Targeted Homomorphic Attribute-Based Encryption

Zvika Brakerski[1(✉)], David Cash[2], Rotem Tsabary[1], and Hoeteck Wee[3]

[1] Weizmann Institute of Science, Rehovot, Israel
{zvika.brakerski,rotem.tsabary}@weizmann.ac.il
[2] Rutgers University, New Brunswick, USA
david.cash@cs.rutgers.edu
[3] ENS, CNRS and Columbia University, Paris, France
wee@di.ens.fr

Abstract. In (key-policy) attribute-based encryption (ABE), messages are encrypted respective to attributes x, and keys are generated respective to policy functions f. The ciphertext is decryptable by a key only if $f(x) = 0$. Adding homomorphic capabilities to ABE is a long standing open problem, with current techniques only allowing compact homomorphic evaluation on ciphertext respective to the same x. Recent advances in the study of multi-key FHE also allow cross-attribute homomorphism with ciphertext size growing (quadratically) with the number of input ciphertexts.

We present an ABE scheme where homomorphic operations can be performed compactly across attributes. Of course, decrypting the resulting ciphertext needs to be done with a key respective to a policy f with $f(x_i) = 0$ for *all* attributes involved in the computation. In our scheme, the *target policy* f needs to be known to the evaluator, we call this *targeted homomorphism*. Our scheme is secure under the polynomial hardness of learning with errors (LWE) with sub-exponential modulus-to-noise ratio.

We present a second scheme where there needs not be a single target policy. Instead, the decryptor only needs a set of keys representing policies f_j s.t. for any attribute x_i there exists f_j with $f_j(x_i) = 0$. In this scheme, the ciphertext size grows (quadratically) with the size of the *set of policies* (and is still independent of the number of inputs or attributes). Again, the target set of policies needs to be known at evaluation time. This latter scheme is secure in the random oracle model under the polynomial hardness of LWE with sub-exponential noise ratio.

For the full and most up-to-date version of this work, see Cryptology ePrint Archive http://eprint.iacr.org/2016/691.

Z. Brakerski and R. Tsabary—Supported by the Israel Science Foundation (Grant No. 468/14), the Alon Young Faculty Fellowship, Binational Science Foundation (Grant No. 712307) and Google Faculty Research Award.

H. Wee—Supported by ERC Project aSCEND (H2020 639554) and NSF Award CNS-1445424.

M. Hirt and A. Smith (Eds.): TCC 2016-B, Part II, LNCS 9986, pp. 330–360, 2016.
DOI: 10.1007/978-3-662-53644-5_13

1 Introduction

Consider a situation where a large number of data items $\mu_1, \mu_2, \ldots$ is stored on a remote cloud server. For privacy purposes, the data items are encrypted. The user, who holds the decryption key, can retrieve the encrypted data and decrypt it locally. Using fully homomorphic encryption (FHE) [20,34], it can also ask the server to evaluate a function g on the encrypted data, and produce an encryption of $g(\mu_1, \mu_2, \ldots)$ which can be sent back for decryption, all without compromising privacy. The state of the art homomorphic encryption schemes can be based on the hardness of the learning with errors (LWE) problem, and of particular importance to us is the scheme of Gentry et al. [22]. However, one could consider a case where the data belongs to a big organization, where different position holders have different access permissions to the data. That is, every user can only access some fraction of the encrypted items. A trivial solution would be to duplicate each data item, and encrypt each copy using the public keys of all permitted users. However, this might be unsatisfactory in many cases.

Attribute-based encryption (ABE) [26,35] is a special type of public-key encryption scheme that allows to implement access control.[1] A (master) public key mpk is used for encryption, and users are associated to secret keys sk_f corresponding to policy functions $f : \{0,1\}^\ell \to \{0,1\}$. The encryption of a message μ is labeled with a public attribute $x \in \{0,1\}^\ell$, and can be decrypted using sk_f if and only if $f(x) = 0$.[2] The security guarantee of ABE is collusion resistance: a coalition of users learns nothing about the plaintext message μ if none of their individual keys are authorized to decrypt the ciphertext. Goyal et al. [26] used bilinear maps to construct ABE for log-depth circuits. Gorbunov et al. [23] showed the first ABE scheme where the policies can be arbitrary (a-priori bounded) polynomial circuits, based on LWE. A scheme with improved parameters was presented by Boneh et al. [5].

Using ABE for encrypting our remote data, a user with access permission to a certain data item can retrieve and decrypt it, but what about private processing on the server side? This would require *homomorphic attribute-based encryption* (HABE). Intuitively, we would like a way for a user to specify a set of data items for which it has permission, as well as a function g to be applied, such that the server can evaluate g on those data items. We would like this procedure to be private, i.e. the server learns nothing about the contents, and compact, i.e. the size of the evaluated response is independent of the number of inputs and the complexity of g.

Gentry et al. [22] showed how to achieve this goal in the case where all items of interest have the same attribute x, but cannot allow any homomorphism across attributes, even if the decryptor is allowed to access all of them. It is possible to compose a standard ABE scheme together with *multi-key* FHE [16,27,31] to achieve HABE, at the cost of blowing up the ciphertext size with the number of inputs to the homomorphic function. We provide a proof for this fact in Appendix A.

[1] Throughout this work we will consider the flavor known as "key-policy" ABE.

[2] In the original formulation, the convention was opposite: that $f(x) = 1$ allows to decrypt. However in this work we use $f(x) = 0$ throughout.

1.1 Our Results

We show that under a proper relaxed formulation of the problem, there is a solution that allows cross-attribute evaluation, with the resulting ciphertext size not depending on the number of attributes at all. In the motivating example above, if the remote server holds various encrypted items under various attributes, then the client must specify which of these ciphertexts are allowed to participate in the computation. In our formulation, this is done by providing the server with the *policy* f associated with the user's decryption key (note that this is public information that does not compromise data privacy). The policy is a compact representation that indicates which attributes are accessible by the user and which are not, so the server can tell which ciphertexts are to be included. We call our notion targeted HABE (T-HABE) since the evaluator needs to know the *target policy* which will be used to decrypt the homomorphically evaluated ciphertext. We believe that our formulation can be useful in some situations, as illustrated by the motivating example above.

So far we discussed the case where the decryptor only has one secret key corresponding to a single policy, we call this *single target* (or single policy) HABE (ST-HABE). We extend this notion and consider *multi target* (or multi policy) HABE (MT-HABE), where the decryptor is defined not just by a single policy f, but rather by a collection of policies F. This means that the decryptor holds all $\{\mathsf{sk}_f : f \in F\}$ and is thus allowed to decrypt ciphertexts with attribute x s.t. there exists $f \in F$ with $f(x) = 0$. This can be thought of as a single user with multiple keys, or as a collection of users who wish to perform homomorphic computation on the union of their permitted data items. In this setting, target homomorphism requires F to be known to the homomorphic evaluator. This notion trivially degenerates to the single-policy variant if F is a singleton set. A formal definition of T-HABE appears in Sect. 2.

We construct new ST-HABE and MT-HABE schemes as follows. In the single target setting, our scheme relies on the same hardness assumptions as previous (standard) ABE candidates [5,23], namely the polynomial hardness of learning with errors (LWE) with sub-exponential modulus-to-noise ratio. Our scheme is leveled both for policies and for homomorphic evaluation, which means that at setup time one can specify arbitrary depth bounds, and once they are set, all policies f and homomorphicly evaluated functions g must adhere by these bounds. We note that in terms of assumptions and functionality, our scheme performs as well as any known ABE for circuits and as well as any known FHE scheme (without bootstrapping). In fact, using the composition theorem in [17], we can get non-leveled full homomorphism. However, this requires a non-leveled MK-FHE as a building block, which is only known to exist under a circular security assumption (see e.g. [10]). We note that whereas the [17] result is stated for non-targeted HABE, it applies readily in this setting as well. See an outline of our construction in Sect. 1.2 below, and the full scheme in Sect. 4.

Our MT-HABE scheme relies on the same assumption but in the *random oracle model*, and furthermore the ciphertext size grows quadratically with the cardinality of the set F (i.e. if more policies are involved, more communication is

needed),[3] however the ciphertext size is independent of the number of attributes and the complexity of g. Interestingly, we use the random oracle in order to generate a part of the *secret key*, and we show that security is still maintained. See an outline of our construction in Sect. 1.2 below, and the full scheme in Sect. 5.

1.2 Our Techniques

Previous works [11,16,22] observed that known LWE-based ABE schemes have the following structure. Given the public parameters pp and an attribute x, it is possible to derive a "designated public key" pk_x, which has the same structure as a public-key for Regev's famous encryption scheme [33] (more precisely "dual-Regev", introduced by Gentry et al. [21]), and indeed the encryption process is also identical to the dual-Regev scheme. Therefore, since the FHE scheme of Gentry, Sahai and Waters [22] (henceforth GSW) has the same key distribution as dual-Regev, one can just substitute the encryption procedure from dual-Regev to GSW, and single attribute homomorphism follows. To show that the evaluated ciphertext can be decrypted, GSW notice that the decryption procedure of the [23] ABE scheme can be seen as a two step process: first sk_f is preprocessed together with x to obtained $\mathsf{sk}_{f,x}$ which is a valid dual-Regev secret key for pk_x, and this key is used for standard dual-Regev decryption. This means that this key can also be used to decrypt GSW evaluated ciphertexts. This observation also carries over to the later ABE scheme of Boneh et al. [5]. A similar approach was used by Clear and McGoldrick [16] in conjunction with their multi-key homomorphism to achieve a homomorphic IBE (ABE where the policies are only point functions) where the ciphertext size grows with the number of attributes.

Our starting point is to consider a "dual" two-step decryption process for the [5] ABE, where given a ciphertext c_x relative to an attribute x, it is first preprocessed together with f to obtain $c_{x,f}$ which can then be decrypted by sk_f as a standard dual-Regev ciphertext. This is not a new perspective, in fact this is the original way [5,23] described their decryption process. We would hope, therefore, to apply targeted homomorphism by first preprocessing all input ciphertexts to make them correspond to the same sk_f, and then apply homomorphic evaluation. However, applied naively, preprocessing a GSW ciphertext destroys its homomorphic features. This is the reason GSW needed to reinterpret the decryption process in order for their approach to work even in the single input setting. We show how to modify the encryption procedure so as to allow preprocessing of a ciphertext *for any policy function* f without compromising its homomorphic features, which will allow to achieve targeted homomorphism for single policy (ST-HABE).

Our multi-target solution relies on the multi-key FHE scheme of [16], and in particular we use the simplified variant of Mukherjee and Wichs [31]. Recall that we have a set F of policies, where each attribute x in the computation has at least

[3] As in previous works, part of the ciphertext is redundant for decryption and can be truncated post-evaluation, which will lead to only linear dependence on $|F|$.

one policy $f \in F$ that can decrypt it. The basic idea is to group the ciphertexts according to the f's, preprocess them so all ciphertexts that correspond to a given f are now respective to the same (unknown) secret key sk_f. After preprocessing, the situation is equivalent to multi-key FHE with $|F|$ many users, each with their own key, so it would appear that we are in the clear. However, known LWE-based multi-key FHE schemes require *common public parameters*. In particular, all public keys are matrices which are identical except the last row, all secret keys are vectors with the last element being equal to 1. However, our preprocessing does not produce ciphertexts that conform with this requirement. In particular, our ciphertexts correspond to public keys that all share a prefix, but they differ in much more than a single row. We show that the [16,31] scheme can be generalized to the aforementioned case, however a fraction of the secret key needs to be known at homomorphic evaluation time. Whereas revealing this fraction of the key does not compromise security, it is generated independently for each policy f using the master secret key, and there appears to be no compact way to provide the key fractions for *all* policies in the public parameters. We resolve this using the random oracle heuristic, namely we show that we can generate a fraction of the secret key using the random oracle, which allows the homomorphic evaluator to learn the allowable part of all relevant keys and perform the multi-key homomorphism.

1.3 A More Formal Overview

Syntax. As mentioned earlier, in an ABE, ciphertexts are associated with an attribute x and a message μ, and decryption is possible using sk_f iff $f(x) = 0$. In a single-attribute homomorphic ABE, an evaluator given encryptions of $\mu_1, \mu_2, \ldots$, under the same attribute x and any circuit g, can compute an encryption of $g(\mu_1, \mu_2, \ldots)$ under the same attribute x. In a ST-HABE, an evaluator given encryptions of $\mu_1, \mu_2, \ldots$ under different attributes $x_1, x_2, \ldots$, any circuit g and a "target" f for which $f(x_1) = f(x_2) = \cdots = 0$, outputs a ciphertext that decrypts to $g(\mu_1, \mu_2, \ldots)$ under sk_f.

Prior ABE. We recall that in the [5] ABE, the public parameters contain a matrix $\mathbf{A}$, a vector $\mathbf{v}$ and a set of matrices $\mathbf{B}_1, \ldots, \mathbf{B}_\ell$, where ℓ is the supported attribute length. For all $x \in \{0,1\}^\ell$, we can define $\mathbf{B}_x = [\mathbf{B}_1 - x_1\mathbf{G} \| \cdots \| \mathbf{B}_\ell - x_\ell \mathbf{G}]$, where $\mathbf{G}$ is the special "gadget" matrix, and use dual-Regev to encrypt messages w.r.t $[\mathbf{A} \| \mathbf{B}_x], \mathbf{v}$. Namely the ciphertexts are of the form $\mathbf{c} \approx [\mathbf{A} \| \mathbf{B}_x \| \mathbf{v}]^T \mathbf{s} + \mathbf{y}_\mu$, where $\mathbf{y}_\mu$ is some vector that encodes the message. Furthermore, given f, $\mathbf{B}_1, \ldots, \mathbf{B}_\ell$ can be preprocessed to obtain a matrix $\mathbf{B}_f$, and for all f, x s.t. $f(x) = 0$, there exists a publicly computable low-norm matrix $\mathbf{H} = \mathbf{H}_{f,x,\mathbf{B}_x}$ s.t. $\mathbf{B}_f = \mathbf{B}_x \mathbf{H}$. The secret key is a row vector $\mathsf{sk}_f = \mathbf{r}_f$ s.t. $\mathbf{r}_f [\mathbf{A} \| \mathbf{B}_f]^T = -\mathbf{v}^T$. Decryption proceeds by using $\widehat{\mathbf{H}} = \mathrm{diag}(\mathbf{I}, \mathbf{H}, 1)$ (i.e. a diagonal block matrix whose blocks are $\mathbf{I}, \mathbf{H}, 1$) to compute $\mathbf{c}_f = \widehat{\mathbf{H}}^T \mathbf{c}$ so that $\mathbf{c}_f \approx [\mathbf{A} \| \mathbf{B}_f \| \mathbf{v}]^T \mathbf{s} + \widehat{\mathbf{H}} \mathbf{y}_\mu$, and then using $\mathbf{r}_f$ to decrypt.

