary encryption algorithm takes as input the attribute vector $\mathbf{a}$ and message length.

  1. Sample a fresh FHE secret key $\mathsf{fhe.sk}$ by running $\mathsf{HE.KeyGen}(1^\lambda, 1^{d'}, 1^k)$.
  2. Encrypt the input attribute vector to obtain a ciphertext
  $$\mathsf{fhe.ct} \leftarrow \mathsf{HE.Enc}(\mathsf{fhe.sk}, \mathbf{a}) \in \{0,1\}^\ell$$
  3. Run $\mathsf{PH.Enc}_{\mathrm{Sim}}$ on input $(\mathsf{mpk}, \mathsf{fhe.ct}, 1^{|\mathsf{fhe.sk}|}, 1^{|\mu|})$ to obtain the ciphertext $\mathsf{ct}_{\mathsf{fhe.ct}}$.

Lemma 4.3. *The output of Hybrid 0 is computationally indistinguishable from the Hybrid 1, assuming security of Partially-Hiding Predicate Encryption.*

Proof. Assume there is an adversary Adv and a distinguisher $\mathcal{D}$ that distinguishes the output $(\mathbf{a}, \mu, \alpha)$ produced in either of the two hybrids. We construct an adversary Adv′ and a distinguisher $\mathcal{D}'$ that break the security of the Partially-Hiding Predicate Encryption. The adversary Adv′ does the following.

1. Invoke the adversary Adv to obtain an attribute vector $\mathbf{a}$.
2. Sample a fresh FHE secret key $\mathsf{fhe.sk}$ using $\mathsf{HE.KeyGen}(1^\lambda, 1^{d'}, 1^k)$. Encrypt the attribute vector
   $$\mathsf{fhe.ct} \leftarrow \mathsf{HE.Enc}(\mathsf{fhe.sk}, \mathbf{a})$$
   and output the pair $(\mathsf{fhe.sk}, \mathsf{fhe.ct})$ as the "selective" challenge attribute.
3. Upon receiving mpk, it forwards it to Adv.
4. For each oracle query C that Adv makes which satisfies $C(\mathbf{a}) \neq 0$, Adv′ uses its oracle to obtain secret keys $\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}_\gamma}$ for $\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B$. It outputs $\mathsf{sk}_C = \left(\{\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}_\gamma}\}_{\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B}\right)$.
5. It outputs message μ that Adv produces, obtains a ciphertext $\mathsf{ct}_{\mathsf{fhe.ct}}$ and sends $\mathsf{ct} = (\mathsf{ct}_{\mathsf{fhe.ct}}, \mathsf{fhe.ct})$ back to Adv to obtain α.

We note that given Adv that is admissible, Adv′ is also admissible. That is, for all queries $\widehat{C} \circ \mathsf{IP}_\gamma$ that Adv′ makes satisfies $(\widehat{C} \circ \mathsf{IP}_\gamma)(\mathsf{fhe.sk}, \mathsf{fhe.ct}) = 0$ since $\langle \mathsf{fhe.sk}, \widehat{C}(\mathsf{fhe.ct}) \rangle \neq \gamma$ for $\gamma = \lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B$ by the correctness of FHE in Section 2.1 and the fact that $C(\mathbf{a}) \neq 0$. Finally, the distinguisher $\mathcal{D}'$ on input $(\mathsf{fhe.sk}, \mathsf{fhe.ct}, \mu, \alpha)$ invokes $\mathcal{D}$ and outputs whatever it outputs. Now, in Hybrid 0 the algorithms used as $\mathsf{PH.Setup}, \mathsf{PH.Keygen}, \mathsf{PH.Enc}$ which corresponds exactly to the real security game of PHPE. However, in Hybrid 1 the algorithms correspond exactly to the simulated security game. Hence, we can distinguish between the real and simulated experiments contradicting the security of PHPE scheme. □

Lemma 4.4. *The output of Hybrid 1 and Hybrid 2 are computationally indistinguishable, assuming semantic security of Fully-Homomorphic Encryption Scheme.*

Proof. The only difference in Hybrids 1 and 2 is how the FHE ciphertext is produced. In one experiment, it is computed honestly by encrypting the attribute vector $\mathbf{a}$, while in the other experiment it is always an encryption of $\mathbf{0}$. Hence, we can readily construct an FHE adversary that given $\mathbf{a}$, distinguishes encryption of $\mathbf{a}$ from encryption of $\mathbf{0}$ as follows:

1. Invoke the admissible PE adversary Adv to obtain an attribute vector $\mathbf{a}$.
2. Run the honest $\mathsf{PH.Setup}$ and forwards mpk to Adv.
3. For each oracle query C that Adv makes which satisfies $C(\mathbf{a}) \neq 0$, return $\mathsf{sk}_C = \left(\{\mathsf{sk}_{\widehat{C} \circ \mathsf{IP}_\gamma}\}_{\gamma=\lfloor q/2 \rfloor - B, \ldots, \lfloor q/2 \rfloor + B}\right)$ as computed using the honest $\mathsf{PH.Keygen}$ algorithm.
4. To simulate the ciphertext, first forward the pair $(\mathbf{a}, \mathbf{0})$ to the FHE challenger to obtain a ciphertext $\mathsf{fhe.ct}$. Then, run $\mathsf{PH.Enc}_{\mathsf{Sim}}(\mathsf{mpk}, \mathsf{fhe.ct}, 1^{|\mathsf{fhe.sk}|}, 1^{\mu})$ to obtain a ciphertext $\mathsf{ct}_{\mathsf{fhe.ct}}$ and forward it to Adv
5. Finally, it runs the PE distinguisher on input $(\mathbf{a}, \mu, \alpha)$ and outputs its guess.

The lemma then follows from semantic security of the FHE completing the security proof. We also refer the reader to the full version for the summary of parameters selection.

□

□

Part V

Fully, (Almost) Tightly Secure IBE and Dual System Groups

Jie Chen and Hoeteck Wee
CRYPTO 2013

Abstract. We present the first fully secure Identity-Based Encryption scheme (IBE) from the standard assumptions where the security loss depends only on the security parameter and is independent of the number of secret key queries. This partially answers an open problem posed by Waters (Eurocrypt 2005). Our construction combines Waters' dual system encryption methodology (Crypto 2009) with the Naor-Reingold pseudo-random function (J. ACM, 2004) in a novel way. The security of our scheme relies on the DLIN assumption in prime-order groups. Along the way, we introduce a novel notion of *dual system groups* and a new randomization and parameter-hiding technique for prime-order bilinear groups.

1 Introduction

In an Identity-Based Encryption (IBE) scheme [142], encryption requires only the identity of the recipient (e.g. an email address or an IP address) and a set of global public parameters, thus eliminating the need to distribute a separate public key for each user in the system. The first realizations of IBE were given in 2001; the security of these schemes were based on either Bilinear Diffie-Hellman or QR in the random oracle model [29, 56]. Since then, tremendous progress has been made towards obtaining IBE and HIBE schemes that are secure in the standard model based on pairings [42, 27, 28, 145, 75, 146] as well as lattices [77, 46, 7, 8]. Specifically, starting with [146], we now have very efficient constructions of IBE based on standard assumptions which achieve the strongest security notion of full (adaptive) security, where the adversary may choose the challenge identity after seeing both the public parameters and making key queries.

In this work, we focus on the issue of security reduction and security loss in the construction of fully secure IBE. Consider an IBE scheme with a security reduction showing that attacking the scheme in time t with success probability ϵ implies breaking some conjectured hard problem in time roughly t with success probability ϵ/L; we refer to L as the security loss, and a tight reduction is one where L is a constant. All known constructions of fully secure IBE schemes from standard assumptions incur a security loss that is at least linear in the number of key queries q; the only exceptions are constructions in the random oracle model [29] and those based on q-type assumptions [75]. Motivated by this phenomenon, Waters [145] posed the following problem in 2005 (reiterated in [75, 21]):

> "*Design an IBE with a tight security reduction to a standard assumption.*"

That is, we are interested in constructions based on "static" assumptions like the Decisional Linear (DLIN) assumption or the subgroup decisional assumption and which do not rely on random oracles.

Note that an IBE with a tight security reduction would also imply signatures with a tight security reduction via Naor's transformation [29]; indeed, the latter were the focus in a series of very recent works [1, 104, 96].

We stress that tight reductions are not just theoretical issues for IBE, rather they are of utmost practical importance: as L increases, we need to increase the size of the underlying groups in order to compensate for the security loss, which in turn increases the running time of the implementation. Note that the impact on performance is quite substantial, as exponentiation in a r-bit group takes time roughly $\mathcal{O}(r^3)$.

While the ultimate goal is to achieve constant security loss (i.e. $L = \mathcal{O}(1)$), even achieving $L = \text{poly}(\lambda)$ and independent of q is already of both practical and theoretical interest. For typical settings of parameters (e.g. $\lambda = 128$ and $q = 2^{20}$), λ is much smaller than q. From the theoretical stand-point, we currently have two main techniques for obtaining fully secure IBE from standard assumptions: random partitioning [145] and dual system encryption framework [146]. For the former, we now know that an $\Omega(q)$ security loss is in fact inherent [98]. For the latter, all known instantiations also incur an $\Omega(q)$ security loss; an interesting theoretical question is whether this is in fact inherent to the dual system encryption framework.

1.1 Our results

Our main result is an IBE scheme based on the (generalized) d-LIN assumption with security loss $\mathcal{O}(\lambda)$ for λ-bit identities:

Theorem 1.1. *There exists an IBE scheme for identity space $\{0,1\}^n$ based on the d-LIN assumption with the following property: for any adversary $\mathcal{A}$ that makes at most q key queries against the IBE scheme, there exist an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\text{IBE}}_{\mathcal{A}}(\lambda) \leq (2n+1) \cdot \mathsf{Adv}^{d\text{-LIN}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}$$

and

$$\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathit{poly}(\lambda, n),$$

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

We compare our scheme with prior constructions in Figure 8. Applying Naor's transformation, we also obtain a d-LIN-based signature scheme with constant-size signatures and security loss independent of the number of signature queries. This yields an alternative construction for an analogous result in [96].

1.1.1 Our approach.

The inspiration for our construction comes from a recent connection between predicate encryption and one-time symmetric-key primitives [148] — namely one-time MACs in the case of IBE — via dual system encryption [146]. Our key observation is to extend this connection to "reusable MACs", namely that if we start with an appropriate pseudorandom function (PRF) with security loss L, we may derive an IBE with the security loss $\mathcal{O}(L)$. More concretely, we begin with the Naor-Reingold DDH-based PRF [124] which has security loss n for input domain $\{0,1\}^n$, and obtain a fully secure IBE with security loss $\mathcal{O}(n)$ via a

Reference	$\|\mathsf{MPK}\|$	security loss	additive overhead	assumption
BB1 [27]	$\mathcal{O}(1)$	$\mathcal{O}(2^n)$	$q \cdot \mathrm{poly}(\lambda, n)$	DBDH
Waters [145]	$\mathcal{O}(n)$	$\mathcal{O}(qn)$	$q^2\epsilon^{-2} \cdot \mathrm{poly}(\lambda, n)$	DBDH
Gentry [75]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$q^2 \cdot \mathrm{poly}(\lambda, n)$	q-ABDHE
BR [21]	$\mathcal{O}(n)$	$\mathcal{O}(qn/\epsilon)$	$q \cdot \mathrm{poly}(\lambda, n)$	DBDH
LW[146, 113, 111]	$\mathcal{O}(1)$	$\mathcal{O}(q)$	$q \cdot \mathrm{poly}(\lambda, n)$	DLIN or composite
Ours	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$q \cdot \mathrm{poly}(\lambda, n)$	DLIN or composite
(Sec 5)	$\mathcal{O}(d^2 n)$	$\mathcal{O}(n)$	$d^2 q \cdot \mathrm{poly}(\lambda, n)$	d-LIN

Figure 8: Comparison amongst IBE schemes, where $\{0,1\}^n$ is the identity space, q is the number of adversary's key queries, and ϵ is the adversary's advantage. In all of these constructions, $|\mathsf{SK}| = |\mathsf{CT}| = \mathcal{O}(1)$.

novel variant of the dual system encryption methodology. Our IBE scheme is essentially that obtained by embedding Waters' fully secure IBE based on DBDH [145] into composite-order groups, and then converting this to a prime-order scheme following [51, 127, 111, 64] (along with some new technical ideas). Here, we exploit the fact that Waters' IBE and the Naor-Reingold PRF share a similar algebraic structure based on bit-by-bit encoding of the identity and PRF input respectively.

1.2 Technical overview

We provide a more technical overview of our main results, starting with the proof idea and then the construction. Here, we assume some familiarity with prior works.

1.2.1 Proof idea.

Our security proof combines Waters' dual system encryption methodology [146] with ideas from the analysis of the Naor-Reingold PRF. In a dual system encryption scheme [146], there are two types of keys and ciphertexts: normal and semi-functional. A key will decrypt a ciphertext properly unless both the key and the ciphertext are semi-functional, in which case decryption will fail with overwhelming probability. The normal keys and ciphertexts are used in the real system, and keys are gradually introduced in the hybrid security proof, one at a time. Ultimately, we arrive at a security game in which the simulator only has to produce semi-functional objects and security can be proved directly. In all prior instantiations of this methodology, the semi-functional keys are introduced one at a time. As a result, we require q hybrid games to switch all of the keys from normal to semi-functional, leading to an $\Omega(q)$ security loss, since each step requires a computational assumption.

We deviate from the prior paradigm by using only n hybrid games, iterating over the bits in the bit-by-bit encoding of the identity, as was done in the Naor-Reingold PRF. That is, we introduce n types of semi-functional ciphertexts and keys, where type i objects appear in game i, while gradually increasing the entropy in the semi-functional components in each game. This strategy introduces new challenges specific to the IBE setting, namely that the adversary could potentially use the challenge ciphertext to test whether we have switched from type $i-1$ keys to type i keys. Prior works exploit the fact that we only switch a *single* key in each step, whereas we could be switching up to q keys in each step.

Property	Where it is used	
	nested dual system groups	dual system groups
projective	correctness	correctness
	normal to type 0 (Lemma 4.2)	normal to semi-functional CT
associative	correctness	correctness
orthogonality	normal to type 0 (Lemma 4.3)	final transition
non-degeneracy	final transition (Lemma 4.5)	pseudo-normal to pseudo-SF keys
		final transition
$\mathbb{H}$-subgroup	type $i-1$ to type i (Lemma 4.4)	key delegation
left subgroup	normal to type 0 (Lemma 4.2)	normal to semi-functional CT
nested-hiding	type $i-1$ to type i (Lemma 4.4)	*unavailable*
right subgroup	*unavailable*	normal to pseudo-normal keys
		pseudo-SF to semi-functional keys
parameter-hiding	*unavailable*	pseudo-normal to pseudo-SF keys

Figure 9: Summary of dual system groups (c.f. Section 3 and Section 6)

We overcome this difficulty as follows. At step i of the hybrid game, we guess the i'th bit b_i of the challenge identity ID^*, and abort if our guess is incorrect. This results in a security loss of 2, which we can afford. If our guess b_i is correct,

- for all identities whose i'th bit equals b_i, the corresponding type $i-1$ and type i object are the same;
- for all other identities, we increase the entropy of the keys going from type $i-1$ to type i (via a tight reduction to a computational assumption).

The first property implies that the adversary cannot use the challenge ciphertext to distinguish between type $i-1$ and type i keys; in the proof, the simulator will not be able to generate type $i-1$ or type i ciphertexts for identities whose i'th bit is different from b_i (c.f. Remark 3.3 and Section 4.4). Interestingly, decryption capabilities remain unchanged throughout the hybrid games: a type i key for ID^* can decrypt a type i ciphertext for ID^* (c.f. Remark 4.2). This is again different from prior instantiations of the dual system encryption methodology where decryption fails for semi-functional objects.

In the final transition, a semi-functional type n object for identity ID has semi-functional component $R_n(\mathrm{ID})$ where R_n is a truly random function. In particular, the semi-functional ciphertext has semi-functional component $R_n(\mathrm{ID}^*)$. Moreover, $R_n(\mathrm{ID}^*)$ is truly random from the adversary's view-point because it only learns $\mathrm{SK}_{\mathrm{ID}}$ and thus $R_n(\mathrm{ID})$ for $\mathrm{ID} \neq \mathrm{ID}^*$. We can then argue that the message which is masked by $R_n(\mathrm{ID}^*)$ is information-theoretically hidden.

1.2.2 Construction.

To achieve a modular analysis, we introduce a novel notion of nested dual system groups (see Section 3.1 for an overview). Our construction proceeds into two steps: the first builds an (almost) tight IBE from nested dual system groups where we rely on the Naor-Reingold PRF argument and the dual system encryption methodology; the second builds nested dual system groups from d-LIN where we handle all of the intricate linear algebra associated with simulating composite-order groups in prime-order groups from [51, 111] and with achieving a tight reduction via random self-reducibility.

1.2.3 Perspective.

In spite of the practical motivation for tight security reductions, we clarify that our contributions are largely of theoretical and conceptual interest. This is because any gain in efficiency from using smaller groups is overwhelmed by the loss from the bit-by-bit encoding of identities. Our work raises the following open problems:

- Can we reduce the size of the public parameters to a constant?
- Can we achieve tight security, namely $L = \mathcal{O}(1)$?

We note that progress on either problem would likely require improving on the Naor-Reingold PRF: namely, reducing respectively the seed length and the security loss to a constant, both of which are long-standing open problems. We also note that the present blow-up in public parameters and security loss arise only in using the Naor-Reingold approach to build an IBE from nested dual system groups; our instantiation of nested dual system groups do achieve tight security.

1.3 Additional results

As a pre-cursor to nested dual system groups, we introduce a basic notion of *dual system groups*. We present

- a generic construction of compact HIBE from dual system groups similar to the Lewko-Waters scheme over composite-order groups [113]; and
- instantiations of dual system groups under the d-LIN assumption in prime-order bilinear groups and the subgroup decisional assumption in composite-order bilinear groups respectively. Along the way, we provide a new randomization and parameter-hiding technique for prime-order groups.

Putting the two together, we obtain a new construction of compact HIBE in prime-order groups (see Figures 10 and 11), as well as new insights into the structural properties needed for Waters' dual system encryption methodology [146]. We proceed to present an overview of dual system groups, our new techniques for prime-order groups and then an overview of nested dual system groups.

Reference	$\lvert\text{MPK}\rvert$	$\lvert\text{SK}\rvert$	$\lvert\text{CT}\rvert$	T_{KeyGen}	T_{Enc}	T_{Dec}	assumption
Wat05 [145]	$(\lambda+4)\lvert G_1\rvert$	$2\lvert G_2\rvert$	$2\lvert G_1\rvert+\lvert G_T\rvert$	$2E_2$	$2E_1+E_T$	$2P$	DBDH
Wat09 [146]	$13\lvert G_1\rvert+\lvert G_T\rvert$	$8\lvert G_2\rvert+\lvert\mathbb{Z}_p\rvert$	$9\lvert G_1\rvert+\lvert G_T\rvert+\lvert\mathbb{Z}_p\rvert$	$8E_2$	$14E_1+E_T$	$9P+E_T$	DLIN
Lewko [111]	$24\lvert G_1\rvert+\lvert G_T\rvert$	$6\lvert G_2\rvert$	$6\lvert G_1\rvert+\lvert G_T\rvert$	$6E_2$	$24E_1+E_T$	$6P$	DLIN
RCS [135]	$9\lvert G_1\rvert+\lvert G_T\rvert$	$6\lvert G_2\rvert+\lvert\mathbb{Z}_p\rvert$	$7\lvert G_1\rvert+\lvert G_T\rvert+\lvert\mathbb{Z}_p\rvert$	$6E_2$	$10E_1+E_T$	$7P+E_T$	XDH + DLIN
CLL+ [52]	$8\lvert G_1\rvert+\lvert G_T\rvert$	$4\lvert G_2\rvert$	$4\lvert G_1\rvert+\lvert G_T\rvert$	$4E_2$	$8E_1+E_T$	$4P$	SXDH
Ours	$6\lvert G_1\rvert+\lvert G_T\rvert$	$4\lvert G_2\rvert$	$4\lvert G_1\rvert+\lvert G_T\rvert$	$4E_2$	$6E_1+E_T$	$4P$	SXDH
(Sec 7)	$18\lvert G_1\rvert+2\lvert G_T\rvert$	$6\lvert G_2\rvert$	$6\lvert G_1\rvert+\lvert G_T\rvert$	$6E_2$	$18E_1+2E_T$	$6P$	DLIN

Figure 10: Comparison amongst IBE schemes based on asymmetric bilinear groups of prime order p with pairing $e : G_1 \times G_2 \to G_T$ and security parameter λ, where (E_1, E_2, E_T, P) denote G_1-exponentiation, G_2-exponentiation, G_T-exponentiation and a pairing respectively. For KeyGen, we assume that we store exponents instead of group elements in MSK. Here, we omitted the G_2 terms in MPK in our scheme, which are not needed for the correctness of the scheme.

1.3.1 Dual system groups.

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

(subgroup indistinguishability.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". An analogous statement holds for $\mathbb{H}^{n+1}$.

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

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

(parameter-hiding.) Both normal distributions can be efficiently sampled given the public parameters; on the other hand, given only the public parameters, the higher-entropy distributions contain n "units" of information-theoretic entropy (in the semi-functional component), one unit for each of the n elements in the $(n+1)$-tuple apart from the first.

The key novelty in the framework lies in identifying the role of associativity in the prior instantiations of the dual system encryption methodology in composite-order groups [113].

1.3.2 Instantiation in prime-order groups.

We present a new randomization and parameter-hiding technique for prime-order bilinear groups, which we use to instantiate dual system groups. This technique allows us to hide arbitrarily large

Reference	$\lvert\mathrm{CT}\rvert$	T_{KeyGen}	T_{Enc}	T_{Dec}	assumption
BBG [31]	$2\lvert G_1\rvert + \lvert G_T\rvert$	$(n+1)E_2$	$(n+2)E_1 + E_T$	$2P$	n-DBDHE
Wat09 [146]	$(n+8)\lvert G_1\rvert + \lvert G_T\rvert$	$(2n+7)E_2$	$(3n+11)E_1 + E_T$	$(2n+7)P + nE_T$	DLIN
LW [113]	$2\lvert G_N\rvert + \lvert G_T\rvert$	$(n+1)E_N$	$(n+2)E_N + E_T$	$2P$	composite
OT10 [128]	$(7n+5)\lvert G_1\rvert + \lvert G_T\rvert$	$(7n+5)E_2$	$(21n+15)E_1 + E_T$	$(7n+5)P$	DLIN
CLL+ [52]	$(4n+3)\lvert G_1\rvert + \lvert G_T\rvert$	$(4n+3)E_2$	$(8n+6)E_1 + E_T$	$(4n+3)P$	SXDH
OT11 [129]	$13\lvert G_1\rvert + \lvert G_T\rvert$	$(16n-3)E_2$	$(8n+13)E_1 + E_T$	$13P$	DLIN
Ours	$4\lvert G_1\rvert + \lvert G_T\rvert$	$2(n+1)E_2$	$2(n+1)E_1 + E_T$	$4P$	SXDH
(Sec 7)	$6\lvert G_1\rvert + \lvert G_T\rvert$	$3(n+1)E_2$	$6(n+1)E_1 + 2E_T$	$6P$	DLIN
	$2(d+1)\lvert G_1\rvert + \lvert G_T\rvert$	$(d+1)(n+1)E_2$	$d(d+1)(n+1)E_1 + dE_T$	$2(d+1)P$	d-LIN

Figure 11: Comparison between existing and our HIBE schemes, where n is the depth parameter; in addition, E_N denotes G_N-exponentiation. In all of the prime-order constructions, $\lvert\mathrm{MPK}\rvert = \mathcal{O}(n\lvert G_1\rvert + n\lvert G_2\rvert + \lvert G_T\rvert)$ and $\lvert\mathrm{SK}\rvert = \mathcal{O}(n\lvert G_2\rvert)$. For T_{Dec}, we omitted the overhead of $\mathcal{O}(n)$ exponentiations associated with delegating a key before decrypting. Apart from [31], all of the schemes achieve full security.

amounts of entropy while working with a vector space of constant dimensions, whereas prior works require a linear blow-up in dimensions.

