Lattice-Inspired Broadcast Encryption and Succinct Ciphertext-Policy ABE

Zvika Brakerski* Vinod Vaikuntanathan†

Abstract

We propose a candidate Ciphertext-Policy Attribute Based Encryption (CP-ABE) scheme for circuits, with ciphertext size that depends only on the depth of the policy circuit (and not its size). This, in particular, gives us a Broadcast Encryption (BE) scheme where all parameters (in particular, the size of the keys and ciphertexts) have a poly-logarithmic dependence on the number of users, a task that was only known to be achievable assuming ideal multilinear maps or indistinguishability obfuscation. Our construction relies on techniques from lattice-based (and in particular LWE-based) cryptography, but we are unable to provide a security proof. We analyze some attempts at cryptanalysis.

1 Introduction

Broadcast Encryption (BE) [FN93] is an important multi-user generalization of public-key encryption where a *broadcaster* can send the same message m to an arbitrary subset $S \subseteq \mathcal{U}$, where $\mathcal{U}$ is the universe of all the N possible users. A trivial, communication-inefficient, way of achieving this would involve the broadcaster encrypting m separately with the public keys of all users in S, resulting in a ciphertext of size $O(|S|)$ (ignoring dependence on the security parameter). Broadcast encryption seeks to achieve the same end goal with a much better ciphertext size, ideally, $O(\lambda)$ independent of the size of the set S or the universe $\mathcal{U}$. (In the context of BE, it is assumed that the decryptor knows the set S, either out-of-band, or as an addendum to the ciphertext. In any case, the description of S is not counted towards the ciphertext size or the communication complexity.) One could ask for more: that is, that all parameters of the system, including the (common) public key, the users' private keys as well as the ciphertext grow polylogarithmically in $|\mathcal{U}|$.

The first solution to the broadcast encryption problem was proposed by Boneh, Gentry and Waters [BGW05] using bilinear maps on elliptic curves.[1] Their construction had ciphertexts of size $O(1)$ and user secret keys of size $O(1)$, however the public key had size $O(N)$. Subsequently, there were constructions from indistinguishability obfuscation (iO) [BZ17] and from multilinear

*Weizmann Institute of Science. Supported by the Binational Science Foundation (Grant No. 2016726), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).

†MIT. Research supported in part by NSF Grants CNS-1350619 and NSF/BSF grant CNS-1718161, the US/Israel Binational Science Foundation and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

[1] Prior to [BGW05], solutions were either for the case of small sets S, having $O(|S|)$ size ciphertexts, or a line of work (starting from [NNL01]) that handled very large sets, that is sets of size $N-r$ with $O(r)$ size ciphertexts. Both solutions degrade to $O(N)$ ciphertext size for "typical" size broadcast sets.

maps [BWZ14], the latter of which achieved $\mathrm{polylog}(N)$ size for ciphertexts, secret keys and the public key. (We recently learned of a related work that is not yet published, see Section 1.4.)

Whereas multilinear maps do not yet have a secure instantiation, there has been a long and productive line of research that, loosely speaking, translates schemes instantiated based on either iO or multilinear maps, into ones based on more standard assumptions, typically the learning with errors assumption (LWE) [Reg05]. This leads us to ask:

Can we construct (quasi-)optimal broadcast encryption systems from LWE?

While we do not provide a complete answer in this paper, we propose a novel "LWE-inspired" construction based on a natural extension of known learning with errors structures. We hope that both the construction and the assumption will prove useful, both in constructing truly LWE-based broadcast encryption and in other LWE-based constructions of advanced cryptographic objects.

A More General Problem: Succinct CP-ABE. It is possible to cast the BE problem as a special case of another cryptographic task which is by itself very interesting, namely that of constructing *succinct ciphertext-policy* attribute-based encryption schemes, as explained below.

Consider a setting where the set S is specified succinctly, say via a Boolean circuit f that takes as input $x \in \{0,1\}^{\log N}$, and encodes membership in S via the relation $f(x) = 1$ *if and only if* $x \in S$. The BE succinctness property now translates naturally to the requirement that the parameters of the scheme scale *polylogarithmically* with N, equivalently polynomial in the bit-length of x, regardless of the description size of f (where, as above, we assume that the circuit that computes f is known to the decryptor and does not count towards the ciphertext size). We note that in this context we can allow N to grow even exponentially with the security parameter, so long as the circuit representing S is bounded by $\mathrm{poly}(\lambda)$. This notion is called ciphertext-policy ABE (CP-ABE).

To describe the state of the art on constructing ABE from LWE, it is useful to introduce the notion of *key-policy* ABE (KP-ABE). This is the dual notion to CP-ABE in which the index x is specified as a part of the ciphertext, and the functions f are associated with decryption keys.

It is shown in [GVW13,BGG+14] how to construct KP-ABE when f is computable by (a-priori bounded) polynomial depth Boolean circuits. To be more precise, these schemes [GVW13,BGG+14] show that for any polynomials in the security parameter $k = k(\lambda)$, $d = d(\lambda)$ there exists a KP-ABE scheme where the class of policies is the class of all $k(\lambda)$-input depth-$d(\lambda)$ circuits. The parameters of the scheme scale polynomially with k, d, and in addition, the key size may also scale with the size of the circuit representing f. In particular, in [GVW13] the key size scales with $|f| \cdot \mathrm{poly}(\lambda, k, d)$, but in a later construction of Boneh et al. [BGG+14] the key size is a fixed polynomial $\mathrm{poly}(\lambda, k, d)$ independent of $|f|$ (but knowledge of f itself is of course needed for decryption). This is the starting point of our work.