Warm-Up. Recall that in GSW style FHE, an encryption of μ under a secret key $\mathbf{r}$ is a matrix $\mathbf{D}+\mu\mathbf{G}$ where $\mathbf{rD} \approx \mathbf{0}^T$, where $\mathbf{G}$ is a gadget matrix of appropriate dimensions. As a warm-up, suppose we encrypt μ as

$$\mathbf{C} \approx [\mathbf{A}\|\mathbf{B}_x\|\mathbf{v}]^T\mathbf{S} + \mu\mathbf{G}.$$

That is, each column in the new ciphertext is essentially a ciphertext of the aforementioned ABE scheme (with different $\mathbf{y}$ in each column). Observe that $[\mathbf{r}_f\|1][\mathbf{A}\|\mathbf{B}_x\|\mathbf{v}]^T\mathbf{S} \approx \mathbf{0}^T$, so $\mathbf{C}$ is indeed a GSW style encryption of μ under the secret key $[\mathbf{r}_f\|1]$.

In order to achieve cross-attribute homomorphism, we would like to replace the matrix $[\mathbf{A}\|\mathbf{B}_x\|\mathbf{v}]^T\mathbf{S}$ in $\mathbf{C}$ with one that depends only on f and not x. Towards this goal, observe that

$$\widehat{\mathbf{H}}^T\mathbf{C} \approx [\mathbf{A}\|\mathbf{B}_f\|\mathbf{v}]^T\mathbf{S} + \mu\widehat{\mathbf{H}}^T\mathbf{G}.$$

Unfortunately, this is not a GSW style FHE ciphertext as described above because we have $\widehat{\mathbf{H}}^T\mathbf{G}$ instead of $\mathbf{G}$. In fact, GSW style homomorphic evaluation can be still made to work if we can ensure that $\widehat{\mathbf{H}}^T\mathbf{G}$ behaves like a gadget matrix (e.g. if the matrix $\widehat{\mathbf{H}}^T$ has a low-norm inverse, which is not true for a general $\mathbf{H}_{f,x,\mathbf{B}_x}$); instead, we provide a simpler fix that also yields shorter ciphertexts.

Our ST-HABE Scheme. Our ST-HABE ciphertext has two components. The first one is independent of x: $\mathbf{C} \approx [\mathbf{A}\|\mathbf{B}_0\|\mathbf{v}]^T\mathbf{S} + \mu\mathbf{G}$, where $\mathbf{B}_0$ is another matrix, like the other $\mathbf{B}_i$'s, which is added to the public parameters. The second one is similar to an ABE encryption of 0, with the same $\mathbf{S}$: $\mathbf{C}_x \approx \mathbf{B}_x^T\mathbf{S}$. Now, observe that

$$\mathbf{C}_f := \mathbf{C} + [\mathbf{0}\|\mathbf{H}^T\mathbf{C}_x\|\mathbf{0}] \approx [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_f\|\mathbf{v}]^T\mathbf{S} + \mu\mathbf{G},$$

since $\mathbf{H}^T\mathbf{B}_x^T = \mathbf{B}_f^T$. Note that $\mathbf{C}_f$ is now indeed a GSW FHE ciphertext under the key $[\mathbf{r}_f\|1]$, where $\mathbf{r}_f$ is the modified ABE secret key satisfying

$$\mathbf{r}_f[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_f]^T = \mathbf{v}^T.$$

The proof of security for the modified ABE scheme is very similar to that of [5] (in the simulation, we program $\mathbf{B}_0$ as $\mathbf{AR}_0$). See Sect. 4 for more details.

Our MT-HABE Scheme. For the multi-policy setting, assume for simplicity that we only have two attributes x, x' and two policies f, f' s.t. $f(x) = 0$, $f'(x') = 0$ (generalization is straightforward). After applying the transformation as above, we have $\mathbf{C}_f \approx [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_f\|\mathbf{v}]^T\mathbf{S} + \mu\mathbf{G}$ and likewise for f'. In the background there are the secret keys $\mathbf{r}_f, \mathbf{r}_{f'}$. Let us partition $\mathbf{r}_f = [\mathbf{r}_1, \mathbf{r}_2]$, s.t. $\mathbf{r}_1\mathbf{A}^T + \mathbf{r}_2(\mathbf{B}_0 + \mathbf{B}_f)^T = -\mathbf{v}^T$. Likewise $\mathbf{r}_{f'} = [\mathbf{r}'_1, \mathbf{r}'_2]$. We show that the methods of [16,31] for achieving multi-key homomorphism generalize fairly straightforwardly whenever the value of the cross multiplication $\mathbf{r}_f[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{f'}\|\mathbf{v}]^T$ is publicly

computable (note that the secret key for f is multiplied by the public key for f', and vice versa). One can verify that if the $\mathbf{r}_2$ components of the two keys are known, then this is indeed the case. Our approach is therefore to achieve multi-policy homomorphism by releasing the $\mathbf{r}_2$ components of the keys. This approach might seem risky, since information about the secret key is revealed. To see why this is not a problem, we recall that the key $\mathbf{r}_f$ is generated using a trapdoor for $\mathbf{A}$ such that $\mathbf{r}_f$ is distributed like a discrete Gaussian, conditioned on $\mathbf{r}_1\mathbf{A}^T + \mathbf{r}_2(\mathbf{B}_0 + \mathbf{B}_f)^T = -\mathbf{v}^T$. One can verify that the marginal distribution of $\mathbf{r}_2$ is Gaussian and completely independent of f (this fact had been utilized in [1,14]). Therefore there seems to be hope that releasing it might not hurt security. Another serious problem is that $\mathbf{r}_2$ is generated using secret information, and is not known to the evaluator. Unfortunately, we are only able to resolve this difficulty in the random oracle model, by generating $\mathbf{r}_2$ using the random oracle. Specifically, we apply the random oracle to $(\mathbf{A}, f)$ to obtain $\mathbf{r}_2$ for f. In a nutshell, producing $\mathbf{r}_2$ using a random oracle is secure since the security reduction can always program the response of the random oracle: if the call is on a function f s.t. $f(x^*) = 1$ (where x^* is the challenge attribute) then returning $\mathbf{r}_2$ is similar to answering a key generation query, and if $f(x^*) = 0$ then a random value can be returned, since a key generation query to f will never be issued and therefore no consistency issues arise. However, as described so far, this solution requires a special random oracle: one that samples from a discrete Gaussian distribution. We would like to rely on the standard binary random oracle. To this end, we will set $\mathbf{r}_f = [\mathbf{r}_1, \mathbf{r}_2]$ such that $\mathbf{r}_1$ is Gaussian and $\mathbf{r}_2$ is *binary*, conditioned on $\mathbf{r}_1\mathbf{A}^T + \mathbf{r}_2(\mathbf{B}_0 + \mathbf{B}_f)^T = -\mathbf{v}^T$. This will allow us to use a standard binary random oracle for the generation of $\mathbf{r}_2$.[4] In the proof of security, we use the discrete Gaussian sampler of Lyubashevsky and Wichs [28] instead of the Gaussian sampler of [2,30]. This sampler, which is based on rejection sampling, allows to sample from "partially Gaussian" distributions which is exactly what we need in order for the proof of security to go through. See Sect. 5 for more details. We note that for the sake of consistency, we also use this distribution of $\mathbf{r}_f$ in our single target scheme.

1.4 Other Related Work

Other works on homomorphic ABE include the works of Clear and McGoldrick [15,17]. In the former, program obfuscation is used to enhance the homomorphic ABE of [22] to support evaluating circuits of arbitrary depth. Still, cross-attribute homomorphism is not addressed. In the latter, it is shown how to use bootstrapping to leverage cross-attribute homomorphism into evaluating circuits without a depth bound. This result can be used in conjunction with our construction from Appendix A to achieve a non-compact solution, or in conjunction with our targeted scheme as explained above.

[4] Alternatively we could have shown that the Gaussian random oracle model is implied by the standard random oracle model. However this requires a fairly involved argument that we chose to avoid.

Brakerski and Vaikuntanathan [13] show how to extend the [5] ABE scheme to support attributes of unbounded polynomial length, and to provide semi-adaptive security guarantee. This was generalized by Goyal et al. [25] to a generic transformation that does not rely on the specific properties of the ABE scheme. Whereas the semi-adaptive transformation appears to be applicable here, it is not clear whether we can support unbounded attribute length using their methods and still maintain homomorphism. We leave this avenue of research for future work.

2 Targeted Homomorphic ABE

In this work, we define a notion of homomorphic ABE where the homomorphic evaluation process depends on the policy (or policies) that are used to decrypt the resulting ciphertext, we refer to such schemes as Targeted Homomorphic ABE (T-HABE). We start by defining the syntax of a T-HABE scheme, and proceed with definitions of correctness and security.

Definition 1 (Targeted Homomorphic ABE). *A* Targeted Homomorphic Attribute Based Encryption *(T-HABE) scheme is a tuple of* PPT *algorithms* $\mathsf{THABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{TEval}, \mathsf{Dec})$ *with the following syntax:*

- $\mathsf{THABE.Setup}(1^\lambda)$ *takes as input the security parameter (and possibly in addition some specification of the class of policies and class of homomorphic operations supported). It outputs a master secret key* msk *and a set of public parameters* pp.
- $\mathsf{THABE.Enc}_{\mathsf{pp}}(\mu, x)$ *uses the public parameters* pp *and takes as input a message* $\mu \in \{0,1\}$ *and an attribute* $x \in \{0,1\}^*$. *It outputs a ciphertext* ct.
- $\mathsf{THABE.Keygen}_{\mathsf{msk}}(f)$ *uses the master secret key* msk *and takes as input a policy* $f \in \mathcal{F}$. *It outputs a secret key* sk_f.
- $\mathsf{THABE.TEval}_{\mathsf{pp}}(F, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$ *uses the public parameters* pp *and takes as input a set* F *of target policies,* k *ciphertexts* $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}$ *and a function* $g \in \mathcal{G}$. *It outputs a ciphertext* ct^g.
- $\mathsf{THABE.Dec}(\mathsf{sk}_F, \mathsf{ct}^g)$ *takes as input a set of secret keys* sk_F *for a set of policies* F, *with* $\mathsf{sk}_F = \{\mathsf{sk}_f : f \in F\}$, *and a ciphertext* ct^g. *It outputs a message* $\mu \in \{0,1\}$.

We will also consider a restriction of the above definition to the single-target setting, where the set F *is only allowed to contain a single function. We call this Single Target HABE (ST-HABE). Explicit reference to the multi target setting is denoted MT-HABE.*

Our correctness guarantee is that given the set of keys for the policy set F, an evaluated ciphertext decrypts correctly to the intended value.

Definition 2 (Correctness of T-HABE). *Let* $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a class of policy functions and* $\{\mathcal{G}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a class of evaluation functions. We say that* $\mathsf{THABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{TEval}, \mathsf{Dec})$ *is correct w.r.t* $\mathcal{F}, \mathcal{G}$ *if the following holds.*

Let $(\mathsf{msk}, \mathsf{pp}) = \mathsf{THABE.Setup}(1^\lambda)$. *Consider a set of functions* $F \subseteq \mathcal{F}_\lambda$ *of* $\mathrm{poly}(\lambda)$ *cardinality, and its matching set of secret keys* $\mathsf{sk}_F = \{\mathsf{sk}_f = \mathsf{THABE.Keygen}_{\mathsf{msk}}(f) : f \in F\}$, *a sequence of* $k \geq 1$ *messages and attributes* $\{(\mu^{(i)} \in \{0,1\}, x^{(i)} \in \{0,1\}^*)\}_{i\in[k]}$ *such that* $\forall x^{(i)}.\ \exists f \in F.\ f(x^{(i)}) = 0$, *and the sequence of their encryptions* $\{\mathsf{ct}^{(i)} = \mathsf{THABE.Enc}_{\mathsf{pp}}(\mu^{(i)}, x^{(i)})\}_{i\in[k]}$.

Then letting $g \in \mathcal{G}$ *for some* $g \in \{0,1\}^k \to \{0,1\}$, *and computing* $\mathsf{ct}^g = \mathsf{THABE.TEval}(F, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$, *it holds that*

$$\Pr[\mathsf{THABE.Dec}(\mathsf{sk}_F, \mathsf{ct}^g) \neq g(\mu^{(1)}, \ldots, \mu^{(k)})] = \mathrm{negl}(\lambda),$$

where the probability is taken over all of the randomness in the experiment.

We note that similarly to the definition of correctness of homomorphic encryption, we do not require correctness for ciphertexts that did not undergo homomorphic evaluation. However, this can be assumed w.l.o.g since the class $\mathcal{G}$ will always contain the identity function which will allow decryption by first evaluating identity and then decrypting.

Security is defined using the exact same experiment as standard ABE.

Definition 3 (Security for ABE/T-HABE). *Let* THABE *be an T-HABE scheme as above, and consider the following game between the challenger and adversary.*

1. *The adversary sends an attribute* x^* *to the challenger.*
2. *The challenger generates* $(\mathsf{msk}, \mathsf{pp}) = \mathsf{THABE.Setup}(1^\lambda)$ *and sends* pp *to the adversary.*
3. *The adversary makes arbitrarily many key queries by sending functions* f_i *(represented as circuits) to the challenger. Upon receiving such function, the challenger creates a key* $\mathsf{sk}_i = \mathsf{THABE.Keygen}_{\mathsf{msk}}(f_i)$ *and sends* sk_i *to the adversary.*
4. *The adversary sends a pair of messages* μ_0, μ_1 *to the challenger. The challenger samples* $b \in \{0,1\}$ *and computes* $\mathsf{ct}^* = \mathsf{THABE.Enc}_{\mathsf{pp}}(\mu_b, x^*)$. *It sends* ct^* *to the adversary.*
5. *The adversary makes arbitrarily many key queries as in Step 3 above.*
6. *The adversary outputs* $\tilde{b} \in \{0,1\}$.
7. *Let* legal *denote the event where all key queries of the adversary are such that* $f_i(x^*) = 1$. *If* legal, *the output of the game is* $b' = \tilde{b}$, *otherwise the output* b' *is a uniformly random bit.*

The advantage *of an adversary* $\mathcal{A}$ *is* $|\Pr[b' = b] - 1/2|$, *where* b, b' *are generated in the game played between the challenger and the adversary* $\mathcal{A}(1^\lambda)$.

