Keeping Secrets from Those You Work With: Constructions and Characterizations of Encryption

by

David A. Wilson

S.B., Massachusetts Institute of Technology (2004)
S.B., Massachusetts Institute of Technology (2004)
M.Eng., Massachusetts Institute of Technology (2005)

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author .... Signature redacted ....
Department of
Electrical Engineering and Computer Science
May 21, 2014

Certified by......... Signature redacted ....
Shafi Goldwasser
RSA Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .... Signature redacted ....
Professor Leslie A. Kolodziejski
Chair, Department Committee on Graduate Theses

Keeping Secrets from Those You Work With: Constructions and Characterizations of Encryption

by
David A. Wilson

Submitted to the Department of
Electrical Engineering and Computer Science
on May 21, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

With the growth of theoretical cryptography, more and more attention has been given to expanding the models in which encrypted messages are sent (leading to constructions such as identity-based encryption) and the additional functionalities supported by encryption schemes (such as homomorphic operations). This thesis explores the relations between several of these primitives and models, both in terms of *generic* constructions, and constructions based on *specific* hardness assumptions.

First, we define *bounded-collusion identity-based encryption* (BC-IBE), a variant of IBE in which the adversary is only allowed to make a limited number of key queries. This restriction allows more general constructions; specifically, we give three distinct generic constructions of BC-IBE from public-key encryption with short ciphertext size. Each of these constructions requires slightly different properties of the underlying public-key scheme; we give specific instantiations of each of these constructions, thus achieving BC-IBE from the DDH, LWE, NTRU, and QR assumptions.

Second, we explore the relationship between *obfuscation* and *fully-homomorphic encryption.* We define a notion of secure obfuscation for a family of functions known as the *f-reencryption functionality*, and prove that a secure obfuscator for this functionality generically yields a fully-homomorphic encryption scheme. Furthermore, we relate this new definition to previous definitions of obfuscation, and give an instantiation of such an obfuscator based on the LWE assumption, yielding an FHE construction.

Thesis Supervisor: Shafi Goldwasser
Title: RSA Professor of Computer Science and Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern cryptography is moving at an exciting pace. The entire field of cryptography–rigorously analyzed as a discipline–is only a few decades old, and is continuously covering more ground. Cryptographers consider more and more complex interactions, functionalities, and adversarial models. At the same time, the basic primitives and definitions in the field are being scrutinized and evaluated. While we consider more complicated functionalities, we ultimately want to show reductions from these more complex functionalities to simpler protocols, ideally in a generic fashion.

The best-known basic cryptographic functionality is *encryption*, a process designed to protect *privacy* of information. The notion of *public-key encryption* (PKE), first introduced in the classic paper of Diffie and Hellman [29], depicts a simple model in which one user wishes to send messages to a second user, such that the messages cannot be read by an eavesdropper.

The basic PKE model, while extremely useful, satisfies a basic and specific use case: privacy of messages sent to one party, in a world in which their public key is known a priori to any potential senders. It does not encompass a wide variety of other use cases; for example:

- The case where a sender knows the *identity* of the desired recipient, but does not know a trustworthy source for the recipient's public key;
- The case where a sender wishes to encrypt to a user *before* that user ever joins a multi-user cryptographic system (and thus does not even have a key);
- The case where a user wants to do more than simply ensure privacy of a static message; for example, they want to allow others to compute on private data, or allow others to perform a computation that is itself kept secret.

Thus, over the past few decades, cryptographers have developed many new notions of encryption, with new models, definitions, functionalities, and security notions. Two such ideas which are central to the work studied in this thesis are *identity-based encryption* and *fully-homomorphic encryption.*

Identity-based Encryption In order to overcome the issue of public key distribution, Shamir proposed the model of *identity-based encryption* [68]. This model describes a world with many users, each of whom has an *identity* (frequently an arbitrary string). A central authority publishes public parameters common to the entire system, and one can encrypt messages to any user knowing only the global public parameters and the user's identity (in particular, without requiring a user-specific public key). The user can obtain a secret key associated with their identity from the central authority, allowing them to decrypt.

Identity-based encryption schemes were first constructed in the random oracle model by Boneh and Franklin [11], under specific number-theoretic hardness assumptions. A line of work expanded the known IBE results ([23, 9, 74, 42, 12, 47, 21, 2], ...), and constructions are now known under several specific hardness assumptions, both in the standard and random oracle models. However, no generic IBE construction is known, and the breadth of specific IBE constructions is not nearly as wide as that of PKE constructions.

Fully-Homomorphic Encryption A separate question from the model of users in the system is the question of what functions one can compute on encrypted data. It has long been known that some encryption schemes exhibit *partial homomorphism.* That is, one can perform certain operations on the ciphertexts in order to perform known operations on the underlying plaintexts. For example, the ElGamal public-key cryptosystem [35] is *multiplicatively homomorphic*; one can combine two ciphertexts to obtain an encryption of the product of the underlying plaintexts.

The idea of *fully-homomorphic encryption* was first proposed by Rivest, Adelman, and Dertouzos [65]. A fully-homomorphic encryption scheme allows *arbitrary* computations to be performed on encrypted data. The existence of fully-homomorphic encryption schemes remained an open question for three decades, until Gentry gave a construction in 2009 [43]. Since then, a line of work ([73, 25, 26, 18, 17, 14, 46, 45, 48], ...) has made FHE simpler, more efficient, and based on worst-case assumptions, as well as coming up with additional uses for the primitive.

For example, one can use FHE to *delegate* computation to an untrusted server. Such an application inherently protects the privacy of the data being computed on (since it is always encrypted); however, FHE can also be used to verify that the server has performed the computation correctly [41]. It has also been employed outside of this direct outsourcing-of-computation use case, such as to achieve succinct functional encryption [50].

Modern encryption relies on *hardness assumptions.* A major goal of cryptography is to characterize the assumptions needed to construct specific primitives, and to give constructions based on the weakest and most general assumptions possible. In order to make these connections, we rely on *reductions* between problems–that is, a way to translating a successful attack on the cryptographic construction to a method of solving a given problem. This problem can either be a specific computational problem that we assume to be hard, or can be a violation of the security of a simpler functionality. In the realm of encryption, public-key cryptography is known to be generically achievable from *trapdoor permutations* (and *trapdoor predicates*). However, no such generic constructions are known for IBE or FHE—the only known constructions of these schemes are based on specific hardness assumptions stemming from number theory, geometry, and coding theory.

We would like to use generic assumptions for several reasons: from a theoretical perspective, we would like to characterize the abstract properties that enable us to construct cryptographic objects. From a more practical perspective, if a specific hardness assumption turns out to be false, we can substitute a different assumption into a generic assumption. (For example, if scalable quantum computers are built, then they will be able to efficiently factor large integers and take discrete logarithms [69], but no efficient way to solve lattice-based assumptions is known.) Much of this thesis is thus geared toward the goal of generically characterizing IBE and FHE.

1.1 Roadmap to This Thesis

This thesis will give a number of generic relations and constructions between several newer, more complex encryption primitives, in hopes of better characterizing the relations between these primitives and their place in the larger landscape of cryptography. The following subsections give a high-level overview of the results of each chapter; each chapter additionally contains similar prefatory material such that it can also be read as a standalone work.

1.2 Bounded-Collusion Identity-Based Encryption

Chapter 2 will describe a variant of identity-based encryption known as *bounded-collusion identity-based encryption* (BC-IBE). It will describe three different generic constructions of BC-IBE from PKE schemes, and provide a number of concrete instantiations starting from specific PKE schemes (that each rely on a specific hardness assumption). This chapter is based on joint work with Goldwasser and Lewko [51] and joint work with Tessaro [72].

Identity-Based Encryption

An identity-based encryption scheme is defined as a 4-tuple of algorithms (IBEGen, IBEExtract, IBEEnc, IBEDec). A central authority uses IBEGen to create the global parameters of the system (including both public parameters and a master secret key). That authority can then use the IBEExtract algorithm with its master secret key to generate secret keys for specific identities. One can encrypt using IBEEnc knowing only the global public parameters and a user's identity, and a user can decrypt messages encrypted to their identity by using IBEDec with their identity-specific secret key.

The security model for identity-based encryption assumes the possibility of *collusion*. That is, other valid users of the system (with their own identities) might pool their resources in order to try to gain information about messages encrypted to an honest user. This is theoretically modeled by allowing the adversary in the security game to obtain secret keys for arbitrary identities, representing a collusion of those users. The formal security game is detailed in Figure 2-2; in essence, one still wishes the adversary to gain no information about messages to any user for whom he did not specifically request the key.

The IBE model was first proposed by Shamir in 1984 [68], and the first constructions were proposed in 2001 by Boneh and Franklin [11] and by Cocks [23], both in the random oracle model. Subsequent work [19, 8, 9, 74, 12, 47, 21, 2] gave additional constructions in the random oracle and standard models, based on several different specific hardness assumptions. To date, IBE schemes have been constructed using the Decisional Bilinear Diffie-Hellman assumption and similar assumptions in bilinear groups, the Learning With Errors Assumption, and the Quadratic Residuosity assumption.

Bounded-Collusion IBE

As an attempt to come up with constructions under a wider range of assumptions, cryptographers began looking at a variant of IBE known as *Bounded-Collusion IBE* (BC-IBE). In this model, one only guarantees security against an adversary who obtains secret keys associated with at most t identities, where the size of the parameters of the system are allowed to depend on t. Falling short of achieving full security, the bounded-collusion model can be a realistic assumption in many settings, and is in fact a necessary restriction to achieve the more general notion of functional encryption [54]. Additionally, it has been studied in other settings, notably broadcast encryption and revocation (e.g. [37, 38, 39, 59, 61, 56, 30]).

The first construction of BC-IBE came in [33], in the context of their study of the problem of a bounded number of secret key exposures in public-key encryption. To remedy the latter problem, they introduced the notion of *key-insulated* PKE systems and show its equivalence to *IBEs semantically secure against a bounded number of colluding identities*. This equivalence, coupled with constructions of key-insulated PKEs by [33], yields a generic combinatorial construction which converts any semantic secure PKE to a bounded-collusion semantic secure IBE, without needing a random oracle. This paper gave a general reduction from any semantically secure public-key cryptosystem to a BC-IBE scheme. However, their construction suffers from a large ciphertext-size blowup – the resulting ciphertext length is a factor $\omega(t)$ larger than that of the underlying encryption scheme.

PKE systems with key homomorphism.

In recent years, several PKE schemes were proposed with interesting homomorphisms over the public keys and the underlying secret keys. These were constructed for the purpose of showing circular security and leakage resilience properties. In particular, for both the scheme of Boneh, Halevi, Hamburg, and Ostrovsky [13] and the scheme of Brakerski and Goldwasser [16], it can be shown that starting with two valid (public-key, secret-key) pairs $(pk_1, sk_1), (pk_2, sk_2)$, one can obtain a third valid pair as $(pk_1 \cdot pk_2, sk_1 + sk_2)$. Similar homomorphisms can be considered for other cryptosystems as well, such as the classic ElGamal cryptosystem [35] and the encryption scheme of

Regev [64].

One elegant example of the usefulness of homomorphism in cryptography was demonstrated by Rothblum [66], who used the homomorphism of an encryption scheme to change from private-key to public-key.

New Results

In this section, we seek for generic constructions of BC-IBE. In doing so, we use the key-homomorphic properties above in order to achieve small ciphertext size. We explore systems which rely on encryption schemes that solely satisfy the *standard* security notion of semantic security in addition to some syntactical, non-security-related, properties which can be easily verified. In addition, we see if efficiency can be improved by starting from stronger security assumptions. Our constructions have the added benefit of conceptual simplicity, and the resulting instantiations from concrete assumptions either outperform or abstract existing BC-IBE constructions along different axes.

In summary, we make several main contributions:

1. We give a generic approach (the GLW construction) that can be used to construct BC-IBE with short ciphertexts from any PKE satisfying certain syntactic properties. We prove *selective* security of this construction assuming semantic security of the PKE scheme using a mapping ϕ satisfying *cover-freeness* (a notion introduced in [36] and used in several other works, e.g. [59, 27, 32], among others). While being strictly weaker than the notion of full security, selective security is sufficient for some applications. In particular, using the transformation of Boneh *et al.* [10], we can construct a *bounded-CCA-secure* PKE scheme from any selectively-secure BC-IBE scheme.

2. Whenever the underlying semantically-secure scheme satisfies an additional new property – which we call *weak multi-key malleability* – we prove that the GLW construction achieves *full* BC-IBE security, i.e., confidentiality holds even with respect an identity chosen adaptively after learning the parameters of the schemes as well as secret keys for at most t other identities. Roughly, our malleability property states that given the encryption of $c = \mathsf{Enc}(\mathsf{pk}, m)$ of an *unknown* message m under a known public-key pk, and given an additional public-key / secret-key pair $(\mathsf{pk}', \mathsf{sk}')$, we can efficiently produce a ciphertext which is indistinguishable from an encryption of m under $\mathsf{pk} \cdot \mathsf{pk}'$. An example scheme with this property is ElGamal encryption – hence we directly obtain a DDH-based BC-IBE scheme from ElGamal encryption.

3. We provide a new, alternative construction that relies on a different form of malleability (which we simply call multi-key malleability), and does not require any explicit key-homomorphic structure. Intuitively, our notion requires that given $c = \mathsf{Enc}(\mathsf{pk}, m)$ for an unknown message m, and another public key pk', we can obtain a new ciphertext c which decrypts to m under a combination of the secret

keys sk and sk' associated with pk and pk'. We provide an efficient instantiation based on NTRU [57], exploiting its multi-key homomorphic properties recently observed by Lopez-Alt *et al.* [60]. This is of particular interest due to the fact that no fully-secure NTRU-based IBE scheme is known to date. Moreover, our constructions support homomorphic evaluation of ciphertexts, and this is the only construction of identity-based fully homomorphic encryption beyond the recent result by Gentry, Sahai, and Waters [48].

4. Finally, we give another construction of BC-IBE given a PKE that exhibits key-homomorphic structure and satisfies a security notion called *linear related-key security*, a new notion that is slightly stronger than standard semantic security. At a high level, this property entails that a cryptosystem remains secure even if an adversary obtains several keys that have a known relationship with the target secret key, as long as the target key is linearly independent of the others. We use an algebraic technique to prove this property in several cryptosystems that is reminiscent of hash proofs [28]. By assuming this stronger security property, we are able to use a t-wise linearly independent mapping ϕ instead of a cover-free map, resulting in smaller public parameter size.

In each of these constructions, the ciphertext size is small (comparable to a single ciphertext of the underlying public-key system), and we provide concrete instantiations based on several common hardness assumptions (such as DDH, QR, LWE, and NTRU).

1.3 Obfuscation and Fully-Homomorphic Encryption

Chapter 3 will investigate the relation between obfuscation and fully-homomorphic encryption. Namely, it will provide a new definition of obfuscation for a specific functionality (the *f-reencryption functionality*), and show that a secure obfuscation under that definition generically implies a fully-homomorphic encryption scheme. We then provide a concrete example of such an obfuscated reencryption for an LWE-based PKE scheme, obtaining an FHE scheme. This chapter is based on joint work with Tessaro, which was subsequently merged with work by Alwen, Barbosa, Farshim, Gennaro, and Gordon [3].

Fully-Homomorphic Encryption.

The discovery of fully-homomorphic encryption schemes (FHE) has been a key development in modern cryptography. FHE schemes allow arbitrary computation on encrypted data without decrypting. The notion was first proposed by Rivest, Adleman, and Dertouzos [65], but it took more than three decades for the first schemes to be developed. Several FHE schemes have now been developed, first under somewhat nonstandard lattice assumptions [43, 70], then under hardness assumptions for approximate GCD [73, 25, 26], and finally under various forms of the Learning With Errors assumption [18, 17, 15, 14, 46, 45, 48] or other lattice-based assumptions [44].

At the same time, no general construction is known from smaller primitives, even for the case of *leveled FHE schemes.* A d-leveled FHE scheme allows computation of depth-d circuits on encrypted data, allowing its public key size to be a polynomial function in d. In this paper, we address the question of finding a primitive which allows a generic construction of FHE on top of a suitable encryption scheme, and revisit existing works in terms of instantiations of this blueprint.

A (Possibly-)Related Primitive: Obfuscation

Another cryptographic primitive, separate from encryption, is *obfuscation.* Obfuscation attempts to formally characterize the high-level idea of being able to run an algorithm without knowing what it is doing. That is, one can take a computation (frequently represented as a circuit, though obfuscation for other computation models such as RAM programs and Turing machines have also been studied) and generate an *obfuscated* computation of the same type. The obfuscated program should have the same input/output behavior as the original program, but should give no other information about the computation. This allows the computation to be performed by an untrusted party. The idea of allowing a computation to be performed by a party while simultaneously "hiding" some information about the computation from them is reminiscent of FHE, and indeed we will draw a formal connection between the two in Chapter 3.

Multiple different definitions of obfuscation exist, with various constructions and possibility/impossibility results. The foundational treatment of obfuscation was performed by Barak et al [5], in which they formally define both *virtual-black-box (VBB) obfuscation* and *indistinguishability obfuscation* (iO). Virtual black-box obfuscation is a very strong definition that captures the intuition that someone viewing the obfuscated circuit knows nothing about what computation is being performed other than the input/output behavior (and an upper bound on the size of the circuit); it states that having the obfuscated circuit is equivalent to being able to query an oracle that simply computes the function and returns the answer. Indistinguishability obfuscation states that if two circuits have exactly the same input/output behavior, then their obfuscations are indistinguishable.[1] Both of these notions have sparked a series of works detailing both constructions and uses of obfuscation [75, 20, 67, 40]. There are also impossibility results known for VBB obfuscation [5]. Of note, in some natural extensions of VBB it is impossible to obfuscate pseudorandom functions and other natural cryptographic functionalities [24]. While no such impossibility is known for indistinguishability definition, this latter definition places requirements on how the input computations behave on *every* input.

Both the VBB and iO definitions are limited to obfuscating deterministic computations. Cryptographic functionalities require randomness, and furthermore are

[1] This is also known as "best-possible obfuscation" [53], since if any strong notion of obfuscation (such as VBB) is possible for a certain functionality, indistinguishability obfuscation provides equivalent guarantees. This is due to the fact that an iO-obfuscated circuit for that functionality will by definition be indistinguishable from the iO-obfuscation of the *circuit obfuscated under the strongest possible method.*

generally concerned with *average-case* security—that is, while there may be "weak" instances of a functionality, a random instance will be secure with overwhelming probability. Average-case definitions of secure obfuscation, tailored for cryptographic usages, have been explored by Hohenberger et al [58] and subsequently by Chandran et al [22], and defining, instantiating, and using obfuscation remains a very active area of research. In Chapter 3 we will define these notions in more detail, for the purpose of constructing an FHE scheme.

Obfuscating re-encryption.

Our approach relies on the notion of *obfuscated re-encryption*, which has been developed in parallel to FHE. While obfuscation of general functions is impossible [5], there have been several positive results detailing function families that can be obfuscated (e.g. [75, 34, 20], among many others). In particular, there has been a line of research on obfuscation that is secure *on average* (that is, for a random function from a family), rather than for any function in the family ([49, 1], and others); this definition is particularly relevant to cryptographic applications that use randomized functions. Hohenberger et al [58] show a method to obfuscate a re-encryption functionality–that is, a functionality which allows for decryption under one key and encryption under a second–such that the re-encryption procedure can be delegated to a third party who does not learn anything about the re-encrypted messages. Chandran et al [22] extended this work even further, and consider functional re-encryption, in which the second encryption key is a function of the underlying message, in the context of obfuscation of the function (and hiding the message). However, such functionalities have generally only been defined for single-input functions.

New Results

1. Our first contribution is to introduce and define the notion of *many-to-one functional re-encryption* and its obfuscation. More specifically, for a function f, this functionality allows an evaluator to take multiple ciphertexts $c_1, \ldots, c_q$ encrypting messages $m_1, \ldots, m_q$ under the same key pk for some public-key cryptosystem PKE, and computes an encryption of $f(m_1, \ldots, m_q)$ under a different key for some possibly different cryptosystem PKE'.

   Clearly, this functionality is by itself uninteresting, as it can be trivially realized by decrypting the input messages, computing the function, and encrypting the result. However, this functionality becomes interesting if it can be obfuscated and hence delegated to a user without revealing the corresponding secret key. For this reason, we also define a notion of obfuscation for this functionality, which is substantially different than the one proposed by previous works on re-encryption, despite its similar "average-case" perspective: At a high level, our first definition states that for a random circuit computing the re-encryption and for an observer who knows the public key of the source scheme, the obfuscation of that circuit *and* the public key of the target scheme are indistinguishable from the output of a simulator that only knows the public-key of the source scheme.

We also consider a stronger notion, where the simulator does not simulate the public key of the target scheme, but obtains it externally. We show that the latter definition is in fact *implied* by the definition from [58].

2. As one application of many-to-one functional encryption, our second contribution is to show a generic construction of leveled FHE given a semantically-secure encryption scheme such that the corresponding multi-input functional re-encryption functionalities for a complete set of operations (e.g., for the NAND operation) can be obfuscated with respect to the new notions introduced in this paper.

   As an application, we show that Regev-style encryption [64] admits such obfuscated re-encryption for multiplication, which, combined with our main result and the existing additive homomorphism of the encryption yields a level FHE scheme. This scheme corresponds to the one recently proposed by Brakerski [14], for which we provide a more modular abstraction. We also reinterpret the technique of "bootstrapping" ([43] and followup work) as specific implementations of our generic construction.

Chapter 2

Bounded-Collusion Identity-Based Encryption

2.1 Introduction

The last decade in the lifetime of cryptography has been quite exciting. We are witnessing a paradigm shift, departing from the traditional goals of secure and authenticated communication and moving towards systems that are simultaneously highly secure, highly functional, and highly flexible in allowing selected access to encrypted data. As part of this development, different "types" of encryption systems have been conceived and constructed to allow greater ability to meaningfully manipulate and control access to encrypted data, such as bounded and fully homomorphic encryption (FHE), identity-based encryption (IBE), hierarchical identity-based encryption (HIBE), functional encryption (FE), attribute based encryption (ABE), and others. As is typical at any time of rapid innovation, the field is today at a somewhat chaotic state. The different primitives of FHE, IBE, HIBE, FE, and ABE are being implemented based on different computational assumptions and as of yet we do not know of general constructions.

One way to put some order in the picture is to investigate reductions between the various primitives. A beautiful example of such a result was recently shown by Rothblum [66], who demonstrated a simple reduction between any semantically secure private key encryption scheme which possesses a simple homomorphic property over its ciphertexts to a full-fledged semantically secure public key encryption scheme. The homomorphic property requires that the product of a pair of ciphertexts c_1 and c_2, whose corresponding plaintexts are m_1 and m_2, yields a new ciphertext $c_1 \cdot c_2$ which decrypts to $m_1 + m_2 \mod 2$.

In this section, we continue this line of investigation and show how public-key encryption schemes which posses a linear homomorphic property over their keys can be used to construct an efficient identity-based encryption (IBE) scheme that is secure against bounded collusions. The main idea is simple. In a nutshell, the homomorphism over the keys will give us a way to map a set of public keys published by the master authority in an IBE system into a new user-specific public key that is obtained by taking a combination of the published keys. We explore several different methods for taking this combination, and the security properties of each. In the most general case, this combination is simply a subset of the available keys; we are then able to reduce the semantic security of the resulting bounded-collusion IBE system to the semantic security of the underlying PKE system. We also investigate using a linear combination of the published keys as the user-specific key. By taking a linear combination instead of a subset, we are able to achieve smaller public parameters than a strictly combinatorial approach would allow. In this case, the challenge will be to prove that the resulting cyptosystem is secure even in the presence of a specified number of colluding users. For this, we rely on an algebraic hash proof property.

Each of our constructions allows the total number of potential identities to be exponential in the size of the public parameters of the IBE.

