Bounded-Collusion IBE from Key Homomorphism

Shafi Goldwasser[1,⋆], Allison Lewko[2,⋆⋆], and David A. Wilson[3,⋆⋆⋆]

[1] MIT CSAIL and Weizmann Institute
shafi@csail.mit.edu
[2] UT Austin
alewko@cs.utexas.edu
[3] MIT CSAIL
dwilson@mit.edu

Abstract. In this work, we show how to construct IBE schemes that are secure against a bounded number of collusions, starting with underlying PKE schemes which possess linear homomorphisms over their keys. In particular, this enables us to exhibit a new (bounded-collusion) IBE construction based on the quadratic residuosity assumption, without any need to assume the existence of random oracles. The new IBE's public parameters are of size $O(t\lambda \log I)$ where I is the total number of identities which can be supported by the system, t is the number of collusions which the system is secure against, and λ is a security parameter. While the number of collusions is bounded, we note that an exponential number of total identities can be supported.

More generally, we give a transformation that takes any PKE satisfying *Linear Key Homomorphism*, *Identity Map Compatibility*, and the *Linear Hash Proof Property* and translates it into an IBE secure against bounded collusions. We demonstrate that these properties are more general than our quadratic residuosity-based scheme by showing how a simple PKE based on the DDH assumption also satisfies these properties.

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

[⋆] Supported by NSF CCF-0729011, NSF CCF-1018064, DARPA FA8750-11-2-0225.
[⋆⋆] Supported by a Microsoft Research Ph.D. Fellowship.
[⋆⋆⋆] Supported by NSF CCF-1018064, DARPA FA8750-11-2-0225.

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 564–581, 2012.

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 [29], 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 paper, we continue this line of investigation and show how public-key encryption schemes which posses a linear homomorphic property over their keys as well as hash proof system features with certain algebraic structure 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 linear combination of the published keys. By taking a linear combination instead of a subset, we are able to achieve smaller keys than a strictly combinatorial approach would allow. Our constructions allow the total number of potential identities to be exponential in the size of the public parameters of the IBE. 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.

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 [30]. The first constructions were proposed in 2001 by Boneh and Franklin [6] based on the hardness of the bilinear Diffie-Hellman problem and by Cocks [13] 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 [22], computational problems employing bilinear pairings were at the time of [6] 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 [11,4], 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 [21] 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 [5] and Waters [34], and most recently under the LWE assumption by Cash, Hofheinz, Kiltz, and Peikert [12] and Agrawal, Boneh, and Boyen [1]. Constructing a fully secure (or even selectively secure) IBE without resorting to the random oracle model based on classical number theoretic assumptions such as DDH in non-bilinear groups or the hardness of quadratic residuosity assumptions remains open.

A different relaxation of IBE comes up in the work of Dodis, Katz, Xu, and Yung [16] 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 PKE's by [16] yields a generic combinatorial construction which converts any semantic secure PKE to a bounded-collusion semantic secure IBE, without needing a random oracle.

New Results. The goal of our work is to point to a new direction in the construction of IBE schemes: the utilization of homomorphic properties over keys of PKE schemes (when they exist) to obtain IBE constructions. This may provide a way to diversify the assumptions on which IBEs can be based. In particular, we are interested in obtaining IBE constructions based on quadratic residuosity in the standard model.

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 Ostrovski [8] and the scheme of Brakerski and Goldwasser [9], 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 $(pk_1 \cdot pk_2, sk_1 + sk_2)$.

We define properties of a PKE scheme allowing homomorphism over keys that suffice to transform the PKE into an IBE scheme with bounded collusion resistance. As examples of our general framework, we show how to turn the schemes of [8] and a modification of [9] into two IBE schemes in the standard model (that is, without random oracles), which are CPA secure against bounded collusions. Namely, security holds when the adversary is restricted to receive t secret keys for identities of its choice for a pre-specified t. We allow the adversary to choose its attack target adaptively. The security of the scheme we present here is based on the intractability of the quadratic residuosity problem. In the full version of this paper, we also present a second scheme with security based on the intractability of DDH. Letting the public parameters of the IBE be of size $O(n\lambda)$ where λ is a security parameter, the new DDH-based IBE will be secure as long as the adversary is restricted to receive t secret keys for adaptively chosen ID's where $t = n - 1$. The QR-based IBE will be secure as long as the adversary is restricted to receive t secret keys for $t = \frac{n}{\log I} - 1$, where I is the total number of users (or identities) that can be supported by the system. There is no upper bound on I, which can be exponential in the size of public parameters.