The game above is called the selective security *game, because the adversary sends* x^* *before Step 2. The scheme* THABE *is* selectively secure *if any* PPT *adversary* $\mathcal{A}$ *only has negligible advantage in the selective security game.*

Stronger notions of security include semi-adaptive *security where step 1 only happens after step 2, and* adaptive *(or full) security where step 1 only happens after step 3.*

We note that the adversary has no benefit in making key queries for policies for which $f(x^*) = 0$ and therefore we can assume w.l.o.g that such queries are not made (this is obvious for selective and semi-adaptive security and slightly less obvious for adaptive security).

Negated Policies. We note again that as in previous lattice based ABE constructions, we allow decryption when $f(x) = 0$ and require that in the security game all queries are such that $f(x^*) = 1$.

3 Preliminaries

We denote vectors by lower-case bold letters (e.g. $\mathbf{v}$) and matrices by upper-case bold letters (e.g. $\mathbf{A}$). The i'th component of a vector $\mathbf{v}$ is denoted by v_i. The component in the ith row and the jth column of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}[i, j]$. We denote the security parameter by λ and let $\mathrm{negl}(\lambda)$ denote a negligible function. Sets and distributions are usually denoted in plain uppercase. If S is a set, then we also use S to denote the uniform distribution over this set. The distinction will be clear from the context.

Elements of $\mathbb{Z}_q$ are represented by the integers in $(-q/2, q/2]$. In particular the absolute value of $x \in \mathbb{Z}_q$ is defined as $|x| = \min\{|y| : y \in \mathbb{Z}, y = x \pmod q\}$.

As in many previous works relying on the LWE assumption, we rely on distributions that are supported over a bounded domain. A distribution χ over $\mathbb{Z}$ is said to be *B-bounded* if it is supported only over $[-B, B]$. The infinity norm of a matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_\infty = \max_{i,j} |\mathbf{A}[i, j]|$, and we write

$$\mathbf{A} \approx \mathbf{B} \quad (\mathsf{err}\colon B)$$

to denote that $\|\mathbf{A} - \mathbf{B}\|_\infty \leq B$.

3.1 Learning with Errors (LWE)

The *Learning with Errors* (LWE) problem was introduced by Regev [33]. Our scheme relies on the hardness of its decisional version.

Definition 4 (Decisional LWE(DLWE) [33]**).** *Let λ be the security parameter, $n = n(\lambda)$ and $q = q(\lambda)$ be integers and let $\chi = \chi(\lambda)$ be a probability distribution over $\mathbb{Z}$. The* $\mathrm{DLWE}_{n,q,\chi}$ *problem states that for all $m = \mathrm{poly}(n)$, letting $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, it holds that $\left(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right)$ and $\left(\mathbf{A}, \mathbf{u}^T\right)$ are computationally indistinguishable.*

In this work we only consider the case where $q \leq 2^n$. Recall that GapSVP_γ is the (promise) problem of distinguishing, given a basis for a lattice and a parameter d, between the case where the lattice has a vector shorter than d, and the case where the lattice doesn't have any vector shorter than $\gamma \cdot d$. SIVP is the search problem of finding a set of "short" vectors. The best known algorithms for GapSVP_γ ([36]) require at least $2^{\tilde{\Omega}(n/\log\gamma)}$ time. We refer the reader to [32,33] for more information.

There are known reductions between $\mathrm{DLWE}_{n,q,\chi}$ and those problems, which allows us to appropriately choose the LWE parameters for our scheme. We summarize in the following corollary (which addresses the regime of sub-exponential modulus-to-noise ratio).

Corollary 1 ([29,30,32,33]). *For all $\epsilon > 0$ there exist functions $q = q(n) \leq 2^n, \chi = \chi(n)$ and $B = B(n)$ such that χ is B-bounded, $q/B \geq 2^{n^\epsilon}$ and such that $\mathrm{DLWE}_{n,q,\chi}$ is at least as hard as the classical hardness of GapSVP_γ and the quantum hardness of SIVP_γ for $\gamma = 2^{\Omega(n^\epsilon)}$.*

3.2 The Gadget Matrix

Let $\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}_q^{\lceil \log q \rceil}$ and let $N = n \cdot \lceil \log q \rceil$. The *gadget matrix* $\mathbf{G}_n$ is defined as the diagonal concatenation of $\mathbf{g}$ n times. Formally, $\mathbf{G}_n = \mathbf{g} \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n \times N}$. We omit the n when the dimension is clear from the context.

We define the inverse function $\mathbf{G}_n^{-1} : \mathbb{Z}_q^{n \times m} \to \{0,1\}^{N \times m}$ which expands each entry $a \in \mathbb{Z}_q$ of the input matrix into a column of size $\lceil \log q \rceil$ consisting of the bits of the binary representation of a. We have the property that for any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, it holds that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A}$.

3.3 Trapdoors and Discrete Gaussians

Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. For all $\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$ and for any distribution P over $\mathbb{Z}^m$, we let $\mathbf{A}_P^{-1}(\mathbf{V})$ denote the random variable whose distribution is P conditioned on $\mathbf{A} \cdot \mathbf{A}_P^{-1}(\mathbf{V}) = \mathbf{V}$. A P-trapdoor for $\mathbf{A}$ is a procedure that can sample from a distribution within 2^{-n} statistical distance of $\mathbf{A}_P^{-1}(\mathbf{V})$ in time $\mathrm{poly}(n, m, m', \log q)$, for any $\mathbf{V}$. We slightly overload notation and denote a P-trapdoor for $\mathbf{A}$ by $\mathbf{A}_P^{-1}$.

The (centered) discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter τ, denoted $D_{\mathbb{Z}^m,\tau}$, is the distribution over $\mathbb{Z}^m$ where for all $\mathbf{x}$, $\Pr[\mathbf{x}] \propto e^{-\pi \|\mathbf{x}\|^2/\tau^2}$. When P is the Discrete Gaussian $D_{\mathbb{Z}^m,\tau}$, we denote $\mathbf{A}_P^{-1} = \mathbf{A}_\tau^{-1}$.

It had been established in a long sequence of works that it is possible to generate an almost uniform $\mathbf{A}$ together with a trapdoor as formalized below (the parameters are taken from [30] together with the Gaussian sampler of [9,21]).

Corollary 2 (Trapdoor Generation). *There exists an efficient procedure $\mathsf{TrapGen}(1^n, q, m)$ that outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for all $m \geq m_0$ for $m_0 = O(n \log q)$, $\mathbf{A}$ is 2^{-n}-uniform and $\tau_0 = O(\sqrt{n \log q \log n})$. Furthermore, given $\mathbf{A}_{\tau_0}^{-1}$, one can obtain $\mathbf{A}_\tau^{-1}$ for any $\tau \geq \tau_0$.*

We will also use the "mixed" Gaussian-Binary sampler of Lyubashevsky and Wichs [28]. The following corollary is a consequence of example 2 in [28, Sect. 3.2], by adjusting the analysis for general $\mathbf{R}$ instead of random $\{-1, 0, 1\}$ entries.

Corollary 3 (Gaussian-Binary Sampler). *Let n, m, q be such that $m \geq n\lceil \log q \rceil$. With all but $O(2^{-n})$ probability over the choice of $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, for all $\mathbf{R} \in \mathbb{Z}^{m \times N}$ with $N = n\lceil \log q \rceil$, one can obtain $[\mathbf{A} \| \mathbf{AR} + \mathbf{G}_n]_P^{-1}$ for $P = D_{\mathbb{Z}^m, \tau} \times \{0,1\}^N$ with $\tau = O(N\sqrt{mn} \cdot \|\mathbf{R}\|_\infty)$. Furthermore, for all $\mathbf{v}$, it holds that the marginal distribution of the last N coordinates of $[\mathbf{A} \| \mathbf{AR} + \mathbf{G}_n]_P^{-1}(\mathbf{v})$ are $O(2^{-n})$-uniform in $\{0,1\}^N$.*

3.4 Homomorphic Evaluation

We define the basic procedure that will be used for homomorphic evaluation of FHE ciphertexts and also in the ABE scheme [5,22,24].

Definition 5. *Let $n, q \in \mathbb{N}$. Consider $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n \times N}$ where $N = n\lceil \log q \rceil$, and denote $\vec{\mathbf{B}} = [\mathbf{B}_1 \| \cdots \| \mathbf{B}_\ell]$. Let f be a boolean circuit of depth d computing a function $\{0,1\}^\ell \to \{0,1\}$, and assume that f contains only NAND gates. We define $\mathbf{B}_f = \mathsf{Eval}(f, \vec{\mathbf{B}})$ recursively: associate $\mathbf{B}_1, \ldots, \mathbf{B}_\ell$ with the input wires of the circuit. For every wire w in f, let u, v be its predecessors and define*

$$\mathbf{B}_w = \mathbf{G} - \mathbf{B}_u \mathbf{G}^{-1}(\mathbf{B}_v). \tag{1}$$

Finally $\mathbf{B}_f$ is the matrix associated with the output wire.

The properties of Eval are summarized in the following facts.

Fact 1. *Consider $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n \times N}$ and $x \in \{0,1\}^\ell$. Denoting $\vec{\mathbf{B}} = [\mathbf{B}_1 \| \cdots \| \mathbf{B}_\ell]$ and $x\vec{\mathbf{G}} = [x_1\mathbf{G} \| \cdots \| x_\ell \mathbf{G}]$, it holds that there exists an polynomial time algorithm $\mathsf{EvRelation}$ s.t. if $\mathbf{H} = \mathbf{H}_{f,x,\vec{\mathbf{B}}} = \mathsf{EvRelation}(f, x, \vec{\mathbf{B}})$ then $\|\mathbf{H}\|_\infty \leq (N+1)^d$ and furthermore*

$$(\mathbf{B}_f - f(x)\mathbf{G})^T = \mathbf{H}^T \cdot [\vec{\mathbf{B}} - x\vec{\mathbf{G}}]^T$$

where $\mathbf{B}_f = \mathsf{Eval}(f, \vec{\mathbf{B}})$.

In particular, if $\mathbf{B}_i = \mathbf{AR}_i + x_i\mathbf{G}$, i.e. $\vec{\mathbf{B}} = \mathbf{A}\vec{\mathbf{R}} + x\vec{\mathbf{G}}$ for $\vec{\mathbf{R}} = [\mathbf{R}_1 \| \cdots \| \mathbf{R}_\ell]$, then $\mathbf{B}_f = \mathbf{AR}_f + f(x)\mathbf{G}$ for $\mathbf{R}_f = \vec{\mathbf{R}} \cdot \mathbf{H}_{f,x,\vec{\mathbf{B}}}$.

To see why the fact holds, note that for the NAND evaluation in Eq. (1), one can verify that

$$\mathsf{EvRelation}(\text{NAND}, [x_u, x_v], [\mathbf{B}_u \| \mathbf{B}_v]) = \begin{bmatrix} -\mathbf{G}^{-1}(\mathbf{B}_v) \\ -x_u \mathbf{I} \end{bmatrix}.$$

Recursive application implies the general statement.

Fact 2. *Let $\mathbf{r} \in \mathbb{Z}_q^n$, $\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)} \in \mathbb{Z}_q^{n \times N}$ and $\mu^{(1)}, \ldots, \mu^{(k)} \in \{0,1\}$, be such that*

$$\mathbf{r}^T \mathbf{C}^{(i)} \approx \mu^{(i)} \mathbf{r}^T \mathbf{G} \qquad (\mathsf{err} : B).$$

Let g be a boolean circuit of depth d computing a function $\{0,1\}^k \to \{0,1\}$, and assume that g contains only NAND gates. Let $\mathbf{C}_g = \mathsf{Eval}(g, \vec{\mathbf{C}})$, then

$$\mathbf{r}^T \mathbf{C}^{(i)} \approx g(\mu^{(1)}, \ldots, \mu^{(k)}) \mathbf{r}^T \mathbf{G} \qquad (\mathsf{err} : B \cdot (N+1)^d).$$

3.5 Pseudorandom Functions

A pseudorandom function family is a pair of PPT algorithms $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$, such that the key generation algorithm $\mathsf{PRF.Gen}(1^\lambda)$ takes as input the security parameter and outputs a seed $\sigma \in \{0,1\}^\lambda$. The evaluation algorithm $\mathsf{PRF.Eval}(\sigma, x)$ takes a seed $\sigma \in \{0,1\}^\lambda$ and an input $x \in \{0,1\}^*$, and returns a bit $y \in \{0,1\}$.

Definition 6. *A family* PRF *as above is secure if for every polynomial time adversary* $\mathcal{A}$ *it holds that*

$$\left|\Pr[\mathcal{A}^{\mathsf{PRF.Eval}(\sigma,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\mathcal{O}(\cdot)}(1^\lambda) = 1]\right| = \mathrm{negl}(\lambda),$$

where $\sigma = \mathsf{PRF.Gen}(1^\lambda)$ *and* $\mathcal{O}$ *is a random oracle. The probabilities are taken over all of the randomness of the experiment.*

4 A Single Target Homomorphic ABE Scheme

In this section we present our construction of LWE-based Single Target HABE. As in previous works, a constant $\epsilon \in (0,1)$ determines the tradeoff between the hardness of the DLWE problem on which security is based, and the efficiency of the scheme.

The scheme supports any class of policies $\mathcal{F}_{\ell,d_\mathcal{F}} \subseteq \{0,1\}^\ell \to \{0,1\}$, and any class of operations $\mathcal{G}_{d_\mathcal{G}} \subseteq \{0,1\}^* \to \{0,1\}$, where $d_\mathcal{F}, d_\mathcal{G}$ is the bound on the depth of the circuit representation of each function in the set $\mathcal{F}, \mathcal{G}$, respectively. Out scheme works for any $\ell, d_\mathcal{F}, d_\mathcal{G} = \mathrm{poly}(\lambda)$.