This can be converted into a CP-ABE scheme by simply using a universal circuit to represent a function f as an attribute and an input x as a circuit. This transformation will result in CP-ABE for a restricted function class and with undesirable parameters. Concretely, this will only allow for any polynomials s, d in the security parameter, to construct CP-ABE for functions with circuits of size s and depth d (note that this is more restrictive than having one scheme working for *arbitrary* size circuits). Furthermore, the *ciphertext size* will now be $|f| \cdot \mathrm{poly}(\lambda, s, d)$ even in the [BGG+14] scheme and it is not known how to improve this dependence on f. That is, even if an f being encrypted has a very small circuit, the CP-ABE ciphertext will scale with the *worst-case* s, d.[2] This scheme is unsatisfactory for two reasons: one is the global bound on s, and the other is the dependence of the ciphertext size on s (i.e. succinctness). This work addresses both aspects and we refer to "succinct CP-ABE (for circuits)" as a scheme which addresses both these issues.[3] In other words, for any polynomials in the security parameter $k = k(\lambda)$, $d = d(\lambda)$, we wish to construct a CP-ABE scheme where the class of policies is the class of all $k(\lambda)$-input depth-$d(\lambda)$ circuits. Furthermore, the size of the ciphertext encrypting a policy circuit f should have size independent of the circuit size of f.

We note that under assumptions related to bilinear maps, it is known how to construct CP-ABE for the class of all Boolean formulas [GPSW06, LOS+10], however the ciphertext size scales with $\mathrm{poly}(\lambda, |\varphi|)$ where φ is the formula representing the policy. This does not have the succinctness property that we desire and therefore BE is not implied.

1.1 Our Results: LWE-Inspired BE and CP-ABE Candidates

In this work, we present a candidate construction of a succinct CP-ABE scheme, and by extension we obtain a quasi-optimal BE scheme. Our construction is based on a heuristic that allows to "invert" the key-succinctness of the BGG+ KP-ABE scheme. We do not have a security reduction for this heuristic, and we pose its security as an open problem.

A Historical Note. We have been circulating this candidate since 2015 and sharing it with researchers in the field, in attempt to find a proof or an attack. Some approaches for attacks were proposed by us and by others, but it so far appears that the scheme withstands those attempts. The scheme in this paper is identical to the scheme that we circulated in 2015, and we did not need to make any adjustments in order to maintain (heuristic) security. This somewhat boosts our confidence in the security of the scheme, but we were nevertheless unable to come up with a proof, or even to come up with a closed-form assumption that implies the security of our candidate. We therefore believe that at this point it may be justified to publish our candidate and utilize the collective wisdom of the cryptographic community to find a proof or an attack. We outline some of the main cryptanalysis attempts in Section 1.3 and Section 5 and explain the reasons why they do not seem to apply. A technical overview of our construction is provided in Section 1.2 below.

1.2 Technical Overview

As explained above, our scheme starts from the KP-ABE scheme of [BGG+14], which has succinct keys and non-succinct ciphertexts, and attempts to turn it on its head, to achieve CP-ABE with succinct ciphertexts (and keys). We use LWE-inspired techniques in order to (heuristically) guarantee collusion resistance. Details follow.

The BGG+ Key Policy ABE Scheme. The high-level idea of the BGG+ scheme is a method of encrypting the attribute x in a way that allows for a specific type of homomorphic evaluation.

[2] We note that in the context of KP-ABE, the explicit dependence on the input size k was removed in [BV16, GKW16], but this does not lead to removing the dependence on s in the CP-ABE setting due to the use of universal circuits, which require a fixed input length.

[3] Removal of the dependence on d is also important, but this seems problematic even in the KP-ABE setting.

In particular, the encryption algorithm takes as input a master public key $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k)$ where each matrix lives in $\mathbb{Z}_q^{n\times m}$, and attributes $x \in \{0,1\}^k$ and encrypts it as

$$\big(\mathbf{s}\mathbf{A}_0, \mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G}), \ldots, \mathbf{s}(\mathbf{A}_k + x_k\mathbf{G})\big) + \mathbf{e}$$

where $\mathbf{s} \in \mathbb{Z}_q^{1\times n}$ is a uniformly random row vector ("the LWE secret") and $\mathbf{G}$ is the "gadget matrix" from [MP12]. The vector $\mathbf{e}$ is the "LWE noise vector" which contains short entries (Gaussians, almost always); we will omit the noise vector henceforth and take care to only multiply the ciphertext by small values so that the presence of noise does not significantly change the output. We also ignore the way the encrypted message is encoded in the ciphertext. For the purpose of this exposition one can simply think about using the above to encrypt 0, and using a completely random ciphertext for 1, so that the LWE assumption guarantees that the two are indistinguishable to an adversary, but given a proper key they can be distinguished. We note that more elegant encoding methods exist.