Let us compare what we achieve to the constructions obtained by [16]. In their generic combinatorial construction, they start with any PKE and obtain a bounded-collusion IBE, requiring public parameters to be of size $O(t^2 \log I)$ times the size of public keys in the PKE scheme and secret keys to be of size $O(t \log I)$ times the size of secret keys in the PKE scheme for t collusions and I total identities supported. Their approach employs explicit constructions of sets $S_1, \ldots, S_I$ of some finite universe U such that the union of any t of these does not cover any additional set. There is a public key of the PKE scheme generated for each element of U, and each set S_i corresponds to a set of $|S_i|$ PKE secret keys. There are are intrinsic bounds on the values of $I, |U|, t$ for which this works, and [16] note that their values of $|U| = \Theta(t^2 \log I)$ and $|S_i| = \Theta(t \log I)$ for each i are essentially optimal. In contrast, by exploiting the algebraic homomorphisms over the keys, we require public parameters of size roughly $O(t \cdot \log I)$ times the size of public keys and secret keys which are $O(\lambda)$ (within a constant times the size of PKE secret keys) for our quadratic residuosity based scheme. (This is assuming a certain relationship between the security parameter λ and n. See the statement of Theorem 2 for details.)

In [16], they also provide a DDH-based key-insulated PKE scheme which is more efficient than their generic construction. It has $O(t\lambda)$ size public parameters and $O(\lambda)$ size secret keys. Viewing their scheme in the identity based context results in, perhaps surprisingly, the DDH based scheme we obtain by exploiting the homomorphism over the keys in BBHO [8]. In the full version of this paper, we describe this scheme and show it can be proved secure against t collusions using our framework.

1.1 Overview of the Techniques

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

Start with a PKE Π 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 Π and $a, b \in R$, then $ask_1 + bsk_2$ is also a valid secret key of Π, 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 is 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 [8], [9], and [14].

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 Π, publish the n pk_i values, and keep secret the corresponding n sk_i's. Each identity is mapped to a vector $id_1 ... id_n$ 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.

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

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

We refer to the combination of a PKE scheme with this homomorphic property over keys and a mapping for identities as having the *linear key homomorphism* and *identity map compatibility* properties. To prove security for the resulting bounded-collusion IBE construction, one can intuitively see that we need the map taking identities to vectors to produce linearly independent outputs for any distinct $t + 1$ identities. This is required to ensure that any t colluding users will not be able to compute a secret key for another user as a known linear combination of their own secret keys. To obtain our full security proof, we define an algebraic property of the PKE scheme in combination with the identity map, called the *linear hash proof property*, which encompasses this requirement on any $t+1$ images of the map and more. The definition of this property is inspired by the paradigm of hash proof systems (introduced by Cramer and Shoup [14]), though it differs from this in many ways. We define valid and invalid ciphertexts for our systems, where valid ciphertexts decrypt properly and invalid ciphertexts should decrypt randomly over the set of many secret keys corresponding to a single public key. We require that valid and invalid ciphertexts are computationally indistinguishable. So far this is quite similar to the previous uses of hash proof systems. However, the identity-based setting introduces a further challenge in proving security by changing to an invalid ciphertext, since now the adversary's view additionally includes the secret keys that it may request for other identities. Hence, we must prove that an invalid ciphertext decrypts randomly over the subset of secret keys that are consistent not only with the public keys, but also with the received secret keys.

Controlling the behavior over this set of consistent keys in the QR-based setting is particularly challenging, where the mathematical analysis is quite subtle due to the fact that secret keys must be treated as integers in a bounded range while public keys are elements of a subgroup of $\mathbb{Z}_N$. To prove the linear hash proof property for our QR-based system, we employ technical bounds concerning the intersection of a shifted lattice in $\mathbb{Z}^n$ with a "bounding box" of elements of $\mathbb{Z}^n$ whose coordinates all lie within a specified finite range.