- $\mathsf{STHABE.Setup}(1^\lambda, 1^\ell, 1^{d_\mathcal{F}}, 1^{d_\mathcal{G}})$. Choose n, q, B, χ, m as described in Sect. 4.1 below. Let $m = \max\{m_0, (n+1)\lceil \log q\rceil + 2\lambda\}$ (where m_0 is as in Corollary 2), $N = n\lceil \log q\rceil$ and $M = (m+N+1)\lceil \log q\rceil$.
  Generate a matrix-trapdoor pair $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) = \mathsf{TrapGen}(1^n, q, m)$ (see Corollary 2), where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. Generate matrices $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell \xleftarrow{\$} \mathbb{Z}_q^{n\times N}$ and denote $\vec{\mathbf{B}} = [\mathbf{B}_1 \| \ldots \| \mathbf{B}_\ell]$. Generate a vector $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$.
  Set $\mathsf{msk} = \mathbf{A}_{\tau_0}^{-1}$ and $\mathsf{pp} = (\mathbf{A}, \mathbf{B}_0, \vec{\mathbf{B}}, \mathbf{v})$.
- $\mathsf{STHABE.Enc}_{\mathsf{pp}}(\mu, x)$, where $\mathsf{pp} = (\mathbf{A}, \mathbf{B}_0, \vec{\mathbf{B}}, \mathbf{v})$, $\mu \in \{0,1\}$ (however, this procedure is well defined for any $\mu \in \mathbb{Z}_q$ which will be useful for our next scheme) and $x \in \{0,1\}^\ell$.
  Sample a random matrix $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n\times M}$, an error matrix $\mathbf{E}_A \xleftarrow{\$} \chi^{m\times M}$ and an error row vector $\mathbf{e}_v \xleftarrow{\$} \chi^M$.
  Generate $\ell+1$ more error matrices as follows: For all $i \in [\ell]$ and $j \in [M]$, sample $\mathbf{R}_{i,j} \xleftarrow{\$} \{0,1\}^{m\times N}$. Let $\mathbf{E}_0, \ldots, \mathbf{E}_\ell$ be matrices of dimension $N \times M$ defined by $\mathbf{E}_i[j] = \mathbf{R}_{i,j}^T \mathbf{E}_A[j]$, where $\mathbf{E}_i[j]$, $\mathbf{E}_A[j]$ denotes the jth column of $\mathbf{E}_i, \mathbf{E}_A$ respectively. Let

$$\begin{bmatrix} \mathbf{C}_A \\ \mathbf{C}_0 \\ \mathbf{c}_v \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{S} + \begin{bmatrix} \mathbf{E}_A \\ \mathbf{E}_0 \\ \mathbf{e}_v \end{bmatrix} + \mu \mathbf{G}_{m+N+1}.$$

The rest of the ciphertext contains auxiliary information that will allow to decrypt given a proper functional secret key. For all $i \in [\ell]$ let

$$\mathbf{C}_i = [\mathbf{B}_i - x_i \mathbf{G}_n]^T \cdot \mathbf{S} + \mathbf{E}_i.$$

Denote $\mathbf{C}_x = \begin{bmatrix} \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_\ell \end{bmatrix}$ and $\mathbf{E}_x = \begin{bmatrix} \mathbf{E}_1 \\ \vdots \\ \mathbf{E}_\ell \end{bmatrix}$.

The final ciphertext is $\mathsf{ct} = (x, \mathbf{C}_A, \mathbf{C}_0, \mathbf{c}_v, \mathbf{C}_x)$.

– $\mathsf{STHABE.Keygen}_{\mathsf{msk}}(f)$. Given a circuit f computing a function $\{0,1\}^\ell \to \{0,1\}$, the key is generated as follows. Set $\mathbf{B}_f = \mathsf{Eval}(f, \vec{\mathbf{B}})$ (where Eval is as defined in Sect. 3.4).

Generate a random vector $\mathbf{r}'_f \xleftarrow{\$} \{0,1\}^N$. Let $\mathbf{r}_f^T = \mathbf{A}_\tau^{-1}(-(\mathbf{B}_0 + \mathbf{B}_f)\,{\mathbf{r}'_f}^T - \mathbf{v}^T)$, where $\tau = O(m \cdot N\ell \cdot (N+1)^{d_{\mathcal{F}}}) \geq \tau_0$ (the enlargement of τ is needed for the security proof to work). Note that

$$[\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot [\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f \| \mathbf{v}]^T = \mathbf{0}^T.$$

Output $\mathsf{sk}_f = [\mathbf{r}_f \| \mathbf{r}'_f]$.

– $\mathsf{STHABE.ApplyF}_{\mathsf{pp}}(\mathsf{ct}, f)$. This is an auxiliary function that is used for homomorphic evaluation below. It uses the public parameters pp and takes as input a ciphertext $\mathsf{ct} = (x, \mathbf{C}_A, \mathbf{C}_0, \mathbf{c}_v, \mathbf{C}_x)$ and a policy $f \in \mathcal{F}$, such that $f(x) = 0$. It computes and outputs a "functioned" ciphertext ct_f as follows.

Compute the matrix $\mathbf{H} = \mathbf{H}_{f,x,\vec{\mathbf{B}}} \in \mathbb{Z}_q^{\ell N \times N}$ as $\mathbf{H} = \mathsf{EvRelation}(f, x, \vec{\mathbf{B}})$ (see Fact 1), define $\mathbf{C}_f = \mathbf{H}^T \mathbf{C}_x$ and finally set

$$\widehat{\mathbf{C}}_f = \begin{bmatrix} \mathbf{C}_A \\ \mathbf{C}_0 + \mathbf{C}_f \\ \mathbf{c}_v \end{bmatrix}.$$

The "functioned" ciphertext is $\mathsf{ct}_f = \widehat{\mathbf{C}}_f$.

– $\mathsf{STHABE.TEval}_{\mathsf{pp}}(f, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$. Given a policy $f \in \mathcal{F}$, k ciphertexts $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}$ and a function $g \in G$. $g \in \{0,1\}^k \to \{0,1\}$, for each $i \in [k]$ compute the matrix

$$\widehat{\mathbf{C}}_f^{(i)} = \mathsf{STHABE.ApplyF}_{\mathsf{pp}}(\mathsf{ct}^{(i)}, f).$$

Set $\mathsf{ct}^g = \mathbf{C}_f^g = \mathsf{Eval}(g, \widehat{\mathbf{C}}_f^{(1)}, \ldots, \widehat{\mathbf{C}}_f^{(k)})$ (see Definition 5 in Sect. 3.4).

– $\mathsf{STHABE.Dec}(\mathsf{sk}_f, \mathsf{ct}^g)$. Given $\mathsf{sk}_f = [\mathbf{r}_f \| \mathbf{r}'_f]$ and $\mathsf{ct}^g = \mathbf{C}_f^g$, compute the vector $\mathbf{c} = [\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot \mathbf{C}_f^g$. Let $\mathbf{u}^T = (0, \ldots, 0, \lfloor q/2 \rceil) \in \mathbb{Z}_q^{(m+N+1)}$. Compute $\tilde{\mu} = \mathbf{c}\mathbf{G}^{-1}(\mathbf{u})$. Output $\mu' = 0$ if $|\tilde{\mu}| \leq q/4$ and $\mu' = 1$ otherwise.

4.1 Choice of Parameters

The DLWE parameters n, q, B, χ are chosen according to constraints from the correctness and security analyses that follow. We require that $n \geq \lambda$, $q \leq 2^n$ and

recall that $\ell = \text{poly}(\lambda) \leq 2^\lambda$. We recall that $m \geq m_0$ where $m_0 = O(n \log q)$, $N = n\lceil \log q \rceil$ and $M = (m + N + 1)\lceil \log q \rceil$, and we require that

$$2^{n^\epsilon} \geq 8 \cdot (N+1)^{2d_\mathcal{F}} \cdot (M+1)^{d_\mathcal{G}} \cdot \ell^2 \cdot P(m, N, M, \lceil \log q \rceil)$$

for $P(m, N, M, \lceil \log q \rceil) = \text{poly}(m, N, M, \lceil \log q \rceil) = n^{O(1)}$ defined in the correctness analysis below. These constraints can be met by setting $n = \tilde{O}(\lambda + d_\mathcal{F} + d_\mathcal{G})^{1/\epsilon}$.

We then choose q, χ, B accordingly based on Corollary 1, and note that it guarantees that indeed $q \leq 2^n$. Furthermore, this choice guarantees that

$$q/B \geq 2^{n^\epsilon} \geq 8 \cdot (N+1)^{2d_\mathcal{F}} \cdot (M+1)^{d_\mathcal{G}} \cdot \ell^2 \cdot P(m, N, M, \lceil \log q \rceil).$$

4.2 Correctness

Lemma 1. *The scheme* STHABE *with parameters* $\ell, d_\mathcal{F}, d_\mathcal{G}$ *is correct with respect to policy class* $\mathcal{F}_{\ell, d_\mathcal{F}}$ *and homomorphism class* $\mathcal{G}_{d_\mathcal{G}}$.

Proof. Let $(\mathsf{msk}, \mathsf{pp}) = \mathsf{STHABE.Setup}(1^\lambda, 1^\ell, 1^{d_\mathcal{F}}, 1^{d_\mathcal{G}})$. Consider a function $f \in \mathcal{F}$ and a matching secret key $\mathsf{sk}_f = \mathsf{STHABE.Keygen}_{\mathsf{msk}}(f)$, a sequence of $k \geq 1$ messages and attributes $\{(\mu^{(i)} \in \{0,1\}, x^{(i)} \in \{0,1\}^\ell)\}_{i \in [k]}$ such that $\{f(x^{(i)}) = 0\}_{i \in [k]}$, and the sequence of their encryptions respectively $\{\mathsf{ct}^{(i)} = \mathsf{STHABE.Enc}_{\mathsf{pp}}(\mu^{(i)}, x^{(i)})\}_{i \in [k]}$. For each ciphertext it holds that

$$\mathbf{C}_x \approx [\vec{\mathbf{B}} - x\vec{\mathbf{G}}] \qquad (\text{err: } mB)$$

Consider a function $g \in \mathcal{G}$ such that $g \in \{0,1\}^k \to \{0,1\}$, and let $\mathsf{ct}^g = \mathsf{STHABE.TEval}(f, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$. Recall that for each ciphertext, during the execution of $\mathsf{STHABE.ApplyF}(\mathsf{ct}, f)$ we compute the matrix $\mathbf{C}_f^{(i)} = \mathbf{H}^{(i)}\mathbf{C}_x^{(i)}$.

By the properties stated at Fact 1, and since for all $i \in [k]$ $\left\|\mathbf{H}^{(i)}\right\|_\infty \leq (N+1)^{d_\mathcal{F}}$ and $f(x^{(i)}) = 0$, for each ciphertext it holds that

$$\mathbf{C}_f = \mathbf{H}^T\mathbf{C}_x \approx [\mathbf{B}_f - f(x)\mathbf{G}]^T \cdot \mathbf{S} = \mathbf{B}_f^T \cdot \mathbf{S} \qquad \left(\text{err: } (N+1)^{d_\mathcal{F}} \cdot \ell N \cdot mB\right)$$

and hence

$$\widehat{\mathbf{C}}_f \approx [\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f \| \mathbf{v}]^T \cdot \mathbf{S} + \mu\mathbf{G} \qquad \left(\text{err: } mB \cdot (1 + (N+1)^{d_\mathcal{F}} \cdot \ell N)\right) \qquad (2)$$

(Note that Eq. (2) also holds when $\mu \in \mathbb{Z}_q$ instead of $\mu \in \{0,1\}$).

It therefore follows that

$$[\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot \widehat{\mathbf{C}}_f \approx \mu \cdot [\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot \mathbf{G}$$
$$\left(\text{err: } \left\|[\mathbf{r}_f \| \mathbf{r}'_f \| 1]\right\|_\infty \cdot mB \cdot (1 + (N+1)^{d_\mathcal{F}} \cdot \ell N) \cdot (m + N + 1)\right)$$

Now consider a function $g \in \mathcal{G}$ such that $g \in \{0,1\}^k \to \{0,1\}$, and consider the execution of $\mathsf{STHABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_f, \mathsf{ct}^g)$ where $\mathsf{ct}^g = \mathsf{STHABE.TEval}(f, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$.

By Fact 2, denoting $\mu^g = g(\mu^{(1)}, \ldots, \mu^{(k)})$, we get

$$\mathbf{c} = [\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot \mathbf{C}_f^g \approx \mu^g \cdot [\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot \mathbf{G}$$
$$\left(\text{err: } \left\|[\mathbf{r}_f \| \mathbf{r}'_f \| 1]\right\|_\infty \cdot mB \cdot (1 + (N+1)^{d_\mathcal{F}} \cdot \ell N) \cdot (m + N + 1) \cdot (M+1)^{d_\mathcal{G}}\right)$$

and therefore

$$\tilde{\mu} = \mathbf{c}\mathbf{G}^{-1}(\mathbf{u}) \approx \mu^g \lfloor q/2 \rceil \tag{3}$$
$$\left(\text{err: } \left\|[\mathbf{r}_f \| \mathbf{r}'_f \| 1]\right\|_\infty \cdot mB \cdot (1 + (N+1)^{d_\mathcal{F}} \ell N) \cdot (m + N + 1) \cdot (M+1)^{d_\mathcal{G}} \cdot \lceil \log q \rceil\right)$$

We conclude that we get correct decryption as long as the error in Eq. (3) is bounded away from $q/4$. We recall that by the properties of discrete Gaussians and since $\mathbf{r}'_f \in \{0,1\}^N$, it holds that $\left\|[\mathbf{r}_f \| \mathbf{r}'_f]\right\|_\infty \leq \max\{\|\mathbf{r}_f\|_\infty, 1\} \leq \tau\sqrt{m}$ with all but $2^{-(m)} = \text{negl}(\lambda)$ probability, where $\tau = O(\sqrt{mn} \cdot N^2\ell \cdot (N+1)^{d_\mathcal{F}})$. Therefore, with all but negligible probability, the error is at most

$$\begin{aligned}
&\left\|[\mathbf{r}_f \| \mathbf{r}'_f \| 1]\right\|_\infty \cdot mB \cdot (1 + (N+1)^{d_\mathcal{F}} \ell N) \cdot (m + N + 1) \cdot (M+1)^{d_\mathcal{G}} \cdot \lceil \log q \rceil \\
&\quad \leq O(\sqrt{mn} \cdot N^2\ell \cdot (N+1)^{d_\mathcal{F}})\sqrt{m} \cdot mB \cdot (1 + (N+1)^{d_\mathcal{F}} \ell N) \\
&\qquad \cdot (m + N + 1) \cdot (M+1)^{d_\mathcal{G}} \cdot \lceil \log q \rceil \\
&\quad = B \cdot (N+1)^{2d_\mathcal{F}} \cdot (M+1)^{d_\mathcal{G}} \cdot \ell^2 \cdot P(m, N, M, \lceil \log q \rceil).
\end{aligned}$$

Since we set $B \leq q/\left(8 \cdot (N+1)^{2d_\mathcal{F}} \cdot (M+1)^{d_\mathcal{G}} \cdot \ell^2 \cdot P(m, N, M, \lceil \log q \rceil)\right)$, it holds that the error is less than $q/4$. Hence,

$$\Pr[\mathsf{STHABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_f, \hat{\mathsf{ct}}_f) \neq g(\mu^{(1)}, \ldots, \mu^{(k)})] = \text{negl}(\lambda).$$

4.3 Security

Lemma 2. *Under the* $\text{DLWE}_{n,q,\chi}$ *assumption, the scheme* STHABE *is selectively secure for the function classes* $\mathcal{F}, \mathcal{G}$*. Moreover, under this assumptions the scheme has* pseudorandom ciphertexts*: no polynomial time adversary can distinguish between the* $\mathbf{C}_A, \mathbf{C}_0, \mathbf{c}_v, \mathbf{C}_x$ *components of* ct^* *and a set of uniform matrices of the same dimension. Furthermore, this is true even if the encryption algorithm is applied to an arbitrary* $\mu \in \mathbb{Z}_q$*, and not necessarily* $\mu \in \{0,1\}$*.*

The security proof is a straightforward extension of the proof of [5]. In fact, our setup and key generation procedure are identical to the [5] scheme, the only difference is the setting of the LWE parameters and the sampling of $\mathbf{r}'_f$ from the binary distribution rather than Gaussian. The latter issue only requires a minor change in the proof, namely replacing the [2,30] Gaussian sampler for matrices of the form $[\mathbf{A} \| \mathbf{A}\mathbf{R} + \mathbf{G}_n]$ with the [28] sampler which allows to sample from a part Gaussian part binary distribution for matrices of this form.