Before describing the decryption process, we note an important *decomposability* feature of this encryption procedure that was already utilized in [BV16]. The encryption process can be split into an "offline step" that is computed before the attribute x is known, and an "online step" whose complexity, and in particular its dependence on x, are extremely simple. Explicitly, in the offline phase, one can compute $\mathbf{c}_0 = \mathbf{s}\mathbf{A}_0$, and then for all $i \in [k]$ and $b \in \{0,1\}$ the value $\mathbf{c}_{i,b} = \mathbf{s}(\mathbf{A}_i - b\mathbf{G})$. Then, in the offline phase, simply output $\mathbf{c}_0$ and $\mathbf{c}_{i,x_i}$ for all i.

The key technique in [BGG$^+$14] is a two-fold homomorphic evaluation procedure which works as follows. Given $\vec{\mathbf{A}} = (\mathbf{A}_1, \ldots, \mathbf{A}_k)$, a function f, and $x \in \{0,1\}^k$, it is possible to derive a matrix $\mathbf{H}$ with relatively small values (essentially exponential in the depth of f, which should be considered small in our setting), s.t.

$$(\vec{\mathbf{A}} + x\mathbf{G})\mathbf{H} = \mathbf{A}_f + f(x)\mathbf{G}\ .$$

The matrix $\mathbf{A}_f$ itself can be efficiently derived from $\vec{\mathbf{A}}$ and f without any knowledge or dependence on x. (We use $x\mathbf{G}$ as shorthand for $x \otimes \mathbf{G} = (x_1\mathbf{G}, \ldots, x_k\mathbf{G})$.)

Using this evaluation procedure, it is possible, given the ciphertext and f, x, to derive

$$\big(\mathbf{s}\mathbf{A}_0, \mathbf{s}(\vec{\mathbf{A}} + x\mathbf{G})\mathbf{H}\big) = \big(\mathbf{s}\mathbf{A}_0, \mathbf{s}(\mathbf{A}_f + f(x)\mathbf{G})\big) \tag{1}$$

up to (somewhat increased but, with proper choice of parameters, still small) noise.

The secret key for f is a pair of "relatively short" vectors $(\mathbf{r}_0, \mathbf{r}_f)$ s.t. $\mathbf{A}_0\mathbf{r}_0 + \mathbf{A}_f\mathbf{r}_f = 0 \pmod q$. Such vectors can be derived given a so-called lattice trapdoor for $\mathbf{A}_0$. Decryption involves multiplying Eq. (1) by the column vector consisting of $\mathbf{r}_0$ and $\mathbf{r}_f$, and checking that the result is small. It is easy to check that decryption succeeds if $f(x) = 0$.

The resulting scheme is a KP-ABE scheme with succinct keys, since the keys contain, apart from f itself, two vectors with short entries whose dimensions are independent of f. The ciphertexts, however, are not succinct: for each attribute bit we need to include an entire vector $\mathbf{c}_{i,x_i}$.

An Approach for (Ciphertext-Policy) ABE with Succinct Ciphertexts. We would like to turn the aforementioned KP-ABE scheme on its head: if we could only invert the roles of the secret key and ciphertext, we would be in good shape, since now the ciphertexts will encode f and be succinct, and the keys will depend on x. However, duality of this sort does not seem straightforward. We therefore consider the following high level outline.

Whenever we wish to encrypt a message with respect to policy f, we will generate a new instance of the KP-ABE scheme. We will generate a key with respect to f and then toss the master secret key of the scheme. We then produce the offline phase of the encryption, i.e. create all ciphertext pieces $\mathbf{c}_0$ and $\{\mathbf{c}_{i,b}\}_{i\in[k],b\in\{0,1\}}$. Now, seemingly, all we need to do is find a way to encode the $\mathbf{c}_{i,b}$ pieces so that user x can only recover $\mathbf{c}_{i,x_i}$. This can easily be done using IBE or (again) KP-ABE as is done in [BV16], by just encrypting each $\mathbf{c}_{i,b}$ with respect to attribute (i, b) and provide user x with a key that decrypts all ciphertexts with attribute (i, x_i). Indeed, this will provide functionality, and even security against an adversary that only gets access to a single decryption key, but not *collusion resistance*: if users x, x' come together, they will be able to learn, for some i, both $\mathbf{c}_{i,0}$ and $\mathbf{c}_{i,1}$, which will render the ciphertext completely insecure. To solve the collusion resistance problem, we will devise a variant of IBE/ABE that, during the decryption process, modifies the $\mathbf{s}$ part of the ciphertext, so that users x, x' will each receive a variant of $\mathbf{c}_{i,x_i}$ or $\mathbf{c}_{i,x'_i}$ respectively, but such that each of them corresponds to a different $\mathbf{s}$. If we were able to do this in such a way that the $\mathbf{s}$ values are completely uniform and independent, and such that in some way the noise values in the $\mathbf{c}_{i,x_i}$ pieces are also randomized, then we will have solved the problem. Unfortunately we cannot achieve this and we therefore resort to a heuristic.