To explain our results in the context of the known literature, let us quickly review some relevant highlights in the history of IBEs. The Identity-Based Encryption model was conceived by Shamir in the early 1980s [68]. The first constructions were proposed in 2001 by Boneh and Franklin [11] based on the hardness of the bilinear Diffie-Hellman problem and by Cocks [23] based on the hardness of the quadratic residuosity problem. Both works relied on the random oracle model. Whereas the quadratic residuosity problem has been used in the context of cryptography since the early eighties [52], computational problems employing bilinear pairings were at the time of [11] relative newcomers to the field. Indeed, inspired by their extensive usage within the context of IBEs, the richness of bilinear group problems has proved tremendously useful for solving other cryptographic challenges (e.g. in the area of leakage-resilient systems).

Removing the assumption that random oracles exist in the construction of IBEs and their variants was the next theoretical target. A long progression of results ensued. At first, partial success for IBE based on bilinear group assumptions was achieved by producing IBEs in the standard model provably satisfying a more relaxed security condition known as selective security [19, 8], whereas the most desirable of security guarantees is that any polynomial-time attacker who can request secret keys for identities of its choice cannot launch a successful chosen-ciphertext attack (CCA) against a new adaptively-chosen challenge identity. Enlarging the arsenal of computational complexity bases for IBE, Gentry, Peikert, and Vaikuntanathan [47] proposed an IBE based on the intractability of the learning with errors (LWE) problem, still in the random oracle model. Ultimately, fully (unrelaxed) secure IBEs were constructed in the standard model (without assuming random oracles) under the decisional Bilinear Diffie-Hellman assumption by Boneh and Boyen [9] and Waters [74], and most recently under the LWE assumption by Cash, Hofheinz, Kiltz, and Peikert [21] and Agrawal, Boneh, and Boyen [2]. While a standard-model construction of IBE under an "interactive quadratic residuosity" assumption was achieved by Boneh et al [12], constructing a fully secure (or even selectively secure) standard-model IBE based on classical number theoretic assumptions such as DDH in non-bilinear groups or the hardness of quadratic residuosity assumptions remains open.

Bounded-collusion IBEs. The security model for IBE assumes that the adversary can adaptively obtain an arbitrary number of secret keys for users in the system, and requires that messages encrypted to any other user still be indistinguishable to the adversary. This models the idea that an individual's messages are still secure even if an arbitrary number of other users of the system collude against that user. As an attempt to come up with constructions under a wider range of assumptions, cryptographers began looking at a variant of IBE known as *Bounded-Collusion IBE* (BC-IBE). In this model, one only guarantees security against an adversary who obtains secret keys associated with at most t identities, where the size of the parameters of the system are allowed to depend on t. Falling short of achieving full security, the bounded-collusion model can be a realistic assumption in many settings, and is in fact a necessary restriction to achieve the more general notion of functional encryption [54]. Additionally, it has been studied in other settings, notably broadcast encryption and revocation (e.g. [37, 38, 39, 59, 61, 56, 30]).

The first construction of BC-IBE came in [33], in the context of their study of the problem of a bounded number of secret key exposures in public-key encryption. To remedy the latter problem, they introduced the notion of *key-insulated* PKE systems and show its equivalence to *IBEs semantically secure against a bounded number of colluding identities.* This equivalence, coupled with constructions of key-insulated PKEs by [33], yields a generic combinatorial construction which converts any semantic secure PKE to a bounded-collusion semantic secure IBE, without needing a random oracle. This paper gave a general reduction from any semantically secure public-key cryptosystem to a BC-IBE scheme. However, their construction suffers from a large ciphertext-size blowup – the resulting ciphertext length is a factor $\omega(t)$ larger than that of the underlying encryption scheme.

PKE systems with key homomorphism. In recent years, several PKE schemes were proposed with interesting homomorphisms over the public keys and the underlying secret keys. These were constructed for the purpose of showing circular security and leakage resilience properties. In particular, for both the scheme of Boneh, Halevi, Hamburg, and Ostrovsky [13] and the scheme of Brakerski and Goldwasser [16], it can be shown that starting with two valid (public-key, secret-key) pairs $(pk_1, sk_1), (pk_2, sk_2)$, one can obtain a third valid pair as $(pk_1 \cdot pk_2, sk_1 + sk_2)$. Similar homomorphisms can be considered for other cryptosystems as well, such as the classic ElGamal cryptosystem [35] and the encryption scheme of Regev [64].

New Results In this section, we seek for generic constructions of BC-IBE. In doing so, we use the key-homomorphic properties above in order to achieve small ciphertext size. We explore systems which rely on encryption schemes that solely satisfy the *standard* security notion of semantic security in addition to some syntactical, non-security-related, properties which can be easily verified. In addition, we see if efficiency can be improved by starting from stronger security assumptions. Our constructions have the added benefit of conceptual simplicity, and the resulting instantiations from concrete assumptions either outperform or abstract existing BC-IBE constructions along different axes.

In summary, we make several main contributions:

1. We give a generic approach (the GLW construction) that can be used to construct BC-IBE with short ciphertexts from any PKE satisfying certain syntactic properties. We prove *selective* security of this construction assuming semantic security of the PKE scheme using a mapping ϕ satisfying *cover-freeness* (a notion introduced in [36] and used in several other works, e.g. [59, 27, 32], among others). While being strictly weaker than the notion of full security, selective security is sufficient for some applications, as discussed below.

2. Whenever the underlying semantically-secure scheme satisfies an additional new property – which we call *weak multi-key malleability* – we prove that the GLW construction achieves *full* BC-IBE security, i.e., confidentiality holds even with respect an identity chosen adaptively after learning the parameters of the schemes as well as secret keys for at most t other identities. Roughly, our malleability property states that given the encryption of $c = \mathsf{Enc}(\mathsf{pk}, m)$ of an *unknown* message m under a known public-key pk, and given an additional public-key / secret-key pair $(\mathsf{pk}', \mathsf{sk}')$, we can efficiently produce a ciphertext which is indistinguishable from an encryption of m under $\mathsf{pk} \cdot \mathsf{pk}'$. An example scheme with this property is ElGamal encryption – hence we directly obtain a DDH-based BC-IBE scheme from ElGamal encryption.

3. We provide a new, alternative construction that relies on a different form of malleability (which we simply call multi-key malleability), and does not require any explicit key-homomorphic structure. Intuitively, our notion requires that given $c = \mathsf{Enc}(\mathsf{pk}, m)$ for an unknown message m, and another public key pk', we can obtain a new ciphertext c which decrypts to m under a combination of the secret keys sk and sk' associated with pk and pk'. We provide an efficient instantiation based on NTRU [57], exploiting its multi-key homomorphic properties recently observed by Lopez-Alt *et al.* [60]. This is of particular interest due to the fact that no fully-secure NTRU-based IBE scheme is known to date. Moreover, our constructions support homomorphic evaluation of ciphertexts, and this is the only construction of identity-based fully homomorphic encryption beyond the recent result by Gentry, Sahai, and Waters [48].

4. Finally, we give another construction of BC-IBE given a PKE that exhibits key-homomorphic structure and satisfies a security notion called *linear related-key security*, a new notion that is slightly stronger than standard semantic security. At a high level, this property entails that a cryptosystem remains secure even if an adversary obtains several keys that have a known relationship with the target secret key, as long as the target key is linearly independent of the others. We use an algebraic technique to prove this property in several cryptosystems that is reminiscent of hash proofs [28]. By assuming this stronger security property, we are able to use a t-wise linearly independent mapping ϕ instead of a cover-free map, resulting in smaller public parameter size.

In each of these constructions, the ciphertext size is small (comparable to a single ciphertext of the underlying public-key system), and we provide concrete instantiations based on several common hardness assumptions (such as DDH, QR, LWE, and NTRU).

Construction	Assumptions	Ciphertext size	PK size
[33]	Semantic-secure PKE	$\Theta(t \log \lvert\mathcal{ID}\rvert)$ PKE ciphertexts	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ PKE PKs
[51]	PKE w/linear hash proof and key homomorphism	Same as underlying PKE	$\Theta(t \log \lvert\mathcal{ID}\rvert)$ PKE PKs
[72]	Semantic-secure PKE; key homomorphism, weak multi-key malleability	Same as underlying PKE	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ PKE PKs
[72]	Semantic-secure PKE; multi-key malleability	Same as underlying PKE	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ PKE PKs
[33]	DDH	3 group elements	$\Theta(t)$ group elements
[51]	DDH	3 group elements	$\Theta(t \log \lvert\mathcal{ID}\rvert)$ group elements
[72]	DDH	2 group elements	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ group elements
[51]	QR	2 RSA group elements	$\Theta(t \log \lvert\mathcal{ID}\rvert)$ group elements
[72]	LWE	Same as GPV [47]	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ GPV PKs
[72]	NTRU	Same as NTRUEncrypt [57]	$\Theta(t^2 \log \lvert\mathcal{ID}\rvert)$ NTRU PKs

Table 2.1: **Comparison with previous works on BC-IBE.** Here t is the collusion parameter and $|\mathcal{ID}|$ is the total number of identities in the system. PK and ciphertext size implicitly include the security parameter. The above portion of the table considers generic constructions, whereas the lower section describes existing constructions from concrete assumptions. Note that linear hash proof property implies semantic security, while being strictly stronger than it.

FROM IBE TO CCA-SECURITY. A somewhat related problem is that of building bounded-CCA secure public-key encryption [27]: Concretely, for t-bounded CCA security, semantic security must hold also for attackers which can decrypt up to t ciphertexts other than the challenge ciphertext for which we attempt to break confidentiality. We note that by re-interpreting a result of Boneh *et al.* [10], every construction of a BC-IBE scheme *selectively* secure against t-collusions directly yields a t-bounded CCA secure PKE. Hence, our BC-IBE constructions also directly yield better bounded-CCA-secure constructions, in terms of ciphertext size and/or conceptual simplicity. When applying our framework to the ElGamal scheme, for example, we obtain a construction which is equivalent to the one proposed in [27], for which a direct security proof was given. Moreover, our instantiation from NTRU is indeed more efficient than the best fully CCA-secure construction from NTRU given by Steinfeld *et al.* [71].

2.2 Overview of the Techniques

The basic idea is to exploit homomorphism over the keys in a PKE system PKE. The high-level overview is as follows.

Start with a PKE PKE with the following properties:

1. The secret keys are vectors of elements in a ring R with operations $(+, \cdot)$ and the public keys consist of elements in a group G.

2. If (pk_1, sk_1) and (pk_2, sk_2) are valid keypairs of PKE and $a, b \in R$, then $ask_1 + bsk_2$ is also a valid secret key of PKE, with a corresponding public key that can be efficiently computed from pk_1, pk_2, a, b. (For the schemes we present, this public key will usually be computed as $pk_1^a \cdot pk_2^b$).

We note that many existing cryptosystems have this property, or can be made to have this property with trivial modifications, including [13], [16], and [28].

The trusted master authority in an IBE will then choose n pairs of (pk_i, sk_i) $(i = 1, ..., n)$ using the key generation algorithm of PKE, publish the n pk_i values, and keep secret the corresponding n sk_i's. Each identity is mapped to a vector $\phi(ID) \in R^n$ (we abuse terminology slightly here since R is only required to be a ring and not a field, but we will still call these "vectors"). The secret key for the identity is computed as a coordinate-wise linear combination of the vectors $sk_1, \ldots, sk_n$, with coefficients $id_1, \ldots, id_n$ respectively, i.e.

$$\mathrm{SK}_{ID} := \sum_{i=1}^{n} (sk_i \cdot \phi(ID)[i])$$

where all additions take place in R.

Anyone can compute the matching public key PK_{ID} using the key homomorphism and the published pk_i values. Since by the key homomorphism (PK_{ID}, SK_{ID}) is still a valid key pair for the original PKE, encryption and decryption can function identically

Game CPA for $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:	**Game** CCA for $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:
$(\mathsf{pk}, \mathsf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}$	$(\mathsf{pk}, \mathsf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}$
$b \overset{\$}{\leftarrow} \{0,1\}$	$b \overset{\$}{\leftarrow} \{0,1\}$
$(m_0, m_1, \mathsf{st}) \overset{\$}{\leftarrow} \mathbf{A}(\mathsf{pk})$	$(m_0, m_1, \mathsf{st}) \overset{\$}{\leftarrow} A^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk})$
$c^* \overset{\$}{\leftarrow} \mathsf{Enc}(\mathsf{pk}, m_b)$	$c^* \overset{\$}{\leftarrow} \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \overset{\$}{\leftarrow} \mathbf{A}(c^*, \mathsf{st})$	$b' \overset{\$}{\leftarrow} \mathbf{A}^{\mathsf{Dec}(\mathsf{sk},\cdot)}(c^*, \mathsf{st})$
Win iff $b' = b$	Win iff $b' = b$.

Figure 2-1: **Semantic Security for Public-Key Encryption.** Security games defining semantic security of public-key encryption against chosen-plaintext attacks (left) and against chosen-ciphertext attacks (right). On the right, we tacitly assume that in the second phase of the game, the decryption oracle $\mathsf{Dec}(\mathsf{sk}, \cdot)$ answers to queries c^* with $\perp$. Also, in both games, the challenge ciphertext c^* is set to $\perp$ if $|m_0| \neq |m_1|$.

to before. The encryptor simply runs the encryption algorithm for PKE using PK_{ID}, and the decryptor runs the decryption algorithm for PKE using SK_{ID}.

The details come in the choice of mapping function ϕ as well as the specifics of the properties required by PKE. Different mapping functions yield different parameters (in terms of e.g. public parameter size), and induce different security requirements on PKE. At the same time, our different constructions demonstrate multiple different syntactic properties of PKE, each of which is sufficient to construct a BC-IBE scheme.

2.3 Preliminaries

SECURITY OF PKE. We define *security against chosen-plaintext attacks* (for short, *IND-CPA security*) [52, 6] for a PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ via the security game depicted on the left of Figure 2-1. It involves an adversary $\mathbf{A}$ which is initially given the public key pk, and subsequently outputs a pair of equal-length messages m_0, m_1. The adversaries continues after receiving a *challenge ciphertext* $c^* \overset{\$}{\leftarrow} \mathsf{Enc}(\mathsf{pk}, m_b)$ for a random secret bit b, and then finally outputs a guess b' for b. We say that PKE is (τ, ε)*-ind-cpa-secure* if all attackers $\mathbf{A}$ with time complexity at most τ guess the right bit (i.e., $b' = b$) with probability at most $\frac{1+\varepsilon}{2}$. Moreover, it is simply *ind-cpa secure* if for all polynomials p, there exists a negligible function ν such that the scheme is $(p(k), \nu(k))$-ind-cpa-secure for all values of the security parameter k. We also consider *security against chosen ciphertext attacks* (for short, *IND-CCA security*), where the adversary is additionally able to decrypt ciphertexts under the constraint that a decryption query for the challenge ciphertext is never asked. (The associated game is on the right of Figure 1.) We say that PKE is (τ, t, ε)-ind-cca-secure if any attacker with time complexity τ and making at most t decryption queries guesses b with probability at most $\frac{1+\varepsilon}{2}$. The asymptotic notion of *t-ind-cca-secure* is defined accordingly.

2.3.1 Identity-based Encryption

Recall that an *identity-based encryption (IBE)* scheme for identity set $\mathcal{ID}$ is a 4-tuple of algorithms $\mathsf{IBE} = (\mathsf{IBEGen}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ satisfying the following syntactical properties:

- IBEGen is the randomized *parameter generator algorithm* which returns a pair $(\mathsf{msk}, \mathsf{pp})$, where msk is the so-called *master secret key*, and pp are the *public parameters*.

- The *extraction algorithm* $\mathsf{IBEExtract}$, on input the master secret-key msk and a valid identity $\mathrm{ID} \in \mathcal{ID}$ returns a secret key $\mathsf{sk}_{\mathrm{ID}} \overset{\$}{\leftarrow} \mathsf{IBEExtract}(\mathsf{msk}, \mathrm{ID})$ associated with this identity.

- The encryption algorithm IBEEnc takes as inputs the public parameters pp, an identity $\mathrm{ID} \in \mathcal{ID}$, and a message m, and returns a ciphertext $c \overset{\$}{\leftarrow} \mathsf{IBEEnc}(\mathsf{pp}, \mathrm{ID}, m)$ such that for the associated deterministic algorithm IBEDec, $\mathsf{IBEDec}(\mathsf{sk}_{\mathrm{ID}}, \mathsf{IBEEnc}(\mathsf{pp}, \mathrm{ID}, m)) = m$ with overwhelming probability for each $(\mathsf{pp}, \mathsf{msk})$ output by Gen and sk_{ID} output by $\mathsf{IBEExtract}(\mathsf{msk}, \mathrm{ID})$.

The notion of IND-CPA security is extended to the setting of IBE. The adversary, given the public parameters pp, can obtain keys $\mathsf{sk}_{\mathrm{ID}}$ for identities ID of its choice (via so-called *extraction queries*), and outputs at some point a pair of equal-length challenge messages m_0, m_1, together with a *challenge identity* ID^* for which no extraction query has been issued. It then obtains an encryption of m_b for the challenge identity ID^* and for a random bit b. The adversary is asked to guess b, constrained on not asking a key extraction query for ID^*. The game is given in Figure 2-2, on the right. We also consider a weaker security notion, called *selective IND-CPA security*, for which the corresponding security game is given on the left of Figure 2-2. Here, the adversary is required to choose its challenge identity *beforehand*, and only subsequently learns the public parameters and is given access to the $\mathsf{IBEExtract}$ oracle.

In analogy to the case of conventional PKE, we say that IBE is (τ, t, ε)-cpa-secure if all τ-time adversaries $\mathbf{A}$ making t extraction queries output b with probability at most $\frac{1+\varepsilon}{2}$ in the CPA-security game above. Similarly, we define (τ, t, ε)-selective-cpa-secure likewise for the selective-CPA game above, as well as the asymptotic notions of t-cpa and t-selective-cpa security.

2.4 From PKE to Bounded-Collusion IBE: General Conditions and Construction 1

We start with a public key scheme and an efficiently computable mapping ϕ on identities that jointly have the following useful properties. We separate the public keys of the PKE into public parameters (distributed independently of the secret key) and user-specific data; the latter is referred to as the "public key".

```
Game selective-CPA for IBE:
$(\mathsf{pp}, \mathsf{msk}) \overset{\$}{\leftarrow} \mathsf{IBEGen}$
$b \overset{\$}{\leftarrow} \{0,1\}$
$(\mathrm{ID}^*, \mathsf{st}) \overset{\$}{\leftarrow} \mathbf{A}$
$(m_0, m_1, \mathsf{st}') \overset{\$}{\leftarrow} \mathbf{A}^{\mathsf{IBEExtract}(\mathsf{msk},\cdot)}(\mathsf{pp}, \mathsf{st})$
$c^* \overset{\$}{\leftarrow} \mathsf{IBEEnc}(\mathsf{pp}, \mathrm{ID}^*, m_b)$
$b' \overset{\$}{\leftarrow} \mathbf{A}^{\mathsf{IBEExtract}(\mathsf{msk},\cdot)}(c^*, \mathsf{st}')$
```

```
Game CPA for IBE:
$(\mathsf{pp}, \mathsf{msk}) \overset{\$}{\leftarrow} \mathsf{IBEGen}$
$b \overset{\$}{\leftarrow} \{0,1\}$
$(m_0, m_1, \mathrm{ID}^*, \mathsf{st}) \overset{\$}{\leftarrow} \mathbf{A}^{\mathsf{IBEExtract}(\mathsf{msk},\cdot)}(\mathsf{pp})$
$c^* \overset{\$}{\leftarrow} \mathsf{IBEEnc}(\mathsf{pp}, \mathrm{ID}^*, m_b)$
$b' \overset{\$}{\leftarrow} \mathbf{A}^{\mathsf{IBEExtract}(\mathsf{msk},\cdot)}(c^*, \mathsf{st})$
```

Figure 2-2: **Semantic Security for IBE.** Security games defining semantic security of identity-based encryption, for selective attacks (left) and for full security (right). In both games, we assume that extraction queries for ID^* are answered by $\perp$, and that $c^* = \perp$ if $|m_0| \neq |m_1|$, or (for full security) if ID^* was previously queried to $\mathsf{Extract}(\mathsf{msk}, \cdot)$.

Secret-key to Public-key Homomorphisms. Throughout this section, we (tacitly) consider only public-key cryptosystems $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the property that secret and public keys are elements of groups G and H, respectively. For convenience and ease of distinction, we will denote the group operations on G and H as $+$ and $\cdot$, respectively.

Definition (Secret-key to public-key homomorphism). *We say that* PKE *admits a* secret-key to public-key homomorphism *if there exists a map* $\mu : G \to H$ *such that:*

- **(i)** *μ is a homomorphism, i.e., for all* $\mathsf{sk}, \mathsf{sk}' \in G$, *we have* $\mu(\mathsf{sk}+\mathsf{sk}') = \mu(\mathsf{sk})\cdot\mu(\mathsf{sk}')$;
- **(ii)** *Every output* $(\mathsf{sk}, \mathsf{pk})$ *of* Gen *satisfies* $\mathsf{pk} = \mu(\mathsf{sk})$.

We stress that we are *not* requiring that every element $\mathsf{sk} \in G$ is a valid secret key output by Gen. This will be important in our LWE instantiation below. In this case, we still want to make sure that decryption is correct: In particular, we say below that μ *satisfies n-correctness* if for any $n' \leq n$ valid secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_{n'}$ output by Gen, the probability $\mathsf{P}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mu(\mathsf{sk}), m)) \neq m]$ is negligible for all messages m, where the probability is over the coins of Enc and where $\mathsf{sk} = \mathsf{sk}_1 + \cdots + \mathsf{sk}_{n'}$. (This property is implicitly satisfied for all n if all elements of G are valid secret keys.)

Also note that the map μ does *not* need to be efficiently computable for our applications, even though the map is often very efficient. Additionally, we observe that in case the scheme depends on some explicit public parameter (like a generator or a matrix, as will be the case in our examples below), μ is indeed allowed to be parameter-dependent.

The GLW Construction. We first present a generic approach to build a bounded-collusion secure IBE from a public-key encryption scheme admitting a secret-key to public-key homomorphism (based on work with Goldwasser and Lewko [51]). Specifically, let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be such a public-key encryption scheme with homomorphism $\mu : G \to H$ satisfying n-correctness, and let $\phi : \mathcal{ID} \to \{0,1\}^n$ be a polynomial-time computable function, called the *identity map.* (With a slight abuse of notation, it will be convenient to consider the output ϕ as a subset of $\{1, \ldots, n\}$, encoded in the canonical way as an n-bit string.) Then, the GLW construction for PKE and ϕ gives rise to the following IBE scheme $\mathsf{IBE} = (\mathsf{IBEGen}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ with identities from the set $\mathcal{ID}$ defined as follows:

$\underline{\mathsf{IBEGen}}$	$\underline{\mathsf{IBEExtract}(\mathsf{msk} = \mathbf{sk}, \mathrm{ID})}$	$\underline{\mathsf{IBEEnc}(\mathsf{pp} = \mathbf{pk}, \mathrm{ID}, m)}$	$\underline{\mathsf{IBEDec}(\mathsf{sk}_{\mathrm{ID}}, c)}$
$(\mathbf{pk}, \mathbf{sk}) \xleftarrow{\$} \mathsf{Gen}^n$ $\mathsf{msk} \leftarrow \mathbf{sk}$ $\mathsf{pp} \leftarrow \mathbf{pk}$ Return $(\mathsf{msk}, \mathsf{pp})$	$\mathsf{sk}_{\mathrm{ID}} = \sum_{i \in \phi(\mathrm{ID})} \mathbf{sk}[i]$ Return $\mathsf{sk}_{\mathrm{ID}}$	$\mathsf{pk}_{\mathrm{ID}} = \prod_{i \in \phi(\mathrm{ID})} \mathbf{pk}[i]$ $c \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_{\mathrm{ID}}, m)$ Return c	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}_{\mathrm{ID}}, c)$ Return m'

The notation $(\mathbf{pk}, \mathbf{sk}) \xleftarrow{\$} \mathsf{Gen}^n$ denotes running Gen n times, with independent random coins, and $\mathbf{pk}, \mathbf{sk}$ are vectors such that $(\mathbf{pk}[i], \mathbf{sk}[i])$ is the output of the i-th execution of Gen. First note that correctness of IBE follows trivially from the correctness of PKE and the existence of a secret-key to public-key homomorphism μ with n-correctness, since $\mathsf{pk}_{\mathrm{ID}} = \mu(\mathsf{sk}_{\mathrm{ID}})$ holds for all IDâĂŹs and $\mathsf{sk}_{\mathrm{ID}}$ is the sum of at most n valid secret keys. We stress that a central advantage of the above construction is that IBE ciphertexts are ciphertexts of the underlying encryption scheme PKE. Also, note that if PKE relies on some public parameters, these are generated once and used across all uses of Gen, Enc, and Dec.