As for our ciphertexts, they are of the form $\widetilde{\mathbf{A}}^T\mathbf{S} + \widetilde{\mathbf{E}} + \mathbf{Y}_\mu$, where $\widetilde{\mathbf{A}}$ is derived from the public parameters, $\widetilde{\mathbf{E}}$ is noise, and $\mathbf{Y}_\mu$ is a matrix that is determined

by the message μ. In [5], the ciphertext is of the form $\widetilde{\mathbf{A}}^T\mathbf{s}+\widetilde{\mathbf{e}}+\mathbf{y}'_\mu$. That is, we can almost think about our ciphertext as a matrix whose every column is a [5] ciphertext. The difference is that the encoding of the message $\mathbf{y}'_\mu$ is different from our $\mathbf{Y}_\mu$. However, [5] prove that their ciphertexts are *pseudorandom* and this means that they can mask $\mathbf{Y}_\mu$ regardless of its specific definition. The security of our scheme thus follows. The full proof follows.

Proof. Consider the selective security game as per Definition 3. Recall that in our scheme, an encryption of a message can be expressed as $\mathsf{ct} = (x, \mathbf{C})$, where

$$\mathbf{C} = \widetilde{\mathbf{A}}_x^T\mathbf{S} + \widetilde{\mathbf{E}} + \widetilde{\mathbf{Y}}_\mu = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \\ (\vec{\mathbf{B}} - x\vec{\mathbf{G}})^T \end{bmatrix} \mathbf{S} + \begin{bmatrix} \mathbf{E}_A \\ \mathbf{E}_0 \\ \mathbf{e}_v \\ \mathbf{E}_x \end{bmatrix} + \begin{bmatrix} \mu\mathbf{G} \\ \mathbf{0} \end{bmatrix}.$$

We note that all columns of $\mathbf{S}$ are identically and independently distributed and the same holds for all columns of $\widetilde{\mathbf{E}}$. We intend to prove security using a hybrid on the columns of $\mathbf{C}$. That is, we will consider a modified game which is identical to the selective security game, except for the challenge phase, where the adversary gets either $\mathbf{c} = \widetilde{\mathbf{A}}_{x^*}^T\mathbf{s} + \widetilde{\mathbf{e}}$ or a completely uniform vector, and needs to distinguish the two cases. Specifically, $\mathbf{s}$ is a uniform vector, and $\widetilde{\mathbf{e}} = [\mathbf{e}_A^T \| \mathbf{e}_0^T \| e_v \| \mathbf{e}_1^T \| \cdots \| \mathbf{e}_\ell^T]^T$, where the entries of $\mathbf{e}_A$ and e_v are sampled from χ and $\mathbf{e}_i = \mathbf{R}_i^T\mathbf{e}_A$ for $\mathbf{R}_0, \ldots, \mathbf{R}_\ell$ which are uniform in $\{0,1\}^{m\times N}$ (recall that we choose the matrices $\mathbf{R}_i$ independently for each column of the ciphertext). We will refer to this game as the *column game* and denote the advantage of an adversary $\mathcal{A}'$ in this game as $|\Pr[b'=1|\mathbf{c}] - \Pr[b'=1|\text{uniform}]|$.

We start by showing that under the DLWE assumption, no polynomial time adversary can have noticeable advantage against the column game. Afterwards we will show that this implies the security of the scheme.

Consider an adversary $\mathcal{A}'$ for the column game discussed above, and let $\mathrm{Adv}[\mathcal{A}']$ denote its advantage in the column game. The proof will proceed by a sequence of hybrids, denote by $\mathrm{Adv}_{\mathcal{H}}[\mathcal{A}']$ the advantage of $\mathcal{A}'$ in the experiment described in hybrid $\mathcal{H}$.

Hybrid $\mathcal{H}_0$. This is the column game. By definition $\mathrm{Adv}[\mathcal{A}'] = \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}']$.

Hybrid $\mathcal{H}_1$. We now change the way the matrices $\mathbf{B}_0$ and $\vec{\mathbf{B}}$ are generated. Recall that $\widetilde{\mathbf{e}} = [\mathbf{e}_A^T \| \mathbf{e}_0^T \| e_v \| \mathbf{e}_1^T \| \cdots \| \mathbf{e}_\ell^T]^T$, where there exist $\mathbf{R}_0, \ldots, \mathbf{R}_\ell$ which are uniform in $\{0,1\}^{m\times N}$ s.t. $\mathbf{e}_i = \mathbf{R}_i^T\mathbf{e}_A$. In this hybrid, we set $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G}_n$ instead of generating the $\mathbf{B}_i$ matrices uniformly.

Indistinguishability will follow from the extended leftover hash lemma as in [1, Lemma 13] (also used in [5]), since $m \geq (n+1)\lceil \log q \rceil + 2\lambda$.[5] We point out

[5] We note that they stated their lemma only for prime q, but in fact any q works for us since $\mathbf{R}_i$ have $\{0,1\}$ entries and since ± 1 are units over any ring $\mathbb{Z}_q$. Therefore matrix multiplication is a universal hash function for any distribution of binary vectors.

that the lemma can be used even though $\mathbf{A}$ is not uniform but only statistically close to uniform, since the argument here is information theoretic.

$$|\mathrm{Adv}_{\mathcal{H}_1}[\mathcal{A}'] - \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}']| = \mathrm{negl}(\lambda).$$

We notice that in this hybrid, we now have that $\vec{\mathbf{B}} = \mathbf{A}\vec{\mathbf{R}} + x\vec{\mathbf{G}}$, where $\vec{\mathbf{R}} = [\mathbf{R}_1 \| \cdots \| \mathbf{R}_\ell]$.

Hybrid $\mathcal{H}_2$. In this hybrid we switch from generating sk_f using $\mathbf{A}_{\tau_0}^{-1}$ to generating them using $\mathbf{R}_0$ and $\vec{\mathbf{R}}$. We recall that we are only required to generate keys for f s.t. $f(x^*) = 1$, otherwise the adversary loses in the selective security game.

We recall that by definition, $\mathsf{sk}_f = [\mathbf{r}_f \| \mathbf{r}'_f]$ where $\mathbf{r}'_f \xleftarrow{\$} \{0,1\}^N$ and $\mathbf{r}_f = \mathbf{A}_\tau^{-1}(-\mathbf{v} - (\mathbf{B}_0 + \mathbf{B}_f)\mathbf{r}'^T_f)$. Corollary 3 asserts that this is equivalent to sampling $[\mathbf{r}_f \| \mathbf{r}'_f] \xleftarrow{\$} [\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}(-\mathbf{v})$ for $P = D_{\mathbb{Z}^m,\tau} \times \{0,1\}^N$, since the marginal distribution of $\mathbf{r}'_f$ is uniform binary, and the conditional distribution of $\mathbf{r}_f$ given $\mathbf{r}'_f$ is therefore the discrete Gaussian over the appropriate coset of the integer lattice. Denoting $\mathbf{H} = \mathbf{H}_{f,x^*,\vec{\mathbf{B}}}$, it holds that

$$\mathbf{B}_f - f(x^*)\mathbf{G}_n = \left(\vec{\mathbf{B}} - x^*\vec{\mathbf{G}}\right)\mathbf{H}.$$

Since $f(x^*) = 1$, we get that

$$\mathbf{B}_f = \mathbf{A}\vec{\mathbf{R}}\mathbf{H} + \mathbf{G}_n.$$

It also holds that

$$\mathbf{A}\mathbf{R}_0 + \mathbf{A}\vec{\mathbf{R}}\mathbf{H} = \mathbf{A}(\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H})$$

Therefore, $[\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f] = [\mathbf{A} \| \mathbf{A}\mathbf{R}_0 + \mathbf{A}\vec{\mathbf{R}}\mathbf{H} + \mathbf{G}_n] = [\mathbf{A} \| \mathbf{A}(\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H}) + \mathbf{G}_n]$. By Corollary 3, given $\mathbf{R}_0, \vec{\mathbf{R}}$ and the computable matrix $\mathbf{H}$, we can sample from $[\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}$, with $P = D_{\mathbb{Z}^m,\tau} \times \{0,1\}^N$ for all values of $\tau \geq \tau'$ for $\tau' = O\left(\sqrt{mn}N \cdot \left\|(\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H})\right\|_\infty\right)$. This is true for all but $O(2^{-n})$ probability for random $\mathbf{A}$ and therefore, since $\mathsf{TrapGen}$ produces a distribution on $\mathbf{A}$ that is 2^{-n} uniform, it also holds for such matrices with all but $O(2^{-n})$ probability. Plugging in the bounds $\|\mathbf{H}\|_\infty \leq (N+1)^{d_\mathcal{F}}$, $\|\mathbf{R}_i\|_\infty = 1$, we get that $\left\|\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H}\right\|_\infty \leq N\ell \cdot (N+1)^{d_\mathcal{F}}$ and therefore

$$\tau' = O(\sqrt{mn} \cdot N^2\ell \cdot (N+1)^{d_\mathcal{F}}).$$

Recall that we need to sample with $\tau = O(\sqrt{mn} \cdot N^2\ell \cdot (N+1)^{d_\mathcal{F}})$ and therefore, by appropriately setting τ, we can sample from $[\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}$ up to $O(2^{-n})$ statistical distance.

It follows that after changing our method of sampling sk_f, the view of the adversary remains unchanged up to statistical distance of $\mathrm{poly}(\lambda) \cdot 2^{-n} = \mathrm{negl}(\lambda)$, since with all but $O(2^{-n})$ probability, our alternative sampler outputs a proper sample from a distribution that is within $O(2^{-n})$ statistical distance of $[\mathbf{A} \| \mathbf{B}_0 +$

$\mathbf{B}_f]_P^{-1}(-\mathbf{v})$. Since the number of key queries is at most $\text{poly}(\lambda)$, the result follows. We conclude that

$$|\text{Adv}_{\mathcal{H}_2}[\mathcal{A}'] - \text{Adv}_{\mathcal{H}_1}[\mathcal{A}']| = \text{negl}(\lambda).$$

We notice that in this hybrid, the challenger does not require $\mathbf{A}_{\tau_0}^{-1}$ at all.

Hybrid $\mathcal{H}_3$. In this hybrid, we change the distribution of $\mathbf{A}$ and sample it uniformly from $\mathbb{Z}_q^{n\times m}$ rather than via $\mathsf{TrapGen}$. Since $\mathsf{TrapGen}$ samples $\mathbf{A}$ which is statistically indistinguishable from uniform, we conclude that the distribution produced in the two hybrids are statistically indistinguishable as well.

$$|\text{Adv}_{\mathcal{H}_3}[\mathcal{A}'] - \text{Adv}_{\mathcal{H}_2}[\mathcal{A}']| = \text{negl}(\lambda).$$

Hybrid $\mathcal{H}_4$. We change the contents of the challenge ciphertext as follows. We generate $\mathbf{s}, \mathbf{e}_A, e_v$ as before, and set $\mathbf{d} = \mathbf{A}^T\mathbf{s} + \mathbf{e}_A$, $d_v = \mathbf{v}^T\mathbf{s} + e_v$. The components of the vector $\mathbf{c}$ can now be expressed in terms of $\mathbf{d}, d_v$ since $\mathbf{c}^T = [\mathbf{d}^T \| \mathbf{d}^T\mathbf{R}_0 \| d_v \| \mathbf{d}^T\mathbf{R}_1 \| \cdots \| \mathbf{d}^T\mathbf{R}_\ell]$. This hybrid is in fact identical to the previous one, only notation had been changed.

$$\text{Adv}_{\mathcal{H}_4}[\mathcal{A}'] = \text{Adv}_{\mathcal{H}_3}[\mathcal{A}'].$$

We note that in this hybrid, given $\mathbf{d}, d_v$, the challenger does not need to know the values of $\mathbf{s}, \mathbf{e}_A, e_v$ since they are not used directly.