To motivate the new technique, we begin with a review of composite-order bilinear groups. Let (G_N, G_T) denote a composite-order bilinear group of order $N = p_1 p_2$ which is the product of two primes, endowed with an efficient bilinear map $e : G_N \times G_N \to G_T$. Let g denote an element of G_N of order p_1. A useful property of composite-order groups, especially in the context of dual system encryption [113, 115], is that we can perform randomization by raising a group element to the power of a random exponent $a \leftarrow_{\mathrm{R}} \mathbb{Z}_N$. This operation satisfy the following useful properties:

(parameter-hiding.) given g, g^a, the quantity $a \pmod{p_2}$ is completely hidden;

(associativity.) for all $u \in G_N$, we have $e(g^a, u) = e(g, u^a)$.

We show how to achieve randomization in the prime-order setting under the d-LIN assumption. Fix a prime-order bilinear group (G, G_T) of order p, endowed with an efficient bilinear map $e : G \times G \to G_T$. Let g denote an element of G of order p. Elements in G_N correspond to elements in G^{d+1} and we consider the bilinear map $e : G^{d+1} \times G^{d+1} \to G_T$ given by $e(g^{\mathbf{x}}, g^{\mathbf{y}}) := e(g,g)^{\mathbf{x}^\top \mathbf{y}}$. Following [127, 64], we pick a random pair of orthogonal basis $(\mathbf{B}, \mathbf{B}^*) \leftarrow_{\mathrm{R}} \mathsf{GL}_{d+1}(\mathbb{Z}_p) \times \mathsf{GL}_{d+1}(\mathbb{Z}_p)$ so that $\mathbf{B}^\top \mathbf{B}^*$ is the identity matrix. We consider the projection maps π_L, π_R that map a $(d+1) \times (d+1)$ matrix to the left d columns and right-most column; they correspond to projecting $a \in \mathbb{Z}_N$ to $a \pmod{p_1}$ and $a \pmod{p_2}$ respectively.

We randomize a basis $(\mathbf{B}, \mathbf{B}^*)$ as follows: pick a random $\mathbf{A} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)}$ and replace $(\mathbf{B}, \mathbf{B}^*)$ with $(\mathbf{BA}, \mathbf{B}^* \mathbf{A}^{-\top})$. Observe that this transformation satisfy the following properties similar to those in the composite-order setting:

(parameter-hiding.) given $g^{\pi_L(\mathbf{B})}, g^{\pi_L(\mathbf{BA})}, g^{\pi_L(\mathbf{B}^*)}, g^{\pi_L(\mathbf{B}^* \mathbf{A}^{-\top})}$, the bottom-right entry of $\mathbf{A}$ is completely hidden;

(associativity.) for all $(\mathbf{B}, \mathbf{B}^*)$ and all $\mathbf{A} \in \mathbb{Z}_p^{(d+1)\times(d+1)}$, we have

$$e(g^{\mathbf{BA}}, g^{\mathbf{B}^*}) = e(g^{\mathbf{B}}, g^{\mathbf{B}^*\mathbf{A}^\top}) \left(= e(g,g)^{\mathbf{A}^\top}\right)$$

where $e(g^{\mathbf{X}}, g^{\mathbf{Y}}) := e(g,g)^{\mathbf{X}^\top\mathbf{Y}}$.

We also establish a subspace indistinguishability assumption similar to those in prior works [128, 111, 52].

1.3.3 Nested dual system groups.

In nested dual system groups, we require a so-called *nested-hiding* property. Roughly speaking, this property says that it is computationally infeasible to distinguish q samples from some distribution with another; specifically, it allows us to boost the entropy of the semi-functional components. In the instantiation, we will need to establish this property with a tight reduction to some standard assumption. The nested-hiding property allows us to "embed" the Naor-Reingold analysis into the semi-functional space of a dual system encryption scheme. We stress that the nested-hiding property even for $q = 1$ is *qualitatively* different from right subgroup indistinguishability in dual system groups.

We outline the instantiations of nested dual system groups in the composite-order and prime-order settings:

- The composite-order instantiation is very similar to that as before. We rely on composite-order group whose order is the product of three primes p_1, p_2, p_3. The subgroup G_{p_1} of order p_1 serves as the "normal space" and G_{p_2} of order p_2 serves as the "semi-functional space". We also require a new static, generically secure assumption, which roughly speaking, states that DDH is hard in the G_{p_2} subgroup. Here, we extend the techniques from [124] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction. Our IBE analysis may also be viewed as instantiating the Naor-Reingold PRF in the G_{p_2} subgroup.
- For the prime-order instantiation based on d-LIN, we extend the prior instantiation in several ways. First, we work with $2d \times 2d$ matrices instead of $(d+1) \times (d+1)$ matrices. In both constructions, the first d dimensions serve as the "normal space"; in our construction, we require a d-dimensional semi-functional space instead of a 1-dimensional one so that we may embed the d-LIN assumption into the semi-functional space. Next, we extend the techniques from [124, 112] to establish nested-hiding indistinguishability without losing a factor of q in the security reduction.

Perspective.

In developing the framework for dual system groups, we opted to identify the minimal properties needed for the application to dual system encryption in the most basic setting of (H)IBE; we adopted an analogous approach also for nested dual system groups. An alternative approach would have been to maximize the properties satisfied by both the composite-order and prime-order instantiations, with the hope of capturing a larger range of applications. In choosing the minimalist approach, we believe we can gain better insights into how and why dual system encryption works, as well as guide potential lattice-based instantiations. In addition, we wanted the framework to be as concise as possible and the

instantiations to be as simple as possible. Nonetheless, the framework remains fairly involved and we hope to see further simplifications in future work.

Organization.

We present nested dual system groups in Section 3, our IBE scheme in Section 4 and a self-contained description of our d-LIN-based scheme in Section 5. For completeness, we included a formal description of dual system group in Section 6 and the d-LIN-based compact HIBE we derive from it in Section 7. We defer all other details to the full versions of this paper [49, 51].

2 Preliminaries

Notation.

We denote by $s \leftarrow_R S$ the fact that s is picked uniformly at random from a finite set S and by $x, y, z \leftarrow_R S$ that all x, y, z are picked independently and uniformly at random from S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout, we use 1^λ as the security parameter. We use $\cdot$ to denote multiplication (or group operation) as well as component-wise multiplication. We use lower case boldface to denote (column) vectors over scalars or group elements and upper case boldface to denote vectors of group elements as well as matrices. Given a group G, we use $\mathrm{ord}(G)$ to denote the smallest positive integer c such that $g^c = 1$ for all $g \in G$.

Identity-Based Encryption.

An IBE scheme consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda, 1^n) \to (\mathrm{MPK}, \mathrm{MSK})$. The setup algorithm takes in the security parameter 1^λ and the length parameter 1^n. It outputs public parameters MPK and a master secret key MSK.

$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m) \to \mathrm{CT}_{\mathbf{x}}$. The encryption algorithm takes in the public parameters MPK, an identity $\mathbf{x}$, and a message m. It outputs a ciphertext $\mathrm{CT}_{\mathbf{x}}$.

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y}) \to \mathrm{SK}_{\mathbf{y}}$. The key generation algorithm takes in the public parameters MPK, the master secret key MSK, and an identity $\mathbf{y}$. It outputs a secret key $\mathrm{SK}_{\mathbf{y}}$.

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}}) \to m$. The decryption algorithm takes in the public parameters MPK, a secret key $\mathrm{SK}_{\mathbf{y}}$ for an identity $\mathbf{y}$, and a ciphertext $\mathrm{CT}_{\mathbf{x}}$ encrypted under an identity $\mathbf{x}$. It outputs a message m if $\mathbf{x} = \mathbf{y}$.

Correctness.

For all $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, all identities $\mathbf{x}$, all messages m, all decryption keys $\mathrm{SK}_{\mathbf{y}}$, all $\mathbf{x}$ such that $\mathbf{x} = \mathbf{y}$, we have

$$\Pr[\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathsf{Enc}(\mathrm{MPK}, \mathbf{x}, m)) = m] = 1.$$

Security Model.

The security game is defined by the following experiment, played by a challenger and an adversary $\mathcal{A}$.

Setup. The challenger runs the setup algorithm to generate $(\mathrm{MPK}, \mathrm{MSK})$. It gives MPK to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively requests keys for any identity $\mathbf{y}$ of its choice. The challenger responds with the corresponding secret key $\mathrm{SK}_{\mathbf{y}}$, which it generates by running $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$.

Challenge. The adversary $\mathcal{A}$ submits two messages m_0 and m_1 of equal length and a challenge identity $\mathbf{x}^*$ with the restriction that $\mathbf{x}^*$ is not equal to any identity requested in the previous phase. The challenger picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$, and encrypts m_β under $\mathbf{x}^*$ by running the encryption algorithm. It sends the ciphertext to the adversary $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to issue key queries for any identity $\mathbf{y}$ as in Phase 1 with the restriction that $\mathbf{y} \neq \mathbf{x}^*$.

Guess. The adversary $\mathcal{A}$ must output a guess β' for β.

The advantage $\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda)$ of an adversary $\mathcal{A}$ is defined to be $|\Pr[\beta' = \beta] - 1/2|$.

Definition 2.1. *An IBE scheme is* fully secure *if all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.*

3 Nested Dual System Groups

In this section, we present nested dual system groups, a variant of dual system groups with a notable difference: we require (computational) nested-hiding indistinguishability, in place of (computational) right subgroup indistinguishability and (information-theoretic) parameter-hiding. As noted in the introduction, the nested-hiding property even for $q = 1$ is *qualitatively* different from right subgroup indistinguishability in dual system groups.

3.1 Overview

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

(left subgroup $\mathbb{G}$.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". We sample the normal distribution using SampG and the semi-functional components using $\widehat{\mathsf{SampG}}$.

(right subgroup $\mathbb{H}$.) There is a single algorithm SampH to sample correlated $(n+1)$-tuples from $\mathbb{H}^{n+1}$. We should think of these tuples as already having semi-functional components, generated by some distinguished element $h^* \in \mathbb{H}$. It is convenient to think of h^* as being orthogonal to each component in the normal distribution over $\mathbb{G}$ (c.f. orthogonality and Remark 3.1). On the other hand, we require that h^* is *not* orthogonal to the semi-functional components in $\mathbb{G}$ (c.f. non-degeneracy) in order to information-theoretically hide the message in the final transition.

(nested-hiding.) We require a computational assumption over $\mathbb{H}$ which we refer to as *nested-hiding*, namely that for each $i = 1, \ldots, n$,

$$(h_0, h_i) \quad \text{and} \quad (h_0, h_i \cdot (h^*)^\gamma)$$

are computationally indistinguishable, where $(h_0, h_1, \ldots, h_n)$ is sampled using SampH and γ is a random exponent. In the formal definition, we provide the adversary with q samples from these distributions, and in the instantiations, we provide a tight reduction (independent of q) to a static assumption such as DLIN.

(associativity.) For all $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ sampled using SampG and SampH respectively, we have that for all $i = 1, \ldots, n$,

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

We require this property for correctness.

3.2 Definitions

3.2.1 Syntax.

Nested dual system groups consist of five randomized algorithms given by $(\mathsf{SampP}, \mathsf{SampGT}, \mathsf{SampG}, \mathsf{SampH})$ along with $\widehat{\mathsf{SampG}}$:

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

- PP contains a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a linear map μ defined on $\mathbb{H}$, along with some additional parameters used by $\mathsf{SampG}, \mathsf{SampH}$;
- given PP, we know $\mathrm{ord}(\mathbb{H})$ (i.e. the order of the group, which is independent of n) and can uniformly sample from $\mathbb{H}$;
- SP contains $h^* \in \mathbb{H}$ (where $h^* \neq 1$), along with some additional parameters used by $\widehat{\mathsf{SampG}}$;

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

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

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

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

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

3.2.2 Correctness.

The requirements for correctness are as follows:

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

(associative.) For all

$$(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP}),\ (h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathrm{PP}),$$

and for all $i = 1, \ldots, n$, we have $e(g_0, h_i) = e(g_i, h_0)$.

3.2.3 Security.

The requirements for security are as follows (we defer a discussion to the end of this section):

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

(non-degeneracy.) With probability $1 - 2^{-\Omega(\lambda)}$ over $\hat{g}_0 \leftarrow \widehat{\mathsf{SampG}}_0(\mathrm{PP}, \mathrm{SP})$, we have that $e(\hat{g}_0, h^*)^\alpha$ is identically distributed to the uniform distribution over $\mathbb{G}_T$, where $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$.

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

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

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

where

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

For any $\mathbf{g} = (g_0, \ldots, g_n) \in \mathbb{G}^{n+1}$, and any $i \in [n]$, we use $\mathbf{g}_{-i}$ to denote $(g_0, \ldots, g_{i-1}, g_{i+1}, \ldots, g_n) \in \mathbb{G}^n$.

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

$$\begin{aligned}\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{A}}(\lambda, q) \quad := \quad & \max_{i\in[n]} |\Pr[\,\mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}^1, \ldots, \mathbf{h}^q}) = 1\,] \\ & - \Pr[\,\mathcal{A}(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-i}, \boxed{\mathbf{h}'^1, \ldots, \mathbf{h}'^q}) = 1\,]|\end{aligned}$$

where

$$\begin{aligned}&(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\\ &\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP});\\ &\mathbf{h}^j := (h_{0,j}, h_{1,j}, \ldots, \boxed{h_{i,j}}, \ldots, h_{n,j}) \leftarrow \mathsf{SampH}(\mathrm{PP}),\ j = 1, \ldots, q;\\ &\mathbf{h}'^j := (h_{0,j}, h_{1,j}, \ldots, \boxed{h_{i,j} \cdot (h^*)^{\gamma_j}}, \ldots, h_{n,j}),\ \gamma_j \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})},\ j = 1, \ldots, q.\end{aligned}$$

Discussion.

We provide additional justification and discussion on the preceding security properties.

Remark 3.1 (orthogonality)**.** *We may deduce from $\mu(h^*) = 1$ that $e(g_0, h^*) = 1$ for all $g_0 = \mathsf{SampG}_0(\mathrm{PP}; s)$: for all $\gamma \in \{0, 1\}$,*

$$\begin{aligned} e(g_0, (h^*)^\gamma) &= \mathsf{SampGT}(\mu((h^*)^\gamma); s) && \textit{(by} \text{ projective}\textit{)} \\ &= \mathsf{SampGT}(\mu(h^*)^\gamma; s) && \textit{(by linearity of } \mu\textit{)} \\ &= \mathsf{SampGT}(1; s) && \textit{(by} \text{ orthogonality}\textit{)} \end{aligned}$$

Thus, we have $e(g_0, h^*) = e(g_0, 1) = 1$. For the instantiation from composite-order groups, h^* is orthogonal to each element in the output of SampG*, that is,*

$$e(g_0, h^*) = e(g_1, h^*) = \cdots = e(g_n, h^*) = 1$$

for all $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathrm{PP})$. On the other hand, for the instantiation from prime-order groups, h^* is in general not orthogonal to $g_1, \ldots, g_n$.

Remark 3.2 ($\mathbb{H}$-subgroup)**.** *We rely on $\mathbb{H}$-subgroup to re-randomize the secret keys in the proof of security for queries that share the same i-bit prefix; see Section 4.4 case 3.*

Remark 3.3 (indistinguishability)**.** *Observe that in left subgroup indistinguishability, the distinguisher does not get h^*; otherwise, it is possible to distinguish between the two distributions using orthogonality. It is also crucial that for nested-hiding, the distinguisher gets $\hat{\mathbf{g}}_{-i}$ and not $\hat{\mathbf{g}} := (\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_n)$. (Looking ahead to the proof in Section 4.4, not having $\hat{\mathbf{g}}$ means that the simulator cannot generate ciphertexts to distinguish between Type $i-1$ and Type i secret keys.) Otherwise, given $\hat{g}_i$, it is possible to distinguish between $\mathbf{h}^j$ and $\mathbf{h}'^j$ by using the relation:*

$$e(g_0 \cdot \hat{g}_0, h_{i,j}) = e(g_i \cdot \hat{g}_i, h_{0,j}).$$

This relation follows from associative and left subgroup indistinguishability.

4 (Almost) Tight IBE from Nested Dual System Groups

We provide a construction of an IBE scheme from nested dual system groups where the ciphertext comprises two group elements in $\mathbb{G}$ and one in $\mathbb{G}_T$.

4.0.4 Overview.

We begin with an informal overview of the scheme. Fix a bilinear group with a pairing $e : G \times G \to G_T$. The starting point of our scheme is the following variant of Waters' IBE [145] with identity space $\{0,1\}^n$:

$$\begin{aligned} \text{MPK} &:= (g, u_1, \ldots, u_{2n}, e(g,g)^{\alpha}) \\ \text{CT}_{\mathbf{x}} &:= (g^s, (\prod_{k=1}^{n} u_{2k-x_k})^s, e(g,g)^{\alpha s} \cdot m) \\ \text{SK}_{\mathbf{y}} &:= (g^r, \text{MSK} \cdot (\prod_{k=1}^{n} u_{2k-y_k})^r) \end{aligned}$$

Note that MPK contains $2n+1$ group elements in G, which we will generate using $\mathsf{SampP}(1^\lambda, \boxed{1^{2n}})$. We will use $\mathsf{SampG}(\text{PP})$ to generate the terms $(g^s, u_1^s, \ldots, u_{2n}^s)$ in the ciphertext, and $\mathsf{SampH}(\text{PP})$ to generate the terms $(g^r, u_1^r, \ldots, u_{2n}^r)$ in the secret key.

4.1 Construction

Let $\{0,1\}^n$ be the identity space.

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

  $$(\text{PP}, \text{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^{2n}).$$

  Pick $\text{MSK} \leftarrow_R \mathbb{H}$ and output the master public and secret key pair

  $$\text{MPK} := \big(\text{PP},\ \mu(\text{MSK})\big) \quad \text{and} \quad \text{MSK}.$$

- $\mathsf{Enc}(\text{MPK}, \mathbf{x}, m)$: On input an identity $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$ and $m \in \mathbb{G}_T$, sample

  $$(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\text{PP}; s),\ g_T' \leftarrow \mathsf{SampGT}(\mu(\text{MSK}); s)$$

  and output

  $$\text{CT}_{\mathbf{x}} := \big(\, C_0 := g_0,\ C_1 := g_{2-x_1} \cdots g_{2n-x_n},\ C_2 := g_T' \cdot m \,\big) \in (\mathbb{G})^2 \times \mathbb{G}_T.$$

- $\mathsf{KeyGen}(\text{MPK}, \text{MSK}, \mathbf{y})$: On input an identity $\mathbf{y} \in \{0,1\}^n$, sample

  $$(h_0, h_1, \ldots, h_{2n}) \leftarrow \mathsf{SampH}(\text{PP})$$

  and output

  $$\text{SK}_{\mathbf{y}} := \big(\, K_0 := h_0,\ K_1 := \text{MSK} \cdot h_{2-y_1} \cdots h_{2n-y_n} \,\big) \in (\mathbb{H})^2.$$

- $\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{x} = \mathbf{y}$, compute

$$e(g_0, \mathrm{MSK}) \leftarrow e(C_0, K_1)/e(C_1, K_0)$$

and recover the message as

$$m \leftarrow C_2 \cdot e(g_0, \mathrm{MSK})^{-1} \in \mathbb{G}_T.$$

Correctness.

Fix $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$, observe that

$$\begin{aligned}
&e(C_0, K_1)/e(C_1, K_0) \\
&= e(g_0, \mathrm{MSK} \cdot h_{2-x_1} \cdots h_{2n-x_n}) \cdot e(g_{2-x_1} \cdots g_{2n-x_n}, h_0)^{-1} \\
&= e(g_0, \mathrm{MSK}) \cdot \Big(e(g_0, h_{2-x_1}) \cdots e(g_0, h_{2n-x_n})\Big) \cdot \Big(e(g_{2-x_1}, h_0) \cdots e(g_{2n-x_n}, h_0)\Big)^{-1} \\
&= e(g_0, \mathrm{MSK})
\end{aligned}$$

where the last equality relies on *associative*, namely, $e(g_0, h_{2i-x_i}) = e(g_{2i-x_i}, h_0)$. In addition, by *projective*, we have $g'_T = e(g_0, \mathrm{MSK})$. Correctness follows readily.

4.2 Proof of Security

We prove the following theorem:

Theorem 4.1. *Under the left subgroup and nested-hiding indistinguishability (described in Section 3) and the additional requirement that* $\mathsf{ord}(\mathbb{H})$ *is prime, our IBE scheme in Section 4.1 is fully secure (in the sense of Definition 2.1). More precisely, for any adversary* $\mathcal{A}$ *that makes at most* q *key queries against the IBE scheme, there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2$ *such that:*

$$\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda) + 2n \cdot \mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q) + 2^{-\Omega(\lambda)}$$

and

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

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

Remark 4.1. *In our instantiations of nested dual system groups, the quantity* $\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q)$ *will be related to the advantage function corresponding to some static assumption, with a constant overhead independent of* q*. Putting the two together, this means that* $\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda)$ *is independent of* q*, as stated in Theorem 1.1.*

The proof follows via a series of games, summarized in Figure 12. To describe the games, we must first define semi-functional keys and ciphertexts. Following [51, 148], we first define two auxiliary algorithms, and define the semi-functional distributions via these auxiliary algorithms.

4.2.1 Auxiliary algorithms.

We consider the following algorithms:

$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m; \mathrm{MSK}', \mathbf{t})$: On input $\mathbf{x} := (x_1, \ldots, x_n) \in \{0,1\}^n$, $m \in \mathbb{G}_T$, $\mathrm{MSK}' \in \mathbb{H}$, and $\mathbf{t} := (T_0, T_1, \ldots, T_{2n}) \in \mathbb{G}^{2n+1}$, output

$$\mathrm{CT}_{\mathbf{x}} := \left(T_0,\ \prod_{k=1}^{n} T_{2k-x_k},\ e(T_0, \mathrm{MSK}') \cdot m \right).$$

$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}', \mathbf{y}; \mathbf{t})$: On input $\mathrm{MSK}' \in \mathbb{H}$, $\mathbf{y} := (y_1, \ldots, y_n) \in \{0,1\}^n$, and $\mathbf{t} := (T_0, T_1, \ldots, T_{2n}) \in \mathbb{H}^{2n+1}$, output

$$\mathrm{SK}_{\mathbf{y}} := \left(T_0,\ \mathrm{MSK}' \cdot \prod_{k=1}^{n} T_{2k-y_k} \right).$$

4.2.2 Auxiliary distributions.

For $i = 0, 1, \ldots, n$, we pick a random function $R_i : \{0,1\}^i \to \langle h^* \rangle$ (we use $\{0,1\}^0$ to denote the singleton set containing just the empty string ε). More concretely, given (PP, h^*), we sample the function R_i by first choosing a random function $R_i' : \{0,1\}^i \to \mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$ (via lazy sampling), and define $R_i(x) := (h^*)^{R_i'(x)}$ for all $x \in \{0,1\}^i$.