1.2 Other Related Work

In addition to those referenced above, constructions of IBE schemes in the standard model in the bilinear setting were also provided by Gentry [20] under the q-ABHDE assumption, and by Waters [35] under the bilinear Diffie-Hellman and decisional linear assumptions. Another construction based on quadratic residuosity in the random oracle model was provided by Boneh, Gentry, and Hamburg [7]. Leakage-resilient IBE schemes in various models have also been constructed, for example by Alwen, Dodis, Naor, Segev, Walfish, and Wichs [2], by Brakerski, Kalai, Katz, and Vaikuntanathan [10], and by Lewko, Rouselakis, and Waters [26].

The property we require for our PKE schemes in addition to key homomorphism is a variant of the structure of hash proof systems, which were first introduced by Cramer and Shoup as a paradigm for proving CCA security of PKE schemes [14]. Hash proof systems have recently been used in the context of leakage-resilience as well ([28], for example), extending to the identity-based setting in [2]. We note that the primitive of identity-based hash proof systems introduced in [2] takes a different direction than our work, and the instantiation they provide from the quadratic residuosity assumption relies on the random oracle model.

The relaxation to bounded collusion resistance has also been well-studied in the context of broadcast encryption and revocation schemes, dating back to the introduction of broadcast encryption by Fiat and Naor [17]. This work and several follow up works employed combinatorial techniques [31,32,33,18,25,19]. Another combinatorial approach, the subset cover framework, was introduced by Naor, Naor, and Lopspeich [27] to build a revocation scheme. In this framework, users are associated with subsets of keys. The trusted system designer can then broadcast an encrypted message by selecting a family of subsets which covers all the desired recipients and none of the undesired ones. An improvement to the NNL scheme was later given by Halevy and Shamir [24], and these techniques were then extended to the public key setting by Dodis and Fazio [15].

2 Preliminaries

2.1 IND-CPA Security for Bounded-Collusion IBE

We define IND-CPA security for bounded-collusion IBE in terms of the following game between a challenger and an attacker. We let t denote our threshold parameter for collusion resistance. The game proceeds in phases:

Setup Phase. The challenger runs the setup algorithm to produce the public parameters and master secret key. It gives the public parameters to the attacker.

Query Phase I. The challenger initializes a counter to be 0. The attacker may then submit key queries for various identities. In response to a key query, the challenger increments its counter. If the resulting counter value is $\leq t$, the challenger generates a secret key for the requested identity by running the key generation algorithm. It gives the secret key to the attacker. If the counter value is $> t$, it does not respond to the query.

Challenge Phase. The attacker specifies messages m_0, m_1 and an identity ID^* that was not queried in the preceding query phase. The challenger chooses a random bit $b \in \{0,1\}$, encrypts m_b to identity ID^* using the encryption algorithm, and gives the ciphertext to the attacker.

Query Phase II. The attacker may again submit key queries for various identities *not equal to* ID^*, and the challenger will respond as in the first query phase. We note that the same counter is employed, so that only t total queries in the game are answered with secret keys.

Guess. The attacker outputs a guess b' for b.

We define the advantage of an attacker $\mathcal{A}$ in the above game to be $Adv_{\mathcal{A}} = \left|Pr[b = b'] - \frac{1}{2}\right|$. We say a bounded-collusion IBE system with parameter t is *secure* if any PPT attacker $\mathcal{A}$ has only a negligible advantage in this game.

2.2 Complexity Assumption

We formally state the QR assumption. We let λ denote the security parameter.

Quadratic Residuosity Assumption. 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 $\frac{\phi(N)}{2}$ and $\frac{\phi(N)}{4}$ respectively. We note that $\frac{\phi(N)}{4}$ 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 in [14], 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 $\mathbb{J}_N$.

2.3 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 $\mathbb{Z}_2$. The proofs of the following lemmas can be found in the full version.

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

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

3 From PKE to Bounded Collusion IBE: General Conditions and Construction

We start with a public key scheme and an efficiently computable mapping f 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".

3.1 Linear Key Homomorphism

We say a PKE has linear key homomorphism if the following requirements hold. First, its secret keys are generated randomly from d-tuples of a ring R for some positive integer d, with a distribution that is independent and uniform in each coordinate over some subset R' of R. Second, starting with any two secret keys sk_1, sk_2 each in R^d and any $r_1, r_2 \in R$, the component-wise R-linear combination formed by $r_1sk_1+r_2sk_2$ also functions as a secret key, with a corresponding public key that can be computed efficiently from r_1, r_2 and the public keys pk_1 and pk_2 of sk_1 and sk_2 respectively, fixing the same public parameters. We note that $r_1sk_1 + r_2sk_2$ may not have all entries in R', but it should still function properly as a key.