Hybrid $\mathcal{H}_5$. We change the distribution of $\mathbf{d}, d_v$ to be uniform in $\mathbb{Z}_q^m, \mathbb{Z}_q$. Indistinguishability follows by definition from the $\text{DLWE}_{n,q,\chi}$ assumption. We have

$$|\text{Adv}_{\mathcal{H}_5}[\mathcal{A}'] - \text{Adv}_{\mathcal{H}_4}[\mathcal{A}']| = \text{negl}(\lambda).$$

Hybrid $\mathcal{H}_6$. Finally, we change the distribution of $\mathbf{c}$ to uniform. By the leftover hash lemma, for all i it holds that $(\mathbf{A}, \mathbf{d}^T, \mathbf{A}\mathbf{R}_i, \mathbf{d}^T\mathbf{R}_i)$ are statistically close to uniform. Therefore this hybrid is statistically indistinguishable from the previous. We have that

$$|\text{Adv}_{\mathcal{H}_6}[\mathcal{A}] - \text{Adv}_{\mathcal{H}_5}[\mathcal{A}]| = \text{negl}(\lambda).$$

Clearly, in this hybrid the adversary has no advantage in the column game since $\mathbf{c}$ itself is uniform, so there is no difference between the two cases. It follows therefore that

$$\text{Adv}_{\mathcal{H}_6}[\mathcal{A}'] = 0,$$

and therefore

$$\text{Adv}[\mathcal{A}'] = \text{negl}(\lambda).$$

Having established the hardness of the column game, a straightforward hybrid argument over the columns of the ciphertext shows that no polynomial time adversary can have non-negligible advantage in a game that is identical to the selective security game, except $\widetilde{\mathbf{A}}_{x^*}^T\mathbf{S} + \widetilde{\mathbf{E}}$ in the generation of ct^* is replaced with a uniform matrix. Pseudorandomness of the ciphertext, and thus selective security, follows.

5 A Multi Target Homomorphic ABE Scheme

Using the multi-key FHE technique presented in [16,31], we generalize the single-target HABE scheme of the previous section to support homomorphic evaluations targeted to a *set of policies* instead of just one. In this variant, homomorphic evaluation is performed with respect to a set of policy functions $F = \{f_1, \ldots, f_d\}$ that "covers" all of the participating attributes. That is, any participating ciphertext's attribute zeros at least one function in F. The resulting ciphertext can be decrypted only with the set of keys corresponding to the set F.

We start in Sect. 5.1 by presenting a generalization to the [16,31] scheme that will be useful for our construction. Section 5.2 contains a description of the scheme, and the choice of parameters is in Sect. 5.3. Correctness and security analyses appear in Sects. 5.4 and 5.5.

5.1 A Generalized Multi-key FHE

We start with a describing a generalized version of the [16,31] MK-FHE scheme. Consider matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{B}_1, \ldots, \mathbf{B}_d \in \mathbb{Z}_q^{n\times N}$ and a vector $\mathbf{v} \in \mathbb{Z}_q^n$. For all $j \in [d]$ let $\mathbf{r}_j, \mathbf{r}'_j$ be vectors of dimensions m, N respectively, such that $[\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot [\mathbf{A} \| \mathbf{B}_j \| \mathbf{v}]^T = \mathbf{0}$ and $\left\| [\mathbf{r}_j \| \mathbf{r}'_j \| 1] \right\|_\infty \leq B'$.

Let $\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)} \in \mathbb{Z}_q^{(m+N+1)\times M}$ be GSW-style encryptions of $\mu^{(1)}, \cdots, \mu^{(k)} \in \{0,1\}$. That is, for all $i \in [k]$ there exists and index $j \in [d]$ and a matrix $\mathbf{S}^{(i)} \in \mathbb{Z}_q^{n\times M}$ for which

$$\mathbf{C}^{(i)} \approx [\mathbf{A} \| \mathbf{B}_j \| \mathbf{v}]^T \mathbf{S}^{(i)} + \mu^{(i)} \mathbf{G} \qquad (\text{err: } B) \tag{4}$$

(recall that $M = (m + N + 1)\lceil \log q \rceil$).

For all $i \in [k]$ let $\mathcal{X}^{(i)} = \{\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{n,M}\}$ be a set of GSW-style encryptions of the entries of $\mathbf{S}^{(i)}$ under the same public key $[\mathbf{A} \| \mathbf{B}_j \| \mathbf{v}]^T$. So for all $\mathbf{X}_{a,b} \in \mathcal{X}^{(i)}$ we have

$$\mathbf{X}_{a,b} \approx [\mathbf{A} \| \mathbf{B}_j \| \mathbf{v}]^T \widetilde{\mathbf{S}}^{(i)}_{a,b} + \mathbf{S}^{(i)}[a,b]\mathbf{G} \qquad (\text{err: } B)$$

for some matrix $\widetilde{\mathbf{S}}^{(i)}_{a,b} \in \mathbb{Z}_q^{n\times M}$. Therefore,

$$[\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{X}_{a,b} \approx \mathbf{S}^{(i)}[a,b] \cdot [\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{G} \qquad (\text{err: } B' \cdot B \cdot (m + N + 1))$$

Let $\mathsf{LComb}(\mathcal{X}, \mathbf{u})$ be an algorithm that takes as input $\mathcal{X} = (\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{n,M})$ as defined above and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and outputs a matrix $\mathbf{X} \in \mathbb{Z}_q^{(m+N+1)\times M}$ computed as follows:

For each $a \in [n], b \in [M]$ define a matrix $\mathbf{Z}_{a,b} \in \mathbb{Z}_q^{(m+N+1)\times M}$ consisting of zeros, where the only non-zero entry is $\mathbf{Z}_{a,b}[m + N + 1, b] = \mathbf{u}[a]$. Compute and output

$$\mathbf{X} = \sum_{a,b}^{n,M} \mathbf{X}_{a,b} \mathbf{G}^{-1}(\mathbf{Z}_{a,b}).$$

Lemma 3. *Consider the properties states above and let* $\mathbf{X}^{(i)} = \mathsf{LComb}\left(\mathcal{X}^{(i)}, \mathbf{u}\right)$ *for some vector* $\mathbf{u} \in \mathbb{Z}_q^n$*. Then for all* $i \in [k]$*, it holds that*

$$[\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{X}^{(i)} \approx \mathbf{u}\mathbf{S}^{(i)} \qquad (\mathsf{err} : B' \cdot B \cdot (m+N+1) \cdot nM \cdot \lceil \log q \rceil)$$

Proof. For all $i \in [k]$ It holds that

$$\begin{aligned}
[\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{X}^{(i)} &= [\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \sum_{a,b}^{n,N} \mathbf{X}_{a,b} \mathbf{G}^{-1}(\mathbf{Z}_{a,b}) \\
&\approx \sum_{a,b}^{n,M} \mathbf{S}[a,b] \cdot [\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{G}\mathbf{G}^{-1}(\mathbf{Z}_{a,b}) \\
&= \sum_{a,b}^{n,M} \mathbf{S}[a,b] \cdot (0, \ldots, 0, \mathbf{u}[a], 0, \ldots, 0) \quad (\text{Where } \mathbf{u}[a] \text{ is in the } b\text{th position}). \\
&= \mathbf{u}\mathbf{S}^{(i)} \qquad (\mathsf{err}\text{: } B' \cdot B \cdot (m+N+1) \cdot nM \cdot \lceil \log q \rceil)
\end{aligned}$$

Denoting $\mathsf{params} = (\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_d, \mathbf{v})$, consider the following algorithm: $\mathsf{Expand}_{\mathsf{params}}(\mathbf{C}, \mathcal{X}, (\mathbf{r}'_1, \ldots, \mathbf{r}'_d), j)$ uses the parameters params and gets as input a ciphertext $\mathbf{C}$ together with its auxiliary data $\mathcal{X}$ (as defined above), a sequence of vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_d$ of dimension N and an index $j \in [d]$. It computes and outputs an "expanded" ciphertext $\widehat{\mathbf{C}}$ as follows:

For all $t \in [d] \backslash \{j\}$ compute $\mathbf{X}_t = \mathsf{LComb}\left(\mathcal{X}, \mathbf{r}'_t(\mathbf{B}_t - \mathbf{B}_j)^T\right)$. Construct and output the expanded matrix $\widehat{\mathbf{C}}$ as a $d \times d$ block matrix, where each block $\widehat{\mathbf{C}}_{a,b} \in \mathbb{Z}_q^{(m+N+1)\times M}$ for $a, b \in [d]$ is defined as:

$$\widehat{\mathbf{C}}_{a,b} = \begin{cases} \mathbf{C} & a = b \\ \mathbf{X}_b & a = j, b \neq j \\ \mathbf{0} & o.w. \end{cases}$$

Fact 3. *Consider the properties stated above. For all* $i \in [k]$ *let* $j \in [d]$ *such that Eq. (4) holds and let* $\widehat{\mathbf{C}}^{(i)} = \mathsf{Expand}_{\mathsf{params}}(\mathbf{C}^{(i)}, \mathcal{X}^{(i)}, (\mathbf{r}'_1, \ldots, \mathbf{r}'_d), j)$*. Let* $g \in \{0,1\}^k \to \{0,1\}$ *be a circuit consisting of* NAND *gates of depth at most* $d_{\mathcal{G}}$*, and let* $\widehat{\mathbf{C}}^g = \mathsf{Eval}(g, \widehat{\mathbf{C}}^{(1)}, \ldots, \widehat{\mathbf{C}}^{(k)})$*. Then denoting* $\mathbf{r} = [\mathbf{r}_1 \| \mathbf{r}'_1 \| 1 \| \cdots \| \mathbf{r}_d \| \mathbf{r}'_d \| 1]$ *and* $\mu^g = g(\mu^{(1)}, \ldots, \mu^{(k)})$*, it holds that*

$$\begin{aligned}
&\mathbf{r} \cdot \widehat{\mathbf{C}}^g \approx \mu^g \cdot \mathbf{r} \cdot \mathbf{G}_{d(m+N+1)} \\
&\qquad (\mathsf{err} : B' \cdot B \cdot (m+N+1)^2 \cdot (1 + nM \cdot \lceil \log q \rceil) \cdot kdM \cdot (dM+1)^{d_{\mathcal{G}}})
\end{aligned}$$

Proof. For all $i \in [k]$ it holds that

$$\begin{aligned}
\widehat{\mathbf{C}}^{(i)} \approx \mathbf{I}_d \otimes \left([\mathbf{A} \| \mathbf{B}_j \| \mathbf{v}]^T \mathbf{S}^{(i)}\right) + \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{X}_1 \cdots \mathbf{X}_{j-1} & \mathbf{0} & \mathbf{X}_{j+1} \cdots \mathbf{X}_d \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^{(i)} \\
+ \mu^{(i)} \mathbf{G}_{d(m+N+1)} \qquad (\mathsf{err}\text{: } B)
\end{aligned}$$

where for all $t \in [d] \backslash \{j\}$, by Lemma 3 we have

$$[\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{X}_t^{(i)} \approx \mathbf{r}'_t (\mathbf{B}_t - \mathbf{B}_j)^T \cdot \mathbf{S}^{(i)} \qquad (\text{err: } B' \cdot B \cdot (m+N+1) \cdot nM \cdot \lceil \log q \rceil)$$

and therefore

$$[\mathbf{r}_t \| \mathbf{r}'_t \| 1] \cdot \mathbf{C}^{(i)} + [\mathbf{r}_j \| \mathbf{r}'_j \| 1] \cdot \mathbf{X}_t^{(i)} \approx \mu^{(i)} \cdot [\mathbf{r}_t \| \mathbf{r}'_t \| 1] \cdot \mathbf{G}$$
$$(\text{err: } B' \cdot B \cdot (m+N+1) \cdot (1 + nM \cdot \lceil \log q \rceil))$$

from which it follows that

$$\begin{aligned} \mathbf{r} \cdot \widehat{\mathbf{C}}^{(i)} &\approx \left[\mu^{(i)} \cdot [\mathbf{r}_1 \| \mathbf{r}'_1 \| 1] \cdot \mathbf{G} \| \ldots \| \mu^{(i)} \cdot [\mathbf{r}_d \| \mathbf{r}'_d \| 1] \cdot \mathbf{G} \right] \\ &= \mu^{(i)} \cdot \mathbf{r} \cdot \mathbf{G}_{d(m+N+1)} \\ &\quad (\text{err: } B' \cdot B \cdot (m+N+1) \cdot (1 + nM \cdot \lceil \log q \rceil)) \end{aligned}$$

Now let $g \in \{0,1\}^k \to \{0,1\}$, where g is of depth $d_{\mathcal{G}}$, and let $\widehat{\mathbf{C}}^g = \mathsf{Eval}(g, \widehat{\mathbf{C}})$. By Fact 2 we get

$$\mathbf{r} \cdot \widehat{\mathbf{C}}^g \approx \mu^g \cdot \mathbf{r} \cdot \mathbf{G}$$
$$(\text{err: } B' \cdot B \cdot (m+N+1)^2 \cdot (1 + nM \cdot \lceil \log q \rceil) \cdot kdM \cdot (dM+1)^{d_{\mathcal{G}}})$$

which completes the proof.

5.2 Our Scheme

Our Random Oracle. We consider a uniform random oracle $\mathcal{O}$. Namely, for every input $x \in \{0,1\}^*$, the value $\mathcal{O}(x)$ is a random variable that is uniformly distributed over $\{0,1\}^N$. The dimension of the vector N will be specified in the description of the scheme.

The Scheme. As in the STHABE construction, the scheme is parameterized with a security vs. efficiency trade-off constant $\epsilon \in (0,1)$, and supports a policies class $\mathcal{F}_{\ell,d_{\mathcal{F}}} \subseteq \{0,1\}^\ell \to \{0,1\}$ and homomorphism class $\mathcal{G}_{d_{\mathcal{G}}} \subseteq \{0,1\}^* \to \{0,1\}$. The scheme works for any $\ell, d_{\mathcal{F}}, d_{\mathcal{G}} = \mathrm{poly}(\lambda)$. We consider a family of pseudorandom functions PRF with seed length λ.