Pseudo-normal ciphertext.

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m; \mathrm{MSK}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}),$$

where $\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $\boxed{\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})}$; we can also write this distribution more explicitly as

$$\left(g_0 \cdot \hat{g}_0,\ \prod_{k=1}^{n} (g_{2k-x_k} \cdot \hat{g}_{2k-x_k}),\ e(g_0 \cdot \hat{g}_0, \mathrm{MSK}) \cdot m \right),$$

where $(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$.

Semi-functional ciphertext type i (for $i = 0, 1, \ldots, n$).

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}, m; \boxed{\mathrm{MSK} \cdot R_i(\mathbf{x}|_i)}, \mathbf{g} \cdot \hat{\mathbf{g}}),$$

where $\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $\hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$ and $\mathbf{x}|_i$ denotes the i-bit prefix of $\mathbf{x}$; we can also write this distribution more explicitly as

$$\left(g_0 \cdot \hat{g}_0,\ \prod_{k=1}^{n} (g_{2k-x_k} \cdot \hat{g}_{2k-x_k}),\ e(g_0 \cdot \hat{g}_0, \mathrm{MSK} \cdot R_i(\mathbf{x}|_i)) \cdot m \right),$$

where $(g_0, g_1, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(\mathrm{PP})$ and $(\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP})$.

Semi-functional secret key type i (for $i = 0, 1, \ldots, n$).

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot R_i(\mathbf{y}|_i)}, \mathbf{y}; \mathbf{h}),$$

Game	Ciphertext $\mathrm{CT}_{\mathbf{x}^*}$	Secret Key $\mathrm{SK}_{\mathbf{y}}$
0	$\mathsf{Enc}(\mathrm{MPK}, \mathbf{x}^*, m_\beta)$ $(g_0, \prod g_{2k-x_k}, e(g_0, \mathrm{MSK}) \cdot m_\beta)$	$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$ $(h_0, \mathrm{MSK} \cdot \prod h_{2k-y_k})$
1	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}})$ $(g_0\hat{g}_0, \prod (g_{2k-x_k}\hat{g}_{2k-x_k}), e(g_0\hat{g}_0, \mathrm{MSK}) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathbf{h})$ $(—, —)$
2,i	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \boxed{\mathrm{MSK} \cdot R_i(\mathbf{x}^*\vert_i)}, \mathbf{g} \cdot \hat{\mathbf{g}})$ $(—, —, e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_i(\mathbf{x}^*\vert_i)) \cdot m_\beta)$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \boxed{\mathrm{MSK} \cdot R_i(\mathbf{y}\vert_i)}, \mathbf{y}; \mathbf{h})$ $(—, \mathrm{MSK} \cdot R_i(\mathbf{y}\vert_i) \cdot \prod h_{2k-y_k})$
3	$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, \boxed{\texttt{random}}; \mathrm{MSK} \cdot R_n(\mathbf{x}^*), \mathbf{g} \cdot \hat{\mathbf{g}})$ $(—, —, e(g_0\hat{g}_0, \mathrm{MSK} \cdot R_n(\mathbf{x}^*)) \cdot \texttt{random})$	$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot R_n(\mathbf{y}), \mathbf{y}; \mathbf{h})$ $(—, \mathrm{MSK} \cdot R_n(\mathbf{y}) \cdot \prod h_{2k-y_k})$

Figure 12: Sequence of games, where we drew a box to highlight the differences between each game and the preceding one, a dash (—) means the same as in the previous game. Recall that $R_i : \{0,1\}^i \to \langle h^* \rangle$ is a random function. Here, the product Π denotes $\Pi_{k=1}^n$. We transition from Game_0 to Game_1 and from $\mathsf{Game}_{2,i-1}$ to $\mathsf{Game}_{2,i}$ using a computational argument via left subgroup and nested-hiding respectively; for the remaining transitions, we use a statistical argument via orthogonality and non-degeneracy.

where a fresh $\mathbf{h} \leftarrow \mathsf{SampH}(\mathrm{PP})$ is chosen for each secret key; we can also write this distribution more explicitly as

$$\left(h_0,\ \mathrm{MSK} \cdot R_i(\mathbf{x}\vert_i) \cdot \prod_{k=1}^{n} h_{2k-y_k} \right)$$

where $(h_0, h_1, \ldots, h_{2n}) \leftarrow \mathsf{SampH}(\mathrm{PP})$.

Remark 4.2 (decryption capabilities)**.** *As noted in the introduction, decryption capabilities remain the same through the hybrid games. Observe that a type i secret key for $\mathbf{x}^*$ can decrypt a type i ciphertext for $\mathbf{x}^*$ since they share $R_i(\mathbf{x}^*\vert_i)$. In addition, a type i secret key for $\mathbf{x}^*$ can decrypt a normal ciphertext for $\mathbf{x}^*$ because $e(g_0, R_i(\mathbf{x}^*\vert_i)) = 1$, which follows readily from $R_i(\mathbf{x}^*\vert_i) \in \langle h^* \rangle$ and $e(g_0, h^*) = 1$ (see Remark 3.1).*

Game sequence.

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

- Game_0: is the real security game (c.f. Section 2).
- Game_1: is the same as Game_0 except that the challenge ciphertext is pseudo-normal.
- $\mathsf{Game}_{2,i}$ for i from 0 to n, $\mathsf{Game}_{2,i}$ is the same as Game_1 except that the challenge ciphertext and all secret keys are of type i.
- Game_3: is the same as $\mathsf{Game}_{2,n}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.

In Game_3, the view of the adversary is statistically independent of the challenge bit β. Hence, $\mathsf{Adv}_3(\lambda) = 0$. We complete the proof by establishing the following sequence of lemmas.

4.3 Normal to Pseudo-Normal to Type 0

Lemma 4.2 (Game_0 to Game_1)**.** *For any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_1$ such that:*

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

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

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

$$(\mathrm{PP}, \mathbf{t}),$$

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

$$\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP}),\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP}),$$

and proceeds as follows:

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

$$\mathrm{MPK} := \left(\mathrm{PP},\ \mu(\mathrm{MSK})\right).$$

Key Queries. On input the j'th secret key query $\mathbf{y}$, output

$$\mathrm{SK}_{\mathbf{y}} \leftarrow \widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK}, \mathbf{y}; \mathsf{SampH}(\mathrm{PP})).$$

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

$$\mathrm{CT}_{\mathbf{x}^*} \leftarrow \widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK}, \mathbf{t}).$$

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

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

□

Lemma 4.3 (Game_1 to $\mathsf{Game}_{2,0}$)**.** *For any adversary $\mathcal{A}$,*

$$\mathsf{Adv}_1(\lambda) = \mathsf{Adv}_{2,0}(\lambda)$$

Proof. Observe that MSK and $\mathrm{MSK} \cdot R_0(\varepsilon)$ (where $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$) are identically distributed, so we may replace MSK in Game_1 by $\mathrm{MSK} \cdot R_0(\varepsilon)$. The resulting distribution is identically distributed to that in $\mathsf{Game}_{2,0}$ except we use $\mu(\mathrm{MSK} \cdot R_0(\varepsilon))$ instead of $\mu(\mathrm{MSK})$ in MPK. Now, by *orthogonality*, these two quantities are in fact equal. □ □

4.4 Type $i-1$ to Type i

We begin with an informal overview of our proof strategy. For simplicity, suppose the adversary only requests secret keys for two identities $\mathbf{y}_0$ and $\mathbf{y}_1$ that differ only in the i'th bit, that is,

$$\mathbf{y}_0 = (y_1, \ldots, y_{i-1}, \boxed{0}, y_{i+1}, \ldots, y_n) \quad \text{and} \quad \mathbf{y}_1 = (y_1, \ldots, y_{i-1}, \boxed{1}, y_{i+1}, \ldots, y_n)$$

Recall that Type $i-1$ secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are of the form:

$$\mathrm{SK}_{\mathbf{y}_0} = \left(h_0, \mathrm{MSK} \cdot \boxed{R_{i-1}(y_1, \ldots, y_{i-1})} \cdot h_{2-y_1} \cdots \boxed{h_{2i}} \cdots h_{2n-y_n} \right) \quad \text{and}$$
$$\mathrm{SK}_{\mathbf{y}_1} = \left(h_0, \mathrm{MSK} \cdot \boxed{R_{i-1}(y_1, \ldots, y_{i-1})} \cdot h_{2-y_1} \cdots \boxed{h_{2i-1}} \cdots h_{2n-y_n} \right)$$

whereas Type i secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are of the form:

$$\mathrm{SK}_{\mathbf{y}_0} = \left(h_0, \mathrm{MSK} \cdot \boxed{R_i(y_1, \ldots, y_{i-1}, 0)} \cdot h_{2-y_1} \cdots \boxed{h_{2i}} \cdots h_{2n-y_n} \right) \quad \text{and}$$
$$\mathrm{SK}_{\mathbf{y}_1} = \left(h_0, \mathrm{MSK} \cdot \boxed{R_i(y_1, \ldots, y_{i-1}, 1)} \cdot h_{2-y_1} \cdots \boxed{h_{2i-1}} \cdots h_{2n-y_n} \right)$$

In order to show that Type $i-1$ and Type i secret keys for $\mathbf{y}_0$ and $\mathbf{y}_1$ are indistinguishable, it suffices to show that

$$(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i}, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1}) \quad \text{and}$$
$$(R_i(y_1, \ldots, y_{i-1}, 0) \cdot h_{2i}, R_i(y_1, \ldots, y_{i-1}, 1) \cdot h_{2i-1})$$

are computationally indistinguishable (*).

Now, suppose for simplicity that the i'th bit of the identity $\mathbf{x}^*$ for challenge ciphertext is 1. Then, *nested-hiding indistinguishability* with index $2i$ tells us that

$$h_{2i} \quad \text{and} \quad h_{2i} \cdot (h^*)^\gamma$$

are computationally indistinguishable, where $\gamma \leftarrow_R \mathbb{Z}_{|\mathbb{H}|}$. Moreover, this holds even if the distinguisher is given $\hat{\mathbf{g}}_{-2i}$, which we will need to simulate the semi-functional ciphertext for $\mathbf{x}^*$. (On the other hand, given only $\hat{\mathbf{g}}_{-2i}$, we cannot simulate semi-functional ciphertext for identities whose i'th bit is 0.) This means that

$$(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i}, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1}) \quad \text{and}$$
$$(R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i} \cdot (h^*)^\gamma, R_{i-1}(y_1, \ldots, y_{i-1}) \cdot h_{2i-1})$$

are computationally indistinguishable, even given the semi-functional ciphertext for $\mathbf{x}^*$.

To achieve (*), we can then implicitly set:

$$R_i(y_1, \ldots, y_{i-1}, 0) := R_{i-1}(y_1, \ldots, y_{i-1}) \cdot (h^*)^\gamma \quad \text{and}$$
$$R_i(y_1, \ldots, y_{i-1}, 1) := R_{i-1}(y_1, \ldots, y_{i-1})$$

This corresponds to Case 2 and Case 1 below respectively.

More generally, we guess at random the i'th bit of $\mathbf{x}^*$ to be b_i and use nested-hiding indistinguishability with index $2i - \overline{b_i}$. In addition, we need to handle q keys and not just two keys, along with an additional complication arising from the fact that multiple queries may share the same i-bit prefix (see Case 3 below).

Lemma 4.4 ($\mathsf{Game}_{2,i-1}$ to $\mathsf{Game}_{2,i}$)**.** *For $i = 1, \ldots, n$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_2$ such that:*

$$|\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)| \leq 2\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q),$$

and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + q \cdot poly(\lambda, n)$ *where* $poly(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. On input $i \in [n]$, $\mathcal{B}_2$ picks a random bit $b_i \leftarrow_{\mathrm{R}} \{0,1\}$ (that is, it guesses the i'th bit of the challenge identity $\mathbf{x}^*$) and requests nested-hiding instantiation for index $2i - \overline{b_i}$. The adversary $\mathcal{B}_2$ gets as input

$$\left(\mathrm{PP}, h^*, \hat{\mathbf{g}}_{-(2i-\overline{b_i})}, \mathbf{t}_1, \ldots, \mathbf{t}_q\right),$$

where $(\mathbf{t}^1, \ldots, \mathbf{t}^q)$ is either $(\mathbf{h}^1, \ldots, \mathbf{h}^q)$ or $(\mathbf{h}'^1, \ldots, \mathbf{h}'^q)$ and

$$\mathbf{h}^j := (h_{0,j}, h_{1,j}, \ldots, h_{2n,j}) \leftarrow \mathsf{SampH}(\mathrm{PP}),$$
$$\mathbf{h}'^j := (h_{0,j}, h_{1,j}, \ldots, h_{2i-\overline{b_i},j} \cdot (h^*)^{\gamma_j}, \ldots, h_{2n,j}),$$

and proceeds as follows:

Setup. Pick $\mathrm{MSK} \leftarrow_{\mathrm{R}} \mathbb{H}$, and output

$$\mathrm{MPK} := \left(\mathrm{PP},\ \mu(\mathrm{MSK})\right).$$

Programming R_{i-1}, R_i. Pick a random function $\tilde{R}_{i-1} : \{0,1\}^{i-1} \to \langle h^* \rangle$ (which we use to program R_{i-1}, R_i). Recall that we can sample a uniformly random element in $\langle h^* \rangle$ by raising h^* to a uniformly random exponent in $\mathbb{Z}_{\mathrm{ord}(\mathbb{H})}$. For all prefixes $\mathbf{x}' \in \{0,1\}^{i-1}$, we implicitly set

$$R_i(\mathbf{x}' \| b_i) := \tilde{R}_{i-1}(\mathbf{x}') \quad \text{and} \quad R_{i-1}(\mathbf{x}') := \tilde{R}_{i-1}(\mathbf{x}').$$

(We set $R_i(\mathbf{x}' \| \overline{b_i})$ later.) This means that for any $\mathbf{x} = (x_1, \ldots, x_n)$ such that $x_i = b_i$, we have:

$$R_i(\mathbf{x}|_i) = R_{i-1}(\mathbf{x}|_{i-1}) = \tilde{R}_{i-1}(\mathbf{x}|_{i-1}).$$

Key Queries. On input the j'th secret key query $\mathbf{y} = (\mathbf{y}|_{i-1}, y_i, \ldots, y_n)$, we consider three cases:

- Case 1: $y_i = b_i$. Here, $\mathcal{B}_2$ can compute

  $$R_i(\mathbf{y}|_i) = R_{i-1}(\mathbf{y}|_{i-1}) = \tilde{R}_{i-1}(\mathbf{y}|_{i-1})$$

  and simply outputs

  $$\widetilde{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \tilde{\mathbf{h}}^j),$$

  where $\tilde{\mathbf{h}}^j \leftarrow \mathsf{SampH}(\mathrm{PP})$.

- Case 2: $y_i = \overline{b_i}$ and $R_i(\mathbf{y}|_i)$ has not been previously set. Here, we implicitly set

  $$R_i(\mathbf{y}|_{i-1} \| \overline{b_i}) := \tilde{R}_{i-1}(\mathbf{y}|_{i-1}) \cdot (h^*)^{\gamma_j},$$

  where γ_j is as defined in the nested-hiding instantiation. Observe that this is the correct distribution since $R_i(\mathbf{y}|_{i-1} \| b_i)$ and $R_i(\mathbf{y}|_{i-1} \| \overline{b_i})$ are two independently random values. Then

$\mathcal{B}_2$ outputs:

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \mathbf{t}^j).$$

- Case 3: $y_i = \overline{b_i}$ and $R_i(\mathbf{y}|_i)$ has been previously set. Let j' be the index of key query in which we set $R_i(\mathbf{y}|_i)$, recall that

$$R_i(\mathbf{y}|_{i-1} \| \overline{b_i}) := \tilde{R}_{i-1}(\mathbf{y}|_{i-1}) \cdot (h^*)^{\gamma_{j'}}.$$

Then $\mathcal{B}_2$ outputs:

$$\widehat{\mathsf{KeyGen}}(\mathrm{PP}, \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{y}|_{i-1}), \mathbf{y}; \mathbf{t}^{j'} \cdot \tilde{\mathbf{h}}^j).$$

where $\tilde{\mathbf{h}}^j \leftarrow \mathsf{SampH}(\mathrm{PP})$. Here, we rely on the $\mathbb{H}$-*subgroup* property to re-randomize $\mathbf{t}^{j'}$.

Ciphertext. Upon receiving a challenge identity $\mathbf{x}^* := (x_1^*, \ldots, x_n^*)$ and two equal length messages m_0, m_1 from $\mathcal{A}$, output a random bit and halt if $x_i^* \neq b_i$. Observe that up to the point when $\mathcal{A}$ submits $\mathbf{x}^*$, its view is statistically independent of b_i. Therefore, the probability that we halt is exactly $1/2$. Suppose that we do not halt, which means we have $x_i^* = b_i$. Hence, $\mathcal{B}_2$ knows

$$R_i(\mathbf{x}^*|_i) = R_{i-1}(\mathbf{x}^*|_{i-1}) = \tilde{R}_{i-1}(\mathbf{x}^*|_{i-1}).$$

Then, $\mathcal{B}_2$ picks $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and outputs the semi-functional challenge ciphertext as:

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK} \cdot \tilde{R}_{i-1}(\mathbf{x}^*|_{i-1}), \mathbf{g} \cdot \hat{\mathbf{g}}),$$

Here, $\mathcal{B}_2$ picks $\mathbf{g} \leftarrow \mathsf{SampG}(\mathrm{PP})$, whereas $\mathbf{g}$ is as defined in the nested-hiding instantiation. Observe that $\mathcal{B}_2$ can compute the output of $\widehat{\mathsf{Enc}}$ using just $\hat{\mathbf{g}}_{-(2i-\overline{b_i})}$ since since $x_i^* = b_i$.

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

Suppose $x_i^* = b_i$. Then, when $(\mathbf{t}^1, \ldots, \mathbf{t}^q) = (\mathbf{h}^1, \ldots, \mathbf{h}^q)$, the output is identical to that in $\mathsf{Game}_{2,i-1}$; and when $(\mathbf{t}^1, \ldots, \mathbf{t}^q) = (\mathbf{h}'^1, \ldots, \mathbf{h}'^q)$, the output is identical to that in $\mathsf{Game}_{2,i}$. Hence,

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q) \\
&= \Big| \Pr[x_i^* \neq b_i] \cdot 0 + \Pr[x_i^* = b_i] \\
&\quad \cdot (\Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i-1}] - \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i}]) \Big| \\
&= 1/2 \cdot \Big| \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i-1}] - \Pr[\mathcal{A} \text{ outputs } \beta' = \beta \text{ in } \mathsf{Game}_{2,i}] \Big| \\
&\geq 1/2 \cdot |\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)|.
\end{aligned}$$

We may therefore conclude that $|\mathsf{Adv}_{2,i-1}(\lambda) - \mathsf{Adv}_{2,i}(\lambda)| \leq 2\mathsf{Adv}^{\mathrm{NS}}_{\mathcal{B}_2}(\lambda, q)$. □

□

4.5 Final Transition

Lemma 4.5 ($\mathsf{Game}_{2,n}$ to Game_3)**.** *For any adversary $\mathcal{A}$:*

$$|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \leq 2^{-\Omega(\lambda)}.$$

Proof. Observe that the challenge ciphertext in $\mathsf{Game}_{2,n}$ is given by:

$$\widehat{\mathsf{Enc}}(\mathrm{PP}, \mathbf{x}^*, m_\beta; \mathrm{MSK} \cdot R_n(\mathbf{x}^*), \mathbf{g} \cdot \hat{\mathbf{g}}) = (C_0, C_1, C_2' \cdot m_\beta),$$

where (C_0, C_1) depend only on $\mathbf{g} \cdot \hat{\mathbf{g}} = (g_0 \cdot \hat{g}_0, \ldots)$, and C_2' is given by:

$$C_2' = e(g_0 \cdot \hat{g}_0, \mathrm{MSK} \cdot R_n(\mathbf{x}^*)) = e(g_0 \cdot \hat{g}_0, \mathrm{MSK}) \cdot \boxed{e(\hat{g}_0, R_n(\mathbf{x}^*))},$$

where in the last equality, we use the fact that $e(g_0, R_n(\mathbf{x}^*)) = 1$ (see Remarks 3.1 and 4.2). In addition, MPK and all of the secret key queries reveal no information about $R_n(\mathbf{x}^*)$. Then, by *non-degeneracy*, with probability $1 - 2^{-\Omega(\lambda)}$ over $\hat{g}_0$, we have $e(\hat{g}_0, R_n(\mathbf{x}^*))$ is uniformly distributed over $\mathbb{G}_T$. This implies that the challenge ciphertext is identically distributed to a semi-functional encryption of a random message in $\mathbb{G}_T$, as in Game_3. We may then conclude that: $|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \le 2^{-\Omega(\lambda)}$. □

Remark 4.3. *In our composite-order instantiation, we only have the weaker guarantee that $e(\hat{g}_0, R_n(\mathbf{x}^*))$ has at least 2λ bits of min-entropy, instead of being uniform over $\mathbb{G}_T$. We will modify the IBE scheme as follows: the message space is now $\{0,1\}^\lambda$, and we replace the term $g_T' \cdot m$ in the ciphertext with:*

$$\mathsf{H}(g_T') \oplus m,$$

where $\mathsf{H} : \mathbb{G}_T \to \{0,1\}^\lambda$ is a pairwise independent hash function. By the left-over hash lemma, we still have $|\mathsf{Adv}_{2,n}(\lambda) - \mathsf{Adv}_3(\lambda)| \le 2^{-\Omega(\lambda)}$.

□

5 Concrete (almost) tight IBE scheme from d-LIN in prime-order groups

In this section, we provide a self-contained description of the (almost) tight IBE scheme in [49] under d-LIN assumption in prime-order bilinear groups (G_1, G_2, G_T, e). Recall that $\pi_L : \mathbb{Z}_p^{2d \times 2d} \to \mathbb{Z}_p^{2d \times d}$ is the projection map that maps a $2d \times 2d$ matrix to the left d columns.