Before describing our heuristic, let us point out that even the variant which enjoys only one-key security (and is not collusion resistant) is not fully succinct, in the sense that the ciphertext size grows with k (times a polynomial in the security parameter). However, this level of succinctness is sufficient for all purposes covered in this work, in particular for broadcast encryption.

Our Heuristic: Select on the Left, Decrypt on the Right. We first "beef up" the KP-ABE scheme. Instead of using a single vector $\mathbf{s}$, we will use a matrix $\mathbf{S}$, or equivalently we will generate a number of independent ciphertexts with independent $\mathbf{s}$ values. Intuitively, this will be useful since it will allow us to define, for each x, its own personal $\mathbf{s}_x$ that will be defined as (roughly) a subset sum of the rows of $\mathbf{S}$, with coefficients that will be specified for x. This modification changes very little in the semantics of the scheme. It still has the same offline/online nature, only now the pieces $\mathbf{C}_0 \approx \mathbf{S}\mathbf{A}$ and $\mathbf{C}_{i,b} \approx \mathbf{S}(\mathbf{A}_i + b\mathbf{G})$ are matrices and not vectors. Recall that decryption only involves multiplying the respective ciphertext pieces on the *right*, first by the matrix $\mathbf{H}$ and then by the secret key vectors $(\mathbf{r}_0, \mathbf{r}_f)$. We will therefore mask the ciphertext pieces, using an encoding that is decryptable *on the left*.

Concretely, we will consider matrices $\mathbf{B}_0$ and $\{\mathbf{B}_{i,b}\}$, and set $\hat{\mathbf{C}}_{i,b} = \mathbf{B}_{i,b}\hat{\mathbf{S}} + \mathbf{E}_{i,b} + \mathbf{C}_{i,b}$, and likewise $\hat{\mathbf{C}}_0$ will encrypt $\mathbf{C}_0$. Note that the same $\hat{\mathbf{S}}$ is used in all new ciphertext pieces, but each one uses a separate $\mathbf{B}$ and separate noise. In itself, the LWE assumption implies that without any additional information on the $\mathbf{B}$ matrices, the $\hat{\mathbf{C}}$ pieces are indistinguishable from uniform, even given arbitrary side information on the $\mathbf{C}$ pieces.

Now, in order to allow decryption, we will provide user x with a short vector $\mathbf{t}_x$ which is sampled randomly (from a discrete Gaussian) subject to $\mathbf{t}_x\mathbf{B}_{i,x_i} = 0 \pmod q$ for all i, and in addition $\mathbf{t}_x\mathbf{B}_0 = 0 \pmod q$. Such a vector can be efficiently sampled if the $\mathbf{B}$ matrices are generated together with a trapdoor for the *intersection* of lattices that they represent – the details are not very important at this point.

Note that given $\mathbf{t}_x$, it is possible to derive $\mathbf{t}_x\hat{\mathbf{C}}_{i,x_i} \approx \mathbf{t}_x\mathbf{C}_{i,x_i}$, but for $\hat{\mathbf{C}}_{i,\bar{x}_i}$ the vector $\mathbf{t}_x$ is useless (at least heuristically) and does not enable deriving any useful variant of $\mathbf{C}_{i,\bar{x}_i}$. Since the decryption procedure is performed by right multiplication, correctness is maintained even when applying decryption to the $\mathbf{t}_x\mathbf{C}_{i,x_i}$ values. The hope is that collusion attacks are prevented since an adversary that is given, say, $\mathbf{t}_x, \mathbf{t}_{x'}$ for two attribute vectors x and x' should not be able to correlate the ciphertext pieces that it derives in the selection process.

This completes the description of our scheme. Note that we obtain succinct ciphertexts and succinct keys. We only use the "symmetric key" version of the KP-ABE scheme, and only use it to generate a single key. These are very appealing properties in terms of functionality, but we have to be very careful when arguing about its security. We will discuss this in the following section.

1.3 Security and Attempted Cryptanalysis

Our heuristic scheme is based on similar logic to the one standing behind the [BGG+14] KP-ABE scheme. We mask a set of values using LWE instances of the form $\mathbf{SA} + \mathbf{E}$, and provide a trapdoor that exposes only certain relations between the masked values. Let us try to speculate what would be the nature of a security proof for our scheme, and focus on possible aspects where problems could arise. We consider the weakest notion of security, namely fully selective security, and focus only on the broadcast encryption application rather than consider CP-ABE in full generality. Specifically, we consider a security game, where the adversary selects k which is logarithmic in the security parameter, and then specifies two *disjoint* sets $X, S \subseteq \{0,1\}^k$. The set X indicates "corrupted" x values, and the set S indicates the recipients of the broadcast. The adversary then receives the public parameters of the scheme, the secret keys for all $x \in X$, and a ciphertext encrypting a random bit with respect to the set S. The adversary is required to guess the value of the encrypted bit with non-trivial success probability.