Instantiating the identity map. We still need to discuss how the map ϕ is instantiated; different instantiations will require different properties of the underlying PKE scheme, and will result in different parameters for the BC-IBE. In the first few constructions of this paper, we rely on constructions based on *cover-free sets*, following previous work on bounded-collusion IBE [33], bounded-CCA security [27], and bounded security for FDH signatures [32]. Concretely, let $2^{[n]}$ be the set of subsets of $[n] := \{1, \ldots, n\}$.

Definition (Cover-free sets). *We say that $\phi : \mathcal{ID} \to 2^{[n]}$ is (t, s)-*cover free *if $|\phi(x)| = s$ for all $x \in \mathcal{ID}$, and moreover $\phi(x_t) \setminus \bigcup_{i=1}^{t-1} \phi(x_i) \neq \emptyset$ for all distinct $x_1, \ldots, x_t \in \mathcal{ID}$, i.e., the set $\phi(x_t)$ is not covered by the union of $\phi(x_1), \ldots, \phi(x_{t-1})$.*

In general, we will equivalently think of ϕ as a map $\mathcal{ID} \to \{0,1\}^n$, where we output the characteristic vector of the associated set, instead of the set itself. The following gives the currently best-known construction of cover-free sets.

Theorem 1 ([27]). *For all integers $t \geq 1$, there exists a polynomial-time computable (t, s)-cover-free map $\phi : \mathcal{ID} \to \{0,1\}^n$, where $n = 16t^2 \log |\mathcal{ID}|$ and $s = 4t \log |\mathcal{ID}|$.*

In Section 2.6 we will use a weaker requirement of ϕ that only requires linear independence of the vectors $\phi(x_1), \ldots, \phi(x_t)$. In this case, the output length n can be reduced to $O(t \log |\mathcal{ID}|)$, or even $O(t)$ if we allow both identities as well as components of $\phi(x)$ to be elements of $\mathbb{Z}_p$ for some large prime p. The properties and parameters of this alternate choice of ϕ, along with the resulting implications for the PKE used, will be discussed in that section.

2.4.1 Security

Selective Security of the GLW Construction We start with selective security, which will be important to obtain bounded CCA-secure cryptosystems with short ciphertexts, as we explain below in Section 2.7. In the following, let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an arbitrary public-key encryption scheme which admits secret-key to public-key homomorphism, and let IBE be the IBE scheme resulting from the above construction, using an underlying identity map ϕ.

Theorem 2 (Selective ID Security of GLW). *Assume that* PKE *is ind-cpa-secure, and that* ϕ *is* $(t+1, s)$*-cover free. Then, the GLW construction is* t*-selective-cpa-secure.*

Proof. Let $\mathbf{A}$ be a selective-cpa adversary for IBE which outputs $b' = b$ with probability at least $(1 + n\varepsilon)/2$, and which makes at most t extraction queries. We construct an ind-cpa adversary $\mathcal{B}$ for PKE from $\mathbf{A}$, guessing the bit b with probability $\frac{1+\varepsilon}{2}$. Concretely, the adversary $\mathcal{B}$ first runs $\mathbf{A}$, obtaining the challenge identity ID^*, and chooses an index i^* uniformly at random from the set $S^* = \{i : \mathsf{id}^*_i = 1\}$, where $\phi(\mathrm{ID}^*) = [\mathsf{id}^*_1, ..., \mathsf{id}^*_n]$. It then gets a public key pk^* from the underlying CPA game, and computes $(\mathbf{pk}[j], \mathbf{sk}[j]) \overset{\$}{\leftarrow} \mathsf{Gen}$ for all $j \in [n] \setminus \{i^*\}$. Finally, it sets $\mathbf{pk}[i^*] = \mathsf{pk}^* \cdot \left(\prod_{j \neq i^*} \mathbf{pk}[j]^{-\mathsf{id}^*_j}\right)$.

The adversary $\mathcal{B}$ then gives $\mathsf{pp} = \mathbf{pk}$ to $\mathbf{A}$ and runs it until it outputs a pair (m_0, m_1). In particular, $\mathbf{A}$'s extraction queries for $\mathrm{ID} \neq \mathrm{ID}^* \in \mathcal{ID}$ are replied by computing $[\mathsf{id}_1, \ldots, \mathsf{id}_n] = \phi(\mathrm{ID})$ and, if $\mathsf{id}_{i^*} = 0$, returning $\mathsf{sk}_{ID} := \sum_i \mathsf{id}_i \cdot \mathbf{sk}[i]$. Note that if $\mathsf{id}_{i^*} = 1$, then $\mathcal{B}$ cannot answer the extraction query, as it does not know any corresponding $\mathbf{sk}[i^*]$. In this case, it returns $\perp$, and sets a flag bad to $\mathtt{true}$. When the adversary $\mathbf{A}$ outputs a pair (m_0, m_1) of messages of equal length, $\mathcal{B}$ forwards them to the CPA, obtaining a challenge ciphertext c^* , which it then gives back to $\mathbf{A}$, and its simulated execution is continued until it outputs a bit b'. To conclude, $\mathcal{B}$ outputs the bit b' if bad is not set to $\mathtt{true}$, and returns a random bit otherwise. Note that we have $\mathsf{pk}_{\mathrm{ID}^*} = \mathsf{pk}^*$ by our definition.

Since ϕ is $(t+1, s)$-cover-free, we know that there exists at least one i^* such that $\mathsf{id}^*_{i^*} = 1$, but $\mathsf{id}_{i^*} = 0$ for all vectors $\phi(\mathrm{ID})$ corresponding to the (at most t) extraction queries $\mathrm{ID} \neq \mathrm{ID}^*$. Intuitively, such an index i^* is chosen hence with probability at least $1/|S^*| = 1/s \geq 1/n$, and conditioned on this, the simulation is easily seen to be perfect. Formally, we let $\mathsf{Win}_{\mathsf{PKE}}$ and $\mathsf{Win}_{\mathsf{IBE}}$ be the events that $\mathcal{B}$ and $\mathbf{A}$ guess the bit in the respective security games. Then,

$$\begin{aligned}\mathsf{P}[\mathsf{Win}_{\mathsf{PKE}}] &= \mathsf{P}[\mathsf{Win}_{\mathsf{PKE}} \wedge \neg\mathsf{bad}] + \mathsf{P}[\mathsf{Win}_{\mathsf{PKE}} \wedge \mathsf{bad}] \\ &\geq \mathsf{P}[\neg\mathsf{bad}] \cdot \mathsf{P}\left[\mathsf{Win}_{\mathsf{PKE}} \,\middle|\, \neg\mathsf{bad}\right] + \mathsf{P}[\mathsf{bad}] \cdot \mathsf{P}\left[\mathsf{Win}_{\mathsf{PKE}} \,\middle|\, \mathsf{bad}\right] .\end{aligned}$$

Now, clearly, $\mathsf{P}[\mathsf{bad}] = 1 - \mathsf{P}[\neg\mathsf{bad}]$, and $\mathsf{P}\left[\mathsf{Win}_{\mathsf{PKE}} \,\middle|\, \mathsf{bad}\right] \geq \frac{1}{2}$, since $\mathcal{B}$ outputs a random bit if bad is $\mathtt{true}$. Moreover, one can verify that $\mathsf{P}[\neg\mathsf{bad}] \geq \frac{1}{n}$, and, as the simulation is perfect, $\mathsf{P}\left[\mathsf{Win}_{\mathsf{PKE}} \,\middle|\, \neg\mathsf{bad}\right] = \mathsf{P}[\mathsf{Win}_{\mathsf{IBE}}]$. Formalizing these last two argument actually requires some (standard) extra work, using the fact that all random coins are independent of the choice of i^*, but we dispense with the details. Plugging in terms into the above concludes the proof. □

2.4.2 Full Security of GLW

We note that the above proof strategy used in Theorem 2 fails when we do not know the challenge identity ID* at the point in time when the reduction $\mathcal{B}$ sets the public parameters pp. However, an additional syntactic requirement on the underlying cryptosystem PKE yields full security, as we show below. This requirement is captured by the following definition.

Definition (Weak Multi-Key Malleability). *We say that* PKE *is* weakly n-key malleable *if there exists an efficient algorithm* $\mathsf{Simulate}$ *such that for all messages m, all $I \subseteq [n]$, and all $i \in I$, the probability distributions D_0 and D_1 are computationally indistinguishable, where with $(\mathbf{pk}, \mathbf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}^n$, D_b consists of $(\mathbf{pk}, \mathbf{sk}[[n] \setminus \{i\}], c_b)$ such that*

(1) $c_0 \overset{\$}{\leftarrow} \mathsf{Enc}(\prod_{i \in I} \mathbf{pk}[i], m)$;

(2) $c \overset{\$}{\leftarrow} \mathsf{Enc}(\mathbf{pk}[i], m)$, $c_1 \overset{\$}{\leftarrow} \mathsf{Simulate}(i, I, c, \mathbf{pk}, \mathbf{sk}[[n] \setminus \{i\}])$.

In other words, given a ciphertext c encrypting with public key $\mathbf{pk}[i]$ (where i is part of some set I) an arbitrary *unknown* message m, we can efficiently generate a ciphertext c' encrypting the same message m under the *product* of the keys $\mathbf{pk}[j]$ for $j \in I$ without knowing the secret key $\mathbf{sk}[i]$, but still possibly using $\mathbf{sk}[j]$ for $j \neq i$. The resulting ciphertext has the right distribution in the eyes of a computationally bounded distinguisher.

Theorem 3 (Full Security of GLW). *Assume that* PKE *is ind-cpa-secure and weakly n-key malleable, and that ϕ is $(t+1, s)$-cover free. Then, the GLW construction is t-cpa-secure.*

Proof Sketch. We only sketch the reduction – the analysis is similar to the one in the proof of Theorem 2. Let $\mathbf{A}$ be an ibe-cpa adversary for IBE which guesses the underlying bit b with probability at least $(1 + n \cdot \varepsilon)/2$, and which makes at most t extraction queries. We construct an ind-cpa adversary $\mathcal{B}$ for PKE from $\mathbf{A}$, winning with probability at least $\frac{1+\varepsilon}{2}$. Concretely, the adversary $\mathcal{B}$ starts by choosing an index i^* uniformly at random from the set $[n]$. It then gets a public key pk^* from the underlying IND-CPA game, and computes $(\mathbf{pk}[j], \mathbf{sk}[j]) \overset{\$}{\leftarrow} \mathsf{Gen}$ for all $j \in [n] \setminus \{i^*\}$. Finally, it sets $\mathbf{pk}[i^*] = \mathsf{pk}^*$. It then gives $\mathsf{pp} = \mathbf{pk}$ to $\mathbf{A}$ and starts its execution, until it outputs a target identity ID* as well as a message pair (m_0, m_1). In particular, $\mathbf{A}$'s extraction queries for ID $\in \{0, 1\}$ are replied by computing $[\mathsf{id}_1, \ldots, \mathsf{id}_n] = \phi(\mathrm{ID})$ and, if $\mathsf{id}_{i^*} = 0$, returning $\mathsf{sk}_{\mathrm{ID}} = \sum_i \mathsf{id}_i \mathbf{sk}[i]$. Note that if $\mathsf{id}_{i^*} = 1$, then $\mathcal{B}$ cannot answer the extraction query. In this case, it returns $\perp$, and sets the flag bad to $\mathtt{true}$. Given ID* and (m_0, m_1), let $\phi(\mathrm{ID}^*) = [\mathsf{id}_1^*, \ldots, \mathsf{id}_n^*]$. If $\mathsf{id}_{i^*}^* = 0$, then bad is set to $\mathtt{true}$, and $\perp$ is returned to $\mathbf{A}$. Otherwise, $\mathcal{B}$ forwards (m_0, m_1) to the ind-cpa game, obtaining a challenge ciphertext c. Given this, it outputs $c^* \overset{\$}{\leftarrow} \mathsf{Simulate}(i^*, \phi(\mathrm{ID}^*), c, \mathbf{pk}, \mathbf{sk}[[n] \setminus \{i^*\}])$, which it then gives back to $\mathbf{A}$, continuing it execution, until $\mathbf{A}$ outputs a bit b'. To conclude, $\mathcal{B}$ outputs the bit b' if bad is $\mathtt{false}$, and returns a random bit otherwise. Unlike in Theorem 2, this simulation is not (necessarily) perfect, since we only require computational indistinguishability from $\mathsf{Simulate}$. However, this will only affect $\mathbf{A}$'s output with negligible probability (or else $(\mathcal{B}, \mathbf{A})$ would serve as an efficient distinguisher for the output of $\mathsf{Simulate}$), and thus $\mathcal{B}$'s overall success probability is still nonnegligibly greater than $\frac{1}{2}$. □

2.4.3 Instantiation from DDH

We present a simple instantiation of the above paradigm based on the DDH assumptions and the ElGamal cryptosystem. The resulting scheme has smaller ciphertexts than earlier BC-IBE schemes [51, 33], both requiring *three* group elements.

Concretely, let G be a group with prime order $|G| = q$. Recall that the *Decisional Diffie-Hellman (DDH)* assumption demands that the distributions (g, g^a, g^b, g^{ab}) (for $g \xleftarrow{\$} G$, $a, b \xleftarrow{\$} \mathbb{Z}_q$) and (g, g^a, g^b, g^c) (where $c \xleftarrow{\$} \mathbb{Z}_q$) are computationally indistinguishable. For the same group G, the *ElGamal cryptosystem* has as a public parameter an element $g \xleftarrow{\$} G$, secret key $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_q$, and public key $\mathsf{pk} = g^{sk}$. For a message $m \in G$, the encryption algorithm is $\mathsf{Enc}(\mathsf{pk}, m) = (g^r, m \cdot \mathsf{pk}^r)$, where $r \xleftarrow{\$} Z_q$, whereas $\mathsf{Dec}(\mathsf{sk}, (c_1, c_2)) = c_2 \cdot c_1^{-\mathsf{sk}}$. ElGamal is easily shown to be ind-cpa-secure under the DDH assumption. Moreover, we observe the following two properties of the ElGamal cryptosystem:

1. ElGamal admits a secret-key to public-key homomorphism $\mu : \mathbb{Z}_q \to G$ where $\mu(x) = g^x$, and n-correctness is satisfied for any n.

2. Moreover, it satisfies (perfect) weak n-key malleability: Namely, just consider the algorithm that for all $I \subseteq [n]$, $i \in I$, and secret- and public-key vectors $\mathbf{sk}$ and $\mathbf{pk}$, outputs

$$c^* = \mathsf{Simulate}(i, I, \mathbf{pk}, \mathbf{sk}[[n] \setminus \{i\}], (c_1, c_2)) = (c_1, c_2 \cdot c_1^{\sum_{j \neq i} \mathbf{sk}[j]}) \ . \tag{2.1}$$

In particular, the resulting IBE scheme with identities $\mathcal{ID}$ obtained by plugging ElGamal into the GLW construction, for any $(t+1, s)$-cover-free map $\phi : \mathcal{ID} \to \{0,1\}^n$, is as follows, and Theorem 3 implies its t-ibe-cpa security under the DDH assumption. (The decryption algorithm remains the same as in the original ElGamal scheme.)

$\underline{\mathsf{IBEGen}}$	$\underline{\mathsf{IBEExtract}(\mathsf{msk} = \mathbf{sk}, \mathrm{ID})}$	$\underline{\mathsf{IBEEnc}(\mathsf{pp} = (g, \mathbf{pk}), \mathrm{ID}, m)}$
$g \xleftarrow{\$} G$	$[\mathsf{id}_1, \ldots, \mathsf{id}_n] \leftarrow \phi(\mathrm{ID})$	$[\mathsf{id}_1, \ldots, \mathsf{id}_n] \leftarrow \phi(\mathrm{ID})$
$\mathbf{sk} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{pk}[i] \leftarrow g^{\mathbf{sk}[i]}$	$\mathsf{sk}_{\mathrm{ID}} \leftarrow \sum_{i=1}^n \mathsf{id}_i \cdot \mathbf{sk}[i]$	$r \xleftarrow{\$} \mathbb{Z}_q$
$\mathsf{pp} \leftarrow (g, \mathbf{pk})$, $\mathsf{msk} \leftarrow \mathbf{sk}$	Return $\mathsf{sk}_{\mathrm{ID}}$	$c \leftarrow (g^r, m \cdot \prod_{i=1}^n \mathbf{pk}[i]^{r \cdot \mathsf{id}_i})$
Return $(\mathsf{pp}, \mathsf{msk})$		Return c

2.4.4 Instantiation from LWE

We now turn to a somewhat more involved example based on the GPV cryptosystem [47]. For ease of exposition, we omit a too-detailed discussion of parameters in the following.

THE LWE ASSUMPTION. Let us first recall the *learning with errors (LWE)* problem, introduced by Regev [64]. Let n, q be parameters. For any noise distribution χ on $\mathbb{Z}_q$, and vector $\mathbf{s} \in \mathbb{Z}_q^n$, the oracle $\mathsf{LWE}_{q,n,\chi}(\mathbf{s})$ samples a fresh random n-dimensional vector $\mathbf{a} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$, as well as noise $e \overset{\$}{\leftarrow} \chi$, and returns $(\mathbf{a}, \mathbf{a}^T\mathbf{s}+e)$. The *LWE assumption* with noise χ states that for every PPT distinguisher D,

$$\mathsf{P}\left[\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n : D^{\mathsf{LWE}_{q,n,\chi}(\mathbf{s})} = 1\right] - \mathsf{P}\left[\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n : D^{\mathsf{LWE}_{q,n,U}(\mathbf{s})} = 1\right] = \mathsf{negl}(n) \,, \qquad (2.2)$$

where U is the uniform distribution on $\mathbb{Z}_q$. (In other words, in addition to $\mathbf{a}$, the oracle $\mathsf{LWE}_{q,n,U}(\mathbf{s})$ simply returns uniform random samples, independent of $\mathbf{s}$.) In general, the error distribution χ is chosen to be a discrete Gaussian on $\mathbb{Z}_q$.

THE GPV CRYPTOSYSTEM. The following is a variant of the cryptosystem suggested by Gentry, Peikert, and Vaikuntanathan [47] (and is in fact the dual of Regev's PKE [64]): For a public random parameter $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m\times n}$, where $m \geq n\cdot\log q+\omega(\log n)$, the secret key $\mathsf{sk} = \mathbf{s}$ consists of a vector $\mathbf{s} \overset{\$}{\leftarrow} \{0,1\}^n$ (i.e., the secret key is selected among *binary* vectors), whereas the public key is $\mathsf{pk} = \mathbf{As}$. Then, encryption of a message $b \in \{0,1\}$ is by first computing $\mathbf{r} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$, $\mathbf{e} \overset{\$}{\leftarrow} \chi^n$, $e' \overset{\$}{\leftarrow} \chi'$ (for some distribution χ' related to χ and to be specified below), and then outputting the ciphertext

$$\mathbf{c} = (\mathbf{r}^T\mathbf{A} + \mathbf{e}^T, \mathbf{r}^T\mathsf{pk} + e' + b\cdot(q-1)/2) \,.$$

For decryption of a ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ one takes the secret key $\mathbf{s}$, compute $\mathbf{c}_2 - \mathbf{c}_1\mathbf{s}$, and checks whether the outcome is closer to 0 or $(q-1)/2$ (modulo q). The noise distribution must guarantee that this is indeed true with overwhelming probability. We omit the details of this discussion, but this essentially accounts to showing that $e' - \mathbf{e}^T\mathbf{s}$ is not too large.

One can show the above cryptosystem to be secure if the LWE assumption with distributions χ and χ' is true. (Note that the original GPV cryposystem has $\chi = \chi'$, but this distinction will be necessary for our analysis below.)

SECRET-KEY TO PUBLIC-KEY HOMOMORPHISM. In order to build an IBE scheme via the GLW-construction, we first observe that the cryptosystem admits a secret-key to public-key homomorphism $\mu : \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ such that $\mu(\mathbf{s}) = \mathbf{As}$. Note that for any two valid secret keys $\mathsf{sk}, \mathsf{sk}' \in \{0,1\}^n$ it is *not* necessarily true that $\mathsf{sk} + \mathsf{sk}'$ is still a valid secret key. However, for any ℓ, it is still true that μ satisfies ℓ-correctness as long as χ and χ' are appropriately bounded.

WEAK MULTI-KEY-MALLEABILITY. For weak ℓ-key malleability, we specify the algorithm $\mathsf{Simulate}$ such that, for all $I \subseteq [\ell]$, $i \in I$, $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$, $\mathbf{sk} = (\mathbf{s}_1, , ..., \mathbf{s}_\ell)$ and $\mathbf{pk} = [\mathbf{As}_1, ..., \mathbf{As}_\ell]$, we have

$$\mathbf{c} = \mathsf{Simulate}(i, I, \mathbf{pk}, \mathbf{sk}[[\ell] \setminus \{i\}], (\mathbf{c}_1, \mathbf{c}_2)) = \left(\mathbf{c}_1, \mathbf{c}_2 + \mathbf{c}_1 \cdot \sum_{j \in I\setminus\{i\}} \mathbf{s}_j\right) \,.$$

Note that in contrast to the above DDH-based example, simulation is not perfect. Indeed, the output of $\mathsf{Simulate}$ can indeed be rewritten as

$$\overline{\mathbf{c}}_1 = \left(\mathbf{r}^T\mathbf{A} + \mathbf{e}^T, \mathbf{r}^T\mathbf{A}\sum_{j\in I}\mathbf{s}_j + \mathbf{e}^T\sum_{j\in I\setminus\{i\}}\mathbf{s}_j + e' + b(q-1)/2)\right) .$$

whereas the term $\mathbf{e}^T\sum_{j\in I\setminus\{i\}}\mathbf{s}_j$ is missing in the real ciphertext $\overline{\mathbf{c}}_0$.

However, statistical indistinguishability of $(\mathbf{pk}, \mathbf{sk}[[\ell] \setminus \{i\}], \overline{\mathbf{c}}_b)$ for $b = 0, 1$ is achieved by choosing χ' to be a distribution with a much larger variance than χ. If elements sampled by χ are bounded by B with overwhelming probability, then $\mathbf{e}^T\sum_{j\in I\setminus\{i\}}\mathbf{s}_j$ is at most $|I|{\cdot}n{\cdot}B$. We know that if χ' is a discrete Gaussian distribution with standard deviation βq, then the statistical distance of χ' and $\chi' + \mathbf{e}^T\sum_{j\in I\setminus\{i\}}\mathbf{s}_j$ is at most $|I| \cdot n \cdot B/(\beta q)$ [31, Lemma 3]. Thus, we wish to choose q large enough such that this factor is negligible, yet the LWE problem with distributions χ and χ' is still hard. If we choose $q = 2^{n^{\varepsilon}}$, $\beta = 2^{n^{-\varepsilon/2}}$, and $B = 2^{n^{\varepsilon/4}}$ for some constant $\varepsilon > 0$, and $|I| = \mathsf{poly}(n)$, we can make the statistical distance smaller than any inverse polynomial in n while retaining subexponential hardness in the LWE assumption. We can thus reduce the security of this PKE scheme to the hardness of subexponential approximations of certain lattice problems [63].