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

$$\mathbf{B}, \mathbf{B}^*, \mathbf{R} \leftarrow_{\mathrm{R}} \mathsf{GL}_{2d}(\mathbb{Z}_p),\ \mathbf{A}_1, \ldots, \mathbf{A}_{2n} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(2d)\times(2d)},\ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d}$$

such that $\mathbf{B}^\top \mathbf{B}^* = \mathbf{I}$, and output the master public and secret key pair

$$\begin{aligned}
\mathrm{MPK} &:= \left(g_1^{\pi_L(\mathbf{B})}, g_1^{\pi_L(\mathbf{B}\mathbf{A}_1)}, \ldots, g_1^{\pi_L(\mathbf{B}\mathbf{A}_{2n})}; e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})} \right) \\
&\in (G_1^{2d \times d})^{2n+1} \times G_T^d, \\
\mathrm{MSK} &:= \left(g_2^{\mathbf{k}}, g_2^{\mathbf{B}^*\mathbf{R}}, g_2^{\mathbf{B}^*\mathbf{A}_1^\top \mathbf{R}}, \ldots, g_2^{\mathbf{B}^*\mathbf{A}_{2n}^\top \mathbf{R}} \right) \in G_2^{2d} \times (G_2^{2d \times 2d})^{2n+1}.
\end{aligned}$$

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

$$\mathrm{CT}_{\mathbf{x}} := \left(\begin{array}{l} C_0 := g_1^{\pi_L(\mathbf{B})\mathbf{s}},\ C_1 := g_1^{\pi_L(\mathbf{B}(\mathbf{A}_{2-x_1} + \cdots + \mathbf{A}_{2n-x_n}))\mathbf{s}} \\ C_2 := e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \cdot m \end{array} \right) \in (G_1^{2d})^2 \times G_T.$$

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$: On input an identity vector $\mathbf{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2d}$ and output

$$\mathrm{SK}_{\mathbf{y}} := \left(K_0 := g_2^{\mathbf{B}^* \mathbf{R} \mathbf{r}},\ K_1 := g_2^{\mathbf{k} + \mathbf{B}^* (\mathbf{A}_{2-y_1} + \cdots + \mathbf{A}_{2n-y_n})^\top \mathbf{R} \mathbf{r}} \right) \in (G_2^{2d})^2.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{x} = \mathbf{y}$, compute

$$e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B}) \mathbf{s}} \leftarrow e(C_0, K_1) / e(C_1, K_0),$$

and recover the message as

$$m \leftarrow C_2 \cdot e(g_1, g_2)^{-\mathbf{k}^\top \pi_L(\mathbf{B}) \mathbf{s}} \in G_T.$$

6 Dual System Groups

6.0.1 Syntax.

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

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

- PP contains a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a linear map μ defined on $\mathbb{H}$, along with some additional parameters used by $\mathsf{SampG}, \mathsf{SampH}$;
- given PP, we know $\mathrm{ord}(\mathbb{H})$ (i.e. the order of the group, which is independent of n) and can uniformly sample from $\mathbb{H}$;
- SP contains $h^* \in \mathbb{H}$ (where $h^* \neq 1$), along with some additional parameters used by $\widehat{\mathsf{SampG}}$;

$\mathsf{SampGT} : \mathrm{Im}(\mu) \to \mathbb{G}_{\mathrm{T}}$.

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

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

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

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

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

6.0.2 Correctness.

The requirements for correctness are as follows:

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

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

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

6.0.3 Security.

The requirements for security are as follows:

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

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

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

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

where

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

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

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

where

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

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

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

where

$$(\mathrm{PP}, \mathrm{SP}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);$$
$$\hat{\mathbf{g}} = (\hat{g}_0, \ldots) \leftarrow \widehat{\mathsf{SampG}}(\mathrm{PP}, \mathrm{SP});$$
$$\hat{\mathbf{h}} = (\hat{h}_0, \ldots) \leftarrow \widehat{\mathsf{SampH}}(\mathrm{PP}, \mathrm{SP});$$
$$\gamma_1, \ldots, \gamma_n \leftarrow_{\mathrm{R}} \mathbb{Z}_{\mathrm{ord}(\mathbb{H})};$$
$$\hat{\mathbf{g}}' := (1, \hat{g}_0^{\gamma_1}, \ldots, \hat{g}_0^{\gamma_n}) \in \mathbb{G}^{n+1};$$
$$\hat{\mathbf{h}}' := (1, \hat{h}_0^{\gamma_1}, \ldots, \hat{h}_0^{\gamma_n}) \in \mathbb{H}^{n+1}.$$

7 Compact HIBE scheme from d-LIN in prime-order groups

In this section, we provide a self-contained description of our compact HIBE scheme (c.f. [51]) under d-LIN assumption in prime-order bilinear groups (G_1, G_2, G_T, e). Recall that $\pi_L : \mathbb{Z}_p^{(d+1)\times(d+1)} \to \mathbb{Z}_p^{(d+1)\times d}$ is the projection map that maps a $(d+1)\times(d+1)$ matrix to the left d columns.

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

$$\mathbf{B}, \mathbf{B}^*, \mathbf{R} \leftarrow_{\mathrm{R}} \mathsf{GL}_{d+1}(\mathbb{Z}_p),\ \mathbf{A}_1, \ldots, \mathbf{A}_{n+1} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(d+1)\times(d+1)},\ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{d+1}$$

such that $\mathbf{B}^\top \mathbf{B}^* = \mathbf{I}$ and $\mathbf{R}$ is a diagonal matrix whose bottom-right entry is 1, and output the master public and secret key pair

$$\mathrm{MPK} := \left(\begin{array}{cccc} g_1^{\pi_L(\mathbf{B})}, & g_1^{\pi_L(\mathbf{B}\mathbf{A}_1)}, & \ldots, & g_1^{\pi_L(\mathbf{B}\mathbf{A}_{n+1})} \\ g_2^{\pi_L(\mathbf{B}^*\mathbf{R})}, & g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_1^\top\mathbf{R})}, & \ldots, & g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_{n+1}^\top\mathbf{R})} \end{array} ; e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})} \right)$$
$$\in (G_1^{(d+1)\times d})^{n+2} \times (G_2^{(d+1)\times d})^{n+2} \times G_T^d$$

and

$$\mathrm{MSK} := g_2^{\mathbf{k}} \in G_2^{d+1}.$$

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

$$\mathrm{CT}_{\mathbf{x}} := \left(C_0 := g_1^{\pi_L(\mathbf{B})\mathbf{s}},\ C_1 := g_1^{\pi_L(\mathbf{B}(\mathbf{A}_{n+1} + x_1\mathbf{A}_1 + \cdots + x_\ell\mathbf{A}_\ell))\mathbf{s}},\ C_2 := e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \cdot m \right)$$
$$\in G_1^{d+1} \times G_1^{d+1} \times G_T.$$

$\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathbf{y})$: On input an identity vector $\mathbf{y} := (y_1, \ldots, y_\ell) \in \mathbb{Z}_p^\ell$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and output

$$\mathrm{SK}_{\mathbf{y}} := \left(\begin{array}{l} K_0 := g_2^{\pi_L(\mathbf{B}^*\mathbf{R})\mathbf{r}},\ K_1 := g_2^{\mathbf{k} + \pi_L(\mathbf{B}^*(\mathbf{A}_{n+1} + y_1\mathbf{A}_1 + \cdots + y_\ell\mathbf{A}_\ell)^\top\mathbf{R})\mathbf{r}} \\ K_{\ell+1} := g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_{\ell+1}^\top\mathbf{R})\mathbf{r}},\ \ldots,\ K_n := g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R})\mathbf{r}} \end{array} \right) \in (G_2^{d+1})^{n-\ell+2}.$$

$\mathsf{Dec}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathrm{CT}_{\mathbf{x}})$: If $\mathbf{y}$ is a prefix of $\mathbf{x}$, run

$$\mathrm{SK}_{\mathbf{x}} := (K_0, K_1, \ldots) \leftarrow \mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{x}).$$

Compute

$$e(g_1, g_2)^{\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \leftarrow e(C_0, K_1)/e(C_1, K_0),$$

and recover the message as

$$m \leftarrow C_2 \cdot e(g_1, g_2)^{-\mathbf{k}^\top \pi_L(\mathbf{B})\mathbf{s}} \in G_T.$$

$\mathsf{KeyDel}(\mathrm{MPK}, \mathrm{SK}_{\mathbf{y}}, \mathbf{y}')$: On input a secret key $\mathrm{SK}_{\mathbf{y}} := (K_0, K_1, K_{\ell+1}, \ldots, K_n)$ and an identity vector $\mathbf{y}' := (y_1, \ldots, y_{\ell'}) \in \mathbb{Z}_p^{\ell'}$, first compute

$$\widetilde{\mathrm{SK}}_{\mathbf{y}'} := \left(K_0,\ K_1 \cdot K_{\ell+1}^{y_{\ell+1}} \cdots K_{\ell'}^{y_{\ell'}},\ K_{\ell'+1},\ \ldots,\ K_n \right).$$

Then, pick $\mathbf{r}' \leftarrow_{\mathrm{R}} \mathbb{Z}_p^d$ and compute

$$\mathrm{SK}' := \left(g_2^{\pi_L(\mathbf{B}^*\mathbf{R})\mathbf{r}'},\ g_2^{\pi_L(\mathbf{B}^*(\mathbf{A}_{n+1}+y_1\mathbf{A}_1+\cdots+y_{\ell'}\mathbf{A}_{\ell'})^\top\mathbf{R})\mathbf{r}'},\ g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_{\ell'+1}^\top\mathbf{R})\mathbf{r}'},\ \ldots,\ g_2^{\pi_L(\mathbf{B}^*\mathbf{A}_n^\top\mathbf{R})\mathbf{r}'} \right).$$

Finally, output

$$\mathrm{SK}_{\mathbf{y}'} := \widetilde{\mathrm{SK}}_{\mathbf{y}'} \cdot \mathrm{SK}'$$

where $\cdot$ denotes entry-wise multiplication.

Part VI

Tightly CCA-Secure Encryption without Pairings

Romain Gay and Dennis Hofheinz and Eike Kiltz and Hoeteck Wee
EUROCRYPT 2016, **BEST PAPER AWARD**

Abstract. We present the first CCA-secure public-key encryption scheme based on DDH where the security loss is independent of the number of challenge ciphertexts and the number of decryption queries. Our construction extends also to the standard k-Lin assumption in pairing-free groups, whereas all prior constructions starting with Hofheinz and Jager (Crypto '12) rely on the use of pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half (to only 3 group elements under DDH), in addition to eliminating the use of pairings.

We also show how to use our techniques in the NIZK setting. Specifically, we construct the first tightly simulation-sound designated-verifier NIZK for linear languages without pairings. Using pairings, we can turn our construction into a highly optimized publicly verifiable NIZK with tight simulation-soundness.

1 Introduction

The most basic security guarantee we require of a public key encryption scheme is that of semantic security against chosen-plaintext attacks (CPA) [80]: it is infeasible to learn anything about the plaintext from the ciphertext. On the other hand, there is a general consensus within the cryptographic research community that in virtually every practical application, we require semantic security against adaptive chosen-ciphertext attacks (CCA) [134, 61], wherein an adversary is given access to decryptions of ciphertexts of her choice.

In this work, we focus on the issue of security reduction and security loss in the construction of CPA and CCA-secure public-key encryption from the DDH assumption. Suppose we have such a scheme along with a security reduction showing that attacking the scheme in time t with success probability ϵ implies breaking the DDH assumption in time roughly t with success probability ϵ/L; we refer to L as the security loss. In general, L would depend on the security parameter λ as well as the number of challenge ciphertexts Q_{enc} and the number decryption queries Q_{dec}, and we say that we have a *tight security reduction* if L depends only on the security parameter and is independent of both Q_{enc} and Q_{dec}. Note that for typical settings of parameters (e.g., $\lambda = 80$ and $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{20}$, or even $Q_{\mathsf{enc}}, Q_{\mathsf{dec}} \approx 2^{30}$ in truly large settings), λ is much smaller than Q_{enc} and Q_{dec}.

In the simpler setting of CPA-secure encryption, the ElGamal encryption scheme already has a tight security reduction to the DDH assumption [124, 22], thanks to random self-reducibility of DDH with a tight security reduction. In the case of CCA-secure encryption, the best result is still the seminal Cramer-Shoup encryption scheme [59], which achieves security loss Q_{enc}.[9] This raises the following open problem:

[9] We ignore contributions to the security loss that depend only on a statistical security parameter.

Does there exist a CCA-secure encryption scheme with a tight security reduction to the DDH assumption?

Hofheinz and Jager [96] gave an affirmative answer to this problem under stronger (and pairing-related) assumptions, notably the 2-Lin assumptions in bilinear groups, albeit with large ciphertexts and secret keys; a series of follow-up works [117, 119, 18, 83] leveraged techniques introduced in the context of tightly-secure IBE [48, 25, 99] to reduce the size of ciphertext and secret keys to a relatively small constant. However, all of these works rely crucially on the use of pairings, and seem to shed little insight on constructions under the standard DDH assumption; in fact, a pessimist may interpret the recent works as strong indication that the use of pairings is likely to be necessary for tightly CCA-secure encryption.

We may then restate the open problem as eliminating the use of pairings in these prior CCA-secure encryption schemes while still preserving a tight security reduction. From a theoretical standpoint, this is important because an affirmative answer would yield tightly CCA-secure encryption under qualitatively weaker assumptions, and in addition, shed insight into the broader question of whether tight security comes at the cost of qualitative stronger assumptions.

Eliminating the use of pairings is also important in practice as it allows us to instantiate the underlying assumption over a much larger class of groups that admit more efficient group operations and more compact representations, and also avoid the use of expensive pairing operations. Similarly, tight reductions matter in practice because as L increases, we should increase the size of the underlying groups in order to compensate for the security loss, which in turn increases the running time of the implementation. Note that the impact on performance is quite substantial, as exponentiation in a r-bit group takes time roughly $\mathcal{O}(r^3)$.

1.1 Our Results

We settle the main open problem affirmatively: we construct a tightly CCA-secure encryption scheme from the DDH assumption without pairings. Moreover, our construction improves upon the concrete efficiency of existing schemes, reducing the ciphertext overhead by about half, in addition to eliminating the use of pairings. We refer to Figure 14 for a comparison with prior works.

Overview of our construction. Fix an additively written group $\mathbb{G}$ of order q. We rely on implicit representation notation [62] for group elements: for a fixed generator P of $\mathbb{G}$ and for a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times t}$, we define $[\mathbf{M}] := \mathbf{M}P \in \mathbb{G}^{n\times t}$ where multiplication is done component-wise. We rely on the $\mathcal{D}_k$-MDDH Assumption [62], which stipulates that given $[\mathbf{M}]$ drawn from a matrix distribution $\mathcal{D}_k$ over $\mathbb{Z}_q^{(k+1)\times k}$, $[\mathbf{Mx}]$ is computationally indistinguishable from a uniform vector in $\mathbb{G}^k$; this is a generalization of the k-Lin Assumption.

We outline the construction under the k-Lin assumption over $\mathbb{G}$, of which the DDH assumption is a special case corresponding to $k = 1$.

In this overview, we will consider a weaker notion of security, namely tag-based KEM security against plaintext check attacks (PCA) [126]. In the PCA security experiment, the adversary gets no decryption oracle (as with CCA security), but a PCA oracle that takes as input a tag and a ciphertext/plaintext pair and checks whether the ciphertext decrypts to the plaintext. Furthermore, we restrict the adversary to only query the PCA oracle on tags different from those used in the challenge ciphertexts. PCA security

is strictly weaker than the CCA security we actually strive for, but allows us to present our solution in a clean and simple way. (We show how to obtain full CCA security separately.)

The starting point of our construction is the Cramer-Shoup KEM. The public key is given by $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_0], [\mathbf{M}^\top \mathbf{k}_1])$ for $\mathbf{M} \leftarrow_R \mathbb{Z}_q^{(k+1)\times k}$. Here, $\mathbf{M}$ corresponds to the matrix used in the k-Lin assumption. On input pk and a tag τ, the encryption algorithm outputs the ciphertext/plaintext pair

$$([\mathbf{y}], [z]) = ([\mathbf{M}\mathbf{x}], [\mathbf{x}^\top \mathbf{M}^\top \mathbf{k}_\tau]), \tag{4}$$

where $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ and $\mathbf{x} \leftarrow_R \mathbb{Z}_q^k$. Decryption relies on the fact that $\mathbf{y}^\top \mathbf{k}_\tau = \mathbf{x}^\top \mathbf{M}^\top \mathbf{k}_\tau$. The KEM is PCA-secure under k-Lin, with a security loss that depends on the number of ciphertexts Q (via a hybrid argument) but independent of the number of PCA queries [59, 3].

Following the "randomized Naor-Reingold" paradigm introduced by Chen and Wee on tightly secure IBE [48], our starting point is (4), where we replace $\mathbf{k}_\tau = \mathbf{k}_0 + \tau \mathbf{k}_1$ with

$$\mathbf{k}_\tau = \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$$

and $\mathsf{pk} := ([\mathbf{M}], [\mathbf{M}^\top \mathbf{k}_{j,b}]_{j=1,\ldots,\lambda,b=0,1})$, where $(\tau_1, \ldots, \tau_\lambda)$ denotes the binary representation of the tag $\tau \in \{0,1\}^\lambda$.

Following [48], we want to analyze this construction by a sequence of games in which we first replace $[\mathbf{y}]$ in the challenge ciphertexts by uniformly random group elements via random self-reducibility of MDDH (k-Lin), and then incrementally replace $\mathbf{k}_\tau$ in both the challenge ciphertexts and in the PCA oracle by $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, where RF is a truly random function and $\mathbf{m}^\perp$ is a random element from the kernel of $\mathbf{M}$, i.e., $\mathbf{M}^\top \mathbf{m}^\perp = 0$. Concretely, in Game i, we will replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)$ where RF_i is a random function on $\{0,1\}^i$ applied to the i-bit prefix of τ. We proceed to outline the two main ideas needed to carry out this transition. Looking ahead, note that once we reach Game λ, we would have replaced $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}(\tau)$, upon which security follows from a straight-forward information-theoretic argument (and the fact that ciphertexts and decryption queries carry pairwise different τ).

First idea. First, we show how to transition from Game i to Game $i+1$, under the restriction that the adversary is only allowed to query the encryption oracle on tags whose $i+1$-st bit is 0; we show how to remove this unreasonable restriction later. Here, we rely on an *information-theoretic* argument similar to that of Cramer and Shoup to increase the entropy from RF_i to RF_{i+1}. This is in contrast to prior works which rely on a computational argument; note that the latter requires encoding secret keys as group elements and thus a pairing to carry out decryption.

More precisely, we pick a random function RF'_i on $\{0,1\}^i$, and implicitly define RF_{i+1} as follows:

$$\mathsf{RF}_{i+1}(\tau) = \begin{cases} \mathsf{RF}_i(\tau) & \text{if } \tau_{i+1} = 0 \\ \mathsf{RF}'_i(\tau) & \text{if } \tau_{i+1} = 1 \end{cases}$$

Observe all of the challenge ciphertexts leak no information about RF'_i or $\mathbf{k}_{i+1,1}$ since they all correspond to tags whose $i+1$-st bit is 0. To handle a PCA query $(\tau, [\mathbf{y}], [z])$, we proceed via a case analysis:

- if $\tau_{i+1} = 0$, then $\mathbf{k}_\tau + \mathsf{RF}_{i+1}(\tau) = \mathbf{k}_\tau + \mathsf{RF}_i(\tau)$ and the PCA oracle returns the same value in both

Games i and $i+1$.

- if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies in the span of $\mathbf{M}$, we have

$$\mathbf{y}^\top \mathbf{m}^\perp = 0 \Longrightarrow \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_i(\tau)) = \mathbf{y}^\top(\mathbf{k}_\tau + \mathbf{m}^\perp \mathsf{RF}_{i+1}(\tau)),$$

  and again the PCA oracle returns the same value in both Games i and $i+1$.

- if $\tau_{i+1} = 1$ and $\mathbf{y}$ lies outside the span of $\mathbf{M}$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random given $\mathbf{M}, \mathbf{M}^\top \mathbf{k}_{i+1,1}$. (Here, we crucially use that the adversary does not query encryptions with $\tau_{i+1} = 1$, which ensures that the challenge ciphertexts do not leak additional information about $\mathbf{k}_{i+1,1}$.) This means that $\mathbf{y}^\top \mathbf{k}_\tau$ is uniformly random from the adversary's view-point, and therefore the PCA oracle will reject with high probability in both Games i and $i+1$. (At this point, we crucially rely on the fact that the PCA oracle only outputs a *single* check bit and not all of $\mathbf{k}_\tau + \mathsf{RF}(\tau)$.)

Via a hybrid argument, we may deduce that the distinguishing advantage between Games i and $i+1$ is at most Q/q where Q is the number of PCA queries.

Second idea. Next, we remove the restriction on the encryption queries using an idea of Hofheinz, Koch and Striecks [99] for tightly-secure IBE in the multi-ciphertext setting, and its instantiation in prime-order groups [83]. The idea is to create two "independent copies" of $(\mathbf{m}^\perp, \mathsf{RF}_i)$; we use one to handle encryption queries on tags whose $i+1$-st bit is 0, and the other to handle those whose $i+1$-st bit is 1. We call these two copies $(\mathbf{M}_0^*, \mathsf{RF}_i^{(0)})$ and $(\mathbf{M}_1^*, \mathsf{RF}_i^{(1)})$, where $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{0}$.

Concretely, we replace $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{(k+1)\times k}$ with $\mathbf{M} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k\times k}$. We decompose $\mathbb{Z}_q^{3k}$ into the span of the respective matrices $\mathbf{M}, \mathbf{M}_0, \mathbf{M}_1$, and we will also decompose the span of $\mathbf{M}^\perp \in \mathbb{Z}_q^{3k\times 2k}$ into that of $\mathbf{M}_0^*, \mathbf{M}_1^*$. Similarly, we decompose $\mathbf{M}^\perp \mathsf{RF}_i(\tau)$ into $\mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$. We then refine the prior transition

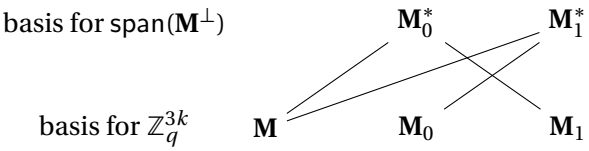


Figure 13: Solid lines mean orthogonal, that is: $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0} = \mathbf{M}^\top \mathbf{M}_1^* = \mathbf{M}_0^\top \mathbf{M}_1^*$.

from Games i to $i+1$ as follows:

- Game $i.0$ (= Game i): pick $\mathbf{y} \leftarrow \mathbb{Z}_q^{3k}$ for ciphertexts, and replace $\mathbf{k}_\tau$ with $\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau)$;
- Game $i.1$: replace $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ with $\mathbf{y} \leftarrow_{\mathrm{R}} \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$;
- Game $i.2$: replace $\mathsf{RF}_i^{(0)}(\tau)$ with $\mathsf{RF}_{i+1}^{(0)}(\tau)$;
- Game $i.3$: replace $\mathsf{RF}_i^{(1)}(\tau)$ with $\mathsf{RF}_{i+1}^{(1)}(\tau)$;
- Game $i.4$ (= Game $i+1$): replace $\mathbf{y} \leftarrow_{\mathrm{R}} \mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ with $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$.

For the transition from Game $i.0$ to Game $i.1$, we rely on the fact that the uniform distributions over $\mathbb{Z}_q^{3k}$ and $\mathsf{span}(\mathbf{M}, \mathbf{M}_{\tau_{i+1}})$ encoded in the group are computationally indistinguishable, even given a random basis for $\mathsf{span}(\mathbf{M}^\perp)$ (in the clear). This extends to the setting with multiple samples, with a tight reduction to the $\mathcal{D}_k$-MDDH Assumption independent of the number of samples.