Let us take the cryptanalyst's perspective and see what kind of potentially-sensitive information can be derived from the adversary's view of the experiment. Assume that the set X is simply a set of randomly selected identities. Then it must be the case that for every index i, roughly half $x \in X$ have $x_i = 0$ and half have $x_i = 1$. This means that the adversary obtains many short $\mathbf{t}_x$ that annihilate $\mathbf{B}_{i,0}$ (and likewise many *other* short vectors that annihilate $\mathbf{B}_{i,1}$). Therefore, the adversary can use these values to reconstruct a trapdoor for all individual matrices $\mathbf{B}_{i,b}$. While this feature appears to be concerning, it does not seem to lead to a break of any sort. Indeed, what is needed in order to learn meaningful information is a trapdoor that *simultaneously* annihilates, e.g. both $\mathbf{B}_{i,0}$ and $\mathbf{B}_{i,1}$ for some i. This perspective can be generalized further to notice that as a necessary condition for security, it should not be possible, given secret keys for the set X, to efficiently generate a valid key for any $x^* \notin X$. Doing so would immediately violate security. This again translates to a question on "mixing and matching" or "malleability" of lattice trapdoors. We show in Section 5 that learning with error trapdoors have a "non-malleability" property so that it is computationally intractable to mix-and-match lattice trapdoors in a way that will compromise the hardness of LWE on a lattice to which an explicit trapdoor has not been given. This is true even if this target lattice is an intersection of lattices for which we provide a trapdoor. This suggests that this attack direction might not be successful.

Of course, so far we only considered very limited attack strategies, namely ones that only observe the set of keys of colluding adversarial parties, and not the challenge ciphertext itself. Indeed, when trying to convert ABE schemes into the stronger notions of Functional Encryption (FE) or Witness Encryption (WE), collusion resistance is lost because even keys that are not authorized to decrypt reveal sufficient information about the *encryption randomness*, which allows to recover it and break security. In this context, we usually consider the set of linear equations that are obtained by applying the keys that the adversary has to the ciphertext. Of course none of the keys individually will decrypt, but each such attempted decryption implies a linear equation over the unknowns which are the randomness in the encryption (in our case the matrix $\mathbf{S}$ and the LWE noise values). The hope is to obtain sufficiently many such equations to recover, say, $\mathbf{S}$. At first glance, this approach appears to be successful. It appears that given sufficiently many keys, it should be possible to generate more equations than variables, and thus supposedly break security. However, a seemingly minor detail plays an important role here. The set of equations that can be obtained in this way is linearly independent *over the integers*, but the equations themselves are only applicable *modulo* q, and it turns out (not surprisingly, as we explain below) that the equations obtained are not full rank modulo q, and thus the information obtained in this way appears to be useless to an attacker. This is indeed a significant difference between our approach and the aforementioned approaches towards FE, WE. In our case, the adversary is unable to obtain equations over the integers, only modular ones (since decrypting with a non-certified key should not yield any functionality, as opposed to FE where all keys have correctness requirement with respect to all ciphertexts). To illustrate this issue, consider a plain LWE instance of the form $(\mathbf{A}, \mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e})$. Now assume that we are given a short full rank matrix $\mathbf{T}$ s.t. $\mathbf{A}\mathbf{T} = 0$. This matrix is a trapdoor for $\mathbf{A}$ and can be used to recover the secrets $\mathbf{s}, \mathbf{e}$ via computing the value $\mathbf{b}\mathbf{T} \pmod q$ and noticing that this is equal to the value $\mathbf{e}\mathbf{T}$ *over the integers*. Since $\mathbf{T}$ is invertible over the integers $\mathbf{e}$ can be recovered, which leads to recovery of $\mathbf{s}$ as well. However, if $\mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{y}$ for a non-short $\mathbf{y}$ ($\mathbf{y}$ can still be very structured, e.g. LWE instance with respect to a known matrix $\mathbf{B}$), this approach fails. This is because now we can only obtain $\mathbf{y}\mathbf{T} \pmod q$, and we note that modulo q the matrix $\mathbf{T}$ is degenerate since it lies in the mod-q kernel of $\mathbf{A}$. It appears that all "linear" cryptanalysis attempts fall into this framework, but we are unable to show a formal statement of this form.

To conclude, it appears that attacks that were successful in other contexts are not applicable to our scheme. This is of course far from constituting a proof of security. As explained above, we hope that one of the readers of this manuscript will be able to advance it, either in the direction of a proof, or in the direction of finding new attacks.

1.4 Other Related Works

We recently learned of a yet unpublished work by Agrawal and Yamada [AY20] that construct broadcast encryption and succinct ciphertext-policy ABE relying on the LWE assumption in addition to groups with bilinear maps. Their proof of security relies on the *generic model* for groups with bilinear maps. Let us point out a few points of comparison.

As explained above, our scheme predates this work. In terms of techniques, their scheme also starts with a KP-ABE scheme with succinct keys and attempts to "turn it on its head". They use the bilinear structure to prevent collusion by only allowing the parties to see their ciphertext in the exponent of a group generator which is distinct for each user.

In terms of the final result, our work relies only on the lattice structure, allows CP-ABE for all functions, but has no proof of security. Their work relies on (standard) LWE and generic bilinear groups (and therefore immediately broken by quantum attacks). Their CP-ABE scheme only supports NC^1 functions, whereas we support all polynomial-time computable functions, with an a-prior bounded (polynomial) depth. However it has the significant advantage of being provably secure in a reasonable (yet heuristic) attack model.