The final LWE-based IBE Scheme. Consequently, every $(s, t+1)$-cover-free map $\phi : \mathcal{ID} \to \{0,1\}^{\ell}$, every $n \geq m\log q + \omega(n)$, and noise distributions χ, χ' as above yield the following scheme with identity set $\mathcal{ID}$, which, by Theorem 3, is t-ibe-cpa secure under the LWE assumption for distribution χ:

$\underline{\mathsf{IBEGen}}$	$\mathsf{IBEExtract}(\mathsf{msk} = \mathbf{sk}, \mathrm{ID})$	$\mathsf{IBEEnc}(\mathsf{pp} = (\mathbf{A}, \mathbf{pk}), \mathrm{ID}, b)$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ $\mathbf{sk}[1], \ldots, \mathbf{sk}[\ell] \xleftarrow{\$} \mathbb{Z}_q^n$ $\mathbf{pk}[i] \leftarrow \mathbf{A}\,\mathbf{sk}[i]\ (i = 1, \ldots, \ell)$ $\mathsf{pp} \leftarrow (\mathbf{A}, \mathbf{pk})$, $\mathsf{msk} \leftarrow \mathbf{sk}$ Return $(\mathsf{pp}, \mathsf{msk})$	$[\mathsf{id}_1, \ldots, \mathsf{id}_\ell] \leftarrow \phi(\mathrm{ID})$ $\mathsf{sk}_{\mathrm{ID}} \leftarrow \sum_{i=1}^{\ell} \mathsf{id}_i \cdot \mathbf{sk}[i]$ Return $\mathsf{sk}_{\mathrm{ID}}$	$[\mathsf{id}_1, \ldots, \mathsf{id}_n] \leftarrow \phi(\mathrm{ID})$ $\mathsf{pk} = \sum_{i=1}^{\ell} \mathsf{id}_i \cdot \mathbf{pk}[i]$ $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^m$, $\mathbf{e} \xleftarrow{\$} \chi^n$, $e' \xleftarrow{\$} \chi'$ $\mathbf{c}_1 \leftarrow \mathbf{r}^T \cdot \mathbf{A} + \mathbf{e}$ $\mathbf{c}_2 \leftarrow \mathbf{r}^T\mathsf{pk} + e' + b{\cdot}(q-1)/2$ Return c

Also, we note that if we are only interested in proving selective security using Theorem 2, then weak ℓ-key malleability is unnecessary, and we can fix $\chi = \chi'$. This allows to choose a polynomial modulus, avoiding the subexponential LWE assumption.

2.5 Construction 2: Multi-Key Malleability

2.5.1 Bounded-IBE Construction

We present a further construction of BC-IBE from PKE schemes which satisfy a different notion of key malleability than the one given above, which we first introduce. Our notion requires that given an encryption of a message under one public key, we are asking for the ability to produce a new ciphertext of the same message which decrypts under a combination of secret keys (e.g., the product) for which we only know the corresponding public keys. Note that we are only asking for decryptability under the combination of the secret keys. In particular, in contrast to the above notion of weak key-malleability, the distribution of the resulting ciphertext may not be a valid encryption under some well-defined combination of the corresponding public keys, and moreover, we require ability to compute this ciphertext without knowledge of any secret keys.

Definition (Multi-Key Malleability). *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme. We say that* PKE *is* n-key malleable *if there exist algorithms* Modify *and* $\mathsf{Combine}$ *such that the following properties hold:*

(i) *For all valid messages* m*, all* $I \subseteq [n]$*, and all* $i \in I$*, the following probability is negligible (taken over the coins of* Enc*):*

$$\mathsf{P}\left[\begin{matrix}(\mathbf{pk}, \mathbf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}^n, c \overset{\$}{\leftarrow} \mathsf{Enc}(\mathbf{pk}[i], m), \\ c' \overset{\$}{\leftarrow} \mathsf{Modify}(i, I, \mathbf{pk}, c)\end{matrix} \quad : \mathsf{Dec}(\mathsf{Combine}(I, \mathbf{sk}), c') \neq m\right] .$$

(ii) *For all* $I \subseteq [n]$*,* $\mathsf{Combine}(I, \mathbf{sk})$ *does* not *depend on* $\mathbf{sk}[i]$ *for* $i \notin I$*.*

(iii) *For all* $I \subseteq [n]$ *and all valid public-key / secret-key vectors* $(\mathbf{pk}, \mathbf{sk})$*, for all* $i, j \in I$*, the values* $\mathsf{Modify}(i, I, \mathbf{pk}, \mathsf{Enc}(\mathbf{pk}[i], m))$ *and* $\mathsf{Modify}(j, I, \mathbf{pk}, \mathsf{Enc}(\mathbf{pk}[j], m))$ *are equally distributed.*

We note that Property **(iii)** above is not really necessary (a computational relaxation would suffice), but will make the presentation somewhat simpler and is true in the only instantiation we give below.

THE IBE CONSTRUCTION AND ITS SECURITY. For an identity map $\phi : \mathcal{ID} \to \{0,1\}^n$, we now propose a construction of an identity-based encryption scheme $\mathsf{IBE} = (\mathsf{IBEGen}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ from an n-key malleable encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The decryption algorithm is unaltered, i.e., $\mathsf{IBEDec} = \mathsf{Dec}$, and moreover the construction consists of the following algorithms. (Note that the choice of i as $\min\{\phi(\mathrm{ID})\}$ below within IBEEnc is purely arbitrary.)

$\underline{\mathsf{IBEGen}}$	$\underline{\mathsf{IBEExtract}(\mathsf{msk} = \mathbf{sk}, \mathrm{ID})}$	$\underline{\mathsf{IBEEnc}(\mathsf{pp} = \mathbf{pk}, \mathrm{ID}, m)}$
$(\mathbf{pk}, \mathbf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}^n$ $\mathsf{msk} \leftarrow \mathbf{sk}$ $\mathsf{pp} \leftarrow \mathbf{pk}$ Return $(\mathsf{msk}, \mathsf{pp})$	$\mathsf{sk}_{\mathrm{ID}} \leftarrow \mathsf{Combine}(\phi(\mathrm{ID}), \mathbf{sk})$ Return $\mathsf{sk}_{\mathrm{ID}}$	$i \leftarrow \min\{\phi(\mathrm{ID})\}$ $c' \overset{\$}{\leftarrow} \mathsf{Enc}(\mathbf{pk}[i], m)$ $c \leftarrow \mathsf{Modify}(i, \phi(\mathrm{ID}), \mathbf{pk}, c')$ Return c

Correctness of the scheme follows by Property **(i)** above. The following theorem establishes security of our new construction.

Theorem 4. *Assume that* PKE *is ind-cpa-secure and* n*-key malleable, and that* ϕ *is* $(t+1, s)$*-cover free. Then,* IBE *is* t*-ibe-cpa-secure.*

Proof. Let $\mathbf{A}$ be an ibe-cpa-adversary for IBE making t extraction queries and which succeeds with probability $\frac{1+n\varepsilon}{2}$ in guessing the right bit b. We build an ind-cpa adversary $\mathcal{B}$ for PKE as follows: The adversary $\mathcal{B}$ initially chooses an index $i^* \xleftarrow{\$} [n]$, and is then given pk^*. It sets $\mathbf{pk}[i^*] = \mathsf{pk}$, and then samples $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathsf{Gen}$ for all $i \neq i^*$, and then runs $\mathbf{A}$ with public parameters $\mathsf{pp} = \mathbf{pk}$. Then, $\mathbf{A}$'s extraction queries for ID are simulated as follows: If $i^* \notin \phi(\mathrm{ID})$, then it returns $\mathsf{Combine}(\phi(\mathrm{ID}), \mathbf{sk})$. (Note that this can be done by Property (ii), since the output of $\mathsf{Combine}$ does not depend on $\mathsf{sk}[i^*]$.) Else it returns $\perp$ if $i^* \in \phi(\mathrm{ID})$, and sets a flag bad to $\mathtt{true}$. Moreover, on input a triple $(m_0, m_1, \mathrm{ID}^*)$, $\mathcal{B}$ forwards m_0, m_1 to the ind-cpa game, obtaining $c \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}^*, m_b)$, and then, if $i^* \in \phi(ID^*)$ and $c \neq \perp$, it gives $c^* = \mathsf{Modify}(i^*, \phi(\mathrm{ID}^*), \mathbf{pk}, c)$ back to $\mathbf{A}$. If $c = \perp$, then it sets $c^* = \perp$. Otherwise, it gives simply $\perp$ back and sets bad to $\mathtt{true}$. Finally, $\mathcal{B}$ outputs $\mathbf{A}$'s final output b' if bad was never set, and a random bit is returned otherwise.

We now turn to the analysis of the success probability of $\mathcal{B}$ in winning the ind-cpa game for PKE. Let bad be the event that the bad flag is set, and let good be its complement. First note that $\mathsf{P}\left[b' = b \,\middle|\, \mathsf{good}\right] \geq \frac{1+n\cdot\varepsilon}{2}$, because as long as the bad flag is never set, all extraction queries have been replied as in the original ibe-cpa game. Also note that $\mathsf{P}\left[b' = b \,\middle|\, \mathsf{bad}\right] \geq 1/2$, since in this case $\mathcal{B}$ outputs a uniform bit b'.

It remains to prove a lower bound on the probability of good happening. Note that since ϕ is $(t+1, s)$-cover-free, then there must exist an index $i \in \phi(\mathrm{ID}^*)$ such that $i \notin \phi(\mathrm{ID})$ for all (at most t) extraction queries $\mathrm{ID} \neq \mathrm{ID}^*$. If i^* takes such a value, then the bad flag is never set, i.e., good holds. It is not hard to show that the probability that such an index is hit is at least $\frac{1}{n}$, even though this requires some (standard) work (which we omit) due to the fact that i^* is chosen *before* $\mathbf{A}$'s execution starts. To conclude, we obtain

$$\mathsf{P}\left[b' = b\right] > \frac{1}{n} \cdot \frac{1+n\varepsilon}{2} + \left(1 - \frac{1}{n}\right) \cdot \frac{1}{2} = \frac{1+\varepsilon}{2} ,$$

which contradicts ind-cpa security of PKE. □

2.5.2 NTRU-Based Instantiation and Fully-Homomorphic IBE

We provide an instantiation of the above constructing using the multi-key homomorphic properties of NTRU-based public-key encryption [60], which we first review. For some parameters r, n and q (where q is a prime), consider the ring of polynomials $R = \mathbb{Z}[x]/(x^r + 1)$, and let χ be a B-bounded distribution on R, i.e., with overwhelming probability, χ samples a polynomial from R whose coefficients are all at most B in absolute value. All operations on polynomials are to be understood as over the ring $R_q = R/qR$. The NTRU cryptosystem is such that key generation Gen samples $f, g \xleftarrow{\$} \chi$ subject to the constraint that $f \equiv 1 \pmod 2$, and sets $\mathsf{pk} = 2g/f$ and $\mathsf{sk} = f$. (Possibly, f needs to be resampled until it admits an inverse in R_q, and χ is such that this happens with good probability.) The message $b \in \{0, 1\}$ is encrypted as

$$\mathsf{Enc}(\mathsf{pk}, m) = h \cdot \mathsf{pk} + 2e + b ,$$

where $h, e \overset{\$}{\leftarrow} \chi$. Finally, decryption, given c, outputs $\mathsf{Dec}(\mathsf{sk}, c) = \mathsf{sk} \cdot c \pmod 2$. To see why decryption is correct, note that

$$\mathsf{sk} \cdot c \equiv f \cdot (2h \cdot g/f + 2e + b) \equiv 2h \cdot g + 2e \cdot f + f \cdot b \pmod q .$$

If $B \leq \sqrt{q/2}/r$, then all coefficients from $h \cdot g$ and $e \cdot f$ are of size at most $r^2B^2 < q/2$. Consequently, $2hg$ and $2ef$ only have even coefficients, and are 0 modulo 2. And finally, $f \cdot b$ clearly always equals b modulo 2.

The scheme was proven ind-cpa-secure under a fairly ad-hoc assumption in [60], where it was also shown to have strong homomorphic properties we address below, and which we exploit for our construction.

The IBE Scheme. We turn now to building an IBE scheme from the above NTRU-based PKE scheme PKE using the above generic approach. In the following, we assume that r is our security parameter, $q = 2^{n^\varepsilon}$ for some constant $\varepsilon < 1$, $B = \mathsf{poly}(r)$, and $n = \Theta(r^\delta)$ for some constant $\delta < 1$.

We first show ℓ-key malleability exploiting the multi-key homomorphic properties of NTRU shown in [60]. To this end, we define the algorithm $\mathsf{Combine}$ which given $I \subseteq [\ell]$ and $\mathbf{sk} \in R_q^\ell$ outputs

$$\mathsf{Combine}(I, \mathbf{sk}) = \prod_{i \in I} \mathbf{sk}[i] .$$

Moreover, we also define the (randomized) function Modify, which given $I \subseteq [\ell]$, $i \in I$, $c \in R_q$, and $\mathbf{pk} \in R_q^\ell$, outputs

$$\mathsf{Modify}(i, I, c, \mathbf{pk}) = c + \sum_{j \in I \setminus \{i\}} h_j \cdot \mathbf{pk}[j] ,$$

where h_j for $j \in I \setminus \{i\}$ are sampled independently from the B-bounded distribution χ as above. Now, Properties (ii) and (iii) in Definition 2.5.1 are immediate to verify. Moreover, for Property (i), fix $I \subseteq [\ell]$ and $i \in I$, and $\mathbf{pk}, \mathbf{sk} \in R_q^\ell$, each consisting of ℓ B-bounded polynomials as components, then define c as

$$c = \mathsf{Modify}(i, I, \mathsf{Enc}(\mathbf{pk}[i], b), \mathbf{pk}) = \sum_{j \in I} h_j \cdot \mathbf{pk}[j] + 2e + b ,$$

and observe that

$$\mathsf{Dec}(\mathsf{Combine}(I, \mathbf{sk}), c) = \left(\prod_{i \in I} \mathbf{sk}[i]\right) \cdot \left(\sum_{j \in I} h_j \cdot \mathbf{pk}[j] + 2e + b\right) \pmod 2 .$$

In particular,

$$\left(\prod_{i \in I} \mathbf{sk}[i]\right) \cdot \left(\sum_{j \in I} h_j \cdot \mathbf{pk}[j] + 2e + b\right) \equiv \sum_{j \in I} 2h_j \cdot g_j \cdot \prod_{i \in I \setminus \{j\}} f_\ell + \left(2e \cdot \prod_{i \in I} f_\ell\right) + b \cdot \left(\prod_{i \in I} f_\ell\right) .$$

Note that in the above sum, only products of at most $|I|+1$ B-bounded polynomials occurs. The coefficients of the resulting products have size at most $r^{|I|} \cdot B^{|I|+1}$, which (given previous parameter choices) is smaller than $q/2$ as long as $|I| = o(n^{\varepsilon})$. This yields correct decryption as no wraparound (modulo q) occurs.

THE FINAL SCHEME. Overall, this yields to the following scheme, for any identity mapping $\phi : \mathcal{ID} \to \{0,1\}^{\ell}$ which is $(s, t+1)$-cover-free for some $s = o(n^{\varepsilon})$, which is t-ind-cpa secure by Theorem 4.

IBEGen	IBEExtract(msk $= \mathbf{sk}$, ID)	IBEEnc(pp $= \mathbf{pk}$, ID, m)
$f_1, \dots, f_n \xleftarrow{\$} \chi$, $f_i \equiv 1 \pmod 2$, $f_i \in R_q^*$ $g_1, \dots, g_n \xleftarrow{\$} \chi$ msk $\leftarrow (f_1, \dots, f_n)$ pp $\xleftarrow{\$} (2g_1/f_1, \dots, 2g_n/f_n)$ Return (msk, pp)	$\mathsf{sk}_{\mathrm{ID}} \leftarrow \prod_{i \in \phi(\mathrm{ID})} \mathbf{sk}[i]$ Return $\mathsf{sk}_{\mathrm{ID}}$	$h_1, \dots, h_n, e \xleftarrow{\$} \chi$ $c \leftarrow \sum_{i \in \phi(\mathrm{ID})} \mathbf{pk}[i] \cdot h_i + 2e + m$ Return c

FULLY-HOMOMORPHIC IBE. The above instantiation has additionally the property of being fully-homomorphic in the following sense: Given encryptions IBEEnc(ID, m_1), ..., IBEEnc(ID, m_t), and a function $f : \{0,1\}^t \to \{0,1\}$, we can compute a ciphertext which decrypts to $f(m_1, \dots, m_t)$ under $\mathsf{sk}_{\mathrm{ID}}$ using the homomorphic-evaluation procedures given in [60].

We note that in general one can provide a construction, along the lines given above, from multi-key fully-homomorphic encryption to fully-homomorphic identity-based encryption for bounded collusions. We defer a full discussion,noting in passing that the above is the only instantiation of this paradigm we are aware of.

2.6 Construction 3: Smaller Parameters using Linear Independence

In the previous constructions we used a cover-free map ϕ to map identities to *subsets* of a set of keys (expressed using the 0-1 characteristic vector). However, since we have a homomorphism over the keys, we can alternately use a different ϕ that maps identities to *linear combinations* of keys. Instead of the cover-freeness property (where a group of t keys obtained by the adversary does not "cover" any honest user's key), we will require ϕ to be *t-wise linearly independent.* That is, a group of t keys obtained by the adversary, viewed as t vectors of coefficients of the underlying keys, does not contain any honest user's key in its span. As we will see, this will enable smaller public parameters than constructions using cover-free maps, at the cost of stricter security requirements on the underlying PKE.

2.6.1 Linear Related-Key Security

We define the stronger notion of security we require, which we call linear related-key security. Here we assume that the secret keys are linear combinations of underlying

```
Game LRKA for PKE = (Gen, Enc, Dec)
$(\mathsf{pk}_1, \mathsf{sk}_1), ...(\mathsf{pk}_n, \mathsf{sk}_n) \xleftarrow{\$} \mathsf{Gen}^n$
$b \xleftarrow{\$} \{0, 1\}$
$(m_0, m_1, \vec{v}, \mathsf{st}) \xleftarrow{\$} \mathbf{A}^{LinComb(\vec{\mathsf{sk}}, \cdot)}(\vec{\mathsf{pk}})$
$c^* \xleftarrow{\$} \mathsf{Enc}(\prod_{i=1}^{n} v_i \mathsf{pk}_i, m_b)$
$b' \xleftarrow{\$} \mathbf{A}(c^*, \mathsf{st})$
Win iff $b' = b$ and $\vec{v} \notin span(queries(\mathbf{A}))$
```

```
Oracle $LinComb(\vec{\mathsf{sk}}, \cdot)$
On input $\vec{v} \in R^n$ ($R$ is the ring of $\mathsf{sk}$s):
Return $\prod_{i=1}^{n} v_i \mathsf{sk}_i$
```

Figure 2-3: **Linear Related-Key Security for Public-Key Encryption.** Security game defining linear related-key security of public-key encryption against chosen-plaintext attacks (left) and against chosen-ciphertext attacks (right). Secret keys are now linear combinations of values in the secret key space; the adversary is allowed to request arbitrary linear combinations as well as specifying the linear combination corresponding to the challenge key. Once again, the challenge ciphertext c^* is set to $\perp$ if $|m_0| \neq |m_1|$.

hidden values. The adversary is allowed to request arbitrary linear combinations of these values, as well as specifying which combination to use as the challenge key.

If the coefficients in each of these linear combinations are viewed as vectors of elements in the secret key space, then we want ciphertexts to remain indistinguishable if they are encrypted to a vector that is outside of the span of the vectors queried by the adversary. A game-based definition is given in Fig. 2-3.

2.6.2 Mapping Identities to Linearly Independent Vectors

To employ our strategy of transforming PKE schemes with homomorphic properties over keys into IBE schemes with polynomial collusion resistance, we first need methods for efficiently mapping identities to linearly independent vectors over various fields. This can be done using generating matrices for the Reed-Solomon codes over $\mathbb{Z}_p$ and dual BCH codes over $\mathbb{Z}_2$.

Lemma 5. *For any prime p and any $t+1 < p$, there exists an efficiently-computable mapping $\phi : \mathbb{Z}_p \to \mathbb{Z}_p^{t+1}$ such that for any distinct $x_1, x_2, ...x_{t+1} \in \mathbb{Z}_p$, the vectors $\phi(x_1), \phi(x_2), ...\phi(x_{t+1})$ are linearly independent.*

Proof. Let $\phi(x) = (1, x, x^2, ...x^t)$. Clearly, this is efficiently computable.

Consider the matrix formed by taking any $t+1$ distinct elements $x_1, x_2, ...x_{t+1}$ of Z_p and letting their images under ϕ form the rows. This is a Vandermonde matrix; it therefore has a well-known formula for the determinant

$$\prod_{1 \leq i < j \leq t+1} (x_j - x_i)$$

which is nonzero modulo p (since the elements are distinct). Thus, the Vandermonde matrix is full rank, and therefore $\phi(x_1), \phi(x_2), ...\phi(x_{t+1})$ are linearly independent over

$\mathbb{Z}_p$. □

Lemma 6. *For any positive integer k and any $t+1 < 2^k$, there exists an efficiently-computable mapping $\phi : \{0,1\}^k \to \{0,1\}^{(t+1)k}$ such that for any distinct $x_1, x_2, ... x_{t+1} \in \{0,1\}^k$, the vectors $\phi(x_1), \phi(x_2), ... \phi(x_{t+1})$ are linearly independent over $\mathbb{Z}_2$.*

Proof. Let $g(x) = (1, x, x^2, ... x^t)$, where multiplication occurs over $GF(2^k)$. Let $\phi(x)$ be the bitwise expansion of $g(x)$ (that is, the output is considered as $k(t+1)$ bits instead of $t+1$ elements of $\{0,1\}^k$). Here, we represent each element $GF(2^k)$ as an element of $\mathbb{Z}_2^k$, and this representation is an additive homomorphism. Clearly, ϕ is efficiently computable.

Again we consider the matrix A formed by taking the images of $t+1$ distinct arbitrary elements $x_1, x_2, ... x_{t+1}$ as rows. Assume that this matrix has rank at most t over $\mathbb{Z}_2$. Then there exists some nonzero linear combination of rows that sums to the zero vector; that is, that there is some nonzero vector v over $\mathbb{Z}_2$ such that $v^{\mathrm{T}} \cdot A = \vec{0}^{\mathrm{T}}$ modulo 2.

Consider the Vandermonde matrix B over $GF(2^k)$ whose rows are $g(x_1)$, $g(x_2)$, $\ldots$, $g(x_{t+1})$. Since addition over $GF(2^k)$ corresponds directly to bitwise addition over $GF(2)^k$, the bitwise computations will be the same when computing $v^{\mathrm{T}} \cdot A$ and $v^{\mathrm{T}} \cdot B$, and we can conclude that $v^{\mathrm{T}} \cdot B = \vec{0}^{\mathrm{T}}$ over $GF(2^k)$ as well. (Here, the 0,1 entries of v are interpreted as 0,1 in $GF(2^k)$). However, this is a contradiction, since B is a square Vandermonde matrix with distinct rows over $GF(2^k)$, and is thus full rank (similarly to the proof of Lemma 5). Therefore, A must be full rank over $\mathbb{Z}_2$, and thus the vectors $\phi(x_1), \phi(x_2), ... \phi(x_{t+1})$ are linearly independent. □

2.6.3 Construction

Given a PKE scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an identity mapping ϕ having the properties defined above, we now construct a bounded-collusion IBE scheme. We let t denote our collusion parameter, and n will be the dimension of the image of ϕ.

$\mathsf{IBEGen}(1^\lambda) \to \mathrm{PP}, \mathrm{MSK}$ The setup algorithm for the IBE scheme calls the key generation algorithm of the PKE scheme to generate n random $sk_1, pk_1, \ldots, sk_n, pk_n$ pairs, sharing the same public parameters. The public parameters PP of the IBE scheme are defined to be these shared public parameters as well as $pk_1, \ldots, pk_n$. The master secret key MSK is the collection of secret keys $sk_1, \ldots, sk_n$.

$\mathsf{IBEExtract}(ID, \mathrm{MSK}) \to \mathrm{SK}_{ID}$ The key generation algorithm takes an identity in the domain of ϕ and first maps it into R^n as $\phi(ID) = (id_1, \ldots, id_n)$. It then computes SK_{ID} as an R-linear combination of $sk_1, \ldots, sk_n$, with coefficients $id_1, \ldots, id_n$: $\mathrm{SK}_{ID} = \sum_{i=1}^{n} id_i sk_i$.

$\mathsf{IBEEnc}(m, \mathrm{PP}, ID) \to \mathrm{CT}$ The encryption algorithm takes in a message in the message space of the PKE scheme. From the public parameters PP, it computes a public key corresponding to SK_{ID} using the linear key homomorphism property (we note that the mapping ϕ is known and efficiently computable). It then runs the PKE encryption algorithm on m with this public key to produce CT.