For the transition from Game $i.1$ to $i.2$, we rely on an information-theoretic argument like the one we just outlined, replacing $\mathsf{span}(\mathbf{M})$ with $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ and $\mathbf{M}^\perp$ with $\mathbf{M}_0^*$ in the case analysis. In particular, we will exploit the fact that if $\mathbf{y}$ lies outside $\mathsf{span}(\mathbf{M}, \mathbf{M}_1)$, then $\mathbf{y}^\top \mathbf{k}_{i+1,1}$ is uniformly random even given $\mathbf{M}, \mathbf{M}\mathbf{k}_{i+1,1}, \mathbf{M}_1, \mathbf{M}_1\mathbf{k}_{i+1,1}$. The transition from Game $i.2$ to $i.3$ is completely analogous.

From PCA to CCA. Using standard techniques from [59, 110, 107, 32, 4], we could transform our basic tag-based PCA-secure scheme into a "full-fledged" CCA-secure encryption scheme by adding another hash proof system (or an authenticated symmetric encryption scheme) and a one-time signature scheme. However, this would incur an additional overhead of several group elements in the ciphertext. Instead, we show how to directly modify our tag-based PCA-secure scheme to obtain a more efficient CCA-secure scheme with the minimal additional overhead of a single symmetric-key authenticated encryption. In particular, the overall ciphertext overhead in our tightly CCA-secure encryption scheme is merely *one* group element more than that for the best known non-tight schemes [110, 97].

To encrypt a message M in the CCA-secure encryption scheme, we will (i) pick a random $\mathbf{y}$ as in the tag-based PCA scheme, (ii) derive a tag τ from $\mathbf{y}$, (iii) encrypt M using a one-time authenticated encryption under the KEM key $[\mathbf{y}^\top \mathbf{k}_\tau]$. The naive approach is to derive the tag τ by hashing $[\mathbf{y}] \in \mathbb{G}^{3k}$, as in [110]. However, this creates a circularity in Game $i.1$ where the distribution of $[\mathbf{y}]$ depends on the tag. Instead, we will derive the tag τ by hashing $[\overline{\mathbf{y}}] \in \mathbb{G}^k$, where $\overline{\mathbf{y}} \in \mathbb{Z}_q^k$ are the top k entries of $\mathbf{y} \in \mathbb{Z}_q^{3k}$. We then modify $\mathbf{M}_0, \mathbf{M}_1$ so that the top k rows of both matrices are zero, which avoids the circularity issue. In the proof of security, we will also rely on the fact that for any $\mathbf{y}_0, \mathbf{y}_1 \in \mathbb{Z}_q^{3k}$, if $\overline{\mathbf{y}}_0 = \overline{\mathbf{y}}_1$ and $\mathbf{y}_0 \in \mathsf{span}(\mathbf{M})$, then either $\mathbf{y}_0 = \mathbf{y}_1$ or $\mathbf{y}_1 \notin \mathsf{span}(\mathbf{M})$. This allows us to deduce that if the adversary queries the CCA oracle on a ciphertext which shares the same tag as some challenge ciphertext, then the CCA oracle will reject with overwhelming probability.

Alternative view-point. Our construction can also be viewed as applying the BCHK IBE→PKE transform [32] to the scheme from [99], and then writing the exponents of the secret keys in the clear, thereby avoiding the pairing. This means that we can no longer apply a computational assumption and the randomized Naor-Reingold argument to the secret key space. Indeed, we replace this with an information-theoretic Cramer-Shoup-like argument as outlined above.

Prior approaches. Several approaches to construct tightly CCA-secure PKE schemes exist: first, the schemes of [96, 5, 6, 118, 117, 119] construct a tightly secure NIZK scheme from a tightly secure signature scheme, and then use the tightly secure NIZK in a CCA-secure PKE scheme following the Naor-Yung double encryption paradigm [125, 61]. Since these approaches build on the public verifiability of the used NIZK scheme (in order to faithfully simulate a decryption oracle), their reliance on a pairing seems inherent.

Next, the works of [48, 25, 99, 18, 83] used a (Naor-Reingold-based) MAC instead of a signature scheme to design tightly secure IBE schemes. Those IBE schemes can then be converted (using the

Reference	$\|\mathsf{pk}\|$	$\|\mathsf{ct}\| - \|m\|$	security loss	assumption	pairing
CS98 [59]	$\mathcal{O}(1)$	3	$\mathcal{O}(Q)$	DDH	no
KD04, HK07 [110, 97]	$\mathcal{O}(1)$	2	$\mathcal{O}(Q)$	DDH	no
C:HofJag12 [96]	$O(1)$	$O(\lambda)$	$\mathcal{O}(1)$	2-Lin	yes
LPJY15 [117, 119]	$\mathcal{O}(\lambda)$	47	$\mathcal{O}(\lambda)$	2-Lin	yes
AHY15 [18]	$\mathcal{O}(\lambda)$	12	$\mathcal{O}(\lambda)$	2-Lin	yes
GCDCT15 [83]	$\mathcal{O}(\lambda)$	10 (resp. $6k+4$)	$\mathcal{O}(\lambda)$	SXDH (resp. k-Lin)	yes
Ours §4	$\mathcal{O}(\lambda)$	3 (resp. $3k$)	$\mathcal{O}(\lambda)$	DDH (resp. k-Lin)	no

Figure 14: Comparison amongst CCA-secure encryption schemes, where Q is the number of ciphertexts, $|\mathsf{pk}|$ denotes the size (i.e the number of groups elements, or exponent of group elements) of the public key, and $|\mathsf{ct}| - |m|$ denotes the ciphertext overhead, ignoring smaller contributions from symmetric-key encryption. We omit [99] from this table since we only focus on prime-order groups here.

BCHK transformation [32]) into tightly CCA-secure PKE schemes. However, the derived PKE schemes still rely on pairings, since the original IBE schemes do (and the BCHK does not remove the reliance on pairings).

In contrast, our approach directly fuses a Naor-Reingold-like randomization argument with the encryption process. We are able to do so since we substitute a computational randomization argument (as used in the latter line of works) with an information-theoretic one, as described above. Hence, we can apply that argument to *exponents* rather than group elements. This enables us to trade pairing operations for exponentiations in our scheme.

Efficiency comparison with non-tightly secure schemes. We finally mention that our DDH-based scheme compares favorably even with the most efficient (non-tightly) CCA-secure DDH-based encryption schemes [110, 97]. To make things concrete, assume $\lambda = 80$ and a setting with $Q_{\mathsf{enc}} = Q_{\mathsf{dec}} = 2^{30}$. The best known reductions for the schemes of [110, 97] lose a factor of $Q_{\mathsf{enc}} = 2^{30}$, whereas our scheme loses a factor of about $4\lambda \leq 2^9$. Hence, the group size for [110, 97] should be at least $2^{2\cdot(80+30)} = 2^{220}$ compared to $2^{2\cdot(80+9)} = 2^{178}$ in our case. Thus, the ciphertext overhead (ignoring the symmetric encryption part) in our scheme is $3 \cdot 178 = 534$ bits, which is close to $2 \cdot 220 = 440$ bits with [110, 97].[10]

Perhaps even more interestingly, we can compare computational efficiency of encryption in this scenario. For simplicitly, we only count exponentiations and assume a naive square-and-multiply-based exponentiation with no further multi-exponentiation optimizations.[11] Encryption in [110, 97] takes about 3.5 exponentiations (where we count an exponentiation with a $(\lambda + \log_2(Q_{\mathsf{enc}} + Q_{\mathsf{dec}}))$-bit hash value[12] as 0.5 exponentiations). In our scheme, we have about 4.67 exponentiations, where we count the computation of $[\mathbf{M}^\top \mathbf{k}_\tau]$ – which consists of 2λ multiplications – as 0.67 exponentiations.) Since exponentiation (under our assumptions) takes time cubic in the bitlength, we get that encryption with our scheme is actually about 29% *less expensive* than with [110, 97].

However, of course we should also note that public and secret key in our scheme are significantly larger (e.g., $4\lambda + 3 = 323$ group elements in pk) than with [110, 97] (4 group elements in pk).

[10] In this calculation, we do not consider the symmetric authenticated encryption of the actual plaintext (and a corresponding MAC value), which is the same with [110, 97] and our scheme.

[11] Here, optimizations would improve the schemes of [110, 97] and ours similarly, since the schemes are very similar.

[12] It is possible to prove the security of [110, 97] using a *target*-collision-resistant hash function, such that $|\tau| = \lambda$. However, in the multi-user setting, a hybrid argument is required, such that the output size of the hash function will have to be increased to at least $|\tau| = \lambda + \log_2(Q_{\mathsf{enc}} + Q_{\mathsf{dec}})$.

Extension: NIZK arguments. We also obtain tightly simulation-sound non-interactive zero-knowledge (NIZK) arguments from our encryption scheme in a semi-generic way.

Let us start with any designated-verifier quasi-adaptive NIZK (short: DVQANIZK) argument system Π for a given language. Recall that in a designated-verifier NIZK, proofs can only be verified with a secret verification key, and soundness only holds against adversaries who do not know that key. Furthermore, quasi-adaptivity means that the language has to be fixed at setup time of the scheme. Let Π_{PKE} be the variant of Π in which proofs are encrypted using a CCA-secure PKE scheme PKE. Public and secret key of PKE are of course made part of CRS and verification key, respectively. Observe that Π_{PKE} enjoys simulation-soundness, assuming that simulated proofs are simply encryptions of random plaintexts. Indeed, the CCA security of PKE guarantees that authentic Π_{PKE}-proofs can be substituted with simulated ones, while being able to verify (using a decryption oracle) a purported Π_{PKE}-proof generated by an adversary. Furthermore, if PKE is tightly secure, then so is Π_{PKE}.

When using a hash proof system for Π and our encryption scheme for PKE, this immediately yields a tightly simulation-sound DVQANIZK for linear languages (i.e., languages of the form $\{[\mathbf{Mx}] \mid \mathbf{x} \in \mathbb{Z}_q^t\}$ for some matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times t}$ with $t < n$) that does not require pairings. We stress that our DVQANIZK is tightly secure in a setting with many simulated proofs and many adversarial verification queries.

Using the semi-generic transformation of [108], we can then derive a tightly simulation-sound QANIZK proof system (with public verification), that however relies on pairings. We note that the transformation of [108] only requires a DVQANIZK that is secure against a single adversarial verification query, since the pairing enables the public verifiability of proofs. Hence, we can first optimize and trim down our DVQANIZK (such that only a single adversarial verification query is supported), and then apply the transformation. This yields a QANIZK with particularly compact proofs. See Figure 15 for a comparison with relevant existing proof systems.

Reference	type	$\lvert\mathsf{crs}\rvert$	$\lvert\pi\rvert$	sec. loss	assumption	pairing
CCS09 [41]	NIZK	$\mathcal{O}(1)$	$2n+6t+52$	$\mathcal{O}(Q_{\mathsf{sim}})$	2-Lin	yes
C:HofJag12 [96]	NIZK	$\mathcal{O}(1)$	$\gg 500$	$\mathcal{O}(1)$	2-Lin	yes
LPJY14 [118]	QANIZK	$\mathcal{O}(n+\lambda)$	20	$\mathcal{O}(Q_{\mathsf{sim}})$	2-Lin	yes
KW15 [108]	QANIZK	$\mathcal{O}(kn)$	$2k+2$	$\mathcal{O}(Q_{\mathsf{sim}})$	k-Lin	yes
LPJY15 [119]	QANIZK	$\mathcal{O}(n+\lambda)$	42	$\mathcal{O}(\lambda)$	2-Lin	yes
Ours (full version)	DVQANIZK	$\mathcal{O}(t+k\lambda)$	$3k+1$	$\mathcal{O}(\lambda)$	k-Lin	no
Ours (full version)	QANIZK	$\mathcal{O}(k^2\lambda+kn)$	$2k+1$	$\mathcal{O}(\lambda)$	k-Lin	yes

Figure 15: (DV)QANIZK schemes for subspaces of $\mathbb{G}^n$ of dimension $t < n$. $\lvert\mathsf{crs}\rvert$ and $\lvert\pi\rvert$ denote the size (in group elements) of the CRS and of proofs. Q_{sim} is the number of simulated proofs in the simulation-soundness experiment. The scheme from [108] (as well as our own schemes) can also be generalized to matrix assumptions [62], at the cost of a larger CRS.

Roadmap. We recall some notation and basic definitions (including those concerning our algebraic setting and for tightly secure encryption) in Section 2. Section 3 presents our basic PCA-secure encryption scheme and represents the core of our results. In Section 4, we present our optimized CCA-secure PKE scheme. Our NIZK-related applications are presented in the full version of this paper.

2 Preliminaries

2.1 Notations

If $\mathbf{x} \in \mathcal{B}^n$, then $|\mathbf{x}|$ denotes the length n of the vector. Further, $x \leftarrow_R \mathcal{B}$ denotes the process of sampling an element x from set $\mathcal{B}$ uniformly at random. For any bit string $\tau \in \{0,1\}^*$, we denote by τ_i the i'th bit of τ. We denote by λ the security parameter, and by $\mathsf{negl}(\cdot)$ any negligible function of λ. For all matrix $\mathbf{A} \in \mathbb{Z}_q^{\ell \times k}$ with $\ell > k$, $\overline{\mathbf{A}} \in \mathbb{Z}_q^{k \times k}$ denotes the upper square matrix of $\mathbf{A}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_q^{\ell-k \times k}$ denotes the lower $\ell - k$ rows of $\mathbf{A}$. With $\mathsf{span}(\mathbf{A}) := \{\mathbf{Ar} \mid \mathbf{r} \in \mathbb{Z}_q^k\} \subset \mathbb{Z}_q^\ell$, we denote the span of $\mathbf{A}$.

2.2 Collision resistant hashing

A hash function generator is a PPT algorithm $\mathcal{H}$ that, on input 1^λ , outputs an efficiently computable function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$.

Definition 2.1 (Collision Resistance)**.** *We say that a hash function generator $\mathcal{H}$ outputs collision-resistant functions H if for all PPT adversaries $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{A}) := \Pr\left[x \neq x' \wedge \mathsf{H}(x) = \mathsf{H}(x') \;\middle|\; \begin{array}{l} \mathsf{H} \leftarrow_R \mathcal{H}(1^\lambda), \\ (x, x') \leftarrow \mathcal{A}(1^\lambda, \mathsf{H}) \end{array} \right] = \mathsf{negl}(\lambda).$$

2.3 Prime-order groups

Let GGen be a probabilistic polynomial time (PPT) algorithm that on input 1^λ returns a description $\mathcal{G} = (\mathbb{G}, q, P)$ of an additive cyclic group $\mathbb{G}$ of order q for a λ-bit prime q, whose generator is P.

We use implicit representation of group elements as introduced in [62]. For $a \in \mathbb{Z}_q$, define $[a] = aP \in \mathbb{G}$ as the *implicit representation* of a in $\mathbb{G}$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_q^{n \times m}$ we define $[\mathbf{A}]$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}$:

$$[\mathbf{A}] := \begin{pmatrix} a_{11}P & \dots & a_{1m}P \\ & & \\ a_{n1}P & \dots & a_{nm}P \end{pmatrix} \in \mathbb{G}^{n \times m}$$

We will always use this implicit notation of elements in $\mathbb{G}$, i.e., we let $[a] \in \mathbb{G}$ be an element in $\mathbb{G}$. Note that from $[a] \in \mathbb{G}$ it is generally hard to compute the value a (discrete logarithm problem in $\mathbb{G}$). Obviously, given $[a], [b] \in \mathbb{G}$ and a scalar $x \in \mathbb{Z}_q$, one can efficiently compute $[ax] \in \mathbb{G}$ and $[a+b] \in \mathbb{G}$.

2.4 Matrix Diffie-Hellman Assumption

We recall the definitions of the Matrix Decision Diffie-Hellman (MDDH) Assumption [62].

Definition 2.2 (Matrix Distribution)**.** *Let $k, \ell \in \mathbb{N}$, with $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices in $\mathbb{Z}_q^{\ell \times k}$ of full rank k in polynomial time. We write $\mathcal{D}_k := \mathcal{D}_{k+1,k}$.*

Without loss of generality, we assume the first k rows of $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$ form an invertible matrix. The $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman problem is to distinguish the two distributions $([\mathbf{A}], [\mathbf{Aw}])$ and $([\mathbf{A}], [\mathbf{u}])$ where $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_R \mathbb{Z}_q^k$ and $\mathbf{u} \leftarrow_R \mathbb{Z}_q^\ell$.

Definition 2.3 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman Assumption $\mathcal{D}_{\ell,k}$-MDDH)**.** *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. We say that the $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) Assumption holds relative to* GGen *if for all PPT adversaries $\mathcal{A}$,*

$$\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) := \\ |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{Aw}]) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}], [\mathbf{u}]) = 1]| = \mathsf{negl}(\lambda),$$

where the probability is over $\mathcal{G} \leftarrow_R \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}, \mathbf{w} \leftarrow_R \mathbb{Z}_q^k, \mathbf{u} \leftarrow_R \mathbb{Z}_q^\ell$.

For each $k \geq 1$, [62] specifies distributions $\mathcal{L}_k$, $\mathcal{SC}_k$, $\mathcal{C}_k$ (and others) over $\mathbb{Z}_q^{(k+1)\times k}$ such that the corresponding $\mathcal{D}_k$-MDDH assumptions are generically secure in bilinear groups and form a hierarchy of increasingly weaker assumptions. $\mathcal{L}_k$-MDDH is the well known k-Linear Assumption k-Lin with 1-Lin = DDH. In this work we are mostly interested in the uniform matrix distribution $\mathcal{U}_{\ell,k}$.

Definition 2.4 (Uniform distribution)**.** *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. We denote by $\mathcal{U}_{\ell,k}$ the uniform distribution over all full-rank $\ell \times k$ matrices over $\mathbb{Z}_q$. Let $\mathcal{U}_k := \mathcal{U}_{k+1,k}$.*

Lemma 2.1 ($\mathcal{U}_k$-MDDH $\Leftrightarrow$ $\mathcal{U}_{\ell,k}$-MDDH)**.** *Let $\ell, k \in \mathbb{N}$, with $\ell > k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ (and vice versa) such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B})$.*

Proof. This follows from the simple fact that a $\mathcal{U}_{\ell,k}$-MDDH instance $([\mathbf{A}], [\mathbf{z}])$ can be transformed into an $\mathcal{U}_k$-MDDH instance $([\mathbf{A}'] = [\mathbf{TA}], [\mathbf{z}'] = [\mathbf{Tz}])$ for a random $(k+1) \times \ell$ matrix $\mathbf{T}$. If $\mathbf{z} = \mathbf{Aw}$, then $\mathbf{z}' = \mathbf{TAw} = \mathbf{A}'\mathbf{w}$; if $\mathbf{z}$ is uniform, so is $\mathbf{z}'$. Similarly, a $\mathcal{U}_k$-MDDH instance $([\mathbf{A}'], [\mathbf{z}'])$ can be transformed into an $\mathcal{U}_{\ell,k}$-MDDH instance $([\mathbf{A}] = [\mathbf{T}'\mathbf{A}'], [\mathbf{z}] = [\mathbf{T}'\mathbf{z}'])$ for a random $\ell \times (k+1)$ matrix $\mathbf{T}'$. □

Among all possible matrix distributions $\mathcal{D}_{\ell,k}$, the uniform matrix distribution $\mathcal{U}_k$ is the hardest possible instance, so in particular k-Lin $\Rightarrow$ $\mathcal{U}_k$-MDDH.

Lemma 2.2 ($\mathcal{D}_{\ell,k}$-MDDH $\Rightarrow$ $\mathcal{U}_k$-MDDH, [62])**.** *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{D}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B})$.*

Let $Q \geq 1$. For $\mathbf{W} \leftarrow_R \mathbb{Z}_q^{k\times Q}, \mathbf{U} \leftarrow_R \mathbb{Z}_q^{\ell\times Q}$, we consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption which consists in distinguishing the distributions $([\mathbf{A}], [\mathbf{AW}])$ from $([\mathbf{A}], [\mathbf{U}])$. That is, a challenge for the Q-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption consists of Q independent challenges of the $\mathcal{D}_{\ell,k}$-MDDH Assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$). In [62] it is shown that the two problems are equivalent, where (for $Q \geq \ell - k$) the reduction loses a factor $\ell - k$. In combination with Lemma 2.1 we obtain the following tighter version for the special case of $\mathcal{D}_{\ell,k} = \mathcal{U}_{\ell,k}$.

Lemma 2.3 (Random self-reducibility of $\mathcal{U}_{\ell,k}$-MDDH, [62])**.** *Let $\ell, k, Q \in \mathbb{N}$ with $\ell > k$. For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and*

$$\mathbf{Adv}^{Q\text{-mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{B}) + \frac{1}{q-1}$$

where $\mathbf{Adv}^{\text{Q-mddh}}_{\mathcal{U}_{\ell,k},\mathsf{GGen}}(\mathcal{B}) := |\Pr[\mathcal{B}(\mathcal{G},[\mathbf{A}],[\mathbf{AW}])=1] - \Pr[\mathcal{B}(\mathcal{G},[\mathbf{A}],[\mathbf{U}])=1]|$ *and the probability is over* $\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda), \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{U}_{\ell,k}, \mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{k\times Q}, \mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{\ell\times Q}$.

2.5 Public-Key Encryption

Definition 2.5 (PKE)**.** *A* Public-Key Encryption *(PKE) consists of three PPT algorithms* $\mathsf{PKE} = (\mathsf{Param}_{\mathsf{PKE}}, \mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$*:*

- *The probabilistic key generation algorithm* $\mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)$ *generates a pair of public and secret keys* $(\mathsf{pk},\mathsf{sk})$.
- *The probabilistic encryption algorithm* $\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)$ *returns a ciphertext* ct.
- *The deterministic decryption algorithm* $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk},\mathsf{sk},\mathsf{ct})$ *returns a message* M *or* $\perp$*, where* $\perp$ *is a special rejection symbol.*

We define the following properties:

Perfect correctness. *For all* λ*, we have*

$$\Pr\left[\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk},\mathsf{sk},\mathsf{ct}) = M \;\middle|\; \begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{PKE}}(1^\lambda);\\ \mathsf{ct} \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)\end{array}\right] = 1.$$

Multi-ciphertext CCA security [22]. *For any adversary* $\mathcal{A}$*, we define*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr\left[\; b = b' \;\middle|\; b' \leftarrow \mathcal{A}^{\mathsf{Setup},\mathsf{DecO}(\cdot),\mathsf{EncO}(\cdot,\cdot)}(1^\lambda)\right] - 1/2\right|$$

where:

- Setup *sets* $\mathcal{C}_{\mathsf{enc}} := \emptyset$*, samples* $(\mathsf{pk},\mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$ *and* $b \leftarrow_{\mathrm{R}} \{0,1\}$*, and returns* pk. Setup *must be called once at the beginning of the game.*
- $\mathsf{DecO}(\mathsf{ct})$ *returns* $\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk},\mathsf{sk},\mathsf{ct})$ *if* $\mathsf{ct} \notin \mathcal{C}_{\mathsf{enc}}$*,* $\perp$ *otherwise.*
- *If* M_0 *and* M_1 *are two messages of equal length,* $\mathsf{EncO}(M_0, M_1)$ *returns* $\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M_b)$ *and sets* $\mathcal{C}_{\mathsf{enc}} := \mathcal{C}_{\mathsf{enc}} \cup \{\mathsf{ct}\}$.

We say PKE *is* IND-CCA *secure if for all* PPT *adversaries* $\mathcal{A}$*, the advantage* $\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}}(\mathcal{A})$ *is a negligible function of* λ.