Acknowledgements. We thank Shweta Agrawal, Dan Boneh, Yilei Chen, Sam Kim, Alex Lombardi, Rotem Tsabary, and Hoeteck Wee for discussions about the scheme and attempts at proofs

and cryptanalysis.

2 Preliminaries

2.1 Attribute Based Encryption (ABE)

Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_\lambda$ be an ensemble of function classes such that $\mathcal{F}_\lambda \subseteq \{0,1\}^* \to \{0,1\}$. We assume that the functions are represented as Boolean circuits. A ciphertext-policy attribute based encryption (CP-ABE) scheme is defined by PPT algorithms $\mathsf{ABE} = (\mathsf{ABE.Params}, \mathsf{ABE.Enc}, \mathsf{ABE.Keygen}, \mathsf{ABE.Dec})$ such that:

- The setup algorithm $\mathsf{ABE.Params}(1^\lambda)$ takes the security parameter and the input length of the supported functions as input and outputs a master secret key msk and a set of public parameters pp.
- The encryption algorithm $\mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, f)$ uses the public parameters pp and takes as input a function $f \in \mathcal{F}_\lambda$ and a message μ from a message space $\mathcal{M} = \mathcal{M}_\lambda$. It outputs a ciphertext $\mathsf{ct} \in \{0,1\}^*$.
- The key generation algorithm $\mathsf{ABE.Keygen}_{\mathsf{msk}}(x)$ uses the master secret key msk and takes as input an attribute vector $x \in \{0,1\}^k$. It outputs a secret key sk_x.
- The decryption algorithm $\mathsf{ABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_x, x, f, \mathsf{ct}_f)$ takes as input a function secret key sk_f, an attribute $x \in \{0,1\}^*$, a function f and a ciphertext ct_f, and outputs a message $\mu' \in \mathcal{M}$.

Remark 1. *It is often the case that a construction of ABE (whether key or ciphertext policy) is applicable to a parameterized class of functions. For example, we may be able to construct ABE for all $\mathcal{F}^{(c)} = \{\mathcal{F}^{(c)}_\lambda\}_\lambda$, where $\mathcal{F}^{(c)}_\lambda$ denotes the set of all circuits of depth λ^c, but such that the parameters of the scheme vary with c. In such cases, we sometimes consider c (or λ^c in this case), as an additional input to the setup algorithm rather than an external specification of the function class. Usually, and specifically in the context of this manuscript, when the setup algorithm takes additional inputs, we explain how these inputs refer to the function classes supported by the construction.*

Remark 2. *We note that the ABE setup algorithm needs to know (and can run in time polynomial in) the input length of the functions, but* not the size of the circuit computing the supported functions*. This is a key feature of CP-ABE schemes which cannot be obtained generically from KP-ABE schemes, flipping the role of the function and input using a universal circuit.*

Definition 2.1 (Correctness of CP-ABE)**.** *A scheme* ABE *is* correct *if the following holds. Consider a sequence of functions $\{f_\lambda \in \mathcal{F}_\lambda\}_\lambda$ and a sequence of attributes $\{x_\lambda \in \{0,1\}^*\}_\lambda$, such that for all λ, the input size of f_λ is* exactly $|x_\lambda|$ *and $f_\lambda(x_\lambda) = 0$.*[4] *For all such sequences and for any sequence $\{m_\lambda \in \mathcal{M}_\lambda\}_\lambda$, it holds that*

$$\Pr[\mathsf{ABE.Dec}_{\mathsf{pp}}(\mathsf{sk}_x, x, f, \mathsf{ct}_f) \neq \mu] = \mathrm{negl}(\lambda) ,$$

where $(\mathsf{msk}, \mathsf{pp}) = \mathsf{ABE.Params}(1^\lambda, 1^k)$, $\mathsf{ct} = \mathsf{ABE.Enc}_{\mathsf{pp}}(\mu, f)$, $\mathsf{sk}_f = \mathsf{ABE.Keygen}_{\mathsf{msk}}(x)$.

[4] Recall our convention that $f(x) = 0$ is the event when decryption succeeds.

Definition 2.2 (Security for CP-ABE)**.** *Let* ABE *be a CP-ABE encryption scheme as above, and consider the following game between the challenger and adversary.*

1. *The challenger generates* $(\mathsf{msk}, \mathsf{pp}) = \mathsf{ABE.Params}(1^\lambda, 1^k)$*, and sends* pp *to the adversary.*

2. *The adversary makes arbitrarily many key queries by sending attributes* x_i *to the challenger. Upon receiving such an attribute, the challenger creates* $\mathsf{sk}_i = \mathsf{ABE.Keygen}_{\mathsf{msk}}(x_i)$ *and sends* sk_i *to the adversary.*

3. *The adversary sends a function* f *and a pair of messages* μ_0, μ_1 *to the challenger. The challenger samples* $b \in \{0, 1\}$ *and computes the challenge ciphertext* $\mathsf{ct}^* = \mathsf{ABE.Enc}_{\mathsf{pp}}(\mu_b, f)$*. It sends* ct^* *to the adversary.*

4. *The adversary makes arbitrarily many key queries as in Step 2 above.*

5. *The adversary outputs* $\tilde{b} \in \{0, 1\}$*.*

6. *Let* legal *denote the event where all key queries of the adversary are such that* $f(x_i) = 1$*. 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 adaptive security game *for ABE, and it has relaxed variants. In the* selective security game*, the adversary sends* f *before Step 1.*

The scheme ABE *is* adaptively/selectively secure *if any* PPT *adversary* $\mathcal{A}$ *only has negligible advantage in the adaptive/selective security game (respectively).*

Negated Policies. We allow decryption when $f(x) = 0$ and require that in the security game all queries are such that $f(x_i) = 1$. In LWE-based constructions it is often much more convenient to work with this negated version of the policy, which explains the apparent strangeness. This variant is obviously equivalent.