3.2 Identity Map Compatibility

We say the identity mapping f is compatible with a PKE scheme with linear key homomorphism if f maps identities into n-tuples of elements of R. Letting I denote the number of identities, the action of f can be represented by a $I \times n$ matrix with entries in R. We denote this matrix by F and its rows by $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_I$.

3.3 Linear Hash Proof Property

We now define the strongest property we require, which we call the linear hash proof property. This property is inspired by the paradigm of hash proof systems, but we deviate from that 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 f 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.

Fix any $t + 1$ distinct rows of the matrix of identity vectors F, denoted by $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_{t+1}}$. We let $sk_{ID_{i_{t+1}}}$ denote the secret key $\boldsymbol{f}_{i_{t+1}} \cdot S$ and $pk_{ID_{i_{t+1}}}$ denote the corresponding public key (computed via the key homomorphism). We let $Ker_R(\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_t})$ denote the kernel of the $t \times n$ submatrix of F formed by these rows; that is, it consists of the vectors $\boldsymbol{v} \in R^n$ such that $\boldsymbol{f}_{i_j} \cdot \boldsymbol{v} = 0$ 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 $Ker_R(\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_t})$ 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 $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_t}$, adding an element of W to S does not change any of the secret keys $\boldsymbol{f}_{\boldsymbol{i_j}} S$. 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 $\boldsymbol{f}_{i_1} \cdot S, \ldots, \boldsymbol{f}_{i_t} \cdot S$. 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 compatible map** f 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. With all but negligible probability over the choice of $sk_1, pk_1, \ldots, sk_n, pk_n$ and the random coins of the invalid encryption algorithm, for any choice of distinct rows $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_{t+1}}$ of F, an invalid ciphertext encrypted to $pk_{ID_{i_{t+1}}}$ must decrypt to a message distributed negligibly close to uniform over the message space when decrypted with a secret key chosen at random from $\boldsymbol{f}_{i_{t+1}} \cdot \tilde{S}$. More precisely, an element of $\tilde{S}$ is chosen uniformly at random, and the resulting matrix is multiplied on the left by $\boldsymbol{f}_{i_{t+1}}$ to produce the secret key.

Computational Indistinguishability of Valid/Invalid Ciphertexts. Second, we require valid and invalid ciphertexts are computationally indistinguishable in the following sense. For any fixed (distinct) $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_{t+1}}$, 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 $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_{t+1}}$ in F in any order it likes. (For convenience, we let $\boldsymbol{f}_{i_{t+1}}$ always denote the vector that will be encrypted under, but we note that this may be chosen before some of the other $\boldsymbol{f}_i$'s.) Upon setting an $\boldsymbol{f}_{i_j}$ for $j \neq t+1$, the attacker receives $\boldsymbol{f}_{i_j} \cdot S$. When it sets $\boldsymbol{f}_{i_{t+1}}$, 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 $\boldsymbol{f}_{i_{t+1}} \cdot S$ as follows. We let pk_{ch} denote the public key corresponding to $\boldsymbol{f}_{i_{t+1}} \cdot S$. If $\beta = 0$, it calls Encrypt with m, pk_{ch}. If $\beta = 1$, it calls InvalidEncrypt with $m, \boldsymbol{f}_{i_{t+1}} \cdot S$. 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 $\mathcal{A}$.

3.4 Construction

Given a PKE scheme (KeyGen, Encrypt, Decrypt) and an identity mapping f 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 f.

Setup$(\lambda) \rightarrow \mathrm{PP}, \mathrm{MSK}$. 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 $sk_1, \ldots, sk_n$.

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

Encrypt$(m, \mathrm{PP}, ID) \rightarrow \mathrm{CT}$. 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 f is known and efficiently computable). It then runs the PKE encryption algorithm on m with this public key to produce CT.