2.6 Key-Encapsulation Mechanism

Definition 2.6 (Tag-based KEM)**.** *A tag-based* Key-Encapsulation Mechanism *(KEM) consists of three PPT algorithms* $\mathsf{KEM} = (\mathsf{Gen}_{\mathsf{KEM}}, \mathsf{Enc}_{\mathsf{KEM}}, \mathsf{Dec}_{\mathsf{KEM}})$*:*

- *The probabilistic key generation algorithm* $\mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$ *generates a pair of public and secret keys* $(\mathsf{pk},\mathsf{sk})$.

- *The probabilistic encryption algorithm* $\mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk},\tau)$ *returns a pair* (K,C) *where* K *is a uniformly distributed symmetric key in* $\mathcal{K}$ *and* C *is a ciphertext, with respect to the tag* $\tau \in \mathcal{T}$.
- *The deterministic decryption algorithm* $\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk},\mathsf{sk},\tau,C)$ *returns a key* $K \in \mathcal{K}$.

We define the following properties:

Perfect correctness. *For all* λ*, for all tags* $\tau \in \mathcal{T}$*, we have*

$$\Pr\left[\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk},\mathsf{sk},\tau,C)=K \;\middle|\; \begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda);\\ (K,C) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk},\tau)\end{array}\right] = 1.$$

Multi-ciphertext PCA security [126]. *For any adversary* $\mathcal{A}$*, we define*

$$\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}}(\mathcal{A}) := \left|\Pr\left[\; b=b' \;\middle|\; b' \leftarrow \mathcal{A}^{\mathsf{Setup},\mathsf{DecO}(\cdot,\cdot,\cdot),\mathsf{EncO}(\cdot)}(1^\lambda)\right] - 1/2\right|$$

where:

- Setup *sets* $\mathcal{T}_{\mathsf{enc}} = \mathcal{T}_{\mathsf{dec}} := \emptyset$*, samples* $(\mathsf{pk},\mathsf{sk}) \leftarrow_{\mathrm{R}} \mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)$*, picks* $b \leftarrow_{\mathrm{R}} \{0,1\}$*, and returns* pk*.* Setup *is called once at the beginning of the game.*
- *The decryption oracle* $\mathsf{DecO}(\tau,C,\widehat{K})$ *computes* $K := \mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk},\mathsf{sk},\tau,C)$*. It returns* 1 *if* $\widehat{K} = K \wedge \tau \notin \mathcal{T}_{\mathsf{enc}}$*,* 0 *otherwise. Then it sets* $\mathcal{T}_{\mathsf{dec}} := \mathcal{T}_{\mathsf{dec}} \cup \{\tau\}$*.*
- $\mathsf{EncO}(\tau)$ *computes* $(K,C) \leftarrow_{\mathrm{R}} \mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk},\tau)$*, sets* $K_0 := K$ *and* $K_1 \leftarrow_{\mathrm{R}} \mathcal{K}$*. If* $\tau \notin \mathcal{T}_{\mathsf{dec}} \cup \mathcal{T}_{\mathsf{enc}}$*, it returns* (C,K_b)*, and sets* $\mathcal{T}_{\mathsf{enc}} := \mathcal{T}_{\mathsf{enc}} \cup \{\tau\}$*; otherwise it returns* $\perp$*.*

We say KEM *is* IND-PCA *secure if for all* PPT *adversaries* $\mathcal{A}$*, the advantage* $\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}}(\mathcal{A})$ *is a negligible function of* λ*.*

2.7 Authenticated Encryption

Definition 2.7 (AE [97])**.** *An* authenticated symmetric encryption *(AE) with message-space* $\mathcal{M}$ *and key-space* $\mathcal{K}$ *consists of two polynomial-time deterministic algorithms* $(\mathsf{Enc}_{\mathsf{AE}},\mathsf{Dec}_{\mathsf{AE}})$*:*

- *The encryption algorithm* $\mathsf{Enc}_{\mathsf{AE}}(K,M)$ *generates* C*, encryption of the message* M *with the secret key* K*.*
- *The decryption algorithm* $\mathsf{Dec}_{\mathsf{AE}}(K,C)$*, returns a message* M *or* $\perp$*.*

We require that the algorithms satisfy the following properties:

Perfect correctness. *For all* λ*, for all* $K \in \mathcal{K}$ *and* $M \in \mathcal{M}$*, we have*

$$\mathsf{Dec}_{\mathsf{AE}}(K,\mathsf{Enc}_{\mathsf{AE}}(K,M)) = M.$$

One-time Privacy and Authenticity. *For any PPT adversary* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{A}) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} K \leftarrow_{\mathrm{R}} \mathcal{K}; b \leftarrow_{\mathrm{R}} \{0,1\} \\ b' \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathsf{ot\text{-}EncO}(\cdot,\cdot),\mathsf{ot\text{-}DecO}(\cdot)}(1^\lambda, \mathcal{K}) \end{array} \right] - 1/2 \right|$$

is negligible, where $\mathsf{ot\text{-}EncO}(M_0, M_1)$*, on input two messages* M_0 *and* M_1 *of the same length,* $\mathsf{Enc}_{\mathsf{AE}}(K, M_b)$*, and* $\mathsf{ot\text{-}DecO}(\phi)$ *returns* $\mathsf{Dec}_{\mathsf{AE}}(K, \phi)$ *if* $b = 0$*,* $\perp$ *otherwise.* $\mathcal{A}$ *is allowed at most one call to each oracle* $\mathsf{ot\text{-}EncO}$ *and* $\mathsf{ot\text{-}DecO}$*, and the query to* $\mathsf{ot\text{-}DecO}$ *must be different from the output of* $\mathsf{ot\text{-}EncO}$*.* $\mathcal{A}$ *is also given the description of the key-space* $\mathcal{K}$ *as input.*

3 Multi-ciphertext PCA-secure KEM

In this section we describe a tag-based Key Encapsulation Mechanism $\mathsf{KEM}_{\mathsf{PCA}}$ that is IND-PCA-secure (see Definition 2.6).

For simplicity, we use the matrix distribution $\mathcal{U}_{3k,k}$ in our scheme in Figure 16, and prove it secure under the $\mathcal{U}_k$-MDDH Assumption ($\Leftrightarrow$ $\mathcal{U}_{3k,k}$-MDDH Assumption, by Lemma 2.1), which in turn admits a tight reduction to the standard k-Lin Assumption. However, using a matrix distribution $\mathcal{D}_{3k,k}$ with more compact representation yields a more efficient scheme, secure under the $\mathcal{D}_{3k,k}$-MDDH Assumption (see Remark 3.1).

3.1 Our construction

$\underline{\mathsf{Gen}_{\mathsf{KEM}}(1^\lambda)}$:
$\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$; $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$
$\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$
$\mathsf{pk} := \left(\mathcal{G}, [\mathbf{M}], \left([\mathbf{M}^\top \mathbf{k}_{j,\beta}]\right)_{1 \le j \le \lambda, 0 \le \beta \le 1}\right)$
$\mathsf{sk} := (\mathbf{k}_{j,\beta})_{1 \le j \le \lambda, 0 \le \beta \le 1}$
Return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}_{\mathsf{KEM}}(\mathsf{pk}, \tau)}$:
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $C := [\mathbf{r}^\top \mathbf{M}^\top]$
$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$K := [\mathbf{r}^\top \cdot \mathbf{M}^\top \mathbf{k}_\tau]$
Return $(C, K) \in \mathbb{G}^{1 \times 3k} \times \mathbb{G}$

$\underline{\mathsf{Dec}_{\mathsf{KEM}}(\mathsf{pk}, \mathsf{sk}, \tau, C)}$:
$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
Return $K := C \cdot \mathbf{k}_\tau$

Figure 16: $\mathsf{KEM}_{\mathsf{PCA}}$, an IND-PCA-secure KEM under the $\mathcal{U}_k$-MDDH Assumption, with tag-space $\mathcal{T} = \{0,1\}^\lambda$. Here, GGen is a prime-order group generator (see Section 2.3).

Remark 3.1 (On the use of the $\mathcal{U}_k$-MDDH Assumption)**.** *In our scheme, we use a matrix distribution* $\mathcal{U}_{3k,k}$ *for the matrix* $\mathbf{M}$*, therefore proving security under the* $\mathcal{U}_{3k,k}$-MDDH *Assumption* $\Leftrightarrow$ $\mathcal{U}_k$-MDDH *Assumption (see Lemma 2.2). This is for simplicity of presentation. However, for efficiency, one may want to use an assumption with a more compact representation, such as the* $\mathcal{CI}_{3k,k}$-MDDH *Assumption [123] with representation size* $2k$ *instead of* $3k^2$ *for* $\mathcal{U}_{3k,k}$*.*

3.2 Security proof

Theorem 3.1. *The tag-based Key Encapsulation Mechanism* $\mathsf{KEM}_{\mathsf{PCA}}$ *defined in Figure 16 has perfect correctness. Moreover, if the* $\mathcal{U}_k$-MDDH *Assumption holds in* $\mathbb{G}$, $\mathsf{KEM}_{\mathsf{PCA}}$ *is* IND-PCA *secure. Namely, for any adversary* $\mathcal{A}$*, there exists an adversary* $\mathcal{B}$ *such that* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$\mathbf{Adv}^{\mathsf{ind\text{-}pca}}_{\mathsf{KEM}_{\mathsf{PCA}}}(\mathcal{A}) \leq (4\lambda+1) \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot 2^{-\Omega(\lambda)},$$

where Q_{enc}, Q_{dec} *are the number of times* $\mathcal{A}$ *queries* EncO, DecO*, respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

Proof of Theorem 3.1. Perfect correctness follows readily from the fact that for all $\mathbf{r} \in \mathbb{Z}_q^k$ and $C = \mathbf{r}^\top \mathbf{M}^\top$, for all $\mathbf{k} \in \mathbb{Z}_q^{3k}$:

$$\mathbf{r}^\top (\mathbf{M}^\top \mathbf{k}) = C \cdot \mathbf{k}.$$

We now prove the IND-PCA security of $\mathsf{KEM}_{\mathsf{PCA}}$. We proceed via a series of games described in Figure 18 and 19 and we use $\mathbf{Adv}_i$ to denote the advantage of $\mathcal{A}$ in game G_i. We also give a high-level picture of the proof in Figure 17, summarizing the sequence of games.

Lemma 3.2 (G_0 to G_1)**.** *There exists an adversary* $\mathcal{B}_0$ *such that* $\mathbf{T}(\mathcal{B}_0) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_0) + \frac{1}{q-1},$$

where Q_{enc}, Q_{dec} *are the number of times* $\mathcal{A}$ *queries* EncO, DecO*, respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH assumption to "tightly" switch the distribution of all the challenge ciphertexts.

Proof of Lemma 3.2. To go from G_0 to G_1, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption on $[\mathbf{M}]$ (see Definition 2.4 and Lemma 2.3).

We build an adversary $\mathcal{B}_0'$ against the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption, such that $\mathbf{T}(\mathcal{B}_0') \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq \mathbf{Adv}^{Q_{\mathsf{enc}}\text{-mddh}}_{\mathcal{U}_{3k,k},\mathsf{GGen}}(\mathcal{B}_0').$$

This implies the lemma by Lemma 2.3 (self-reducibility of $\mathcal{U}_{3k,k}$-MDDH), and Lemma 2.1 ($\mathcal{U}_{3k,k}$-MDDH $\Leftrightarrow$ $\mathcal{U}_k$-MDDH).

Upon receiving a challenge $(\mathcal{G}, [\mathbf{M}] \in \mathbb{G}^{3k\times k}, [\mathbf{H}] := [\mathbf{h}_1|\ldots|\mathbf{h}_{Q_{\mathsf{enc}}}] \in \mathbb{G}^{3k\times Q_{\mathsf{enc}}})$ for the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption, $\mathcal{B}_0'$ picks $b \leftarrow_{\mathrm{R}} \{0,1\}$, $\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and simulates Setup, DecO as described in Figure 18. To simulate EncO on its j'th query, for $j = 1, \ldots, Q_{\mathsf{enc}}$, $\mathcal{B}_0'$ sets $[\mathbf{y}] := [\mathbf{h}_j]$, and computes K_b as described in Figure 18. □

Lemma 3.3 (G_1 to $\mathrm{G}_{2.0}$)**.** $|\mathbf{Adv}_1 - \mathbf{Adv}_{2.0}| = 0$.

Proof of Lemma 3.3. We show that the two games are identically distributed. To go from G_1 to $\mathrm{G}_{2.0}$, we change the distribution of $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ for $\beta = 0, 1$, to $\mathbf{k}_{1,\beta} + \mathbf{M}^\perp \mathsf{RF}_0(\varepsilon)$, where $\mathbf{k}_{1,\beta} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, $\mathsf{RF}_0(\varepsilon) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{2k}$,

game	$\mathbf{y}$ uniform in:	$\mathbf{k}'_\tau$ used by EncO and DecO	justification/remark
G_0	$\mathsf{span}(\mathbf{M})$	$\mathbf{k}_\tau$	actual scheme
G_1	$\boxed{\mathbb{Z}_q^{3k}}$	$\mathbf{k}_\tau$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}]$
$\mathrm{G}_{2.i}$	$\mathbb{Z}_q^{3k}$	$\mathbf{k}_\tau + \boxed{\mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i})}$	$\mathrm{G}_1 \equiv \mathrm{G}_{2.0}$
$\mathrm{G}_{2.i.1}$	$\tau_{i+1}=0 : \boxed{\mathsf{span}(\mathbf{M},\mathbf{M}_0)}$ $\tau_{i+1}=1 : \boxed{\mathsf{span}(\mathbf{M},\mathbf{M}_1)}$	$\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i})$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_0]$ $\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_1]$
$\mathrm{G}_{2.i.2}$	$\tau_{i+1}=0 : \mathsf{span}(\mathbf{M},\mathbf{M}_0)$ $\tau_{i+1}=1 : \mathsf{span}(\mathbf{M},\mathbf{M}_1)$	$\mathbf{k}_\tau + \boxed{\mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{\mid i+1})} + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{\mid i})$	Cramer-Shoup argument
$\mathrm{G}_{2.i.3}$	$\tau_{i+1}=0 : \mathsf{span}(\mathbf{M},\mathbf{M}_0)$ $\tau_{i+1}=1 : \mathsf{span}(\mathbf{M},\mathbf{M}_1)$	$\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{\mid i+1}) + \boxed{\mathbf{M}_1^* \mathsf{RF}_{i+1}^{(1)}(\tau_{\mid i+1})}$	Cramer-Shoup argument
$\mathrm{G}_{2.i+1}$	$\boxed{\mathbb{Z}_q^{3k}}$	$\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_{i+1}(\tau_{\mid i+1})$	$\mathcal{U}_{3k,k}$-MDDH on $[\mathbf{M}_0]$ and $[\mathbf{M}_1]$

Figure 17: Sequence of games for the proof of Theorem 3.1. Throughout, we have (i) $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$; (ii) $\mathsf{EncO}(\tau) = ([\mathbf{y}], K_b)$ where $K_0 = [\mathbf{y}^\top \mathbf{k}'_\tau]$ and $K_1 \leftarrow_{\mathrm{R}} \mathbb{G}$; (iii) $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ computes the encapsulation key $K := [\mathbf{y}^\top \cdot \mathbf{k}'_\tau]$. Here, $(\mathbf{M}_0^*, \mathbf{M}_1^*)$ is a basis for $\mathsf{span}(\mathbf{M}^\perp)$, so that $\mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{M}_0^\top \mathbf{M}_1^* = \mathbf{0}$, and we write $\mathbf{M}^\perp \mathsf{RF}_i(\tau_{\mid i}) := \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{\mid i}) + \mathbf{M}_0^* \mathsf{RF}_i^{(1)}(\tau_{\mid i})$. The second column shows which set $\mathbf{y}$ is uniformly picked from by EncO, the third column shows the value of $\mathbf{k}'_\tau$ used by both EncO and DecO.

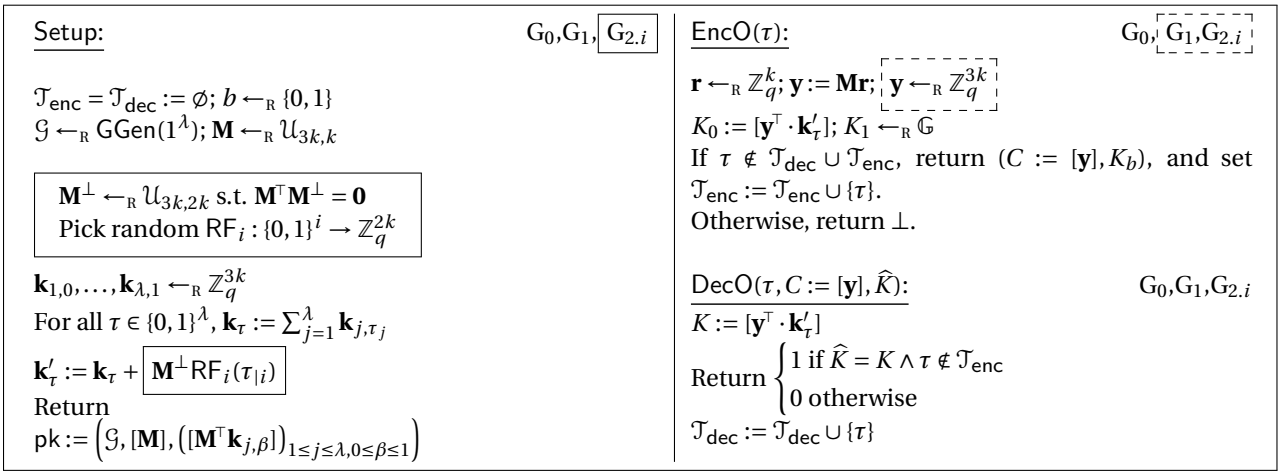


Figure 18: Games $G_0, G_1, G_{2.i}$ (for $1 \le i \le \lambda$) for the proof of multi-ciphertext PCA security of $\mathsf{KEM}_{\mathsf{PCA}}$ in Figure 16. For all $0 \le i \le \lambda$, $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{2k}$ is a random function, and for all $\tau \in \mathcal{T}$, $\tau_{|i}$ denotes the i-bit prefix of τ. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame.

and $\mathbf{M}^\perp \leftarrow_R \mathcal{U}_{3k,2k}$ such that $\mathbf{M}^\top\mathbf{M}^\perp = \mathbf{0}$. Note that the extra term $\mathbf{M}^\perp\mathsf{RF}_0(\varepsilon)$ does not appear in pk, since $\mathbf{M}^\top(\mathbf{k}_{1,\beta} + \mathbf{M}^\perp\mathsf{RF}_0(\varepsilon)) = \mathbf{M}^\top\mathbf{k}_{1,\beta}$. □

Lemma 3.4 ($G_{2.i}$ to $G_{2.i+1}$)**.** *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i}$ such that $\mathbf{T}(\mathcal{B}_{2.i}) \approx \mathbf{T}(\mathcal{A}) + (Q_{enc} + Q_{dec}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i+1}| \le 4 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{2.i}) + \frac{4Q_{dec} + 2k}{q} + \frac{4}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO*, respectively, and* poly(λ) *is independent of $\mathbf{T}(\mathcal{A})$.*

Proof of Lemma 3.4. To go from $G_{2.i}$ to $G_{2.i+1}$, we introduce intermediate games $G_{2.i.1}$, $G_{2.i.2}$ and $G_{2.i.3}$, defined in Figure 19. We prove that these games are indistinguishable in Lemma 3.5, 3.6, 3.7, and 3.8.

Lemma 3.5 ($G_{2.i}$ to $G_{2.i.1}$)**.** *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i.0}$ such that $\mathbf{T}(\mathcal{B}_{2.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{enc} + Q_{dec}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.1}| \le 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}_{2.i.0}) + \frac{2}{q-1},$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO*, respectively, and* poly(λ) *is independent of $\mathbf{T}(\mathcal{A})$.*

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts. We proceed in two steps, first, by changing the distribution of all the ciphertexts with a tag τ such that $\tau_{i+1} = 0$, and then, for those with a tag τ such that $\tau_{i+1} = 1$. We use the MDDH Assumption

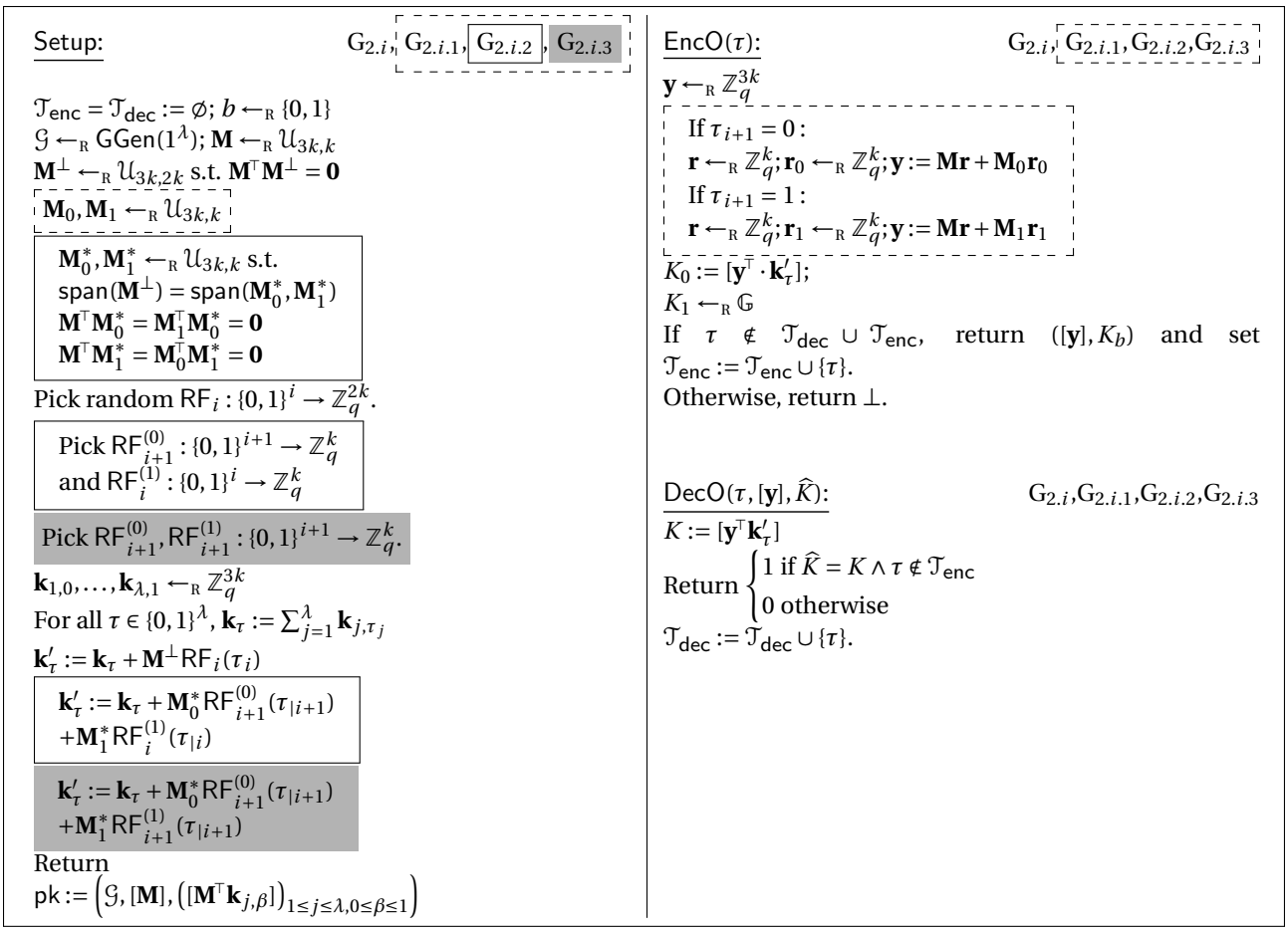


Figure 19: Games $\mathsf{G}_{2.i}$ (for $0 \le i \le \lambda$),$\mathsf{G}_{2.i.1}$, $\mathsf{G}_{2.i.2}$ and $\mathsf{G}_{2.i.3}$ (for $0 \le i \le \lambda - 1$) for the proof of Lemma 3.4. For all $0 \le i \le \lambda$, $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{2k}$, $\mathsf{RF}_i^{(0)}$, $\mathsf{RF}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_q^k$ are random functions, and for all $\tau \in \mathcal{T}$, we denote by $\tau_{|i}$ the i-bit prefix of τ. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame.

with respect to an independent matrix for each step.