$\mathsf{IBEDec}(\mathrm{CT}, \mathrm{SK}_{ID}) \rightarrow m$ The decryption algorithm runs the decryption algorithm of the PKE, using SK_{ID} as the secret key.

Theorem 7. *When a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with linear key homomorphism and a compatible $(t+1)$-wise linearly-independent identity mapping ϕ satisfy linear related-key security, then the construction defined in Section 2.6.3 is a CPA-secure bounded-collusion IBE scheme with collusion parameter t.*

Proof. Given the construction in Section 2.6.3, we note that the security game for LRKA-security (Fig. 2-3) exactly corresponds to the security game for IBE CPA-security (Fig. 2-2). That is, the IBE adversary first takes public parameters, which in the construction above simply consist of a vector of public keys (as in LRKA-security). It makes oracle queries for IBE secret keys, which are simply linear combinations of PKE secret keys. The challenge ciphertext is constructed in the same manner.

Finally, note that the IBE adversary is constrained to make at most t oracle queries, which must all be distinct from the challenge identity id^*. Since ϕ is $(t+1)$-wise linearly-independent, it is thus the case that $\phi(id^*) \notin span(queries(\mathcal{A}))$. Thus, we have a direct reduction between the security games; the reduction can essentially pass the oracle and challenge queries of the IBE adversary to the LRKA challenger unaltered, only applying ϕ to translate IBE identities into LRKA vectors. □

2.6.4 Linear Hash Proofs

The major tool we will use to prove linear related-key security is *linear hash proofs*. This technique is inspired by the paradigm of hash proof systems, which were first introduced by Cramer and Shoup as a paradigm for proving CCA security of PKE schemes [28]. Hash proof systems have recently been used in the context of leakage-resilience as well ([62], for example), extending to the identity-based setting in [4]. We note that the primitive of identity-based hash proof systems introduced in [4] takes a different direction than our work, and the instantiation they provide from the quadratic residuosity assumption relies on the random oracle model.

We deviate from the original hash-proof paradigm in several respects. In hash proof systems, a single public key corresponds to many possible secret keys. There are two encryption algorithms: a valid one and an invalid one. Valid ciphertexts decrypt properly when one uses any of the secret keys associated to the public key, while invalid ciphertexts decrypt differently when different secret keys are used. Our linear hash proof property will consider several public keys at once, each corresponding to a set of many possible secret keys. The adversary will be given these public keys, along with some linear combinations of fixed secret keys corresponding to the public keys. We will also have valid and invalid encryption algorithms. Our valid ciphertexts will behave properly. When an invalid ciphertext is formed for a public key corresponding to a linear combination of the secret keys that is *independent* of the revealed combinations, the invalid ciphertext will decrypt "randomly" when one chooses a random key from the set of secret keys that are consistent with the adversary's view.

To define this property more formally, we first need to define some additional notation. We consider a PKE scheme with linear key homomorphism which comes equipped with a compatible identity map ϕ and an additional algorithm InvalidEncrypt which takes in a message and a *secret key sk* and outputs a ciphertext (note that the invalid encryption algorithm does not necessarily need to be efficient). The regular and invalid encryption algorithms produce two distributions of ciphertexts. We call these *valid* and *invalid* ciphertexts. Correctness of decryption must hold for valid ciphertexts.

We let $(sk_1, pk_1), (sk_2, pk_2), \ldots, (sk_n, pk_n)$ be n randomly generated key pairs, where all of $sk_1, \ldots, sk_n$ are d-tuples in a ring R (here we assume that the key generation algorithm chooses R, d and then generates a key pair. We fix R and then run the rest of the algorithm independently n times to produce the n key pairs). We define S to be the $n \times d$ matrix with entries in R whose i^{th} row contains sk_i.

For notational convenience, we also define a (very large) matrix $\Phi \in R^{|\mathcal{ID}| \times n}$. Let the rows of Φ be $\vec{\phi}(1), \ldots, \vec{\phi}(|ID|)$ for our (yet-to-be-defined) identity map ϕ.

Fix any $t+1$ distinct rows of the matrix of identity vectors Φ, denoted by $\vec{\phi}(i_1), \ldots, \vec{\phi}(i_{t+1})$. We let $sk_{ID_{i_{t+1}}}$ denote the secret key $\vec{\phi}(i_{t+1}) \cdot S$ and $pk_{ID_{i_{t+1}}}$ denote the corresponding public key (computed via the key homomorphism). We let $Ker_R(\vec{\phi}(i_1), \ldots, \vec{\phi}(i_t))$ denote the kernel of the $t \times n$ submatrix of Φ formed by these rows; that is, it consists of the vectors $\vec{v} \in R^n$ such that $\vec{\phi}(i_j) \cdot \vec{v} = 0$ for all j from 1 to t.

Now we consider the set of possible secret key matrices given the public and secret key information available to an adversary who has queried identities $i_1, ..., i_t$. We let W denote the set of matrices in $R^{n \times d}$ whose columns belong to $Ker_R(\vec{\phi}(i_1), \ldots, \vec{\phi}(i_t))$ and whose rows w_i satisfy that $sk_i + w_i$ has the same public key as sk_i for all i. Since W's columns are orthogonal to the identity vectors $\vec{\phi}(i_1), \ldots, \vec{\phi}(i_t)$, adding an element of W to S does not change any of the secret keys $\vec{\phi}(i_j)S$. Furthermore, by construction, adding an element of W to S does not change the public keys associated with the scheme.

We define the subset $\tilde{S}$ of $R^{n \times d}$ to be the set of all matrices in $S + W := \{S + W_0 | W_0 \in W\}$, intersected with the set of all matrices of n secret keys that can be generated by the key generation algorithm (i.e. those with components in R'). Intuitively, $\tilde{S}$ is the set of all possible $n \times d$ secret key matrices that are "consistent" with the n public keys $pk_1, \ldots, pk_n$ and the t secret keys $\vec{\phi}(i_1) \cdot S, \ldots, \vec{\phi}(i_t) \cdot S$. In other words, after seeing these values, even an information-theoretic adversary cannot determine S uniquely - only the set $\tilde{S}$ can be determined.

We say that a **PKE scheme with linear key homomorphism is a linear hash proof system with respect to the identity map** ϕ if the following two requirements are satisfied. We refer to these requirements as *uniform decryption of invalid ciphertexts* and *computational indistinguishability of valid/invalid ciphertexts.*

Uniform Decryption of Invalid Ciphertexts Recall that there are many possible secret keys corresponding to each public key. Any of these keys should correctly decrypt a valid ciphertext; however, when a uniformly-selected key is used to decrypt an invalid ciphertext, we want the resulting distribution of decrypted messages to be (close to) uniform over the message space. Formally, given a public key and a ciphertext that is the output of $InvalidEncrypt$, we want the distribution of possible input messages to be statistically close to uniform.

Given ciphertext c^* and public key $pk_{ID_{i_{t+1}}}$, we require that the probability that c^* was created by running $InvalidEncrypt$ on any particular message is statistically close to uniform over the message space given that the secret key is chosen uniformly from $\vec{\phi}(i_{t+1}) \cdot \tilde{S}$. That is, for any $m_0, m_1 \in \mathcal{M}$,

$$\begin{aligned} Pr[m = m_0 | sk \xleftarrow{\$} \vec{\phi}(i_{t+1}) \cdot \tilde{S}, c^* = InvalidEnc_{sk}(m)] \\ \approx_s Pr[m = m_1 | sk \xleftarrow{\$} \vec{\phi}(i_{t+1}) \cdot \tilde{S}, c^* = InvalidEnc_{sk}(m)] \end{aligned}$$

.

Computational Indistinguishability of Valid/Invalid Ciphertexts Second, we require valid and invalid ciphertexts are computationally indistinguishable in the following sense. For any fixed (distinct) $\vec{\phi}(i_1), \ldots, \vec{\phi}(i_{t+1})$, we consider the following game between a challenger and an attacker $\mathcal{A}$:

$Game_{hp}$: The challenger starts by sampling $(sk_1, pk_1), \ldots, (sk_n, pk_n)$ as above, and gives the attacker the public parameters and $pk_1, \ldots, pk_n$. The attacker may adaptively choose distinct rows $\vec{\phi}(i_1), \ldots, \vec{\phi}(i_{t+1})$ in Φ in any order it likes. (For convenience, we let $\vec{\phi}(i_{t+1})$ always denote the vector that will be encrypted under, but we note that this may be chosen before some of the other $\vec{\phi}(i)$'s.) Upon setting an $\vec{\phi}(i_j)$ for $j \neq t+1$, the attacker receives $\vec{\phi}(i_j) \cdot S$. When it sets $\vec{\phi}(i_{t+1})$, it also chooses a message m. At this point, the challenger flips a coin $\beta \in \{0,1\}$, and encrypts m to the public key corresponding to $\vec{\phi}(i_{t+1}) \cdot S$ as follows. We let pk_{ch} denote the public key corresponding to $\vec{\phi}(i_{t+1}) \cdot S$. If $\beta = 0$, it calls Encrypt with m, pk_{ch}. If $\beta = 1$, it calls InvalidEncrypt with $m, \vec{\phi}(i_{t+1}) \cdot S$. It gives the resulting ciphertext to the attacker, who produces a guess β' for β.

We denote the advantage of the attacker by $Adv_{\mathcal{A}}^{hp} = \left|\mathbb{P}[\beta = \beta'] - \frac{1}{2}\right|$. We require that $Adv_{\mathcal{A}}^{hp}$ be negligible for all PPT attackers $\mathcal{A}$.

Theorem 8. *If a PKE scheme* PKE *is a linear hash proof system with respect to identity map ϕ (that is, if it satisfies uniform decryption of invalid ciphertexts and computational indistinguishability of valid and invalid ciphertexts), then it satisfies LRKA-security.*

Proof. We first change from the real security game defined in Section 2.6.1 (Figure 2-3) to a new game $LRKA'$ in which the challenger calls the invalid encryption algorithm to form an invalid ciphertext. We argue that if the adversary's advantage changes by a non-negligible amount, this violates the computational indistinguishability of valid/invalid ciphertexts. To see this, we consider a PPT adversary $\mathcal{A}$ whose advantage changes non-negligibly. We will construct a PPT adversary $\mathcal{A}'$ against Game$_{hp}$. The challenger for Game$_{hp}$ gives $\mathcal{A}'$ the public parameters and $pk_1, \ldots, pk_n$, which $\mathcal{A}'$ forwards to $\mathcal{A}$. When $\mathcal{A}$ requests a secret key for an identity corresponding to $\vec{\phi}(i_j)$, $\mathcal{A}'$ can forward $\vec{\phi}(i_j)$ to its challenger and obtain the corresponding secret key. When $\mathcal{A}$ declares m_0, m_1 and some ID^* corresponding to $\vec{\phi}(i_{t+1})$, $\mathcal{A}'$ chooses a random bit $b \in \{0,1\}$ and sends $m_b, \vec{\phi}(i_{t+1})$ to its challenger. It receives a ciphertext encrypting m_b, which it forwards to $\mathcal{A}$. We note here that the $t+1$ distinct identities chosen by $\mathcal{A}$ correspond to distinct rows of Φ. If the challenger for $\mathcal{A}'$ is calling the regular encryption algorithm, then $\mathcal{A}'$ has properly simulated the real security game for $\mathcal{A}$. If it is calling the invalid encryption algorithm, then $\mathcal{A}'$ has properly simulated the new game, $LRKA'$. Hence, if $\mathcal{A}$ has a non-negligible change in advantage, $\mathcal{A}'$ can leverage this to obtain a non-negligible advantage in Game$_{hp}$.

In $LRKA'$, we argue that information-theoretically, the attacker's advantage must be negligible. We observe that in our definition of the linear hash proof property, the subset $\tilde{S}$ of $R^{n \times d}$ is precisely the subset of possible MSK's that are consistent with the public parameters and requested secret keys that the attacker receives in the game, and each of these is equally likely. Since the invalid ciphertext decrypts to an essentially random message over this set (endowed with the uniform distribution), the attacker cannot have a non-negligible advantage in distinguishing the message. □

2.6.5 QR-based Construction

We now present a PKE scheme with linear key homomorphism and a compatible identity mapping ϕ such that this is a linear hash proof system with respect to ϕ under the quadratic residuosity assumption.

Quadratic Residuosity Assumption We formally state the QR assumption. We let λ denote the security parameter. We let $N = pq$ where p, q are random λ-bit primes. We require $p, q \equiv 3 \pmod 4$, i.e. N is a Blum integer. We let $\mathbb{J}_N$ denote the elements of $\mathbb{Z}_N^*$ with Jacobi symbol equal to 1, and we let $\mathbb{QR}_N$ denote the set of quadratic residues modulo N. Both of these are multiplicative subgroups of $\mathbb{Z}_N^*$, with orders $\frac{\varphi(N)}{2}$ and $\frac{\varphi(N)}{4}$ respectively.[1] We note that $\frac{\varphi(N)}{4}$ is odd, and that -1 is an element of $\mathbb{J}_N$, but is not a square modulo N. As a consequence, $\mathbb{J}_N$ is isomorphic to $\{+1, -1\} \times \mathbb{QR}_N$. We let u denote an element of $\mathbb{QR}_N$ chosen uniformly at random, and h denote an element of $\mathbb{J}_N$ chosen uniformly at random. For any algorithm $\mathcal{A}$, we define the advantage of $\mathcal{A}$ against the QR problem to be:

$$Adv_N^{\mathcal{A}} \left| Pr\left[\mathcal{A}(N,u) = 1\right] - Pr\left[\mathcal{A}(N,h) = 1\right]\right|.$$

We further restrict our choice of N to values such that $\mathbb{QR}_N$ is cyclic. We note that this is satisfied when p, q are strong primes, meaning $p = 2p' + 1, q = 2q' + 1$, where p, q, p', q' are all distinct odd primes. This restriction was previously imposed

[1] Note that here φ denotes Euler's totient function, whereas we will use ϕ for the identity-mapping function.

in [28], where they note that this restricted version implies the usual formulation of the quadratic residuosity assumption if one additionally assumes that strong primes are sufficiently dense. We say that the QR assumption holds if for all PPT $\mathcal{A}$, $Adv_N^{\mathcal{A}}$ is negligible in λ.

Furthermore, we note that this definition is equivalent to one in which $\mathcal{A}$ receives a random element h of $\mathbb{J}_N \backslash \mathbb{QR}_N$ instead of $\mathbb{J}_N$.

QR-based PKE Construction We define the message space to be $\{-1, 1\}$. The public parameters of the scheme are a Blum integer $N = pq$, where primes $p, q \equiv 3 \bmod 4$ and $\mathbb{QR}_N$ is cyclic, and an element g that is a random quadratic residue modulo N. Our public keys will be elements of $\mathbb{Z}_N$, while our secret keys are elements of the ring $R := \mathbb{Z}$. We define the subset R' to be $[\rho(N)]$. We will later provide bounds for appropriate settings of $\rho(N)$.

- $\mathsf{Gen}(1^\lambda)$: The generation algorithm chooses an element sk uniformly at random in $[\rho(N)]$. This is the secret key. It then calculates the public key as $pk = g^{sk}$.
- $\mathsf{Enc}_{pk}(m)$: The encryption algorithm chooses an odd $r \in [N^2]$ uniformly at random, and calculates $Enc(m) = (g^r, m \cdot pk^r)$.
- $\mathsf{Dec}_{sk}(c_1, c_2)$: The decryption algorithm computes $m = c_2 \cdot (c_1^{sk})^{-1}$.

We additionally define the invalid encryption algorithm:

- $\mathsf{InvalidEnc}_{sk}(m)$: The invalid encryption algorithm chooses a random $h \in \mathbb{J}_N \backslash \mathbb{QR}_N$ (i.e. a random non-square). It produces the invalid ciphertext as $(h, m \cdot h^{sk})$.

Key Homomorphism Considering N, g as global parameters and only $pk = g^{sk}$ as the public key, we have homomorphism over keys through multiplication and exponentiation in G for public keys and arithmetic over the integers for secret keys.

For secret keys $sk_1, sk_2 \in \mathbb{Z}$ and integers $a, b \in \mathbb{Z}$, we can form the secret key $sk_3 := ask_1 + bsk_2$ and corresponding public key $pk_3 = pk_1^a \cdot pk_2^b$ in G.

Compatible Mapping and Resulting IBE Construction Our compatible map ϕ is obtained from Lemma 6 (Section 2.6.2). We may assume that our identities are hashed to $\{0, 1\}^k$ for some k using a collision-resistant hash function, so they are in the domain of ϕ. The image of each identity under ϕ is a vector with 0,1 entries of length $n = k(t + 1)$, where t is our collusion parameter. For every $t + 1$ distinct elements of $\{0, 1\}^k$, their images under ϕ are linearly independent (over $\mathbb{Z}_2$ as well as $\mathbb{Q}$).

A formal description of our construction follows. This is an instance of the general construction in Section 2.6.3, but we state it explicitly here for the reader's convenience. We assume that messages to be encrypted are elements of $\{-1, +1\}$, and identities are elements of $\{0, 1\}^k$. For each identity ID, we let ID^{T} denote the row vector of length n over $\{0, 1\}$ obtained by our mapping from $\{0, 1\}^k$ to binary vectors of length n.

IBEGen The setup algorithm chooses a Blum integer N such that $\mathbb{QR}_N$ is cyclic and a random element $g \in \mathbb{QR}_N$. It then generates n key pairs of the PKE $((pk_1, sk_1), (pk_2, sk_2), ...(pk_n, sk_n))$ using the common g, and publishes the public keys (along with N, g) as the public parameters. The master secret key consists of the corresponding secret keys, $sk_1, \dots, sk_n$. These form an $n \times 1$ vector S with entries in $[\rho(N)]$ (the i^{th} component of S is equal to sk_i for $i = 1 \dots n$).

IBEExtract(ID) The key generation algorithm receives an $ID \in \{0,1\}^k$. By Lemma 6 (Section 2.6.2), we then have a mapping ϕ that takes this ID to a vector $(id_1, id_2, ...id_n)$, such that the vectors corresponding to $t+1$ different ID's are linearly independent. The secret key for ID will be an element of $\mathbb{Z}$, which is computed as a linear combination of the values $sk_1, \dots, sk_n$, with coefficients $id_1, \dots, id_n$ respectively. We express this as $\mathrm{SK}_{ID} := \sum_{i=1}^{n}(sk_i \cdot id_i)$, where the sum is taken over $\mathbb{Z}$. Since the mapping ϕ provided in Section 2.6.2 produces vectors $(id_1, \dots, id_n)$ with 0,1 entries, the value of SK_{ID} is at most $\rho(N)n$. Since n will be much less than $\rho(N)$, this will require roughly $\log \rho(N)$ bits to represent.

IBEEnc(ID, m, PP) We let $\mathrm{PK}_{ID} := \prod_{i=1}^{n}(pk_i^{id_i})$. Anyone can compute this using the multiplicative key homomorphism and the published pk_i values. Since by the key homomorphism $(\mathrm{PK}_{ID}, \mathrm{SK}_{ID})$ is still a valid keypair for the original PKE, encryption and decryption can function as for the PKE. In other words, the encryptor runs the encryption algorithm for the PKE scheme with PK_{ID} as the public key to produce the ciphertext CT.

Note that for ciphertexts, we now have

$$\begin{aligned} Enc_{\mathrm{PK}_{ID}}(m) &= (g^r, m \cdot ((\mathrm{PK}_{ID})^r)) \\ &= \left(g^r, m \cdot \prod_{i=1}^{n}(pk_i^{id_i \cdot r})\right) = \left(g^r, m \cdot \prod_{i=1}^{n} g^{id_i \cdot sk_i \cdot r}\right). \end{aligned}$$

All arithmetic here takes place modulo N.

This can alternately be expressed as: $Enc_{PK_{ID}}(m) = \left(g^r, m \cdot g^{(ID)^{\mathrm{T}} S r}\right)$ where $S = (sk_i)_{n \times 1}$ is a vector over $\mathbb{Z}$ containing the n PKE secret keys of the master secret key.

IBEDec$(\mathrm{CT}, \mathrm{SK}_{ID})$ The decryption algorithm runs the decryption algorithm of the PKE with SK_{ID} as the secret key.

2.6.6 Security of the IBE

We now prove security of IBE scheme up to t collusions. This will follow from Theorem 7 and the theorem below.

Theorem 9. *Under the QR assumption, the PKE construction in Section 2.6.5 is a linear hash proof system with respect to ϕ when $\rho(N)$ is sufficiently large. When $\log(N) = \Omega(n^2 \log n)$, $\rho(N) = N^\ell$ for some constant ℓ suffices.*

We note that when $\rho(N) = N^\ell$, our secret keys are of size $O(\log N) = O(\lambda)$. We prove this theorem in two lemmas.

Lemma 10. *Under the QR assumption, computational indistinguishability of valid and invalid ciphertexts holds.*

Proof. We suppose there exists a PPT adversary $\mathcal{A}$ with non-negligible advantage in Game$_{hp}$. We will create a PPT algorithm $\mathcal{B}$ with non-negligible advantage against the QR assumption. We simplify/abuse notation a bit by letting $\vec{\phi}_1, \dots, \vec{\phi}_{t+1}$ denote the distinct rows of Φ that are chosen adaptively by $\mathcal{A}$ during the course of the game (these were formerly called $\vec{\phi}(i_1), \dots, \vec{\phi}(i_{t+1})$).

$\mathcal{B}$ is given (N, h), where N is a Blum integer such that $\mathbb{QR}_N$ is cyclic and h is either a random element of $\mathbb{J}_N \backslash \mathbb{QR}_N$ or a random element of $\mathbb{QR}_N$. Crucially, $\mathcal{B}$ does not know the factorization of N. $\mathcal{B}$ sets g to be a random element of $\mathbb{QR}_N$.

It chooses an $n \times 1$ vector $S = (sk_i)$, whose entries are chosen uniformly at random from $[\rho(N)]$. For each i from 1 to n, the i^{th} entry of S is denoted by sk_i. It computes $pk_i = g^{sk_i} \bmod N$ and gives the public parameters PP $= (N, g, pk_1, \dots, pk_n)$ to $\mathcal{A}$. We note that $\mathcal{B}$ knows the MSK $= S$, so it can compute $\vec{\phi}_1 \cdot S, \dots, \vec{\phi}_t \cdot S$ and give these to $\mathcal{A}$ whenever $\mathcal{A}$ chooses the vectors $\vec{\phi}_1, \dots, \vec{\phi}_t$.

At some point, $\mathcal{A}$ declares a message m and a vector $\vec{\phi}_{t+1}$ corresponding to identity ID^*. $\mathcal{B}$ encrypts m using the following ciphertext: $\left(h, m \cdot h^{(ID^{*\mathrm{T}})S}\right)$.

We consider two cases, depending on the distribution of h.

Case 1: h is random in $\mathbb{QR}_N$ When h is a random square modulo N, we claim that the ciphertext is properly distributed as a valid ciphertext. More precisely, we claim that the distribution of h and the distribution of g^r for a random odd $r \in [N^2]$ are negligibly close. This follows from the fact that $\mathbb{QR}_N$ is cyclic of order $\frac{\varphi(N)}{4}$, and the reduction of a randomly chosen odd $r \in [N^2]$ modulo $\frac{\varphi(N)}{4}$ will be distributed negligibly close to uniform.

Case 2: h is random in $\mathbb{J}_N \backslash \mathbb{QR}_N$ In this case, $\mathcal{B}$ has followed the specification of the invalid encryption algorithm.