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

3.5 Security

Theorem 1. *When a PKE scheme (KeyGen, Encrypt, Decrypt) with linear key homomorphism and a compatible identity mapping f satisfy the linear hash proof property, then the construction defined in Section 3.4 is a secure bounded-collusion IBE scheme with collusion parameter t.*

Proof. We first change from the real security game defined in Section 2.1 to a new Game′ 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 $\boldsymbol{f}_{i_j}$, $\mathcal{A}'$ can forward $\boldsymbol{f}_{i_j}$ to its challenger and obtain the corresponding secret key. When $\mathcal{A}$ declares m_0, m_1 and some ID^* corresponding to $\boldsymbol{f}_{i_{t+1}}$, $\mathcal{A}'$ chooses a random bit $b \in \{0, 1\}$ and sends $m_b, \boldsymbol{f}_{i_{t+1}}$ 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 F. 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, Game′. 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 Game$'$, 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.

4 QR-Based Construction

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

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 $\rho(N)$.

- $Gen(1^\lambda)$: 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 $pk = g^{sk}$.
- $Enc_{pk}(m)$: The encryption algorithm chooses an odd $r \in [N^2]$ uniformly at random, and calculates $Enc(m) = (g^r, m \cdot pk^r)$.
- $Dec_{sk}(c_1, c_2)$: The decryption algorithm computes $m = c_2 \cdot (c_1^{sk})^{-1}$.

We additionally define the invalid encryption algorithm:

- $InvalidEnc_{sk}(m)$: 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.

4.1 Compatible Mapping and Resulting IBE Construction

Our compatible map f is obtained from Lemma 2 (Section 2.3). 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 f. The image of each identity under f 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 f 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 3.4, 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.

Setup. 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, $sk_1, \ldots, sk_n$. 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 $i = 1 \ldots n$).

KeyGen(ID). The key generation algorithm receives an $ID \in \{0,1\}^k$. By Lemma 2 (Section 2.3), we then have a mapping f 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 $sk_1, \ldots, sk_n$, with coefficients $id_1, \ldots, id_n$ 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 f provided in Section 2.3 produces vectors $(id_1, \ldots, id_n)$ 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.

Encrypt(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 (PK_{ID}, 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

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

All arithmetic here takes place modulo N.

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

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

4.2 Security of the IBE

We now prove security of IBE scheme up to t collusions. This will follow from Theorem 1 and the theorem below.

Theorem 2. *Under the QR assumption, the PKE construction in Section 4 is a linear hash proof system with respect to f 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 3. *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 $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{t+1}$ denote the distinct rows of f that are chosen adaptively by $\mathcal{A}$ during the course of the game (these were formerly called $\boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_{t+1}}$).

$\mathcal{B}$ 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 $\mathrm{PP} = (N, g, pk_1, \ldots, pk_n)$ to $\mathcal{A}$. We note that $\mathcal{B}$ knows the $\mathrm{MSK} = S$, so it can compute $\boldsymbol{f}_1 \cdot S, \ldots, \boldsymbol{f}_t \cdot S$ and give these to $\mathcal{A}$ whenever $\mathcal{A}$ chooses the vectors $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t$.

At some point, $\mathcal{A}$ declares a message m and a vector $\boldsymbol{f}_{t+1}$ 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 $\frac{\phi(N)}{4}$, and the reduction of a randomly chosen odd $r \in [N^2]$ modulo $\frac{\phi(N)}{4}$ 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 4. *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 F, denoted by $\boldsymbol{f}_1, \dots, \boldsymbol{f}_{t+1}$. We must argue that the value of $\boldsymbol{f}_{t+1} \cdot S$ modulo 2 is negligibly close to uniform, conditioned on $\boldsymbol{f}_1 \cdot S, \dots, \boldsymbol{f}_t \cdot S$ and S modulo $\frac{\phi(N)}{4}$. 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 $sk \equiv sk^* \bmod \phi(N)/4$ 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 $\boldsymbol{f}_1 \cdot S, \dots, \boldsymbol{f}_t \cdot S$ and S modulo $\frac{\phi(N)}{4}$, we can restate our goal as showing that the distribution of $\boldsymbol{f}_{t+1} \cdot S' \bmod 2$ is negligibly close to uniform, where S' is chosen uniformly at random from $\tilde{S}$.