Proof of Lemma 3.5. To go from $G_{2.i}$ to $G_{2.i.1}$, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption.

We introduce an intermediate game $G_{2.i.0}$ where $\mathsf{EncO}(\tau)$ is computed as in $G_{2.i.1}$ if $\tau_{i+1}=0$, and as in $G_{2.i}$ if $\tau_{i+1}=1$. Setup, DecO are as in $G_{2.i.1}$. We build adversaries $\mathcal{B}'_{2.i.0}$ and $\mathcal{B}''_{2.i.0}$ such that $\mathbf{T}(\mathcal{B}'_{2.i.0}) \approx \mathbf{T}(\mathcal{B}''_{2.i.0}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}}+Q_{\mathsf{dec}})\cdot \mathrm{poly}(\lambda)$ with $\mathrm{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

Claim 1: $|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.0}| \le \mathbf{Adv}^{Q_{\mathsf{enc}}\text{-mddh}}_{\mathcal{U}_{3,k},\mathsf{GGen}}(\mathcal{B}'_{2.i.0})$.

Claim 2: $|\mathbf{Adv}_{2.i.0} - \mathbf{Adv}_{2.i.1}| \le \mathbf{Adv}^{Q_{\mathsf{enc}}\text{-mddh}}_{\mathcal{U}_{3k,k},\mathsf{GGen}}(\mathcal{B}''_{2.i.0})$.

This implies the lemma by Lemma 2.3 (self-reducibility of $\mathcal{U}_{3k,k}$-MDDH), and Lemma 2.1 ($\mathcal{U}_{3k,k}$-MDDH $\Leftrightarrow$ $\mathcal{U}_k$-MDDH).

Let us prove Claim 1. Upon receiving a challenge $(\mathcal{G}, [\mathbf{M}_0] \in \mathbb{G}^{3k\times k}, [\mathbf{H}] := [\mathbf{h}_1|\ldots|\mathbf{h}_{Q_{\mathsf{enc}}}] \in \mathbb{G}^{3k\times Q_{\mathsf{enc}}})$ for the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption with respect to $\mathbf{M}_0 \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathcal{B}'_{2.i.0}$ does as follows:

Setup**:** $\mathcal{B}'_{2.i.0}$ picks $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, $\mathbf{k}_{1,0},\ldots,\mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and computes pk as described in Figure 19. For each τ queried to EncO or DecO, it computes on the fly $\mathsf{RF}_i(\tau_{|i})$ and $\mathbf{k}'_\tau := \mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i})$, where $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$, $\mathsf{RF}_i : \{0,1\}^i \to \mathbb{Z}_q^{2k}$ is a random function, and $\tau_{|i}$ denotes the i-bit prefix of τ (see Figure 19). Note that $\mathcal{B}'_{2.i.0}$ can compute efficiently $\mathbf{M}^\perp$ from $\mathbf{M}$.

EncO**:** To simulate the oracle $\mathsf{EncO}(\tau)$ on its j'th query, for $j=1,\ldots,Q_{\mathsf{enc}}$, $\mathcal{B}'_{2.i.0}$ computes $[\mathbf{y}]$ as follows:

$$\begin{array}{ll} \text{if } \tau_{i+1}=0: & \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k;\ [\mathbf{y}] := [\mathbf{M}\mathbf{r} + \mathbf{h}_j] \\ \text{if } \tau_{i+1}=1: & [\mathbf{y}] \leftarrow_{\mathrm{R}} \mathbb{G}^{3k} \end{array}$$

This way, $\mathcal{B}'_{2.i.0}$ simulates EncO as in $G_{2.i.0}$ when $[\mathbf{h}_j] := [\mathbf{M}_0\mathbf{r}_0]$ with $\mathbf{r}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$, and as in $G_{2.i}$ when $[\mathbf{h}_j] \leftarrow_{\mathrm{R}} \mathbb{G}^{3k}$.

DecO**:** Finally, $\mathcal{B}'_{2.i.0}$ simulates DecO as described in Figure 19.

Therefore, $|\mathbf{Adv}_{2.i} - \mathbf{Adv}_{2.i.0}| \le \mathbf{Adv}^{Q_{\mathsf{enc}}\text{-mddh}}_{\mathcal{U}_{3k,k},\mathsf{GGen}}(\mathcal{B}'_{2.i.0})$.

To prove Claim 2, we build an adversary $\mathcal{B}''_{2.i.0}$ against the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption with respect to a matrix $\mathbf{M}_1 \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$, independent from $\mathbf{M}_0$, similarly than $\mathcal{B}'_{2.i.0}$. □

Lemma 3.6 ($G_{2.i.1}$ to $G_{2.i.2}$)**.** *For all* $0 \le i \le \lambda - 1$,

$$|\mathbf{Adv}_{2.i.1} - \mathbf{Adv}_{2.i.2}| \le \frac{2Q_{\mathsf{dec}} + 2k}{q},$$

where Q_{dec} *is the number of times* $\mathcal{A}$ *queries* DecO.

Here, we use a variant of the Cramer-Shoup information-theoretic argument to move from RF_i to RF_{i+1}, thereby increasing the entropy of $\mathbf{k}'_\tau$ computed by Setup. For the sake of readability, we proceed

in two steps: in Lemma 3.6, we move from RF_i to an hybrid between RF_i and RF_{i+1}, and in Lemma 3.7, we move to RF_{i+1}.

Proof of Lemma 3.6. In $\mathrm{G}_{2.i.2}$, we decompose $\mathsf{span}(\mathbf{M}^\perp)$ into two subspaces $\mathsf{span}(\mathbf{M}_0^*)$ and $\mathsf{span}(\mathbf{M}_1^*)$, and we increase the entropy of the components of $\mathbf{k}'_\tau$ which lie in $\mathsf{span}(\mathbf{M}_0^*)$. To argue that $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$ are statistically close, we use a Cramer-Shoup argument [59].

Let us first explain how the matrices $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ are sampled. Note that with probability at least $1-\frac{2k}{q}$ over the random coins of Setup, $(\mathbf{M}\|\mathbf{M}_0\|\mathbf{M}_1)$ forms a basis of $\mathbb{Z}_q^{3k}$. Therefore, we have

$$\mathsf{span}(\mathbf{M}^\perp) = \mathsf{Ker}(\mathbf{M}^\top) = \mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_1)^\top\big) \oplus \mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_0)^\top\big).$$

We pick uniformly $\mathbf{M}_0^*$ and $\mathbf{M}_1^*$ in $\mathbb{Z}_q^{3k\times k}$ that generate $\mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_1)^\top\big)$ and $\mathsf{Ker}\big((\mathbf{M}\|\mathbf{M}_0)^\top\big)$, respectively (see Figure 1.1). This way, for all $\tau\in\{0,1\}^\lambda$, we can write

$$\mathbf{M}^\perp \mathsf{RF}_i(\tau_{|i}) := \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}),$$

where $\mathsf{RF}_i^{(0)}, \mathsf{RF}_i^{(1)} : \{0,1\}^i \to \mathbb{Z}_q^k$ are independent random functions.

We define $\mathsf{RF}_{i+1}^{(0)} : \{0,1\}^{i+1} \to \mathbb{Z}_q^k$ as follows:

$$\mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) := \begin{cases} \mathsf{RF}_i^{(0)}(\tau_{|i}) & \text{if } \tau_{i+1}=0 \\ \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathsf{RF}'^{(0)}_i(\tau_{|i}) & \text{if } \tau_{i+1}=1 \end{cases}$$

where $\mathsf{RF}'^{(0)}_i : \{0,1\}^i \to \mathbb{Z}_q^k$ is a random function independent from $\mathsf{RF}_i^{(0)}$. This way, $\mathsf{RF}_{i+1}^{(0)}$ is a random function.

We show that the outputs of EncO and DecO are statistically close in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$. We decompose the proof in two cases (delimited with ■): the queries with a tag $\tau\in\{0,1\}^\lambda$ such that $\tau_{i+1}=0$, and the queries with a tag τ such that $\tau_{i+1}=1$.

Queries with $\tau_{i+1}=0$:
The only difference between $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$ is that Setup computes $\mathbf{k}'_\tau$ using the random function $\mathsf{RF}_i^{(0)}$ in $\mathrm{G}_{2.i.1}$, whereas it uses the random function $\mathsf{RF}_{i+1}^{(0)}$ in $\mathrm{G}_{2.i.2}$ (see Figure 19). Therefore, by definition of $\mathsf{RF}_{i+1}^{(0)}$, for all $\tau\in\{0,1\}^\lambda$ such that $\tau_{i+1}=0$, $\mathbf{k}'_\tau$ is the same in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$, and the outputs of EncO and DecO are identically distributed. ■

Queries with $\tau_{i+1}=1$:
Observe that for all $\mathbf{y}\in\mathsf{span}(\mathbf{M},\mathbf{M}_1)$ and all $\tau\in\{0,1\}^\lambda$ such that $\tau_{i+1}=1$,

$$\begin{aligned}
&\overbrace{\mathbf{y}^\top\Big(\mathbf{k}_\tau + \mathbf{M}_0^*\mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^*\mathsf{RF}_i^{(1)}(\tau_{|i}) + \boxed{\mathbf{M}_0^*\mathsf{RF}'^{(0)}_i(\tau_{|i})}\Big)}^{\mathrm{G}_{2.i.2}} \\
&= \mathbf{y}^\top\Big(\mathbf{k}_\tau + \mathbf{M}_0^*\mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^*\mathsf{RF}_i^{(1)}(\tau_{|i})\Big) + \underbrace{\boxed{\mathbf{y}^\top\mathbf{M}_0^*\mathsf{RF}'^{(0)}_i(\tau_{|i})}}_{=0} \\
&= \overbrace{\mathbf{y}^\top\cdot\Big(\mathbf{k}_\tau + \mathbf{M}_0^*\mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^*\mathsf{RF}_i^{(1)}(\tau_{|i})\Big)}^{\mathrm{G}_{2.i.1}}
\end{aligned}$$

where the second equality uses the fact that $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{M}_1^\top \mathbf{M}_0^* = \mathbf{0}$ and thus $\mathbf{y}^\top \mathbf{M}_0^* = \mathbf{0}$.

This means that:

- the output of EncO on any input τ such that $\tau_{i+1} = 1$ is identically distributed in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$;
- the output of DecO on any input $(\tau, [\mathbf{y}], \widehat{K})$ where $\tau_{i+1} = 1$, and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$ is the same in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$.

Henceforth, we focus on the *ill-formed* queries to DecO, namely those corresponding to $\tau_{i+1} = 1$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_1)$. We introduce intermediate games $\mathrm{G}_{2.i.1.j}$, and $\mathrm{G}'_{2.i.1.j}$ for $j = 0, \ldots, Q_{\mathsf{dec}}$, defined as follows:

- $\mathrm{G}_{2.i.1.j}$: DecO is as in $\mathrm{G}_{2.i.1}$ except that for the first j times it is queried, it outputs 0 to any ill-formed query. EncO is as in $\mathrm{G}_{2.i.2}$.
- $\mathrm{G}'_{2.i.1.j}$: DecO as in $\mathrm{G}_{2.i.2}$ except that for the first j times it is queried, it outputs 0 to any ill-formed query. EncO is as in $\mathrm{G}_{2.i.2}$.

We show that:

$$\mathrm{G}_{2.i.1} \equiv \mathrm{G}_{2.i.1.0} \approx_s \mathrm{G}_{2.i.1.1} \approx_s \ldots \approx_s \mathrm{G}_{2.i.1.Q_{\mathsf{dec}}} \equiv \mathrm{G}'_{2.i.1.Q_{\mathsf{dec}}}$$
$$\mathrm{G}'_{2.i.1.Q_{\mathsf{dec}}} \approx_s \mathrm{G}'_{2.i.1.Q_{\mathsf{dec}}-1} \approx_s \ldots \approx_s \mathrm{G}'_{2.i.1.0} \equiv \mathrm{G}_{2.i.2}$$

where we denote statistical closeness with $\approx_s$ and statistical equality with $\equiv$.

It suffices to show that for all $j = 0, \ldots, Q_{\mathsf{dec}} - 1$:

Claim 1: in $\mathrm{G}_{2.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs 0 with overwhelming probability $1 - 1/q$ (this implies $\mathrm{G}_{2.i.1.j} \approx_s \mathrm{G}_{2.i.1.j+1}$, with statistical difference $1/q$);

Claim 2: in $\mathrm{G}'_{2.i.1.j}$, if the $j+1$-st query is ill-formed, then DecO outputs 0 with overwhelming probability $1 - 1/q$ (this implies $\mathrm{G}'_{2.i.1.j} \approx_s \mathrm{G}'_{2.i.1.j+1}$, with statistical difference $1/q$)

where the probabilities are taken over the random coins of Setup.

Let us prove Claim 1. Recall that in $\mathrm{G}_{2.i.1.j}$, on its $j+1$-st query, $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K})$ computes $K := \left[\mathbf{y}^\top \mathbf{k}'_\tau\right]$, where $\mathbf{k}'_\tau := \left(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_i^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i})\right)$ (see Figure 19). We prove that if $(\tau, [\mathbf{y}], \widehat{K})$ is ill-formed, then K is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO. The reason is that the vector $\mathbf{k}_{i+1,1}$ in sk contains some entropy that is hidden from $\mathcal{A}$. This entropy is "released" on the $j+1$-st query to DecO if it is ill-formed. More formally, we use the fact that the vector $\mathbf{k}_{i+1,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ is identically distributed as $\mathbf{k}_{i+1,1} + \mathbf{M}_0^* \mathbf{w}$, where $\mathbf{k}_{i+1,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$, and $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$. We show that $\mathbf{w}$ is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO.

- The public key pk does not leak any information about $\mathbf{w}$, since

$$\mathbf{M}^\top (\mathbf{k}_{i+1,1} + \boxed{\mathbf{M}_0^* \mathbf{w}}) = \mathbf{M}^\top \mathbf{k}_{i+1,1}.$$

  This is because $\mathbf{M}^\top \mathbf{M}_0^* = \mathbf{0}$.

- The outputs of EncO also hide $\mathbf{w}$.

– For τ such that $\tau_{i+1}=0$, $\mathbf{k}'_\tau$ is independent of $\mathbf{k}_{i+1,1}$, and therefore, so does $\mathsf{EncO}(\tau)$.

– For τ such that $\tau_{i+1}=1$, and for any $\mathbf{y}\in\mathsf{span}(\mathbf{M},\mathbf{M}_1)$, we have:

$$\mathbf{y}^\top(\mathbf{k}'_\tau+\boxed{\mathbf{M}_0^*\mathbf{w}})=\mathbf{y}^\top\mathbf{k}'_\tau \tag{5}$$

since $\mathbf{M}^\top\mathbf{M}_0^*=\mathbf{M}_1^\top\mathbf{M}_0^*=\mathbf{0}$, which implies $\mathbf{y}^\top\mathbf{M}_0^*=\mathbf{0}$.

- The first j outputs of DecO also hide $\mathbf{w}$.

  – For τ such that $\tau_{i+1}=0$, $\mathbf{k}'_\tau$ is independent of $\mathbf{k}_{i+1,1}$, and therefore, so does $\mathsf{DecO}([\mathbf{y}],\tau,\widehat{K})$.

  – For τ such that $\tau_{i+1}=1$ and $\mathbf{y}\in\mathsf{span}(\mathbf{M},\mathbf{M}_1)$, the fact that $\mathsf{DecO}(\tau,[\mathbf{y}],\widehat{K})$ is independent of $\mathbf{w}$ follows readily from Equation (5).

  – For τ such that $\tau_{i+1}=1$ and $\mathbf{y}\notin\mathsf{span}(\mathbf{M},\mathbf{M}_1)$, that is, for an ill-formed query, DecO outputs 0, independently of $\mathbf{w}$, by definition of $\mathrm{G}_{2.i.1.j}$.

This proves that $\mathbf{w}$ is uniformly random from $\mathcal{A}$'s viewpoint.

Finally, because the $j+1$-st query $(\tau,[\mathbf{y}],\widehat{K})$ is ill-formed, we have $\tau_{i+1}=1$, and $\mathbf{y}\notin\mathsf{span}(\mathbf{M},\mathbf{M}_1)$, which implies that $\mathbf{y}^\top\mathbf{M}_0^*\neq\mathbf{0}$. Therefore, the value

$$K=[\mathbf{y}^\top(\mathbf{k}'_\tau+\mathbf{M}_0^*\mathbf{w})]=[\mathbf{y}^\top\mathbf{k}'_\tau+\underbrace{\mathbf{y}^\top\mathbf{M}_0^*}_{\neq\mathbf{0}}\mathbf{w}]$$

computed by DecO is uniformly random over $\mathbb{G}$ from $\mathcal{A}$'s viewpoint. Thus, with probability $1-1/q$ over $K\leftarrow_{\mathrm{R}}\mathbb{G}$, we have $\widehat{K}\neq K$, and $\mathsf{DecO}(\tau,[\mathbf{y}],\widehat{K})=0$.

We prove Claim 2 similarly, arguing than in $\mathrm{G}'_{2.i.1.j}$, the value $K:=\left[\mathbf{y}^\top\mathbf{k}'_\tau\right]$, where $\mathbf{k}'_\tau:=\big(\mathbf{k}_\tau+\mathbf{M}_0^*\mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1})+\mathbf{M}_1^*\mathsf{RF}_i^{(1)}(\tau_{|i})\big)$, computed by $\mathsf{DecO}(\tau,[\mathbf{y}],\widehat{K})$ on its $j+1$-st query, is completely hidden from $\mathcal{A}$, up to its $j+1$-st query to DecO, if $(\tau,[\mathbf{y}],\widehat{K})$ is ill-formed. The argument goes exactly as for Claim 1. ■ □

Lemma 3.7 ($\mathrm{G}_{2.i.2}$ to $\mathrm{G}_{2.i.3}$)**.** *For all* $0\le i\le\lambda-1$,

$$|\mathbf{Adv}_{2.i.2}-\mathbf{Adv}_{2.i.3}|\le\frac{2Q_{\mathsf{dec}}}{q},$$

where Q_{dec} *is the number of times* $\mathcal{A}$ *queries* DecO.

Proof of Lemma 3.7. In $\mathrm{G}_{2.i.3}$, we use the same decomposition $\mathsf{span}(\mathbf{M}^\perp)=\mathsf{span}(\mathbf{M}_0^*,\mathbf{M}_1^*)$ as that in $\mathrm{G}_{2.i.2}$. The entropy of the components of $\mathbf{k}'_\tau$ that lie in $\mathsf{span}(\mathbf{M}_1^*)$ increases from $\mathrm{G}_{2.i.2}$ to $\mathrm{G}_{2.i.3}$. To argue that these two games are statistically close, we use a Cramer-Shoup argument [59], exactly as for Lemma 3.6.

We define $\mathsf{RF}_{i+1}^{(1)}\{0,1\}^{i+1}\to\mathbb{Z}_q^k$ as follows:

$$\mathsf{RF}_{i+1}^{(1)}(\tau_{|i+1}):=\begin{cases}\mathsf{RF}_i^{(1)}(\tau_{|i})+\mathsf{RF}'^{(1)}_i(\tau_{|i}) & \text{if } \tau_{i+1}=0\\ \mathsf{RF}_i^{(1)}(\tau_{|i}) & \text{if } \tau_{i+1}=1\end{cases}$$

where $\mathsf{RF}'^{(1)}_i:\{0,1\}^i\to\mathbb{Z}_q^k$ is a random function independent from $\mathsf{RF}_i^{(1)}$. This way, $\mathsf{RF}_{i+1}^{(1)}$ is a random function.

We show that the outputs of EncO and DecO are statistically close in $\mathrm{G}_{2.i.1}$ and $\mathrm{G}_{2.i.2}$. We decompose the proof in two cases (delimited with ■): the queries with a tag $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$, and the queries with tag τ such that $\tau_{i+1} = 1$.

Queries with $\tau_{i+1} = 1$:
The only difference between $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$ is that Setup computes $\mathbf{k}'_\tau$ using the random function $\mathsf{RF}_i^{(1)}$ in $\mathrm{G}_{2.i.2}$, whereas it uses the random function $\mathsf{RF}_{i+1}^{(1)}$ in $\mathrm{G}_{2.i.3}$ (see Figure 19). Therefore, by definition of $\mathsf{RF}_{i+1}^{(1)}$, for all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 1$, $\mathbf{k}'_\tau$ is the same in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$, and the outputs of EncO and DecO are identically distributed. ■

Queries with $\tau_{i+1} = 0$:
Observe that for all $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$ and all $\tau \in \{0,1\}^\lambda$ such that $\tau_{i+1} = 0$,

$$\begin{aligned}
&\overbrace{\mathbf{y}^\top \left(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) + \boxed{\mathbf{M}_1^* \mathsf{RF}_i'^{(1)}(\tau_{|i})} \right)}^{\mathrm{G}_{2.i.3}} \\
&= \mathbf{y}^\top \left(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) \right) + \underbrace{\boxed{\mathbf{y}^\top \mathbf{M}_1^* \mathsf{RF}_i'^{(1)}(\tau_{|i})}}_{=0} \\
&= \overbrace{\mathbf{y}^\top \cdot \left(\mathbf{k}_\tau + \mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i+1}) + \mathbf{M}_1^* \mathsf{RF}_i^{(1)}(\tau_{|i}) \right)}^{\mathrm{G}_{2.i.2}}
\end{aligned}$$

where the second equality uses the fact $\mathbf{M}^\top \mathbf{M}_1^* = \mathbf{M}_0^\top \mathbf{M}_1^* = \mathbf{0}$, which implies $\mathbf{y}^\top \mathbf{M}_1^* = \mathbf{0}$.

This means that:

- the output of EncO on any input τ such that $\tau_{i+1} = 0$ is identically distributed in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$;
- the output of DecO on any input $(\tau, [\mathbf{y}], \widehat{K})$ where $\tau_{i+1} = 0$, and $\mathbf{y} \in \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$ is the same in $\mathrm{G}_{2.i.2}$ and $\mathrm{G}_{2.i.3}$.