Thus, if $\mathcal{A}$ has a non-negligible advantage in distinguishing between valid and invalid ciphertexts, then $\mathcal{B}$ can leverage $\mathcal{A}$ to obtain non-negligible advantage against the QR assumption. □

Lemma 11. *Uniform decryption of invalid ciphertexts holds when $\rho(N)$ is sufficiently large. When $\log(N) = \Omega(n^2 \log n)$, $\rho(N) = N^\ell$ for some constant ℓ suffices.*

Proof. We choose S with uniformly random entries in $[\rho(N)]$. We then fix any $t+1$ distinct rows of Φ, denoted by $\vec{\phi}_1, \dots, \vec{\phi}_{t+1}$. We must argue that the value of $\vec{\phi}_{t+1} \cdot S$

modulo 2 is negligibly close to uniform, conditioned on $\vec{\phi}_1 \cdot S, \ldots, \vec{\phi}_t \cdot S$ and S modulo $\frac{\varphi(N)}{4}$. To see why this is an equivalent statement of the uniform decryption of invalid ciphertexts property for our construction, note that the decryption of an invalid ciphertext is computed as follows. We let sk denote the secret key the ciphertext was generated with, and sk^* denote another secret key for the same public key used for decryption: $Dec(sk^*, (h, mh^{sk})) = m(-1)^{sk-sk^*}$, since $sk \equiv sk^* \bmod \varphi(N)/4$ in order to both have the same public key. If we think of S as fixed and $\tilde{S}$ as the set of vectors with entries in $[\rho(N)]$ that yield the same values of $\vec{\phi}_1 \cdot S, \ldots, \vec{\phi}_t \cdot S$ and S modulo $\frac{\varphi(N)}{4}$, we can restate our goal as showing that the distribution of $\vec{\phi}_{t+1} \cdot S' \bmod 2$ is negligibly close to uniform, where S' is chosen uniformly at random from $\tilde{S}$.

We know by Lemma 6 that the vectors $\vec{\phi}_1, \ldots, \vec{\phi}_{t+1}$ are linearly independent as vectors over $\mathbb{Z}_2$. This implies that these vectors are linearly independent as vectors over $\mathbb{Q}$ as well. We let $Ker_{\mathbb{Q}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$ denote the $(n-t)$-dimensional kernel of these vectors as a subspace of $\mathbb{Q}^n$.

Our strategy is to prove that this space contains a vector $\vec{p}$ with integer entries that is *not* orthogonal to $\vec{\phi}_{t+1}$ modulo 2. Then, for every S' in $S + W$, $S' + \frac{\varphi(N)}{4}\vec{p}$ is also in $S + W$. Here we are using the notation from Section 2.4 where we defined W. In this instance, $S + W$ is the set of vectors yielding the same values as S for $\vec{\phi}_1 \cdot S, \ldots, \vec{\phi}_t \cdot S$ and S modulo $\frac{\varphi(N)}{4}$. $\tilde{S}$ is then the intersection of $S + W$ with the set of vectors having all of their entries in $[\rho(N)]$.

To complete the argument, we need to prove that for most elements of $S' \in \tilde{S}$ (all but a negligible proportion), $S' + \frac{\varphi(N)}{4}\vec{p}$ will also be in $\tilde{S}$ (i.e. have entries in $[\rho(N)]$). This will follow from showing that there exists a $\vec{p}$ with reasonably bounded entries, and also that the set $\tilde{S}$ contains mostly vectors whose entries stay a bit away from the boundaries of the region $[\rho(N)]$.

We will use the following lemmas.

Lemma 12. *Let A be a $t \times n$ matrix of rank t over $\mathbb{Q}$ with entries in $\{0, 1\}$. Then there exists a basis for the kernel of A consisting of vectors with integral entries all bounded by $n^{\frac{t}{2}} t^{\frac{t}{4}}$.*

Proof. This is an easy consequence of Theorem 2 in [7], which implies the existence of a basis with entries all bounded in absolute value by $\sqrt{det(AA^{\mathrm{T}})}$. We note that AA^{T} is a $t \times t$ matrix with integral entries between 0 and n. Dividing each row by n, we obtain a matrix with rational entries between 0 and 1, and can then apply Hadamard's bound [55] to conclude that the determinant of this rational matrix has absolute value at most $t^{\frac{t}{2}}$. Thus, the determinant of AA^{T} has absolute value at most $n^t t^{\frac{t}{2}}$. Applying Theorem 2 in [7], the lemma follows. □

Lemma 13. *We suppose that M is $d \times n$ matrix with integral entries all of absolute value at most B and rank d over $\mathbb{Q}$. Then there exists another $d \times n$ matrix M' with integral entries of absolute value at most $2^{d-1}B$ that has the same rowspan as M over $\mathbb{Q}$ and furthermore remains rank d when its entries are reduced modulo 2.*

Proof. We provide the following algorithm for obtaining M' from M. This is a variant of Gaussian elimination, tailored to ensure linear independence modulo 2 while avoiding increasing the entries by too much. The main idea is as follows: we start by dividing the first row of M by a power of 2 to ensure that it contains integer entries with at least one of them being odd. We relabel coordinates (swap columns) to move this entry to the top left corner of M. We then proceed to add the first row to the later rows as necessary in order to obtain first column entries which are all even except for the first. We then consider the second row and proceed iteratively.

However, because we sometimes need to divide a row by a power of 2, we design our algorithm to "fix" the earlier entries in a row which were previously set to be even and then become odd upon dividing. To ensure that this process both terminates and does not end up increasing the entries by too much, we alternate between adding and subtracting the higher rows. Our algorithm is formally stated below. We let x_{ij} denote the current value of the i, j entry of our matrix as the algorithm runs. The matrix is initialized to M.

Algorithm

```
For $i = 1$ to $d$:
    Set $z = (z_1, z_2, ...z_{i-1})$ to be a length-$(i-1)$ list of integers, all entries initialized to $-1$.
    While (row $i$ contains all even entries)
        Divide row $i$ by a power of 2 to obtain integer entries, at least one odd.
        For $j = 1$ to $i - 1$:
            If $x_{ij}$ is odd:
                Add $z_j$ times row $j$ to row $i$.
                Set $z_j = -z_j$.
    Row $i$ now has an odd entry in some position $\geq i$. Swap columns so that this entry is in column $i$.
    For $k = i + 1$ to $d$:
        If $x_{ki}$ is odd, add row $i$ to row $k$.
end
```

We now establish the following properties of this algorithm.

Claim 1 Throughout the execution of the algorithm, for all i from 1 to d we have:

$$\max_j |x_{ij}| \leq 2^{i-1}B.$$

We prove this claim by induction on i. For $i = 1$, all the algorithm does is potentially divide the first row by a power of two and swap columns. This will maintain that entries have absolute value at most B. We now assume the claim for $\leq i - 1$ and we show it holds for i.

We note that at the beginning of the while loop for row i, the contents of row i have only been changed by adding in rows j for $j < i$. Since each row has been added at most once and has entries of size at most $2^{j-1}B$, the entries of row i up to the start of the while loop for i are always bounded in absolute value by $B+2^1B+\cdots+2^{i-2}B+B = 2^{i-1}B$, as required.

Now we consider the execution of the while loop for row i. If at the start of the loop row i has an odd entry, then the loop will never run, and the bound therefore holds (note that by construction the first $i-1$ entries of row i will be even, so the odd entry will be in column i or greater as needed).

Otherwise, consider row i after some number of iterations of the while loop. Define w as the total power of 2 divided out of row i over all iterations up to this point (that is, 2^w is the product of all values divided out by the first line of the while loop so far), α_{jk} as the total power of 2 that had been divided out by the k-th time row j was added or subtracted from row i, and k_j as the total number of times row j has been added or subtracted up to the current point. Since row i is divided by at least 2 at the beginning of each iteration, and each of the rows 1 through $(i-1)$ is added or subtracted from row i at most once per iteration of the loop, we know that $\alpha_{jk} < \alpha_{j(k+1)}$ for all k, and that $\alpha_{jk} \leq w$ for all k. Furthermore, we can express element $x_{i\beta}$ (for arbitrary $\beta \in [1,n]$) as

$$x_{i\beta} = \left(m_{i\beta} \pm \sum_{j=1}^{i-1} \left(\sum_{\ell=1}^{k_j} (-1)^\ell 2^{\alpha_{j\ell}} x_{j\beta} \right) \right) / 2^w$$

In this expression, $m_{i\beta}$ denote the original entries of M, and we have accounted for all the changes that occur to row i both before the while loop for i and up to the current point in this loop's execution. Note that $\sum_{\ell=1}^{k_j} (-1)^\ell 2^{\alpha_{j\ell}}$ is an alternating sum of strictly increasing values; thus it is bounded in absolute value by its last term. Thus we have

$$\begin{aligned}
|x_{i\beta}| &\leq \left(|m_{i\beta}| + \sum_{j=1}^{i-1} 2^{\alpha_{jk_j}} |x_{j\beta}| \right) / 2^w \\
&\leq \left(|m_{i\beta}| + \sum_{j=1}^{i-1} 2^w |x_{j\beta}| \right) / 2^w \\
&\leq \left(B + \sum_{j=1}^{i-1} 2^w 2^{j-1} B \right) / 2^w \\
&= B(1 + 2^w(2^{i-1} - 1))/2^w \\
&\leq 2^{i-1} B
\end{aligned}$$

Claim 2 The algorithm terminates in a finite number of steps.

We suppose for contradiction that the algorithm does not terminate. This means the while loop for some row i does not terminate. By Claim 1, the entries of row i remain bounded throughout the process, so there are a finite number of states of row i during the infinite while loop. Hence there must be a cycle of repeating states. We let r_i denote the state of row i at some point in this cycle. The algorithm proceeds by adding/subtracting rows 1 through $i-1$ (which are unchanging throughout this loop) and dividing by powers of 2. If we do not need to divide by any powers of 2, then the while loop will terminate. Since we arrive back at r_i, we must have that r_i is a rational linear combination of itself and the first $i-1$ rows. The coefficient of r_i in this combination will be at most $\frac{1}{2}$, and hence the rational combination of the first $i-1$ rows must be non-zero. This contradicts linear independence of the rows of M over $\mathbb{Q}$. (Note that we never encounter an r_i with all zero entries because of linear independence.)

Claim 1 and Claim 2 show that the algorithm correctly finds a basis modulo 2 with entries bounded by $2^{d-1}B$. We can undo the coordinate-relabeling performed by the algorithm; since the other operations are simply linear combinations of rows, the new matrix spans the same space as M. Thus, we have the desired M'. □

Combining these two lemmas, we may conclude that there exists a basis for $Ker_{\mathbb{Q}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$ with integral entries all having absolute value at most $C := 2^{n-t-1}n^{\frac{t}{2}}t^{\frac{t}{4}}$ that remains of rank $n-t$ when reduced modulo 2. Now, if all of these basis vectors are orthogonal to $\vec{\phi}_{t+1}$ modulo 2, then these form a $(n-t)$-dimensional space that is contained in the kernel of the $(t+1)$-dimensional space generated by $\vec{\phi}_1, \ldots, \vec{\phi}_t, \vec{\phi}_{t+1}$ in $\mathbb{Z}_2^n$. This is a contradiction. Thus, at least one of the basis vectors is not orthogonal to $\vec{\phi}_{t+1}$ modulo 2. Since it is orthogonal to $\vec{\phi}_1, \ldots, \vec{\phi}_t$ over $\mathbb{Q}$ and has integral entries of absolute value at most C, this is our desired $\vec{p}$.

Now, the set of vectors $\tilde{S}$ can be described as the intersection of the set

$$S + \frac{\varphi(N)}{4} Ker_{\mathbb{Z}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$$

with the set of vectors with coordinates all in $[\rho(N)]$, where $Ker_{\mathbb{Z}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$ denotes the vectors in $Ker_{\mathbb{Q}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$ with integral entries. Since we have a bound C on the size of entries an integer basis for the kernel, we can argue that if the coordinates of S are sufficiently bounded away from 0 and $\rho(N)$, then there will be many vectors in $\tilde{S}$, negligibly few of which themselves have entries outside of $(\frac{\varphi(N)}{4}C, \rho(N) - \frac{\varphi(N)}{4}C)$. Both this bound and the probability that S is indeed sufficiently bounded away from 0 and $\rho(N)$ depend only on the relationship between n and $\rho(N)$. We prove the following lemma regarding these parameters:

Lemma 14. *With $\rho(N)$, n, $\vec{p}$, S, and $\tilde{S}$ defined as above, when $\log N = \Omega(n^2 \log n)$, we can set $\rho(N) = N^{\ell}$ for some constant ℓ so that the fraction of $S' \in \tilde{S}$ such that $S' + \frac{\varphi(N)}{4}\vec{p}$ is not also in $\tilde{S}$ is negligible with all but negligible probability over the choice of S.*

Proof. We first derive a lower bound on the number of elements of $\tilde{S}$ that holds with all but negligible probability. We let E denote a positive integer that we will set later so that with all but negligible probability, all of the entries of S lie in the range $(E, \rho(N) - E)$. From now on, we will assume this bound on the entries of S.

We let K be our basis from Lemma 13 multiplied by $\frac{\varphi(N)}{4}$ (so all of these have entries bounded in absolute value by $\frac{\varphi(N)}{4}C$). We note that integral combinations of elements in K will be contained in $\frac{\varphi(N)}{4}Ker_{\mathbb{Z}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$, which is an $(n-t)$-dimensional lattice in $\mathbb{Z}^n$. We let $\vec{\phi}_1, \ldots, \vec{\phi}_{n-t}$ denote vectors in K, and $a_1, \ldots, a_{n-t}$

be integers. We consider

$$S + \sum_{i=1}^{n-t} a_i \vec{k}_i,$$

which is guaranteed to be in $\tilde{S}$ whenever

$$\sum_{i=1}^{n-t} |a_i| \leq \frac{4E}{\varphi(N)C}.$$

We let $D := \frac{4E}{\varphi(N)C}$. The number of $a_1, \ldots, a_{n-t}$ satisfying this condition is lower bounded by the number of ordered $(n-t)$-tuples of non-negative integers that sum to a value $\leq D$. This is calculated as $\binom{D+n-t}{n-t}$. Hence, the number of elements of $\tilde{S}$ is at least:

$$\binom{D+n-t}{n-t} \geq \left(\frac{D}{n-t}\right)^{n-t}. \tag{2.3}$$

Now we provide an upper bound on the number of points in $\tilde{S}$ which have a coordinate in $[0, \frac{\varphi(N)}{4}C] \cup [\rho(N) - \frac{\varphi(N)}{4}C]$. We note that E will be chosen to ensure to that $E > \frac{\varphi(N)}{4}C$, so that S has no coordinates in this range. We fix a basis $\vec{b}_1, \ldots, \vec{b}_{n-t}$ for $Ker_{\mathbb{Z}}(\vec{\phi}_1, \ldots, \vec{\phi}_t)$. Then every element of $\tilde{S}$ can be expressed in the form:

$$S + \frac{\varphi(N)}{4} \sum_{i=1}^{n-t} \alpha_i \vec{b}_i,$$

for coefficients $\alpha_i \in \mathbb{Z}$. We let B denote the $n-t \times n$ matrix formed by letting $\vec{b}_1, \ldots, \vec{b}_{n-t}$ be its rows.

We consider a fixed coordinate $j \in [n]$. If the j^{th} column of B is all zeros, then it is impossible for the j^{th} coordinate of an element of $\tilde{S}$ to be in the range $[0, \frac{\varphi(N)}{4}C] \cup [\rho(N) - \frac{\varphi(N)}{4}C]$, since the j^{th} coordinate of S is not in this range. Otherwise, the j^{th} column of B is non-zero, and can be selected along with some other columns with indices $i_1, \ldots, i_{n-t-1}$ to form a full rank $n-t \times n-t$ submatrix. Thus, if we fix values for the $i_1, \ldots, i_{n-t-1}$ and j coordinates, there is at most one set of values of $\alpha_1, \ldots, \alpha_{n-t}$ such that these coordinates appear for an element of $\tilde{S}$. To get an element of S with j^{th} coordinate in $[0, \frac{\varphi(N)}{4}C] \cup [\rho(N) - \frac{\varphi(N)}{4}C]$ and all remaining coordinates in $[\rho(N)]$, we have at most $2C\rho(N)^{n-t-1}\left(\frac{\varphi(N)}{4}\right)^{-n+t+1}$ possible settings of these $n-t$ coordinates. Hence, an upper bound on the number of points in $\tilde{S}$ which have a coordinate within $\frac{\varphi(N)}{4}C$ of the boundary of the region $[\rho(N)]$ is:

$$n(2C)\rho(N)^{n-t-1}\left(\frac{4}{\varphi(N)}\right)^{n-t-1} \tag{2.4}$$

Combining (2.3) and (2.4), we obtain the following upper bound on the fraction of elements of $\tilde{S}$ that have at least one coordinate in the range $[0, \frac{\varphi(N)}{4}C] \cup [\rho(N) - \frac{\varphi(N)}{4}C]$:

$$\frac{n(2C)\rho(N)^{n-t-1}4^{n-t-1}(n-t)^{n-t}}{\varphi(N)^{n-t-1}D^{n-t}}.$$

Recalling that $D = \frac{4E}{\varphi(N)C}$ and ignoring a constant, we can rewrite this as

$$\frac{n\varphi(N)C^{n-t+1}(n-t)^{n-t}\rho(N)^{n-t-1}}{E^{n-t}}.$$

Since $C = 2^{n-t-1}n^{\frac{t}{2}}t^{\frac{t}{4}} \leq 2^n n^{\frac{3}{4}n}$, we can loosely bound this as

$$\leq \frac{2^{n^2}n^{2n^2}\varphi(N)\rho(N)^{n-t-1}}{E^{n-t}}. \tag{2.5}$$

We now choose

$$E := \varphi(N)^{\frac{1}{n-t}}\rho(N)^{1-\frac{1}{2(n-t)}}.$$

Then (2.5) becomes $\leq \frac{2^{n^2}n^{2n^2}}{\rho(N)^{\frac{1}{2}}}$.

We must set $\rho(N)$ so that this is negligible in the security parameter λ. For this, it suffices to set $\rho(N) > 2^{cn^2 \log n} \cdot \frac{1}{neg(\lambda)}$, for some constant c and some quantity $neg(\lambda)$ that is negligible in λ. We must also ensure that this yields a value of E for which all of the entries of S will be at least E away from the boundary of $[\rho(N)]$ with all but negligible probability. By a union bound, the probability of this failing is upper bounded by:

$$\frac{2nE}{\rho(N)} = \frac{2n\varphi(N)^{\frac{1}{n-t}}}{\rho(N)^{\frac{1}{2(n-t)}}}.$$

For this to be negligible, it suffices to set $\rho(N) > \varphi(N)^2 \cdot \left(\frac{1}{neg(\lambda)}\right)^{2n}$. We can satisfy both requirements by defining a $\rho(N)$ that is bigger than both constraints. We observe that $\rho(N) = N^{\ell}$ for some constant ℓ will suffice in the case that $\log N = \Omega(n^2 \log n)$. □

Thus, ignoring negligible factors, we can consider $\tilde{S}$ as partitioned into pairs of the form S' and $S' + \frac{\varphi(N)}{4}\vec{p}$. For each S', the values of $\vec{\phi}_{t+1} \cdot S'$ and $\vec{\phi}_{t+1} \cdot \left(S' + \frac{\varphi(N)}{4}\vec{p}\right)$ modulo 2 are different. Thus, the distribution of $\vec{\phi}_{t+1} \cdot S' \bmod 2$ over $S' \in \tilde{S}$ is sufficiently close to uniform. □

2.7 Applications: Bounded CCA Security with Short Ciphertexts

In this section, we revisit the generic transform by Boneh, Canetti, Halevi, and Katz [10] in the context of BC-IBE, and use it to obtain constructions of bounded-CCA2 secure encryption schemes with short ciphertexts from any semantically secure scheme with a secret-key to public-key homomorphism.

One-time signatures. Recall that a one-time signature scheme $\mathsf{SS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ consists of a parameter generator algorithm Gen, outputting a pair consisting of the signing key sk and the verification key vk, and a signing algorithm Sign and verification algorithm Verify such that for any message m, Sign returns a signature $\sigma = \mathsf{Sign}(\mathsf{sk}, m)$ such that $\mathsf{Verify}(\mathsf{vk}, \sigma) = \texttt{true}$ with overwhelming probability for the associated verification key vk. We say that SS is *strongly one-time secure* if for all PPT adversaries $\mathbf{A}$, there exists a negligible function ν such that

$$\mathsf{P}\left[\begin{array}{ll}(\mathsf{sk}, \mathsf{vk}) \xleftarrow{\$} \mathsf{Gen}, (m, \mathsf{st}) \xleftarrow{\$} \mathbf{A}(\mathsf{vk}), & \\ \sigma = \mathsf{Sign}(\mathsf{sk}, m), & : (m, \sigma) \neq (m', \sigma') \wedge \mathsf{Verify}(\mathsf{vk}, m', \sigma') = \texttt{true} \\ (m', \sigma') \xleftarrow{\$} \mathbf{A}(\mathsf{st}, \sigma) & \end{array}\right] \leq \nu(k) ,$$

where k is the corresponding (implicit) security parameter.

The BCHK transform . In the following, let $\mathsf{IBE} = (\mathsf{IBEGen}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ be an IBE scheme and let $\mathsf{SS} = (\mathsf{Gen}_{\mathsf{SS}}, \mathsf{Sign}, \mathsf{Verify})$ be a strong signature scheme. Boneh et al [10] presented the following construction of an encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ from IBE:

```
Gen
$(\mathsf{pp}, \mathsf{msk}) \xleftarrow{\$} \mathsf{IBEGen}$
$\mathsf{pk} \leftarrow \mathsf{pp}$
$\mathsf{sk} \leftarrow \mathsf{msk}$
Return $(\mathsf{pk}, \mathsf{sk})$.
```

```
Enc(pk, m)
$(\mathsf{sk}', \mathsf{vk}') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{SS}}$
$c \xleftarrow{\$} \mathsf{IBEEnc}(\mathsf{pp}, \mathsf{vk}', m)$
$\sigma \xleftarrow{\$} \mathsf{Sign}(\mathsf{sk}', c)$
Return $(\mathsf{vk}', c, \sigma)$.
```

```
Dec(sk, (vk, c, σ))
If $\mathsf{Verify}(\mathsf{vk}, c, \sigma) = \texttt{false}$ then
    $m \leftarrow \bot$
Else
    $\mathsf{sk}_{\mathsf{vk}} \leftarrow \mathsf{IBEExtract}(\mathsf{sk}, \mathsf{vk})$
    $m \leftarrow \mathsf{IBEDec}(\mathsf{sk}_{\mathsf{vk}}, c)$
Return $m$
```

Chosen-ciphertext security of this construction was proven in [10, Theorem 1]. Their proof considers two cases: in order to defeat the CCA security of the above construction, an adversary must either forge a signature for SS or defeat the selective security of IBE. They thus provide reductions in two different security games: one with a signature oracle (for the former case), and one with an IBE challenger (for the latter).

Of note is that in their reduction for the IBE case, the reduction makes at most one $\mathsf{IBEExtract}$ query for each decryption query it receives from the adversary, and no other parameters change. Thus, their proof carries through exactly in the bounded-collusion case, yielding:

Theorem 15. *If* IBE *is t-selective-ibe-cpa-secure, and if* SS *is strongly one-time secure, then* PKE *is t-CCA secure.*

Applications. Using previous results, we directly obtain bounded-CCA PKE constructions from DDH, QR, NTRU, and (standard) LWE using the constructions of the previous sections. In particular, note that only standard LWE is required as we only need selective security to instantiate the above paradigm. Moreover, the resulting DDH construction is essentially equivalent to the one presented in [27], and our construction thus provides an abstraction to obtain the same construction.

As an example, we give the t-CCA PKE based on the NTRU assumption that comes from applying Theorem 15 to the BC-IBE of Section 2.5.2. (Here the parameters q, χ, R_q^* are defined as in that section.)