Succinct Ciphertext-Policy ABE. The key succinctness property we care about in this work is that the ciphertext size depends polynomially in the input length of the functions, namely k, and the security parameter, but is otherwise independent of the size of the functions being encrypted. As a result of a technical deficit common to all known LWE based ABE schemes, we will allow the ciphertext size to grow with the depth of the circuits (but not the size).

2.2 Learning with Errors and Lattice Trapdoors

This section summarizes tools from previous works that are used in our construction. This includes the definition of the LWE problem and its relation to worst case lattice problems, the notion of trapdoors for lattices and operations on trapdoors, and homomorphic evaluation of matrices with special properties.

Learning with Errors (LWE). The Learning with Errors (LWE) problem was introduced by Regev [Reg05] as a generalization of "learning parity with noise" [BFKL93, Ale03]. We now define the decisional version of LWE. (Unless otherwise stated, we will treat all vectors as column vectors in this paper).

Definition 2.3 (Decisional LWE (DLWE) [Reg05])**.** *Let λ be the security parameter, $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be integers and $\chi = \chi(\lambda)$ 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 $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, the following distributions are computationally indistinguishable:*

$$\left(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right) \overset{c}{\approx} \left(\mathbf{A}, \mathbf{u}^T\right)$$

There are known quantum (Regev [Reg05]) and classical (Peikert [Pei09]) reductions between $\mathrm{DLWE}_{n,q,\chi}$ and approximating short vector problems in lattices. Specifically, these reductions take χ to be a discrete Gaussian distribution $D_{\mathbb{Z},\alpha q}$ for some $\alpha < 1$. We write $\mathrm{DLWE}_{n,q,\alpha}$ to indicate this instantiation. We now state a corollary of the results of [Reg05, Pei09, MM11, MP12]. These results also extend to additional forms of q (see [MM11, MP12]).

Corollary 1 ([Reg05, Pei09, MM11, MP12])**.** *Let $q = q(n) \in \mathbb{N}$ be either a prime power $q = p^r$, or a product of co-prime numbers $q = \prod q_i$ such that for all i, $q_i = \mathrm{poly}(n)$, and let $\alpha \geq \sqrt{n}/q$. If there is an efficient algorithm that solves the (average-case) $\mathrm{DLWE}_{n,q,\alpha}$ problem, then:*

- *There is an efficient quantum algorithm that solves $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ (and $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$) on any n-dimensional lattice.*
- *If in addition $q \geq \tilde{O}(2^{n/2})$, there is an efficient classical algorithm for $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ on any n-dimensional lattice.*

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 require at least $2^{\tilde{\Omega}(n/\log\gamma)}$ time [Sch87]. We refer the reader to [Reg05, Pei09] for more information.

In this work, we will only consider the case where $q \leq 2^n$. Furthermore, the underlying security parameter λ is assumed to be polynomially related to the dimension n.

Lastly, we derive the following corollary which will allow us to choose the LWE parameters for our scheme. The corollary follows immediately from the fact that the discrete Gaussian $D_{\mathbb{Z},\alpha q}$ is $(\alpha q \cdot t, 2^{-\Omega(t^2)})$-bounded for all t.

Corollary 2. *For all $\epsilon > 0$ there exist functions $q = q(n) \leq 2^n, \chi = \chi(n)$ such that χ is B-bounded for some $B = B(n)$, $q/B \geq 2^{n^\epsilon}$ 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 $\gamma = 2^{\Omega(n^\epsilon)}$.*

The Gadget Matrix. Let $N = n \cdot \lceil \log q \rceil$ and define the "gadget matrix" $\mathbf{G} = \mathbf{g} \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n\times N}$ where $\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}_q^{\lceil \log q \rceil}$. We will also refer to this gadget matrix as the "powers-of-two" matrix. We define the inverse function $\mathbf{G}^{-1} : \mathbb{Z}_q^{n\times m} \to \{0,1\}^{N\times m}$ 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 $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A}$.

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

The following properties had been established in a long sequence of works.