We know by Lemma 2 that the vectors $\boldsymbol{f}_1, \dots, \boldsymbol{f}_{t+1}$ 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 $Ker_{\mathbb{Q}}(\boldsymbol{f}_1, \dots, \boldsymbol{f}_t)$ 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 $\boldsymbol{p}$ with integer entries that is *not* orthogonal to $\boldsymbol{f}_{t+1}$ modulo 2. Then, for every S' in $S+W$, $S' + \frac{\phi(N)}{4}\boldsymbol{p}$ is also in $S+W$. Here we are using the notation from Section 3 where we defined W. In this instance, $S+W$ is the set of vectors yielding the same values as S for $\boldsymbol{f}_1 \cdot S, \dots, \boldsymbol{f}_t \cdot S$ and S modulo $\frac{\phi(N)}{4}$. $\tilde{S}$ 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), $S' + \frac{\phi(N)}{4}\boldsymbol{p}$ will also be in $\tilde{S}$ (i.e. have entries in $[\rho(N)]$). This will follow from showing that there exists a $\boldsymbol{p}$ 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. The proof the second can be found in the full version.

Lemma 5. *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 [3], 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 [23] 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 [3], the lemma follows.

Lemma 6. *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.*

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

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

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

with the set of vectors with coordinates all in $[\rho(N)]$, where $Ker_{\mathbb{Z}}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ denotes the vectors in $Ker_{\mathbb{Q}}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ 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 $(\frac{\phi(N)}{4}C, \rho(N) - \frac{\phi(N)}{4}C)$. 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 $\rho(N)$. In the full version of this paper, we prove the following lemma:

Lemma 7. *With $\rho(N), n, \boldsymbol{p}, S$, 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 $S' + \frac{\phi(N)}{4}\boldsymbol{p}$ is not also in $\tilde{S}$ is negligible with all but negligible probability over the choice of S.*

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

5 Open Problems

It remains to find additional constructions within this framework based on other assumptions; in particular, lattice-based constructions may be possible. 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.

Acknowledgements. We thank David Zuckerman for pointing us to the use of dual BCH codes for the mapping f and Mark Lewko for pointing us to [3].

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-Key Encryption in the Bounded-Retrieval Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
3. Bombieri, E., Vaaler, J.: On siegel's lemma. Inventiones Mathematicae 73, 11–32 (1983), doi:10.1007/BF01393823
4. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)
8. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
9. Brakerski, Z., Goldwasser, S.: Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability (or: Quadratic Residuosity Strikes Back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)
10. Brakerski, Z., Tauman Kalai, Y., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: FOCS, pp. 501–510 (2010)
11. Canetti, R., Halevi, S., Katz, J.: A Forward-secure Public-key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
12. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
13. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
14. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
15. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)

16. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-Insulated Public Key Cryptosystems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 65–82. Springer, Heidelberg (2002)
17. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
18. Gafni, E., Staddon, J., Yin, Y.L.: Efficient Methods for Integrating Traceability and Broadcast Encryption. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 372–387. Springer, Heidelberg (1999)
19. Garay, J.A., Staddon, J., Wool, A.: Long-Lived Broadcast Encryption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 333–352. Springer, Heidelberg (2000)
20. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
22. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
23. Hadamard, J.: Resolution d'une question relative aux determinants. Bull. Sci. Math. 17, 240–246 (1893)
24. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
25. Kumar, R., Rajagopalan, S., Sahai, A.: Coding Constructions for Blacklisting Problems without Computational Assumptions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 609–623. Springer, Heidelberg (1999)
26. Lewko, A., Rouselakis, Y., Waters, B.: Achieving Leakage Resilience through Dual System Encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011)
27. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
28. Naor, M., Segev, G.: Public-Key Cryptosystems Resilient to Key Leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
29. Rothblum, R.: Homomorphic Encryption: From Private-Key to Public-Key. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 219–234. Springer, Heidelberg (2011)
30. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
31. Stinson, D.R.: On some methods for unconditionally secure key distribution and broadcast encryption. Des. Codes Cryptography 12(3), 215–243 (1997)
32. Stinson, D.R., van Trung, T.: Some new results on key distribution patterns and broadcast encryption. Des. Codes Cryptography 14(3), 261–279 (1998)
33. Stinson, D.R., Wei, R.: Combinatorial properties and constructions of traceability schemes and frameproof codes. SIAM J. Discret. Math. 11(1), 41–53 (1998)
34. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
35. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)