- $\mathsf{THABE.Setup}(1^\lambda, 1^\ell, 1^{d_{\mathcal{F}}}, 1^{d_{\mathcal{G}}})$. Choose n, q, B, χ as described in Sect. 5.3 below, and generate $\mathbf{A}_{\tau_0}^{-1}$ and $\mathbf{A}, \mathbf{B}_0, \vec{\mathbf{B}}, \mathbf{v}$ as in STHABE.Setup. Generate a PRF seed $\sigma = \mathsf{PRF.Gen}(1^\lambda)$.
  Set $\mathsf{msk} = (\mathbf{A}_{\tau_0}^{-1}, \sigma)$ and $\mathsf{pp} = (\mathbf{A}, \mathbf{B}_0, \vec{\mathbf{B}}, \mathbf{v})$.
- $\mathsf{THABE.Enc}_{\mathsf{pp}}(\mu, x)$. Let $(\mathbf{C}_A, \mathbf{C}_0, \mathbf{c}_v, \mathbf{C}_x) \leftarrow \mathsf{STHABE.Enc}_{\mathsf{pp}}(\mu, x)$ and denote $\mathbf{S} \in \mathbb{Z}_q^{n \times M}$ the randomness matrix that was generated in the encryption process.
  We now add an ABE-encryption of each entry of the matrix $\mathbf{S}$, respective to the attribute x. For all $a \in [n], b \in [M]$, let

$$\mathbf{X}_{a,b} \leftarrow \mathsf{STHABE.Enc}_{\mathsf{pp}}(\mathbf{S}[a,b], x)$$

As pointed out above, $\mathsf{STHABE.Enc_{pp}}$ is well defined and has some provable features even for $\mu \notin \{0,1\}$, and indeed here we use it with $\mathbf{S}[a,b] \in \mathbb{Z}_q$.
The final ciphertext is $\mathsf{ct} = (\mathbf{C}_A, \mathbf{C}_0, \mathbf{c}_v, \mathbf{C}_x, \mathcal{X} = (\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{n,M}))$.

– $\mathsf{THABE.Keygen_{msk}}(f)$. Set $\mathbf{B}_f = \mathsf{Eval}(f, \vec{\mathbf{B}})$ and query the random oracle $\mathbf{r}'_f = \mathcal{O}(\mathbf{A}, f) \in \{0,1\}^N$.
Let $\mathbf{r}_f^T = \mathbf{A}_\tau^{-1}\left(-(\mathbf{B}_0 + \mathbf{B}_f){\mathbf{r}'_f}^T - \mathbf{v}^T\right)$, where $\tau = O(\sqrt{mn} \cdot N^2 \ell \cdot (N+1)^{d_{\mathcal{F}}}) \geq \tau_0$ (the enlargement of τ is needed for the security proof to work), such that the trapdoor function uses $\mathsf{PRF.Gen}(\sigma, f)$ as its randomness. Note that

$$[\mathbf{r}_f \| \mathbf{r}'_f \| 1] \cdot [\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f \| \mathbf{v}]^T = \mathbf{0}^T.$$

Output $\mathsf{sk}_f = \mathbf{r}_f$.

– $\mathsf{THABE.Eval}(F, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$. Denoting $F = \{f_1, \ldots, f_d\}$, for every $i \in [k]$ let $j \in [d]$ be an index for which $f_j(x^{(i)}) = 0$. Compute

$$\widehat{\mathbf{C}}_f^{(i)} = \mathsf{STHABE.ApplyF_{pp}}(\mathsf{ct}^{(i)}, f_j),$$
$$\mathcal{X}_f^{(i)} = \{\mathsf{STHABE.ApplyF_{pp}}(\mathbf{X}, f_j) : \ \mathbf{X} \in \mathcal{X}^{(i)}\}$$

and for all $t \in [d]$ let $\mathbf{B}_{f_t} = \mathsf{Eval}(f_t, \vec{\mathbf{B}})$ and $\mathbf{r}'_t = \mathcal{O}(\mathbf{A}, f_t)$. Now compute

$$\mathbf{C}_F^{(i)} = \mathsf{Expand_{params}}(\widehat{\mathbf{C}}_f^{(i)}, \mathcal{X}_f^{(i)}, (\mathbf{r}'_1, \ldots, \mathbf{r}'_d), j)$$

where

$$\mathsf{params} = (\mathbf{A}, (\mathbf{B}_0 + \mathbf{B}_{f_1}), \ldots, (\mathbf{B}_0 + \mathbf{B}_{f_d}), \mathbf{v}).$$

Finally, set $\mathsf{ct}^g = \mathbf{C}_F^g = \mathsf{Eval}(g, \mathbf{C}_F)$.

– $\mathsf{THABE.Dec}(\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_d}, \mathsf{ct}^g)$. For all $j \in [d]$ sample $\mathbf{r}'_{f_j} = \mathcal{O}(\mathbf{A}, f_j)$. Construct the concatenated key $\mathbf{r}_F = [\mathbf{r}_{f_1} \| \mathbf{r}'_{f_1} \| 1 \| \cdots \| \mathbf{r}_{f_d} \| \mathbf{r}'_{f_d} \| 1]$ and compute the vector $\mathbf{c} = \mathbf{r}_F \cdot \mathbf{C}_f^g$.
Let $\mathbf{u}^T = (0, \ldots, 0, \lfloor q/2 \rceil) \in \mathbb{Z}_q^{d(m+N+1)}$. Compute $\tilde{\mu} = \mathbf{c}\mathbf{G}^{-1}(\mathbf{u})$. Output $\mu' = 0$ if $|\tilde{\mu}| \leq q/4$ and $\mu' = 1$ otherwise.

5.3 Choice of Parameters

The DLWE parameters n, q, B, χ are chosen according to constraints from the correctness and security analyses that follow. We require that $n \geq \lambda$, $q \leq 2^n$ and recall that $\ell, d = \mathrm{poly}(\lambda) \leq 2^\lambda$. We recall that $m = O(n \log q)$, $N = n\lceil \log q \rceil$ and $M = (m + N + 1)\lceil \log q \rceil$, and we require that

$$2^{n^\epsilon} \geq 8 \cdot (N+1)^{2d_{\mathcal{F}}} \cdot (dM+1)^{d_{\mathcal{G}}} \cdot d^{1.5} \cdot \ell^2 \cdot P(n, m, N, M, \lceil \log q \rceil)$$

for $P(n, m, N, M, \lceil \log q \rceil) = \mathrm{poly}(n, m, N, M, \lceil \log q \rceil) = n^{O(1)}$ defined in the correctness analysis below. These constraints can be met by setting $n = \tilde{O}(d_{\mathcal{F}} + \lambda d_{\mathcal{G}})^{1/\epsilon}$. We then choose q, χ, B accordingly based on Corollary 1. This choice guarantees that

$$q/B \geq 2^{n^\epsilon} \geq 8 \cdot (N+1)^{2d_{\mathcal{F}}} \cdot (dM+1)^{d_{\mathcal{G}}} \cdot d^{1.5} \cdot \ell^2 \cdot P(n, m, N, M, \lceil \log q \rceil).$$

5.4 Correctness

Lemma 4. *The scheme* THABE *with parameters* $\ell, d_{\mathcal{F}}, d_{\mathcal{G}}$ *is correct with respect to policy class* $\mathcal{F}_{\ell,d_{\mathcal{F}}}$ *and homomorphism class* $\mathcal{G}_{d_{\mathcal{G}}}$.

Proof. Let $(\mathsf{msk}, \mathsf{pp}) = \mathsf{THABE.Setup}(1^\lambda, 1^\ell, 1^{d_{\mathcal{F}}}, 1^{d_{\mathcal{G}}})$. Consider a set of $d \geq 1$ functions $F = \{f_1, \ldots, f_d \in\} \subseteq \mathcal{F}$ along with matching secret keys $\{\mathsf{sk}_f = \mathsf{THABE.Keygen}_{\mathsf{msk}}(f)\}_{f\in F}$. Consider a sequence of $k \geq 1$ messages and attributes $\{(\mu^{(i)} \in \{0,1\}, x^{(i)} \in \{0,1\}^\ell)\}_{i\in[k]}$, such that

$$\forall i \in [k]\ \exists j \in [d]: \quad f_j(x^{(i)}) = 0,$$

and the sequence of their encryptions $\{\mathsf{ct}^{(i)} = \mathsf{THABE.Enc}_{\mathsf{pp}}(\mu^{(i)}, x^{(i)})\}_{i\in[k]}$. Let $g \in \mathcal{G}$ and consider the execution of $\mathsf{THABE.Eval}(F, \mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$. By Eq. (2), for all $i \in [k]$ the following holds:

- $\widehat{\mathbf{C}}_f^{(i)} \approx [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{f_j}\|\mathbf{v}]^T\mathbf{S}^{(i)} + \mu^{(i)}\mathbf{G}$ $\quad(\mathsf{err}: mB\cdot(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N))$.
- $\forall\ \mathbf{X}_{a,b} \in \mathcal{X}_f^{(i)}$,
  $\mathbf{X}_{a,b} \approx [\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{f_j}\|\mathbf{v}]^T\widetilde{\mathbf{S}}_{a,b}^{(i)} + \mathbf{S}^{(i)}[a,b]\mathbf{G}$ $\quad(\mathsf{err}: mB\cdot(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N))$
  for some $\widetilde{\mathbf{S}}_{a,b}^{(i)}$.
- $[\mathbf{r}_{f_j}\|\mathbf{r}'_{f_j}\|1]\cdot[\mathbf{A}\|\mathbf{B}_0 + \mathbf{B}_{f_j}\|\mathbf{v}]^T = \mathbf{0}$

Therefore, considering $\mathsf{THABE.Dec}(\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_d}, \mathsf{ct}^g)$, by Fact 3 it holds that

$$\mathbf{c} = \mathbf{r}_F\cdot\mathbf{C}_F^g \approx \mu^g\cdot\mathbf{r}_F\cdot\mathbf{G}$$
$$(\mathsf{err}:\ \|\mathbf{r}_F\|_\infty\cdot mB(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N)\cdot(m+N+1)^2\cdot(1+nM\cdot\lceil\log q\rceil)\cdot (dM+1)^{d_{\mathcal{G}}})$$

and therefore

$$\tilde{\mu} = \mathbf{c}\mathbf{G}^{-1}(\mathbf{u}) \approx \mu^g\lfloor q/2\rceil \tag{5}$$
$$(\mathsf{err}:\ \|\mathbf{r}_F\|_\infty\cdot mB(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N)\cdot(m+N+1)^2\cdot(1+nM\cdot\lceil\log q\rceil)\cdot (dM+1)^{d_{\mathcal{G}}}\lceil\log q\rceil)$$

We conclude that we get correct decryption as long as the error in Eq. (5) is bounded away from $q/4$. We recall that by the properties of discrete Gaussians, it holds that $\|\mathbf{r}_F\|_\infty \leq \tau\sqrt{dm}$ with all but $2^{-dm} = \mathrm{negl}(\lambda)$ probability, where $\tau = O(\sqrt{mn}\cdot N^2\ell\cdot(N+1)^{d_{\mathcal{F}}})$. Therefore, with all but negligible probability, the error is at most

$$\begin{aligned}
&\|\mathbf{r}_F\|_\infty\cdot mB(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N)\cdot(m+N+1)^2\cdot(1+nM\cdot\lceil\log q\rceil).\\
&\quad(dM+1)^{d_{\mathcal{G}}}\cdot\lceil\log q\rceil\\
&\leq O(\sqrt{mn}\cdot N^2\ell\cdot(N+1)^{d_{\mathcal{F}}})\sqrt{dm}\cdot mB(1+(N+1)^{d_{\mathcal{F}}}\cdot\ell N)\cdot(m+N+1)^2\cdot\\
&\quad(1+nM\cdot\lceil\log q\rceil)\cdot(dM+1)^{d_{\mathcal{G}}}\cdot\lceil\log q\rceil\\
&= B\cdot(N+1)^{2d_{\mathcal{F}}}\cdot(dM+1)^{d_{\mathcal{G}}}\cdot d^{1.5}\cdot\ell^2\cdot P(n,m,N,M,\lceil\log q\rceil).
\end{aligned}$$

Since we set (see Sect. 5.3)

$$B \leq q/\left(8 \cdot (N+1)^{2d_{\mathcal{F}}} \cdot (dM+1)^{d_{\mathcal{G}}} \cdot d^{1.5} \cdot \ell^2 \cdot P(n,m,N,M,\lceil \log q \rceil)\right),$$

it holds that the error is less than $q/4$. Hence,

$$\Pr[\mathsf{THABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_F, \mathsf{ct}^g) \neq g(\mu^{(1)}, \ldots, \mu^{(k)})] = \mathrm{negl}(\lambda).$$

5.5 Security

Lemma 5. *In the random oracle model, under the* $\mathrm{DLWE}_{n,q,\chi}$ *assumption the scheme* THABE *is selectively secure for the function classes* $\mathcal{F}, \mathcal{G}$.

Proof. Consider the selective security game as per Definition 1 and let $\mathcal{A}$ be an adversary with advantage $\mathrm{Adv}[\mathcal{A}]$ in the selective security game. We start with a claim on random oracle queries that will be useful down the line. We classify oracle queries as follows. A query is *blind* if it is made before x^* is sent to the challenger. A query is *valid* if it is of the form $(\mathbf{D}, f)$ with $\mathbf{D} = \mathbf{A}$ and $f(x^*) = 1$ (for the matrix $\mathbf{A}$ in the public parameters). Let η be the probability that a blind and valid oracle query is made throughout the experiment. Clearly, since blind queries are made by the adversary before any information on $\mathbf{A}$ is given to him, the probability of any blind query has $\mathbf{D} = \mathbf{A}$ is at most $q^{-nm} = \mathrm{negl}(\lambda)$. Since the total number of queries is $\mathrm{poly}(\lambda)$ it holds that $\eta = \mathrm{negl}(\lambda)$.

The proof proceeds by a sequence of hybrids. Recall that in the random oracle model, the challenger needs to also be able to answer oracle queries at all steps of the security game.

Hybrid $\mathcal{H}_0$. In this hybrid, the challenger executes the selective security game as prescribed. Oracle queries are answered "on the fly": if the query is made for the first time, a fresh $\mathbf{r}$ is sampled uniformly from $\{0,1\}^N$, and if the query had been made before then a consistent response is returned. By definition $\mathrm{Adv}[\mathcal{A}] = \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}]$.

Hybrid $\mathcal{H}_1$. In this hybrid, the challenger, upon receiving x^*, checks whether any of the previous oracle calls had been blind and valid. If any such query had been made, the challenger aborts. Since this happens with negligible probability as analyzed above, the view of the adversary is statistically indistinguishable from the previous hybrid.

$$|\mathrm{Adv}_{\mathcal{H}_1}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_0}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Hybrid $\mathcal{H}_2$. In this hybrid, we no longer use the PRF to generate randomness for the Gaussian sampler in Keygen queries. Instead, the challenger will keep track of all Keygen queries made so far. Given a Keygen query on a function f that was made before, it will answer consistently. When a new query is made, a new

random string is generated and used for the Gaussian sampling. The pseudorandomness property of the PRF guarantees that this hybrid is indistinguishable from the previous one.

$$|\mathrm{Adv}_{\mathcal{H}_2}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_1}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

From this point and on, we assume that a Keygen query is not made with the same f more than once.