Henceforth, we focus on the *ill-formed* queries to DecO, namely those corresponding to $\tau_{i+1} = 0$, and $\mathbf{y} \notin \mathsf{span}(\mathbf{M}, \mathbf{M}_0)$. The rest of the proof goes similarly than the proof of Lemma 3.6. See the latter for further details. ■ □

Lemma 3.8 ($\mathrm{G}_{2.i.3}$ to $\mathrm{G}_{2.i+1}$)**.** *For all $0 \le i \le \lambda - 1$, there exists an adversary $\mathcal{B}_{2.i.3}$ such that $\mathbf{T}(\mathcal{B}_{2.i.3}) \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathbf{Adv}_{2.i.3} - \mathbf{Adv}_{2.i+1}| \le 2 \cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k, \mathsf{GGen}}(\mathcal{B}_{2.i.3}) + \frac{2}{q-1}$$

where Q_{enc}, Q_{dec} are the number of times $\mathcal{A}$ queries EncO, DecO, respectively, and $\mathsf{poly}(\lambda)$ is independent of $\mathbf{T}(\mathcal{A})$.

Here, we use the MDDH Assumption to "tightly" switch the distribution of all the challenge ciphertexts, as for Lemma 3.5. We proceed in two steps, first, by changing the distribution of all the ciphertexts with a tag τ such that $\tau_{i+1} = 0$, and then, the distribution of those with a tag τ such that $\tau_{i+1} = 1$, using the MDDH Assumption with respect to an independent matrix for each step.

Proof of Lemma 3.8. To go from $\mathrm{G}_{2.i.3}$ to $\mathrm{G}_{2.i+1}$, we switch the distribution of the vectors $[\mathbf{y}]$ sampled by EncO, using the Q_{enc}-fold $\mathcal{U}_{3k,k}$-MDDH Assumption. This transition is symmetric to the transition

between $G_{2.i}$ and $G_{2.i.1}$ (see the proof of Lemma 3.5 for further details). Finally, we use the fact that for all $\tau \in \{0,1\}^\lambda$, $\mathbf{M}_0^* \mathsf{RF}_{i+1}^{(0)}(\tau_{|i}) + \mathbf{M}_1^* \mathsf{RF}_{i+1}^{(1)}(\tau_{|i+1})$ is identically distributed to $\mathbf{M}^\perp \mathsf{RF}_{i+1}(\tau_{|i+1})$, where $\mathsf{RF}_{i+1} : \{0,1\}^{i+1} \to \mathbb{Z}_q^{2k}$ is a random function. This is because $(\mathbf{M}_0^*, \mathbf{M}_1^*)$ is a basis of $\mathsf{span}(\mathbf{M}^\perp)$. □

The proof of Lemma 3.4 follows readily from Lemma 3.5, 3.6, 3.7, and 3.8. □

Lemma 3.9 ($G_{2.\lambda}$)**.** $\mathbf{Adv}_{2.\lambda} \leq \frac{Q_{\mathsf{enc}}}{q}$.

Proof of Lemma 3.9. We show that the joint distribution of all the values K_0 computed by EncO is statistically close to uniform over $\mathbb{G}^{Q_{\mathsf{enc}}}$. Recall that on input τ, $\mathsf{EncO}(\tau)$ computes

$$K_0 := [\mathbf{y}^\top (\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau))],$$

where $\mathsf{RF}_\lambda : \{0,1\}^\lambda \to \mathbb{Z}_q^{2k}$ is a random function, and $\mathbf{y} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$ (see Figure 18).

We make use of the following properties:

Property 1: all the tags τ queried to EncO, such that $\mathsf{EncO}(\tau) \neq \perp$, are distinct.

Property 2: the outputs of DecO are independent of $\{\mathsf{RF}(\tau) : \tau \in \mathcal{T}_{\mathsf{enc}}\}$. This is because for all queries $(\tau, [\mathbf{y}], \widehat{K})$ to DecO such that $\tau \in \mathcal{T}_{\mathsf{enc}}$, $\mathsf{DecO}(\tau, [\mathbf{y}], \widehat{K}) = 0$, independently of $\mathsf{RF}_\lambda(\tau)$, by definition of $G_{2.\lambda}$.

Property 3: with probability at least $1 - \frac{Q_{\mathsf{enc}}}{q}$ over the random coins of EncO, all the vectors $\mathbf{y}$ sampled by EncO are such that $\mathbf{y}^\top \mathbf{M}^\perp \neq \mathbf{0}$.

We deduce that the joint distribution of all the values $\mathsf{RF}_\lambda(\tau)$ computed by EncO is uniformly random over $\left(\mathbb{Z}_q^{2k}\right)^{Q_{\mathsf{enc}}}$ (from Property 1), independent of the outputs of DecO (from Property 2). Finally, from Property 3, we get that the joint distribution of all the values K_0 computed by EncO is statistically close to uniform over $\mathbb{G}^{Q_{\mathsf{enc}}}$, since:

$$K_0 := [\mathbf{y}^\top (\mathbf{k}_\tau + \mathbf{M}^\perp \mathsf{RF}_\lambda(\tau)) = [\mathbf{y}^\top \mathbf{k}_\tau + \underbrace{\mathbf{y}^\top \mathbf{M}^\perp}_{\neq \mathbf{0} \text{ w.h.p.}} \mathsf{RF}_\lambda(\tau)].$$

This means that the values K_0 and K_1 are statistically close, and therefore, $\mathbf{Adv}_3 \leq \frac{Q_{\mathsf{enc}}}{q}$. □

Finally, Theorem 3.1 follows readily from Lemmas 3.2, 3.3, 3.4, and 3.9. □

4 Multi-ciphertext CCA-secure Public Key Encryption scheme

4.1 Our construction

We now describe the optimized IND-CCA-secure PKE scheme. Compared to the PCA-secure KEM from Section 3, we add an authenticated (symmetric) encryption scheme $(\mathsf{Enc}_{\mathsf{AE}}, \mathsf{Dec}_{\mathsf{AE}})$, and set the KEM tag τ as the hash value of a suitable part of the KEM ciphertext (as explained in the introduction). A formal definition with highlighted differences to our PCA-secure KEM appears in Figure 20.

We prove the security under the $\mathcal{U}_k$-MDDH Assumption, which admits a tight reduction to the standard k-Lin Assumption.

$\underline{\mathsf{Gen}_{\mathsf{PKE}}(1^\lambda)}$:
$\mathcal{G} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$; $\mathsf{H} \leftarrow_{\mathrm{R}} \mathcal{H}(1^\lambda)$; $\mathbf{M} \leftarrow_{\mathrm{R}} \mathcal{U}_{3k,k}$
$\mathbf{k}_{1,0}, \ldots, \mathbf{k}_{\lambda,1} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^{3k}$
$\mathsf{pk} := \left(\mathcal{G}, [\mathbf{M}], \mathsf{H}, \left([\mathbf{M}^\top \mathbf{k}_{j,\beta}]\right)_{1\le j\le\lambda, 0\le\beta\le 1}\right)$
$\mathsf{sk} := (\mathbf{k}_{j,\beta})_{1\le j\le\lambda, 0\le\beta\le 1}$
Return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}, M)}$:
$\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_q^k$; $\mathbf{y} := \mathbf{M}\mathbf{r}$
$\tau := \mathsf{H}([\overline{\mathbf{y}}])$
$\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$
$K := [\mathbf{r}^\top \cdot \mathbf{M}^\top \mathbf{k}_\tau]$
$\phi := \mathsf{Enc}_{\mathsf{AE}}(K, M)$
Return $([\mathbf{y}], \phi)$

$\underline{\mathsf{Dec}_{\mathsf{PKE}}(\mathsf{pk}, \mathsf{sk}, ([\mathbf{y}], \phi))}$:
$\tau := \mathsf{H}([\overline{\mathbf{y}}])$; $\mathbf{k}_\tau := \sum_{j=1}^{\lambda} \mathbf{k}_{j,\tau_j}$; $K := [\mathbf{y}^\top \mathbf{k}_\tau]$
Return $\mathsf{Dec}_{\mathsf{AE}}(K, \phi)$.

Figure 20: $\mathsf{PKE}_{\mathsf{CCA}}$, an IND-CCA-secure PKE. We color in blue the differences with $\mathsf{KEM}_{\mathsf{PCA}}$, the IND-PCA-secure KEM in Figure 16. Here, GGen is a prime-order group generator (see Section 2.3) , and $\mathsf{AE} := (\mathsf{Enc}_{\mathsf{AE}}, \mathsf{Dec}_{\mathsf{AE}})$ is an Authenticated Encryption scheme with key-space $\mathcal{K} := \mathcal{G}$ (see Definition 2.7).

Theorem 4.1. *The Public Key Encryption scheme* $\mathsf{PKE}_{\mathsf{CCA}}$ *defined in Figure 20 has perfect correctness, if the underlying Authenticated Encryption scheme* AE *has perfect correctness. Moreover, if the* $\mathcal{U}_k$-MDDH *Assumption holds in* $\mathbb{G}$, AE *has one-time privacy and authenticity, and* $\mathcal{H}$ *generates collision resistant hash functions, then* $\mathsf{PKE}_{\mathsf{CCA}}$ *is* IND-CCA *secure. Namely, for any adversary* $\mathcal{A}$*, there exist adversaries* $\mathcal{B}, \mathcal{B}', \mathcal{B}''$ *such that* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') \approx \mathbf{T}(\mathcal{B}'') \approx \mathbf{T}(\mathcal{A}) + (Q_{\mathsf{dec}} + Q_{\mathsf{enc}}) \cdot \mathsf{poly}(\lambda)$ *and*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{PKE}_{\mathsf{CCA}}}(\mathcal{A}) \le\ & (4\lambda+1)\cdot \mathbf{Adv}^{\mathrm{mddh}}_{\mathcal{U}_k,\mathsf{GGen}}(\mathcal{B}) \\ & + ((4\lambda+2)Q_{\mathsf{dec}} + Q_{\mathsf{enc}} + Q_{\mathsf{enc}}Q_{\mathsf{dec}}) \cdot \mathbf{Adv}^{\mathsf{ae\text{-}ot}}_{\mathsf{AE}}(\mathcal{B}') \\ & + \mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{B}'') + Q_{\mathsf{enc}}(Q_{\mathsf{enc}} + Q_{\mathsf{dec}}) \cdot 2^{-\Omega(\lambda)},\end{aligned} \tag{6}$$

where Q_{enc}, Q_{dec} *are the number of times* $\mathcal{A}$ *queries* EncO, DecO, *respectively, and* $\mathsf{poly}(\lambda)$ *is independent of* $\mathbf{T}(\mathcal{A})$.

We note that the Q_{enc} and Q_{dec} factors in (6) are only related to AE. Hence, when using a statistically secure (one-time) authenticated encryption scheme, the corresponding terms in (6) become exponentially small.

Remark 4.1 (Extension to the multi-user CCA security)**.** *We only provide an analysis in the multi-ciphertext (but single-user) setting. However, we remark (without proof) that our analysis generalizes to the multi-user, multi-ciphertext scenario, similar to [22, 96, 99]. Indeed, all computational steps (not counting the steps related to the AE scheme) modify all ciphertexts simultaneously, relying for this on the re-randomizability of the* $\mathcal{U}_k$-MDDH *Assumption relative to a fixed matrix* $\mathbf{M}$*. The same modifications can be made to many* $\mathsf{PKE}_{\mathsf{CCA}}$ *simultaneously by using that the* $\mathcal{U}_k$-MDDH *Assumption is also re-randomizable across many matrices* $\mathbf{M}_i$*. (A similar property for the DDH, DLIN, and bilinear DDH assumptions is used in [22], [96], and [99], respectively.)*

We defer the proof of Theorem 4.1 to the full version of this paper.

References

[1] M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes. In *EUROCRYPT*, pages 572–590, 2012.

[2] M. Abdalla, F. Benhamouda, and D. Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 69–100. Springer, Heidelberg, Apr. 2015. doi: 10.1007/978-3-662-46803-6_3.

[3] M. Abdalla, F. Benhamouda, and D. Pointcheval. Public-key encryption indistinguishable under plaintext-checkable attacks. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 332–352. Springer, Heidelberg, Mar. / Apr. 2015. doi: 10.1007/978-3-662-46447-2_15.

[4] M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *Journal of Cryptology*, 21(1):97–130, Jan. 2008.

[5] M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, Heidelberg, Dec. 2012. doi: 10.1007/978-3-642-34961-4_3.

[6] M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 312–331. Springer, Heidelberg, Feb. / Mar. 2013. doi: 10.1007/978-3-642-36362-7_20.

[7] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[8] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, pages 98–115, 2010.

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

[10] 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.

[11] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption: New perspectives and lower bounds. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 500–518. Springer, Heidelberg, Aug. 2013. doi: 10.1007/978-3-642-40084-1_28.

[12] M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, pages 1–9, 1999.

[13] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.

[14] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 297–314. Springer, Heidelberg, Aug. 2014. doi: 10.1007/978-3-662-44371-2_17.

[15] B. Applebaum, Y. Ishai, and E. Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *ICALP (1)*, pages 152–163, 2010.

[16] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.

[17] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014. doi: 10.1007/978-3-642-55220-5_31.

[18] N. Attrapadung, G. Hanaoka, and S. Yamada. A framework for identity-based encryption with almost tight security. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 521–549. Springer, Heidelberg, Nov. / Dec. 2015. doi: 10.1007/978-3-662-48797-6_22.

[19] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous

credentials. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, Heidelberg, Mar. 2008.

[20] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, Heidelberg, Aug. 2009.

[21] M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In *EUROCRYPT*, pages 407–424, 2009.

[22] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

[23] M. Bellare, V. T. Hoang, and P. Rogaway. Foundations of garbled circuits. In *ACM CCS*, 2012. Also Cryptology ePrint Archive, Report 2012/265.

[24] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[25] O. Blazy, E. Kiltz, and J. Pan. (Hierarchical) identity-based encryption from affine message authentication. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, Aug. 2014. doi: 10.1007/978-3-662-44371-2_23.

[26] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

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

[28] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

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

[30] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

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

[32] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007. Preliminary version in Eurocrypt '04.

[33] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, Mar. 2011.

[34] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014. doi: 10.1007/978-3-642-55220-5_30.

[35] X. Boyen. General *Ad Hoc* encryption from exponent inversion IBE. In *EUROCRYPT*, pages 394–411, 2007.

[36] X. Boyen. Attribute-based functional encryption on lattices. In A. Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Heidelberg, Mar. 2013. doi: 10.1007/978-3-642-36594-2_8.

[37] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, 2011. Cryptology ePrint Archive, Report 2011/344.

[38] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.

[39] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, pages 1–12, 2014.

[40] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[41] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, Heidelberg, Apr. 2009.

[42] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[43] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, Heidelberg, May 2004.

[44] A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In *Pairing-Based Cryptography - Pairing 2010*, pages 347–366, 2010.

[45] D. Cash, D. Hofheinz, and E. Kiltz. How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351, 2009. http://eprint.iacr.org/2009/351.

[46] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[47] M. Chase and S. Meiklejohn. Déjà Q: Using dual systems to revisit q-type assumptions. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 622–639. Springer, Heidelberg, May 2014. doi: 10.1007/978-3-642-55220-5_34.

[48] J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, Aug. 2013. doi: 10.1007/978-3-642-40084-1_25.

[49] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. IACR Cryptology ePrint Archive, Report 2013/803, 2013. Preliminary version in [48].

[50] J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In M. Abdalla and R. D. Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, Sept. 2014. doi: 10.1007/978-3-319-10879-7_16.

[51] J. Chen and H. Wee. Dual system groups and its applications — compact HIBE and more. IACR Cryptology ePrint Archive, Report 2014/265, 2014. Preliminary version in [48].

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

[53] J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, Apr. 2015. doi: 10.1007/978-3-662-46803-6_20.

[54] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, Apr. 2015. doi: 10.1007/978-3-662-46800-5_1.

[55] K.-M. Chung, Y. Kalai, and S. P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO*, pages 483–501, 2010.

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

[57] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.

[58] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, Apr. / May 2002.

[59] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[60] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero

knowledge. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, Heidelberg, Aug. 2001.

[61] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[62] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, Aug. 2013. doi: 10.1007/978-3-642-40084-1_8.

[63] M. Fischlin, B. Libert, and M. Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 468–485. Springer, Heidelberg, Dec. 2011.

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

[65] G. Fuchsbauer. Commuting signatures and verifiable encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 224–245. Springer, Heidelberg, May 2011.

[66] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices and applications. Cryptology ePrint Archive, Report 2012/610, 2012.

[67] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013. Also, Cryptology ePrint Archive, Report 2012/610.

[68] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013. Also, Cryptology ePrint Archive, Report 2013/451.

[69] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pages 479–499, 2013. Also, Cryptology ePrint Archive, Report 2013/128.

[70] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Functional encryption without obfuscation. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511. Springer, Heidelberg, Jan. 2016. doi: 10.1007/978-3-662-49099-0_18.

[71] R. Gay, I. Kerenidis, and H. Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 485–502. Springer, Heidelberg, Aug. 2015. doi: 10.1007/978-3-662-48000-7_24.

[72] R. Gay, P. Méaux, and H. Wee. Predicate encryption for multi-dimensional range queries from lattices. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 752–776. Springer, Heidelberg, Mar. / Apr. 2015. doi: 10.1007/978-3-662-46447-2_34.

[73] R. Gay, D. Hofheinz, E. Kiltz, and H. Wee. Tightly CCA-secure encryption without pairings. In *EUROCRYPT*, 2016. To appear.

[74] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, pages 465–482, 2010.

[75] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[76] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[77] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[78] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92, 2013.

[79] C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. In *TCC*, 2015. Also, Cryptology ePrint Archive, Report 2014/645.

[80] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[81] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: interactive proofs for muggles. In *STOC*, pages 113–122, 2008.

[82] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *STOC*, pages 555–564, 2013.

[83] J. Gong, J. Chen, X. Dong, Z. Cao, and S. Tang. Extended nested dual system groups, revisited. In C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 133–163. Springer, Heidelberg, Mar. 2016. doi: 10.1007/978-3-662-49384-7_6.

[84] S. Gorbunov and D. Vinayagamurthy. Riding on asymmetry: Efficient ABE for branching programs. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 550–574. Springer, Heidelberg, Nov. / Dec. 2015. doi: 10.1007/978-3-662-48797-6_23.

[85] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, Aug. 2012.

[86] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013. Also, Cryptology ePrint Archive, Report 2013/337.

[87] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[88] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, Aug. 2015. doi: 10.1007/978-3-662-48000-7_25.

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

[90] M. Green, S. Hohenberger, and B. Waters. Outsourcing the decryption of abe ciphertexts. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 34–34, Berkeley, CA, USA, 2011. USENIX Association. URL http://dl.acm.org/citation.cfm?id=2028067.2028101.

[91] J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, Dec. 2006.

[92] J. Groth. Fully anonymous group signatures without random oracles. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, Heidelberg, Dec. 2007.

[93] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, Apr. 2008.

[94] J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Heidelberg, May / June 2006.

[95] D. Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 251–281. Springer, Heidelberg, Jan. 2016. doi: 10.1007/978-3-662-49096-9_11.

[96] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, Aug. 2012.

[97] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, Heidelberg, Aug. 2007.

[98] D. Hofheinz, T. Jager, and E. Knapp. Waters signatures with optimal security reduction. In *Public Key Cryptography*, pages 66–83, 2012.

[99] D. Hofheinz, J. Koch, and C. Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 799–822. Springer, Heidelberg, Mar. / Apr. 2015. doi: 10.1007/978-3-662-46447-2_36.

[100] S. Hohenberger, G. N. Rothblum, A. Shelat, and V. Vaikuntanathan. Securely obfuscating re-encryption. *J.*

Cryptology, 24(4):694–719, 2011.

[101] Y. Ishai and H. Wee. Partial garbling schemes and their applications. In J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Heidelberg, July 2014. doi: 10.1007/978-3-662-43948-7_54.

[102] C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2013. doi: 10.1007/978-3-642-42033-7_1.

[103] C. S. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2014. doi: 10.1007/978-3-662-44381-1_17.

[104] S. A. Kakvi and E. Kiltz. Optimal security proofs for full domain hash, revisited. In *EUROCRYPT*, pages 537–553, 2012.

[105] J. Katz and V. Vaikuntanathan. Round-optimal password-based authenticated key exchange. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 293–310. Springer, Heidelberg, Mar. 2011.

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

[107] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, Heidelberg, Mar. 2006.

[108] E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, Apr. 2015. doi: 10.1007/978-3-662-46803-6_4.

[109] E. Kiltz, J. Pan, and H. Wee. Structure-preserving signatures from standard assumptions, revisited. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 275–295. Springer, Heidelberg, Aug. 2015. doi: 10.1007/978-3-662-48000-7_14.

[110] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, Heidelberg, Aug. 2004.

[111] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012. Also Cryptology ePrint Archive, Report 2011/490.

[112] A. B. Lewko and B. Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *ACM CCS 09*, pages 112–120. ACM Press, Nov. 2009.

[113] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, Feb. 2010.

[114] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.

[115] 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.

[116] B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, Aug. 2013. doi: 10.1007/978-3-642-40084-1_17.

[117] B. Libert, M. Joye, M. Yung, and T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 1–21. Springer, Heidelberg, Dec. 2014. doi: 10.1007/978-3-662-45608-8_1.

[118] B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532. Springer, Heidelberg, May 2014.

doi: 10.1007/978-3-642-55220-5_29.

[119] B. Libert, T. Peters, M. Joye, and M. Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, Nov. / Dec. 2015. doi: 10.1007/978-3-662-48797-6_28.

[120] S. Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.

[121] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.

[122] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC*, pages 351–358, 2010.

[123] P. Morillo, C. Ràfols, and J. L. Villar. Matrix computational assumptions in multilinear groups. *IACR Cryptology ePrint Archive*, 2015:353, 2015.

[124] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[125] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[126] T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In D. Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 159–175. Springer, Heidelberg, Apr. 2001.

[127] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 214–231. Springer, Heidelberg, Dec. 2009.

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

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

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

[131] A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

[132] B. Parno, M. Raykova, and V. Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In *TCC*, pages 422–439, 2012.

[133] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342, 2009.

[134] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, Aug. 1992.

[135] S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of waters' dual system primitives using asymmetric pairings - (extended abstract). In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 298–315. Springer, Heidelberg, May 2012.

[136] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[137] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[138] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999.

[139] A. Sahai and H. Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM Conference on Computer and Communications Security*, pages 463–472, 2010.

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

[141] R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003.

[142] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, Aug. 1984.

[143] E. Shi, J. Bethencourt, H. T.-H. Chan, D. X. Song, and A. Perrig. Multi-Dimensional Range Query over Encrypted Data. In *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[144] D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In *ASIACRYPT*, pages 377–394, 2010.

[145] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[146] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, Aug. 2009.

[147] B. Waters. Functional encryption for regular languages. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Heidelberg, Aug. 2012.

[148] H. Wee. Dual system encryption via predicate encodings. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, Feb. 2014. doi: 10.1007/978-3-642-54242-8_26.

[149] H. Wee. Déjà Q: Encore! Un petit IBE. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 237–258. Springer, Heidelberg, Jan. 2016. doi: 10.1007/978-3-662-49099-0_9.

[150] H. Wee. Déjà Q: Encore! Un petit IBE. In *TCC*, pages 237–258, 2016.

[151] T. H. Yuen, S. S. Chow, C. Zhang, and S. M. Yiu. Exponent-inversion signatures and IBE under static assumptions. Cryptology ePrint Archive, Report 2014/311, 2014.