$\underline{\mathsf{Gen}}$	$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$	$\underline{\mathsf{Dec}(\mathsf{sk}, (\mathsf{vk}, c, \sigma))}$
$f_1, \ldots, f_n \overset{\$}{\leftarrow} \chi,$ $\quad f_i \equiv 1 \pmod 2,\ f_i \in R_q^*$ $g_1, \ldots, g_n \overset{\$}{\leftarrow} \chi$ $\mathsf{sk} \leftarrow (f_1, \ldots, f_n)$ $\mathsf{pk} \leftarrow (2g_1/f_1, \ldots, 2g_n/f_n)$ Return $(\mathsf{pk}, \mathsf{sk})$.	$(\mathsf{sk}_{\mathsf{SS}}, \mathsf{vk}_{\mathsf{SS}}) \overset{\$}{\leftarrow} \mathsf{Gen}_{\mathsf{SS}}$ $h_1, \ldots, h_n, e \overset{\$}{\leftarrow} \chi$ $c \leftarrow \sum_{i \in \phi(\mathsf{vk}_{\mathsf{SS}})} \mathbf{pk}[i] \cdot h_i$ $\qquad +2e+m$ $\sigma \overset{\$}{\leftarrow} \mathsf{Sign}(\mathsf{sk}_{\mathsf{SS}}, c)$ Return $(\mathsf{vk}_{\mathsf{SS}}, c, \sigma)$.	If $\mathsf{Verify}(\mathsf{vk}, c, \sigma) = \texttt{false}$ then $\quad m \leftarrow \perp$ Else $\quad \mathsf{sk}_{\mathsf{vk}} \leftarrow \prod_{i \in \phi(\mathsf{vk})} \mathbf{sk}[i]$ $\quad m \leftarrow \mathsf{sk}_{\mathsf{vk}} \cdot c \pmod 2$ Return m

The ciphertext size of the CCA scheme generated by the BCHK transform is the same as the ciphertext size of the IBE scheme (and hence of the NTRU encryption scheme), plus a verification key and signature. Steinfeld *et al.* [71] show a (fully) CCA-secure construction based on NTRU; their ciphertext contains k ciphertexts of the underlying NTRUEncrypt algorithm (where $k = \Theta(1)$ is a parameter that depends on the hardness assumption used, but is at least 4), and additionally a verification key, a signature, and a blinded message. Since the NTRUEncrypt ciphertexts are polynomials in R_q, they will typically be much larger than the other values. Thus, we obtain a constant-factor improvement in ciphertext size by moving to the bounded-query model, in addition to the conceptual simplicity of the proof.

2.8 Open Problems

It remains to find additional constructions within this framework based on other assumptions; in particular, the only known lattice-based constructions make use of the combinatorial constructions rather than the linear hash proof construction (which affords smaller public parameters). It would also be interesting to extend this framework to accommodate stronger security requirements, such as CCA-security. Finally, constructing a fully collusion-resistant IBE from the QR assumption in the standard model remains a challenging open problem.

Separate from the question of constructions is the question of techniques. In these BC-IBE constructions we use homomorphism over the keys to reduce the ciphertext size; though many PKE keys are used in the system, the ciphertext in each BC-IBE construction consists of a single PKE ciphertext. Additional applications of homomorphism over the key space would be exciting developments.

Chapter 3

Obfuscating Functional Re-encryption, and its Connection to Fully Homomorphic Encryption

3.1 Introduction

FULLY-HOMOMORPHIC ENCRYPTION. The discovery of fully-homomorphic encryption schemes (FHE) has been a key development in modern cryptography. FHE schemes allow arbitrary computation on encrypted data without decrypting. The notion was first proposed by Rivest, Adleman, and Dertouzos [65], but it took more than three decades for the first schemes to be developed. Several FHE schemes have now been developed, first under somewhat nonstandard lattice assumptions [43, 70], then under hardness assumptions for approximate GCD [73, 25, 26], and finally under various forms of the Learning With Errors assumption [18, 17, 15, 14, 46, 45, 48] or other lattice-based assumptions [44].

At the same time, no general construction is known from smaller primitives, even for the case of *leveled FHE schemes.* A d-leveled FHE scheme allows computation of depth-d circuits on encrypted data, allowing its public key size to be a polynomial function in d. In this paper, we address the question of finding a primitive which allows a generic construction of FHE on top of a suitable encryption scheme, and revisit existing works in terms of instantiations of this blueprint.

OBFUSCATING RE-ENCRYPTION. Our approach relies on the notion of *obfuscated re-encryption*, which has been developed in parallel to FHE. While obfuscation of general functions is impossible [5], there have been several positive results detailing function families that can be obfuscated (e.g. [75, 34, 20], among many others). In particular, there has been a line of research on obfuscation that is secure *on average* (that is, for a random function from a family), rather than for any function in the family ([49, 1], and others); this definition is particularly relevant to cryptographic applications that use randomized functions. Hohenberger et al [58] show a method to obfuscate a re-encryption functionality–that is, a functionality which allows for decryption under one key and encryption under a second–such that the re-encryption procedure can be delegated to a third party who does not learn anything about the re-encrypted messages. Chandran et al [22] extended this work even further, and consider functional re-encryption, in which the second encryption key is a function of the underlying message, in the context of obfuscation of the function (and hiding the message). However, such functionalities have generally only been defined for single-input functions.

MANY-TO-ONE FUNCTIONAL RE-ENCRYPTION. Our first contribution is to introduce and define the notion of *many-to-one functional re-encryption* and its obfuscation. More specifically, for a function f, this functionality allows an evaluator to take multiple ciphertexts $c_1, \ldots, c_q$ encrypting messages $m_1, \ldots, m_q$ under the same key pk for some public-key cryptosystem PKE, and computes an encryption of $f(m_1, \ldots, m_q)$ under a different key for some possibly different cryptosystem PKE'.

Clearly, this functionality is by itself uninteresting, as it can be trivially realized by decrypting the input messages, computing the function, and encrypting the result. However, this functionality becomes interesting if it can be obfuscated and hence delegated to a user without revealing the corresponding secret key. For this reason, we also define a notion of obfuscation for this functionality, which is substantially different than the one proposed by previous works on re-encryption, despite its similar "average-case" perspective: At a high level, our first definition states that for a random circuit computing the re-encryption and for an observer who knows the public key of the source scheme, the obfuscation of that circuit *and* the public key of the target scheme are indistinguishable from the output of a simulator that only knows the public-key of the source scheme. We also consider a stronger notion, where the simulator does not simulate the public key of the target scheme, but obtains it externally. We show that the latter definition is in fact *implied* by the definition from [58].

FHE FROM MANY-TO-ONE FUNCTIONAL RE-ENCRYPTION. As one application of many-to-one functional encryption, our second contribution is to show a generic construction of leveled FHE given a semantically-secure encryption scheme such that the corresponding multi-input functional re-encryption functionalities for a complete set of operations (e.g., for the NAND operation) can be obfuscated with respect to the new notions introduced in this paper.

As an application, we show that Regev-style encryption [64] admits such obfuscated re-encryption for multiplication, which, combined with our main result and the existing additive homomorphism of the encryption yields a level FHE scheme. This scheme corresponds to the one recently proposed by Brakerski [14], for which we provide a more modular abstraction. We also reinterpret the technique of "bootstrapping" ([43] and followup work) as specific implementations of our generic construction.

3.2 Preliminaries

3.2.1 Public-Key Encryption and Semantic Security

We start by introducing our notation to describe public-key encryption schemes. Specifically, a *public-key* encryption scheme is a triple of algorithms $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where:

- the randomized algorithm Gen is the key generation algorithm, which takes as input the security parameter 1^k, and outputs a public-key / secret-key pair $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^k)$.

- Enc is the randomized encryption algorithm, and Dec is the deterministic decryption algorithm.

We assume that PKE is *correct* if for all valid public-key / secret-key pairs $(\mathsf{pk}, \mathsf{sk})$, and all messages m, the probability $\mathsf{P}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \neq m]$ is negligible, where the probability is taken over the random coins of the encryption algorithm Enc. Moreover, we say that PKE is semantically secure if for all PPT distinguishers $\mathbf{D}$ and all messages m, we have

$$\mathsf{P}\left[(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m))) = 1\right] \\ - \mathsf{P}\left[(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0))) = 1\right] \leq negl(n) .$$

3.2.2 Fully-Homomorphic Encryption

A fully homomorphic encryption (FHE) scheme is an encryption scheme which allows for arbitrary computation on encrypted data. Namely, it consists of a tuple $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ such that Gen outputs a triple of keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$, where evk is the additional *evaluation key*. The correctness requirements for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ are as in traditional public-key encryption. Moreover, Eval is the evaluation algorithm and is such that for every circuit f with q inputs, and messages $m_1, \ldots, m_q$, we have

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{evk}, f, \mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_q))) = f(m_1, \ldots, m_q) ,$$

where $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}$. Informally, we say that FHE is *leveled* (with d levels), if it only evaluates circuits of depth d (in some well defined circuit model), and the parameters are allowed to depend on d. Finally, we say that FHE is semantically secure, if for all PPT distinguishers $\mathbf{D}$ and all messages m, we have

$$\mathsf{P}\left[(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{pk}, \mathsf{evk}, \mathsf{Enc}(\mathsf{pk}, m))) = 1\right] \\ - \mathsf{P}\left[(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{pk}, \mathsf{evk}, \mathsf{Enc}(\mathsf{pk}, 0))) = 1\right] \leq negl(n) .$$

FHE constructions in the literature include [43, 70, 73, 25, 18, 17, 44, 15, 14, 46, 26, 45, 48].

3.3 Many-to-one Functional Re-encryption and its Obfuscation

In this section, we introduce the notion of many-to-one functional re-encryption, as well as a new notion of obfuscation for this functionality which, while tailored at our applications, exhibits natural connections to previous notions.

3.3.1 Many-to-one Functional Re-encryption

We start by defining circuits providing many-to-one functional re-encryption. In the most general case, we are given two public-key encryption schemes PKE and PKE' (where potentially, but not necessarily, $\mathsf{PKE} = \mathsf{PKE}'$). We are interested in families of circuits $R^f_{\mathsf{sk},\mathsf{pk}'}$ indexed by valid *secret* keys sk for PKE and valid *public* keys pk' for PKE' which, given encryptions of messages $m_1, \dots, m_q$ under PKE, produce an encryption of $f(m_1, \dots, m_q)$ for PKE'. Of course, a canonical implementation of such circuit simply decrypts $c_1, \dots, c_q$, and then re-encrypts $f(m_1, \dots, m_q)$ with *fresh randomness*. However, we will not make any further assumptions on these circuits, i.e., they may be randomized or not, and we require them to work in a more general sense, where *any* q ciphertexts $c_1, \dots, c_q$ decrypting to $m_1, \dots, m_q$ under sk will result in a ciphertext decrypting to $f(m_1, \dots, m_q)$.

Definition. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and* $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *be public-key encryption schemes. Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be the message spaces of* PKE *and* PKE'*, respectively, and let* $f : \mathcal{M}^q \to \mathcal{M}'$ *be a function. A* f-re-encryption functionality from PKE to PKE' *is a family of (possibly randomized) circuits* $\mathcal{R}^f = \left\{ R^f_{\mathsf{sk},\mathsf{pk}'} \right\}_{(\mathsf{sk},\mathsf{pk}')}$ *indexed by secret keys* sk *of* PKE *and public keys* pk' *of* PKE' *such that for all valid ciphertexts* $c_1, \dots, c_q$ *for* PKE,

$$\mathsf{Dec}'(\mathsf{sk}', R^f_{\mathsf{sk},\mathsf{pk}'}(c_1, \dots, c_q)) = f(m_1, \dots, m_q) \ ,$$

with overwhelming probability over the random choices of $(\mathsf{pk}, \mathsf{sk}) \overset{\$}{\leftarrow} \mathsf{Gen}$*,* $(\mathsf{pk}', \mathsf{sk}') \overset{\$}{\leftarrow} \mathsf{Gen}'$*, and* R^f*, where* $m_i = \mathsf{Dec}(\mathsf{sk}, c_i)$ *for* $i = 1, \dots, q$.

Without loss of generality, it will be convenient to assume that the description of the circuit $R^f_{\mathsf{sk},\mathsf{pk}'}$ allows one to recover the value of sk and pk' efficiently.

Note that in the case where $q = 1$ and f is the identity, this notion corresponds to the traditional setting of re-encryption introduced by Hohenberger et al [58]. In contrast, the more general setting of functional re-encryption introduced by Chandran et al [22] is different, in that it considers multiple recipients with different keypairs, and a function applied to an attribute associated with the ciphertext determines the *recipient* of the encryption. In their setting, however, no transformation is applied to the plaintext itself.

3.3.2 Obfuscation for Many-to-one Functional Re-encryption

We now define our new notion of secure obfuscation as specifically applied to the many-to-one re-encryption regime, i.e., to a f-re-encryption functionality $\mathcal{R}^f$ from a source scheme PKE to a target scheme PKE'. Following earlier work on obfuscation [75, 34, 1, 58, 20], we want the obfuscated circuit to perform the same computation as the original circuit. However, at the same time, we want to argue that an adversary does not learn any useful information from the obfuscated circuit beyond what it would learn by evaluating its functionality in purely black-box manner. This latter requirement is defined using a simulation-based approach, in contrast to indistinguishability-based obfuscation as in e.g. [1].

We note that for the case of one-argument functions, our notion will differ from the one proposed by Chase et al [22], while still following the same average-case viewpoint. Intuitively, our notion attempts to capture at the same time the fact that the obfuscated re-encryption functionality does not reveal *any* information beyond black-box access to the functionality *and* the fact that black-box access to the functionality does not reveal any information about the messages being encrypted. Still, our notion is connected to (and in many cases implied by) the notion defined in these earlier work, as we explain below.

For now, more concretely, let Obf be a PPT algorithm whose input and output are both circuits. Obf is a secure obfuscator for re-encryption circuit family $\mathcal{R}^f$ if the following definition is satisfied.

Definition (Re-encryption Obfuscation). *We say that* Obf securely obfuscates the f-re-encryption functionality $\mathcal{R}^f$ *from* PKE *to* PKE' *if the following two properties hold:*

- ***Correctness**: For any* $C = R^f_{\mathsf{sk},\mathsf{pk}'} \in \mathcal{R}^f$*, the statistical distance* $\Delta(\mathsf{Obf}(C)(x), C(x))$ *is negligible for all inputs* x.
- ***Simulatability**: There exists a PPT simulator* S *such that for all PPT distinguishers* $\mathbf{D}$ *and security parameter* n,

$$|\mathsf{P}[(\mathsf{sk},\mathsf{pk}) \xleftarrow{\$} \mathsf{Gen}(1^n), (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n) : \mathbf{D}(\mathsf{pk},\mathsf{pk}',\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})) = 1]$$
$$-\mathsf{P}[(\mathsf{sk},\mathsf{pk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{pk}, S(\mathsf{pk})) = 1]| < negl(n)$$

where the probabilities are taken over the coins of Gen *and* S.

This notion is somewhat different than those found in the existing literature on obfuscation; let us discuss this notion a little bit further. Generally, one defines obfuscators as being secure whenever the resulting obfuscation does not help more in computing the function implemented by the underlying circuit than black-box access to the function itself. We note that the definition provides a very strong guarantee, in that it says that an attacker, given $\mathsf{pk}, \mathsf{pk}'$ and the obfuscation $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})$ does not learn *anything* beyond the public key pk of the source scheme. Note that the obfuscation may be a randomized circuit itself, and that the correctness requirements assumes *honest* evaluation of the circuit, i.e., using honestly generated random coins.

We stress that the simulator is required to simulate the public-key pk' *together* with the obfuscation $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}})$. We also discuss a stronger notion of obfuscation where the simulator is restricted to use an externally generate public key pk' for the target scheme.

Definition (Strong Re-encryption Obfuscation). *We say that* Obf strongly securely obfuscates the f-re-encryption functionality $\mathcal{R}^f$ *from* PKE *to* PKE' *if correct-*

ness as above holds, and additionally, the following stronger simulatability requirement holds:

- ***Strong Simulatability**: There exists a PPT simulator S such that for all PPT distinguishers $\mathbf{D}$ and security parameter n,*

$$\left|\mathsf{P}\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), \\ (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n)\end{matrix} : \mathbf{D}(\mathsf{pk},\mathsf{pk}',\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})) = 1]\right]\right.$$
$$\left. - \mathsf{P}\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), \\ (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n)\end{matrix} : \mathbf{D}(\mathsf{pk},\mathsf{pk}',S(\mathsf{pk},\mathsf{pk}')) = 1\right]\right| < negl(n)$$

where the probabilities are taken over the coins of Gen *and* S*.*

Relation to earlier definitions. As mentioned above, previous works on re-encryption [58, 22] considered a different notion of average-case obfuscation which appears at first incomparable to ours, in which the simulator must simulate $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})$, given *black-box* access to $R^f_{\mathsf{sk},\mathsf{pk}'}$ and knowing the public keys pk, pk′. Formally, when translated to our setting of multi-input functional re-encryption, the requirement of these earlier works is as follows:

- **Virtual Black-boxness**: There exists a PPT simulator S such that for all PPT distinguishers $\mathbf{D}$ and security parameter n,

$$\left|\mathsf{P}\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), \\ (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n)\end{matrix} : \mathbf{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}',\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})) = 1\right]\right.$$
$$\left. -\mathsf{P}\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), \\ (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}(1^n)\end{matrix} : \mathbf{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}',S^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}')) = 1\right]\right| < negl(n)$$

where the probabilities are taken over the coins of Gen and S.

We will now prove that strong virtual black-boxness implies our strong obfuscation notion above for natural re-encryption functionalities, hence making it a somewhat stronger notion. More concretely, we say that the f-re-encryption functionality $\mathcal{R}^f = \{R^f_{\mathsf{sk},\mathsf{pk}'}\}$ is *simulatable* if there exists a simulator S' such that for all PPT distinguishers $\mathbf{D}$, we have

$$\left|\mathsf{P}\left[(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n) : \mathbf{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}') = 1\right]\right.$$
$$\left. -\mathsf{P}\left[(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n), (\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'(1^n) : \mathbf{D}^{S'(\mathsf{pk},\mathsf{pk}')}(\mathsf{pk},\mathsf{pk}') = 1\right]\right| < negl(n)\ .$$

For example, the canonical re-encryption functionality is simulatable by semantic security, provided we can efficiently test if a ciphertext input to the functionality is decryptable given pk only. Then, we can show the following:

Lemma 16. *Assume that the obfuscator satisfies the virtual black-boxness property* and *the f-reencryption functionality $\mathcal{R}^f$ is private. Then, the obfuscator satisfies the strong simulatability property.*

Proof. As our new simulator $\hat{S}$ for the strong simulatability property, we use the simulator S for virtual black-boxness, taking pk and pk' as inputs, and use S' guaranteed to exist by simulatability of the functionality $\mathcal{R}^f$ to answer S's queries, i.e., for short, $\hat{S}(\cdot,\cdot) = S^{S'}(\cdot,\cdot)$. Then, if there exists an attacker $\mathbf{D}$ violating strong obfuscability, distinguishing with non-negligible advantage ε, then $\mathbf{D}$ also violates the virtual black-boxness property (without making oracle queries) with distinguishing advantage $\varepsilon - negl(n)$. This is because by the simulatability of $\mathcal{R}^f$, the probabilities that $\mathbf{D}$ outputs one when interacting with either of $(\mathsf{pk}, \mathsf{pk}', \hat{S}(\mathsf{pk}, \mathsf{pk}')) = (\mathsf{pk}, \mathsf{pk}', S^{S'}(\mathsf{pk}, \mathsf{pk}'))$ and $(\mathsf{pk}, \mathsf{pk}', S^{R^f_{\mathsf{sk},\mathsf{pk}}}(\mathsf{pk}, \mathsf{pk}')))$ are negligibly close. □

3.4 Fully Homomorphic Encryption from Many-to-one Functional Re-encryption

In this section, we connect the notion of obfuscated many-to-one functional re-encryption with FHE, by presenting a generic construction from the former to the latter. In particular, we assume the possibility of obfuscating functional-re-encryption for specific families of functions, which we will discuss first.

3.4.1 Universal Operations and Circuits

We define the notion of an (unobfuscated) re-encryption circuit that applies a universal operation to its inputs. In particular, for a message space $\mathcal{M} = \{\mathcal{M}_n\}_{n \in \mathbb{N}}$ (e.g., $\mathcal{M} = \{0,1\}$), let $\mathcal{F} = \{\mathcal{F}_n\}$ be a universal class of functions, i.e., such that $\mathcal{F}_n$ is small enough (i.e., polynomial in n, though usually constant) and such that every function $\mathcal{M}_n^q \to \mathcal{M}_n$ can be computed by circuits having gates implementing functions from $\mathcal{F}_n$. For example, we could have $\mathcal{M}_n = \{0,1\}$ for all $n \in \mathbb{N}$, and $\mathcal{F}_n$ simply contains the NAND function. Similarly, if $\mathcal{M}_n = \mathbb{F}_q$ for some prime power q depending on n, then $\mathcal{F}$ could consists of addition and multiplication in $\mathbb{F}_q$.

As usual, the gates of the circuit with $\mathcal{F}$-gates can be divided into *layers*: any gate whose inputs consist only of input bits to the entire circuit is defined to be in layer 0, and any gate whose input consists only of outputs of layer-i gates is in layer $i+1$. Without loss of generality, we can consider circuits where each layer-i gate only outputs to layer $i+1$.

3.4.2 Main Construction

For $i \in \{0, 1, \ldots, d\}$, let $\mathsf{PKE}_i = (\mathsf{Gen}_i, \mathsf{Enc}_i, \mathsf{Dec}_i)$ be public-key encryption schemes (later to be assumed semantically secure) with common message space $\mathcal{M}$, and let $\mathcal{F}$ be a universal family of functions for $\mathcal{M}$. Also, for all $f \in \mathcal{F}$ and $i \in \{0, 1, \ldots, d-1\}$, let $\mathcal{R}_i^f = \{R^{f,i}_{\mathsf{sk}_i,\mathsf{pk}_{i+1}}\}$ be the a f-re-encryption functionality from PKE_i to PKE_{i+1}. Moreover, assume we have an obfuscator Obf_i^f for $\mathcal{R}_i^f$.

We construct a d-leveled FHE scheme $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ as follows:

- $\mathsf{Gen}(1^n)$: Run $\mathsf{Gen}^{(i)}$ to generate $(\mathsf{pk}_i, \mathsf{sk}_i) \overset{\$}{\leftarrow} \mathsf{Gen}^{(i)}$ for all $i = 0, 1, \dots, d$. Let the public key $\mathsf{pk} = (\mathsf{pk}_0, ...\mathsf{pk}_d)$, and let the evaluation key $\mathsf{evk} = (\{\mathsf{Obf}_0^f(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), ...\mathsf{Obf}_{d-1}^f(R^{f,d-1}_{sk_{d-1},pk_d})\}_{f\in\mathcal{F}})$. The secret key is $\mathsf{sk} = (\mathsf{sk}_0, ...\mathsf{sk}_d)$.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Return $c = \mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m)$.
- $\mathsf{Dec}_{\mathsf{sk}}(c)$: Run $\mathsf{Dec}_{sk_d}(c)$. (For depths i less than d, other sk_i may be used.)
- $\mathsf{Eval}_{\mathsf{evk}}(B, c_1, \dots, c_q)$, where B is a circuit consisting of $\mathcal{F}$ gates of depth at most d and with q inputs: Start with $c_1, \dots, c_q$ as values on the q input wires, and for each r-ary gate f with inputs at layer $i = 0, 1, \dots, d-1$ with value $c'_1, \dots, c'_r$ on the input layers, run $\mathsf{Obf}_i^f(R^{f,i}_{\mathsf{sk}_{i-1},pk_{i+1}})$ on inputs $c'_1, \dots, c'_q$, and assign the resulting value c'' to the output wire.

Remark. In many situations, the encryption schemes PKE_i may present some partial homomorphism properties, i.e., it may allow for computing some function $f \in \mathcal{F}$ (e.g., addition in $\mathbb{F}_q$) without resorting to re-encryption. In these situations, the obvious efficiency improvements can be made for the scheme, avoiding the use of re-encryption to compute f gates. We dispense with a formal specification of the construction in this case.

3.4.3 Security

We will prove the following theorems, which are the main result of this section.

Theorem 17 (Security of the Main Construction). *Assume that* PKE_0 *is semantically secure, and that for all* $i \in \{0, \dots, d-1\}$ *and* $f \in \mathcal{F}$*, the obfuscators* Obf_i^f *strongly securely obfuscate the* f *re-encryption functionality* $\mathcal{R}_i^f$*. Then the Main Construction above is a semantically-secure d-leveled FHE scheme.*

The following result shows that if $\mathcal{F} = \{f\}$, i.e., only one function is contained, then we can instead use the weaker notion of (non-strong) obfuscation.[1]

Theorem 18 (Security of the Main Construction – Single Function Case). *Assume that* PKE_0 *is semantically secure, and that for all* $i \in \{0, \dots, d-1\}$*, the obfuscator* Obf_i^f *securely obfuscates the* f *re-encryption functionality* $\mathcal{R}_i^f$*. Then the Main Construction above is a semantically-secure d-leveled FHE scheme.*

[1] There are multiple reasons why $\mathcal{F}$ may only contain one function: EIther f is the NAND function or the underlying scheme already provides some level of homomorphism (e.g. additions).