Hybrid $\mathcal{H}_3$. We now change the way non-blind and valid oracle queries, as well as Keygen queries, are answered. First, we assume w.l.o.g that any non-blind and valid oracle query is preceded by a Keygen query to the same function f (this is allowed since $f(x^*) = 1$ by definition of a valid query). The Keygen query itself is answered by using $\mathbf{A}_{\tau_0}^{-1}$ to sample $[\mathbf{r}_f \| \mathbf{r}'_f] = [\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}(-\mathbf{v})$ where $P = D_{\mathbb{Z}^m,\tau} \times \{0,1\}^N$. It then stores $\mathbf{r}'_f$ as the answer to the oracle query $(\mathbf{A}, f)$ (which at this point had necessarily not yet been made), and returns $\mathbf{r}_f$ as the response to the $\mathsf{Keygen}(f)$ query.

Since Corollary 3 implies that the marginal distribution of the $\mathbf{r}'$ component of $[\mathbf{A} \| \mathbf{B}_0 + \mathbf{B}_f]_\tau^{-1}(-\mathbf{v})$ is statistically indistinguishable from uniform over $\{0,1\}^N$, it follows that the view of the adversary in this experiment is statistically close to the previous hybrid.

$$|\mathrm{Adv}_{\mathcal{H}_3}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_2}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Hybrid $\mathcal{H}_4$. At this point, we notice that the challenger in $\mathcal{H}_3$ can be simulated via black box access to the challenger of our single key scheme described in Sect. 4. This is because valid and non blind oracle queries are translated into key generation queries, and all other queries are answered randomly. Since in the proof of Lemma 2 we show that the encryption is secure even for non binary messages, we can replace the encryptions of $\mathbf{S}$ in the challenge ciphertext with encryptions of all 0, and asserts that this is indistinguishable to the adversary.

$$|\mathrm{Adv}_{\mathcal{H}_4}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_3}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Hybrid $\mathcal{H}_5$. Now that $\mathbf{S}$ is only used for generating the encryption of the message bit μ, we can again use Lemma 2 to replace this part of the challenge ciphertext with an encryption of 0.

$$|\mathrm{Adv}_{\mathcal{H}_5}[\mathcal{A}] - \mathrm{Adv}_{\mathcal{H}_4}[\mathcal{A}]| = \mathrm{negl}(\lambda).$$

Clearly in this hybrid the adversary has no advantage since its view is independent of μ_b. Therefore $\mathrm{Adv}_{\mathcal{H}_5}[\mathcal{A}] = 1/2$ and it follows that

$$|\mathrm{Adv}[\mathcal{A}] - 1/2| = \mathrm{negl}(\lambda),$$

which completes the proof of security.

Acknowledgments. We thank Vadim Lyubashevsky for numerous insightful discussions.

A A Generic (Non-compact) Homomorphic ABE Construction

We show how to construct a non-targeted homomorphic ABE (HABE) given any ABE scheme and Multi-Key FHE scheme as building blocks. The main disadvantage of this construction is that the ciphertext's size grows at least linearly with the number of participants in the homomorphic evaluation. Interestingly, our method is very similar to the one presented in [17], despite the difference in the scheme's goal. Their construction relies on a leveled homomorphic ABE and uses it to create a non-leveled HABE scheme.

Below are definitions of ABE, MFHE and HABE, followed by our HABE construction and a brief proof of its correctness and security.

Definition 7 (ABE). *An* Attribute Based Encryption *(ABE) scheme is a tuple of* PPT *algorithms* $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Dec})$ *with the following syntax:*

- $\mathsf{ABE.Setup}(1^\lambda)$ *takes as input the security parameter and outputs a master secret key* msk *and a set of public parameters* pp.
- $\mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, x)$ *uses the public parameters* pp *and takes as input a message* $\mu \in \{0,1\}$ *and an attribute* $x \in \{0,1\}^\ell$. *It outputs a ciphertext* ct.
- $\mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$ *uses the master secret key* msk *and takes as input a function* $f \in \mathcal{F}$. *It outputs a secret key* sk_f.
- $\mathsf{ABE.Dec}(\mathsf{sk}_f, \mathsf{ct})$ *takes as input a secret key* sk_f *for a policy* f, *and a ciphertext* ct. *It outputs a message* $\mu \in \{0,1\}$.

Definition 8 (MK-FHE). *A* Multi-Key Fully Homomorphic Encryption *(MK-FHE) scheme is a tuple of* PPT *algorithms* $\mathsf{MFHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Eval}, \mathsf{Dec})$ *with the following syntax:*

- $\mathsf{MFHE.Setup}(1^\lambda)$ *takes as input the security parameter and generates public parameters* pp.
- $\mathsf{MFHE.Keygen}_{\mathsf{pp}}(1^\lambda)$ *uses the public parameters* pp *and outputs a pair of public key and secret key* $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{MFHE.Enc}_{\mathsf{pp}}(\mathsf{pk}, \mu)$ *uses the public parameters* pp *and takes as input a message* $\mu \in \{0,1\}$ *and a public key* pk. *It outputs a ciphertext* ct.
- $\mathsf{MFHE.Eval}_{\mathsf{pp}}((\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}), (\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(k)}), g)$ *uses the public parameters* pp *and takes as input* k *ciphertexts along with their respective public keys* $(\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(k)})$ *and a function* g. *It outputs a ciphertext* ct_g.
- $\mathsf{MFHE.Dec}_{\mathsf{pp}}(\mathsf{sk}^{(1)}, \ldots, \mathsf{sk}^{(k)}, \mathsf{ct}_g)$ *uses the public parameters and takes as input a sequence of* k *secret keys* $\mathsf{sk}^{(1)}, \ldots, \mathsf{sk}^{(k)}$ *and a ciphertext* ct_g. *It outputs a message* $\mu \in \{0,1\}$.

Definition 9 (HABE). *An* Homomorphic ABE *(HABE) scheme is a tuple of* PPT *algorithms* $\mathsf{HABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Eval}, \mathsf{Dec})$ *with the following syntax:*

- $\mathsf{HABE.Setup}(1^\lambda)$ *takes as input the security parameter and outputs a master secret key* msk *and a set of public parameters* pp.

- $\mathsf{HABE.Enc}_{\mathsf{pp}}(\mu, x)$ *uses the public parameters* pp *and takes as input a message* $\mu \in \{0,1\}$ *and an attribute* $x \in \{0,1\}^{\ell}$*. It outputs a ciphertext* ct*.*
- $\mathsf{HABE.Keygen}_{\mathsf{msk}}(f)$ *uses the master secret key* msk *and takes as input a function* $f \in \mathcal{F}$*. It outputs a secret key* sk_f*.*
- $\mathsf{HABE.Eval}(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$ *takes as input* k *ciphertexts* $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}$ *and a function* $g \in \mathcal{G}$*. It outputs a ciphertext* ct^{g}*.*
- $\mathsf{HABE.Dec}(\mathsf{sk}_F, \mathsf{ct}^{g})$ *takes as input a set of secret keys* sk_F *for a set of policies* F*, with* $\mathsf{sk}_F = \{\mathsf{sk}_f : f \in F\}$*, and a ciphertext* ct^{g}*. It outputs a message* $\mu \in \{0,1\}$*.*

Correctness. The correctness guarantee is that given a set of keys for a policy set F and a ciphertext that was evaluated from ciphertexts respective to attributes covered by F, the ciphertext decrypts correctly to the intended value.

Security. Security is defined using the same experiment as standard ABE (see Definition 3).

Construction of HABE. Consider an ABE black box and a MFHE black box. The construction works as follows:

- $\mathsf{HABE.Setup}(1^{\lambda})$
  Let $(\mathsf{pp}_{\mathsf{ABE}}, \mathsf{msk}_{\mathsf{ABE}}) \leftarrow \mathsf{ABE.Setup}(1^{\lambda})$ and $\mathsf{pp}_{\mathsf{MFHE}} \leftarrow \mathsf{MFHE.Setup}(1^{\lambda})$. Output $\mathsf{pp} = (\mathsf{pp}_{\mathsf{ABE}}, \mathsf{pp}_{\mathsf{MFHE}}), \mathsf{msk} = \mathsf{msk}_{\mathsf{ABE}}$.
- $\mathsf{HABE.Enc}_{\mathsf{pp}}(\mu, x)$. Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{MFHE.Keygen}_{\mathsf{pp}}$, where $\mathsf{sk} \in \{0,1\}^{t}$. Compute $\mathsf{ct}_{\mu} \leftarrow \mathsf{MFHE.Enc}_{\mathsf{pp}}(\mathsf{pk}, \mu)$ and $\mathsf{ct}_{\mathsf{sk}} = \{\mathsf{ct}_{\mathsf{sk}_i} = \mathsf{ABE.Enc}_{\mathsf{pp}}(\mathsf{sk}_i, x)\}_{i \in [t]}$. Output $\mathsf{ct} = (\mathsf{ct}_{\mu}, \mathsf{ct}_{\mathsf{sk}}, \mathsf{pk}, x)$.
- $\mathsf{HABE.Keygen}_{\mathsf{msk}}(f)$. Output $\mathsf{ABEk}_f \leftarrow \mathsf{ABE.Keygen}_{\mathsf{msk}}(f)$.
- $\mathsf{HABE.Eval}(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, g)$
  Let $\mathsf{ct}_g \leftarrow \mathsf{MFHE.Eval}_{\mathsf{pp}}((\mathsf{ct}_{\mu}^{(1)}, \ldots, \mathsf{ct}_{\mu}^{(k)}), (\mathsf{pk}^{(1)}, \ldots, \mathsf{pk}^{(k)}), g)$. Output $\mathsf{ct}^{g} = (\mathsf{ct}_g, \mathsf{ct}_{\mathsf{sk}}^{(1)}, \ldots, \mathsf{ct}_{\mathsf{sk}}^{(k)})$.
- $\mathsf{HABE.Dec}(\mathsf{ABEk}_F, \mathsf{ct}^{g})$.
  For all $i \in [k], j \in [t]$ compute $\mathsf{sk}_j^{(i)} = \mathsf{ABE.Dec}_{\mathsf{pp}}(\mathsf{ct}_{\mathsf{sk}_j}^{(i)}, \mathsf{ABEk}_f)$, where $f \in F$ such that $f(x^{(i)}) = 0$. Compute and Output $\mathsf{MFHE.Dec}_{\mathsf{pp}}(\mathsf{sk}^{(1)}, \ldots, \mathsf{sk}^{(k)}, \mathsf{ct}_g)$.

Correctness Proof Sketch. Consider the execution of $\mathsf{HABE.Dec}(\mathsf{ABEk}_F, \mathsf{ct}^{g})$. By the correctness of the ABE scheme we get correct decryptions of $\{\mathsf{sk}^{(i)}\}_{i \in [k]}$, and by the correctness of the MFHE scheme we get a correct decryption of $g(\mu^{(1)}, \ldots, \mu^{(k)})$.

Security Proof Sketch. Consider the ABE selective security game, and assume that in $\mathsf{HABE.Enc}_{\mathsf{pp}}$ the challenger generates ABE encryptions of 0s instead of ABE encryptions of the bits of sk. By the security of the ABE scheme this change is indistinguishable to the adversarys, therefore in this case the ciphertext gives no information other than the MFHE encryption of the message μ. Hence by the security of the MFHE scheme the security of our construction follows.

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. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
3. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May 1996, pp. 99–108. ACM (1996)
4. Alperin-Sheriff, J., Peikert, C.: Faster bootstrapping with polynomial error. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 297–314. Springer, Heidelberg (2014)
5. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Advances in Cryptology - EUROCRYPT –33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, 11–15 May 2014, Proceedings, pp. 533–556 (2014)
6. Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.): Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, 1–4 June 2013. ACM (2013)
7. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: ITCS (2012)
9. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., et al. (eds.) [6], pp. 575–584
10. Brakerski, Z., Perlman, R.: Lattice-based fully dynamic multi-key FHE withshort ciphertexts. IACR Cryptology ePrint Archive, 2016:339 (2016, to appear)
11. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS (2011)
12. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Naor, M. (ed.) Innovations in Theoretical Computer Science, ITCS 2014, Princeton, NJ, USA, 12–14 January 2014, pp. 1–12. ACM (2014)
13. Brakerski, Z., Vaikuntanathan, V.: Circuit-abe from LWE: unbounded attributesand semi-adaptive security. IACR Cryptology ePrint Archive, 2016:118 (2016, to appear)
14. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. J. Cryptology **25**(4), 601–639 (2012)
15. Clear, M., McGoldrick, C.: Bootstrappable identity-based fully homomorphic encryption. In: Gritzalis, D., Kiayias, A., Askoxylakis, I. (eds.) CANS 2014. LNCS, vol. 8813, pp. 1–19. Springer, Heidelberg (2014)
16. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M. (eds.) [19], pp. 630–656

17. Clear, M., McGoldrick, C.: Attribute-based fully homomorphic encryption with a bounded number of inputs. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 307–324. Springer, Heidelberg (2016). doi:10.1007/978-3-319-31517-1_16
18. Fischlin, M., Coron, J.-S. (eds.): EUROCRYPT 2016. LNCS, vol. 9666. Springer, Heidelberg (2016)
19. Gennaro, R., Robshaw, M. (eds.): CRYPTO 2015. LNCS, vol. 9216. Springer, Heidelberg (2015)
20. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 17–20 May 2008, pp. 197–206. ACM (2008)
22. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013)
23. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., et al. (eds.) [6], pp. 545–554
24. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, 14–17 June 2015, pp. 469–477. ACM (2015)
25. Goyal, R., Koppula, V., Waters, B.: Semi-adaptive security and bundling functionalities made generic and easy. Cryptology ePrint Archive, Report 2016/317 (2016). http://eprint.iacr.org/2016/317
26. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, 30 October–3 November 2006, pp. 89–98. ACM (2006)
27. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Karloff, H.J., Pitassi, T. (eds.) Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, 19–22 May 2012, pp. 1219–1234. ACM (2012)
28. Lyubashevsky, V., Wichs, D.: Simple lattice trapdoor sampling from a broad class of distributions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 716–730. Springer, Heidelberg (2015)
29. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
30. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
31. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J. (eds.) [18], pp. 735–763
32. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, 31 May – 2 June 2009, pp. 333–342 (2009)

33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 84–93 (2005)
34. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secure Comput. (1978)
35. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
36. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**, 201–224 (1987)