For both theorems, note that correctness is obvious by the definition of the re-encryption functionality and the correctness properties of the obfuscators. We are going to focus on proving the second theorem, as the proof is in fact more complicated than in the first case.

Therefore, as the core of our proof, we wish to show that the above construction achieves semantic security. Specifically, we show that for all PPT $\mathbf{D}$,

$$\begin{aligned}|\mathsf{P}[(\mathsf{sk},\mathsf{pk},\mathsf{evk}) \leftarrow \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{Enc}_{\mathsf{pk}}(m),\mathsf{pk},\mathsf{evk}) = 1 \\ -\mathsf{P}[(\mathsf{sk},\mathsf{pk},\mathsf{evk}) \leftarrow \mathsf{Gen}(1^n) : \mathbf{D}(\mathsf{Enc}_{\mathsf{pk}}(0),\mathsf{pk},\mathsf{evk}) = 1]| < negl(n)\end{aligned}$$

where the probability is taken over the random coins of Gen and of the encryptions.

To this end, we first prove a useful lemma to show that we can securely chain together obfuscators to perform multiple operations on an underlying message.

Lemma 19. *For all $m \in \mathcal{M}$, there exists PPT simulator S^* such that*

$$\begin{aligned}|\mathsf{P}[(\mathsf{pk},\mathsf{evk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : D(\mathsf{Enc}_{\mathsf{pk}}(m),\mathsf{pk},\mathsf{evk}) = 1] \\ - \mathsf{P}[(\mathsf{sk}_0,\mathsf{pk}_0) \xleftarrow{\$} \mathsf{Gen}^{(0)}(1^n) : \mathbf{D}(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m),\mathsf{pk}_0,S^*(\mathsf{pk}_0)) = 1]| < negl(n)\end{aligned}$$

where the probabilities are taken over the coins of Gen, $\mathsf{Gen}^{(0)}$*, the encryptions, and the simulator S^*.*

Proof. The real distribution $(\mathsf{Enc}_{\mathsf{pk}}(m),\mathsf{pk},\mathsf{evk})$ can be rewritten explicitly as

$$(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m),\mathsf{pk}_0,\mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}),\mathsf{pk}_1,\mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}),\mathsf{pk}_2,...,\mathsf{Obf}^f_{d-1}(R^{fd-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}),\mathsf{pk}_d)\ .$$

We now use a hybrid argument to show that this distribution is computationally indistinguishable from the simulated distribution

$$(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m),\mathsf{pk}_0,S^*(\mathsf{pk}_0))\ ,$$

for a simulator S^* which is given below.

To do this, we construct a series of distributions, and argue that a polynomial-time distinguisher cannot notice a difference at each step, except with negligible probability.

Distribution 0 : The distinguisher is given the "real-world view"

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m),\mathsf{pk}_0,\mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}),\mathsf{pk}_1,\mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}),\mathsf{pk}_2,...\mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}),\mathsf{pk}_d)\ .$$

Distribution 1 : Let S_{d-1} be the simulator guaranteed by the security of Obf^f_{d-1}. The distinguisher is given

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m),\mathsf{pk}_0,\mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}),\mathsf{pk}_1,\mathsf{Obf}(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}),\mathsf{pk}_2,...,\mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}}),\mathsf{pk}_{d-1},S_{d-1}(\mathsf{pk}_{d-1}))$$

That is, the only change from Distribution 0 is that $(\mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d)$ is replaced by $S_{d-1}(\mathsf{pk}_{d-1})$.

By definition, we know that $(\mathsf{pk}_{d-1}, \mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d)$ is computationally indistinguishable from $(\mathsf{pk}_{d-1}, S_{d-1}(\mathsf{pk}_{d-1}))$. The only remaining element of these distributions that depends on the values $(\mathsf{sk}_{d-1}, \mathsf{pk}_{d-1})$ is $\mathsf{Obf}^{f,d-2}(R^{f,d-2}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}})$. Note that this value only depends on pk_{d-1} and not sk_{d-1}. Thus, since we are already giving pk_{d-1} in the clear, an adversary gains no additional information about sk_{d-1} by seeing $\mathsf{Obf}^f_{d-2}(R^{f,d-2}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}})$. The other elements of the distribution are independent of the keys at index $d-1$ and d, so we know that the Distribution 0 is computationally indistinguishable from Distribution 1.

Distribution 2 Again, let S_{d-1} be the simulator guaranteed by the security of Obf^f_{d-1}, and let S'_{d-2} be the simulator guaranteed by the security of Obf^f_{d-2}. Define S_{d-2} as a function that applies S'_{d-2} to its input to get a pair, then applies S_{d-1} to the second element of that pair to get another pair, and outputs the 4-tuple that consists of both pairs. The distinguisher is given

$$\begin{aligned}(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, \mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, \ldots \\ \mathsf{pk}_{d-3}, \mathsf{Obf}^f_{d-3}(R^{f,d-3}_{\mathsf{sk}_{d-3},\mathsf{pk}_{d-2}}), \mathsf{pk}_{d-2}, S_{d-2}(\mathsf{pk}_{d-2}))\end{aligned}$$

This step is different from the previous step since the "pk_{d-1}" used to generate the last two elements is now itself simulated instead of being output by $\mathsf{Gen}^{(d-1)}$ directly. However, if an adversary could distinguish Distribution 2 from Distribution 1, he could use S' to break break the security of the obfuscator itself (by generating the encryption and $\mathsf{pk}_0, \ldots, \mathsf{pk}_{d-3}$ himself, using the challenge as pk_{d-2}, x, y, and running $S_{d-1}(y)$ to generate the final two elements). Thus, Distribution 2 must be computationally indistinguishable from Distribution 1.

We continue replacing pairs with a simulator in this manner until we reach:

Distribution d In Distribution d, we have replaced d (obfuscated circuit, public key) pairs with simulated values, yielding

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0))$$

as desired. By hybrid argument, since each adjacent pair of distributions are computationally indistinguishable, Distribution 0 and Distribution d are computationally indistinguishable. □

We therefore know that the security of the obfuscation algorithm implies that we can use many obfuscated re-encryption algorithms in succession without breaking security. From here on, proving the semantic security of the main construction is straightforward. Indeed, assume an adversary has both pk and evk. We know that

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), m_0, \mathsf{Obf}_0^f(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}_1^f(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, ..., \mathsf{Obf}_{d-1}^f(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d) \\ \approx_c (\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0))$$

for some S^*. Furthermore, since S^* is efficient, we know that the output of $S^*(\mathsf{pk}_0)$ can give no more information about sk_0 to the adversary than pk_0 itself can (since the adversary could have simply run S^* on his own). Since the original encryption scheme is semantically secure, we thus know that $(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0)) \approx_c (\mathsf{Enc}_{\mathsf{pk}_0}(0), \mathsf{pk}_0, S^*(\mathsf{pk}_0))$ to any PPT adversary. Thus, such an adversary can only have negligible advantage at distinguishing encryptions of m and of 0, and the FHE is semantically secure.

3.5 Example Construction

In this section, we exercise our framework by taking the public-key system of Regev [64], which is semantically secure under the Learning With Errors assumption, and give a secure obfuscation algorithm for the multiplication-re-encryption functionality from this scheme to itself. This scheme is naturally additively homomorphic; thus, by the main theorem, this implies a (leveled) fully-homomorphic encryption scheme. Note that the resulting construction is essentially that of [14]; however, we believe that viewing the problem as one of obfuscated re-encryption provides a cleaner approach.

3.5.1 A Public-Key Encryption Scheme

The basic public-key encryption scheme is due to Regev [64]. It is parameterized by n, m, q, χ from the LWE assumption used. We will refer to this scheme as $\mathsf{PKE}_{n,q,\chi}$.

- $\mathsf{Gen}(1^k)$: Choose vector $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, and vector $\mathbf{e} \xleftarrow{\$} \chi^m$. Compute $\mathbf{b} = \mathbf{A}\cdot\mathbf{s}' + \mathbf{e}$. Output secret key $\mathbf{s} = (\mathbf{s}', -1)$ and public key $(\mathbf{A}, \mathbf{b})$.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Given $m \in \{0,1\}$, choose $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$ and output $(\mathbf{A}^T\mathbf{r}, \langle \mathbf{b}, \mathbf{r}\rangle + \lfloor \frac{q}{2} \rfloor \cdot m)$.
- $\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c})$: Compute $(\langle \mathbf{s}, \mathbf{c}\rangle \pmod q)$. Output 0 if this value is closer to 0 and 1 if this value is closer to $\lfloor \frac{q}{2} \rfloor \pmod q$.

This encryption scheme is semantically secure under the $LWE_{q,\chi}$ assumption [64]. Furthermore, it is clearly additively homomorphic over $GF[2]$ (for appropriate choice of χ), since $(\langle \mathbf{s}, \mathbf{c}_1 + \mathbf{c}_2\rangle \pmod q) = \lfloor \frac{q}{2} \rfloor \cdot (m_1 + m_2) - \langle \mathbf{e}, \mathbf{r}_1 + \mathbf{r}_2\rangle \pmod q$.

3.5.2 Re-encryption and Obfuscation

Re-encryption functionality. We consider the family of circuits $\mathcal{R}^\times$, the re-encryption-with-multiplication circuits from $\mathsf{PKE}_{n,q,\chi}$ to $\mathsf{PKE}_{n,q,\chi'}$. (The values n

and q could change as well, if desired.) A circuit $R^{\times}_{\mathsf{sk},\mathsf{pk}'} \in \mathcal{R}^{\times}$ contains the secret key $\mathsf{sk} = \mathbf{s}$ of a scheme in $\mathsf{PKE}_{n,q,\chi}$ and the public key $\mathsf{pk}' = (\mathbf{A}', \mathbf{b}')$ of a scheme in $\mathsf{PKE}_{n,q,\chi'}$, hardwired inside. It takes as input two ciphertexts and applies $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ to each of them to obtain two bits. It multiplies these two bits (corresponding to a logical and), runs $\mathsf{Enc}_{\mathsf{pk}'}(\cdot)$ on the result, and outputs the resulting ciphertext.

Construction of Obf. To construct our obfuscator, we first define transformations $\mathsf{BitDecomp}$ and $\mathsf{PowersOf2}$ (used previously in [17, 15, 14, 48]). If $\mathbf{v} = (v_1, v_2, ...v_\ell) \in \mathbb{Z}_q^\ell$, then:

- $\mathsf{BitDecomp}_q(\mathbf{v}) = (v_{1,0}, v_{1,1}, ...v_{1,\lceil \lg q \rceil}, v_{2,0}, ...v_{\ell,\lceil \lg q \rceil})$, where $v_{i,j}$ is the j-th least significant bit of v_i (that is, $v_i = \sum_j 2^j v_{i,j}$).
- $\mathsf{PowersOf2}_q(\mathbf{v}) = (v_1, 2v_1, 4v_1, ...2^{\lceil \lg q \rceil} \cdot v_1, v_2, 2v_2 ... 2^{\lceil \lg q \rceil} \cdot v_\ell)$.

In the following, we will generally omit the subscript q. Of note is that for any $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathsf{BitDecomp}(\mathbf{u}), \mathsf{PowersOf2}(\mathbf{v}) \rangle$.

We will describe the transformation we want Obf to perform first, and then define its circuit output. We first compute $\tilde{\mathbf{s}} = \frac{2}{q}(\mathsf{BitDecomp}(\mathbf{s}) \otimes \mathsf{BitDecomp}(\mathbf{s}))$, a rational vector of length $((n+1)\lceil \lg q \rceil)^2$. Here $\otimes$ denotes the tensor product.

We then use $\mathsf{pk}' = (\mathbf{A}', \mathbf{b}')$ to "encrypt"[2] each element of $\mathsf{PowersOf2}(\tilde{\mathbf{s}})$. That is, we choose $\mathbf{R} \xleftarrow{\$} \{0,1\}^{((n+1)^2 \lceil \lg q \rceil^3) \times m}$ and compute $\mathbf{D} = [\mathbf{A}' | \mathbf{b}']^T \cdot \mathbf{R} + \frac{q}{2}[\mathbf{0} | \mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T$, where $\mathbf{0}$ is an $m \times n$ matrix of zeroes. (Note that $\mathbf{D}$ is an integer matrix.)

Define $\tilde{\mathbf{c}} = \frac{2}{q}(\mathsf{PowersOf2}(\mathbf{c}_1) \otimes \mathsf{PowersOf2}(\mathbf{c}_2))$. Obf will extract $\mathbf{s}$ and $(\mathbf{A}', \mathbf{b}')$ from its input. Then it constructs a randomized circuit that chooses a random $\mathbf{R}$ as defined above and computes the corresponding $\mathbf{D}$. The circuit takes in two input ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$, computes $\mathbf{D} \cdot \mathsf{BitDecomp}(\lfloor \tilde{\mathbf{c}} \rceil)$, and outputs this value. Obf outputs this circuit as the obfuscation of $R^{\times}_{\mathsf{sk},\mathsf{pk}'}$.

[2] As in [17], this is not true encryption, since the encrypted values are not bits; thus, they cannot be decrypted properly. However, the operation is the same, and the intuition that these values are "encrypted" may be useful.

Correctness. The circuit $\mathsf{Obf}(R^{\times}_{\mathsf{sk},\mathsf{pk}'})$ calculates

$$\begin{aligned}
&\mathbf{D}\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R}\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil) + \frac{q}{2}[\mathbf{0}|\mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' + \frac{q}{2}(0^n, \langle\tilde{\mathbf{s}}, \lfloor\tilde{\mathbf{c}}\rceil\rangle)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' + (0^n, \langle\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s}), \frac{2}{q}(\mathsf{PowersOf2}(\mathbf{c}_1)\otimes\mathsf{PowersOf2}(\mathbf{c}_2))\rangle)) + \mathbf{e}_1'\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' + \frac{2}{q}(0^n, \langle\mathbf{s},\mathbf{c}_1\rangle\cdot\langle\mathbf{s},\mathbf{c}_2\rangle) + \mathbf{e}_1'\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' + \frac{2}{q}(0^n, (\langle\mathbf{e}_1,\mathbf{r}_1\rangle + \frac{q}{2}m_1)(\langle\mathbf{e}_2,\mathbf{r}_2\rangle + \frac{q}{2}m_2)) + \mathbf{e}_1'\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' + \frac{q}{2}(0^n, m_1m_2) + \mathbf{e}_1' + \mathbf{e}_2'
\end{aligned}$$

We wish to show that this is statistically close to the output of $R^{\times}_{\mathsf{sk},\mathsf{pk}'}$ (which is a fresh encryption of m_1m_2). There are two differences: the fact that $\mathbf{r}'$ is not a binary vector, and the presence of an additional additive error term $(\mathbf{e}_1' + \mathbf{e}_2')$.

For the first difference, note that $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' \in \mathbb{Z}_q^n$, and that both $\mathbf{A}'$ and $\mathbf{R}$ are chosen randomly. There are 2^m choices of $\mathbf{r}'' \in \{0,1\}^m$. Thus, for a value $m = \Omega(n \lg q)$, with high probability there exists $\mathbf{r}'' \in \{0,1\}^m$ such that $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}' = [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}''$.

For the second difference, we note that both $\mathbf{e}_1'$ and $\mathbf{e}_2'$ are "small". Specifically, $\mathbf{e}_1'$ comes from rounding error; each element is rounded by at most $1/2$, so its magnitude is bounded[3] by $||\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s})||_1\cdot\frac{1}{2} \leq ((n+1)(\lceil\lg q\rceil+1))^2/2$. $\mathbf{e}_2'$ is due to the presence of $\mathbf{e}_1$ and $\mathbf{e}_2$ in the original ciphertexts; however, the presence of the $\frac{2}{q}$ coefficient means that this term is bounded by $O(m\varepsilon)$, where ε is the original error bound of χ. Note that the magnitude of $(\mathbf{e}_1+\mathbf{e}_2)$ is *independent* of q aside from a logarithmic factor; thus, we can choose the LWE parameters (in particular, q and χ') such that the output distributions of the obfuscated and unobfuscated circuits are statistically close.

Simulatability. We show a simulator S that satisfies the strong simulatability condition for this construction, as defined in section 3.2. Recall that $\mathsf{Obf}(R^{\times}_{\mathsf{sk},\mathsf{pk}'})$ constructs a circuit that only depends on the values $(\mathsf{sk},\mathsf{pk}')$ through a matrix $\mathbf{D}$, defined as $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R} + \frac{q}{2}[\mathbf{0}|\mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T$. The simulator S simply chooses $\mathbf{R} \xleftarrow{\$} \{0,1\}^{((n+1)^2\lceil\lg q\rceil^3)\times m}$ and returns a circuit that uses $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R}$ in place of $\mathbf{D}$.

Note that this is simply a Regev encryption of 0 under the key pk'; indistinguishability holds by the semantic security of the original Regev scheme.

[3] Bounding this error is the reason to introduce BitDecomp and PowersOf2–this allows the vector $\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s})$ to be binary.

3.6 Bootstrapping

Many existing FHE schemes, starting with that of Gentry [43], operate on the principle of "bootstrapping". That is, they first define a "somewhat homomorphic" scheme, which is capable of homomorphically evaluating its own decryption circuit plus a single operation under a single key. They then provide a chain of encrypted keys under this scheme, where the i-th decryption key is encrypted under the $(i+1)$st key. This construction allows for (leveled) fully-homomorphic evaluation: given a ciphertext encrypted under the i-th key, the evaluator encrypts the ciphertext under the $(i+1)$st key and then homomorphically evaluates the decryption circuit on the new ciphertext and the encrypted i-th key, followed by one operation. The net result is an encryption under key $i+1$ of the operation applied to the plaintext corresponding to the input.

The general bootstrapping paradigm can be seen under our framework as providing an obfuscated re-encryption-with-operation functionality. Specifically, given the keys pk_{i+1}, sk_i, one can construct a circuit that encrypts its input under pk_{i+1}, runs the decryption operation homomorphically using a hardcoded value $\mathsf{Enc}_{pk_{i+1}}(sk_i)$, and then homomorphically performs one operation. This circuit performs the same computation as decrypting, performing the operation, and encrypting (by the correctness of the FHE scheme), and does not leak any information about the encrypted data (by the semantic security of the FHE scheme). Thus, at a high level it is an obfuscated re-encryption-with-operation circuit under our definition. However, our definition is more general, since we do not require starting with a "somewhat homomorphic" encryption scheme, but any semantically-secure encryption scheme with a securely-obfuscatable f-re-encryption functionality.

Chapter 4

Conclusion

Encryption has developed rapidly in the past few decades, in models, definitions, and functionalities. From a theoretical perspective, it is of great importance to understand the relationships between various cryptographic primitives in this more complex world.

We take strides to better characterize identity-based encryption by describing bounded-collusion IBE and providing multiple generic constructions from public-key encryption. These constructions explore the idea of using homomorphisms over keys, which reduces the ciphertext size. We additionally show a number of instantiations of these methods based on a variety of well-known cryptographic hardness assumptions.

Additionally, in the realm of characterizing encryption schemes with additional functionality, we give a generic construction of fully-homomorphic encryption starting from a new definition of obfuscation. This provides a mathematical formalization for an intuitive connection between these functionalities–that computing on encrypted data is equivalent to decrypting, computing, and reencrypting, as long as the latter processes are collectively obfuscated.

These advances provide not only generic and concrete constructions of cryptographic primitives, but also new characterizations of these models of encryption. These new ways of thinking about encryption provide a lens with which to analyze the relations between these technologies as they continue to develop.

Bibliography

[1] Ben Adida and Douglas Wikström. How to shuffle in public. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 555–574. Springer, February 2007.

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

[3] Joël Alwen, Manuel Barbosa, Pooya Farshim, Rosario Gennaro, S. Dov Gordon, Stefano Tessaro, and David A. Wilson. On the relationship between functional encryption, obfuscation, and fully homomorphic encryption. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 65–84. Springer, 2013.

[4] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *EUROCRYPT*, pages 113–134, 2010.

[5] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, August 2001.

[6] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, August 1998.

[7] Enrico Bombieri and Jeffrey Vaaler. On siegel's lemma. *Inventiones Mathematicae*, 73:11–32, 1983. 10.1007/BF01393823.

[8] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223 – 238. Springer, 2004.

[9] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, August 2004.

[10] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[11] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, August 2001.

[12] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual Symposium on Foundations of Computer Science*, pages 647–657. IEEE Computer Society Press, October 2007.

[13] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.

[14] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, August 2012.

[15] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011. http://eprint.iacr.org/.

[16] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2010.

[17] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106. IEEE Computer Society Press, October 2011.

[18] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, August 2011.

[19] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.

[20] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, February 2010.

[21] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, May 2010.

[22] Nishanth Chandran, Melissa Chase, and Vinod Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 404 421. Springer, March 2012.

[23] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.

[24] Henry Cohn, Shafi Goldwasser, and Yael Tauman Kalai. The impossibility of obfuscation with a universal simulator. *CoRR*, abs/1401.0348, 2014.

[25] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, August 2011.

[26] Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer, April 2012.

[27] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 502–518. Springer, December 2007.

[28] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[29] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[30] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In *Digital Rights Management Workshop*, pages 61–80, 2002.

[31] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, February 2010.

[32] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign RSA signatures. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 112–132. Springer, March 2012.

[33] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Key-insulated public key cryptosystems. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82. Springer, April / May 2002.

[34] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 654–663. ACM Press, May 2005.

[35] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, August 1985.

[36] P. Erdös, P. Frankel, and Z. Furedi. Families of finite sets in which no set is covered by the union of r others. *Israeli Journal of Mathematics*, 51:79–89, 1985.

[37] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, August 1994.

[38] Eli Gafni, Jessica Staddon, and Yiqun Lisa Yin. Efficient methods for integrating traceability and broadcast encryption. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 372–387. Springer, August 1999.

[39] Juan A. Garay, Jessica Staddon, and Avishai Wool. Long-lived broadcast encryption. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 333–352. Springer, August 2000.

[40] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49. IEEE Computer Society, 2013.

[41] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer, August 2010.

[42] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, May / June 2006.

[43] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, May / June 2009.

[44] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 107–109. IEEE Computer Society Press, October 2011.

[45] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, April 2012.

[46] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer, August 2012.

[47] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.

[48] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

[49] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual Symposium on Foundations of Computer Science*, pages 553–562. IEEE Computer Society Press, October 2005.

[50] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 555–564. ACM, 2013.

[51] Shafi Goldwasser, Allison B. Lewko, and David A. Wilson. Bounded-collusion IBE from key homomorphism. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 564–581. Springer, March 2012.

[52] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[53] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 194–213. Springer, February 2007.

[54] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179. Springer, August 2012.

[55] Jacques Hadamard. Resolution d'une question relative aux determinants. *Bull. Sci. Math*, 17:240–246, 1893.

[56] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, August 2002.

[57] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In Joe Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

[58] Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 233–252. Springer, February 2007.

[59] Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 609–623. Springer, August 1999.

[60] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th Annual ACM Symposium on Theory of Computing*, pages 1219–1234. ACM Press, May 2012.

[61] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, August 2001.

[62] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[63] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342. ACM Press, May / June 2009.

[64] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.

[65] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, editors, *Foundations of Secure Computation*, pages 165–179. Academic Press, 1978.

[66] Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 219–234. Springer, March 2011.

[67] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *IACR Cryptology ePrint Archive*, 2013:454, 2013.

[68] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

[69] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, 41:303–332, January 1999.

[70] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, May 2010.

[71] Ron Steinfeld, San Ling, Josef Pieprzyk, Christophe Tartary, and Huaxiong Wang. NTRUCCA: How to strengthen NTRUEncrypt to chosen-ciphertext security in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Workshop on Theory and Practice in Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 353–371. Springer, May 2012.

[72] Stefano Tessaro and David A. Wilson. Bounded-collusion identity-based encryption from semantically-secure public-key encryption: Generic constructions with short ciphertexts. In Hugo Krawczyk, editor, *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2014.

[73] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, May 2010.

[74] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, May 2005.

[75] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 523–532. ACM Press, May 2005.