Corollary 3 (Properties of Trapdoors [Ajt96, Ajt99, GPV08, ABB10a, CHKP12, ABB10b, MP12])**.** *Lattice trapdoors exhibit the following properties.*

1. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $\mathbf{A}_{\tau'}^{-1}$ for* any *$\tau' \geq \tau$.*
2. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $[\mathbf{A}\|\mathbf{B}]_\tau^{-1}$ and $[\mathbf{B}\|\mathbf{A}]_\tau^{-1}$ for* any *$\mathbf{B}$.*
3. *For all $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{R} \in \mathbb{Z}^{m\times N}$, with $N = n\lceil \log q\rceil$, one can obtain $[\mathbf{AR}+\mathbf{G}\|\mathbf{A}]_\tau^{-1}$ for $\tau = O(m\cdot\|\mathbf{R}\|_\infty)$.*
4. *There exists an efficient procedure $\mathsf{TrapGen}(1^n, q)$ that outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some $m = O(n\log q)$ and is 2^{-n}-uniform, where $\tau_0 = O(\sqrt{n\log q\log n})$.*

3 The [BGG+14] KP-ABE Scheme

In this section, we recall the KP-ABE scheme of Boneh et al. [BGG+14]. This scheme will be the basis of our construction in Section 4. We start by describing the notion of homomorphic evaluation that underlies their construction.

Key-Homomorphic Evaluation. Let f be a boolean circuit of depth d computing a function from $\{0,1\}^k$ to $\{0,1\}$, and assume that f contains only NAND gates. We will show a version of f that computes on matrices. In particular, for every NAND gate whose inputs are associated to matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, we associate to the output the matrix

$$\mathbf{A}_{\mathsf{nand}} := \mathbf{A}_1\mathbf{G}^{-1}(\mathbf{A}_2) - \mathbf{G} \tag{2}$$

Note that

$$\begin{aligned}[\mathbf{A}_1 + x_1\mathbf{G}||\mathbf{A}_2 + x_2\mathbf{G}]\cdot\begin{bmatrix}\mathbf{G}^{-1}(\mathbf{A}_2)\\ -x_1\mathbf{I}\end{bmatrix} &= \mathbf{A}_{\mathsf{nand}} + (1 - x_1x_2)\mathbf{G}\\ &= \mathbf{A}_{\mathsf{nand}} + \mathsf{NAND}(x_1, x_2)\cdot\mathbf{G}\end{aligned}$$

which provides us with a mathematical framework for homomorphic evaluation.

In particular, given matrices $\vec{\mathbf{A}} := (\mathbf{A}_1, \ldots, \mathbf{A}_k)$, we can define the matrix $\mathbf{A}_f$ associated to the output wire of f. This has the associated homomorphic evaluation property that

$$[\mathbf{A}_1 + x_1\mathbf{G}||\ldots||\mathbf{A}_k + x_k\mathbf{G}]\cdot\mathbf{H}_{f,x,\vec{\mathbf{A}}} = \mathbf{A}_f + f(x_1, \ldots, x_k)\cdot\mathbf{G} \tag{3}$$

for some matrix $\mathbf{H}_{f,x,\vec{\mathbf{A}}}$ that has norm $(n\log q)^{O(d)}$.

The KP-ABE Scheme. We are now ready to describe the KP-ABE scheme of [BGG+14].

- $\mathsf{Setup}(1^\lambda)$ generates a master public key

$$\mathsf{BGG.MPK} = (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k)$$

  and a master secret key $\mathsf{BGG.MSK} = \mathbf{T}_{\mathbf{A}_0}$, the lattice trapdoor of $\mathbf{A}_0$. This, as usual, is done by running the trapdoor sampling algorithm $(\mathbf{A}_0, \mathbf{T}_0) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.

- $\mathsf{Enc}(\mathsf{BGG.MPK}, x, \mu)$ where $x \in \{0,1\}^k$ and $\mu \in \{0,1\}$ does the following. Pick a random LWE secret $\mathbf{s} \in \mathbb{Z}_q^n$ and output the ciphertext

$$\mathsf{BGG.ct} := \mathbf{s}^T[\mathbf{A}_0||\mathbf{A}_1 + x_1\mathbf{G}||\ldots||\mathbf{A}_k + x_k\mathbf{G}] + \mathbf{e}^T$$

  where $\mathbf{e}$ is a small Gaussian error.

  The particular method of encrypting the message μ will not be particularly relevant to us; for example, think of computing $\mu \oplus \mathsf{GL}(\mathbf{s})$ where $\mathsf{GL}(\cdot)$ denotes the Goldreich-Levin hardcore bit.

- $\mathsf{KeyGen}(\mathsf{BGG.MSK}, f)$: Outputs a random trapdoor for the matrix $[\mathbf{A}_0||\mathbf{A}_f]$. This can be generated using $\mathbf{T}_{\mathbf{A}_0}$ using a trick due to [CHKP12] (see Corollary 3.)

- $\mathsf{Dec}(\mathsf{BGG.sk}_f, \mathsf{BGG.ct})$ uses Equation 3 to compute

$$\left(\mathbf{s}^T[\mathbf{A}_1 + x_1\mathbf{G}||\ldots||\mathbf{A}_k + x_k\mathbf{G}] + \mathbf{e}^T\right) \cdot \mathbf{H}_{f,x,\vec{\mathbf{A}}} \approx \mathbf{s}^T(\mathbf{A}_f + f(\vec{x})\mathbf{G})$$

  Thus, when $f(x) = 0$, we have (approximately) $\mathbf{s}^T[\mathbf{A}_0||\mathbf{A}_f]$ which can be decoded using the functional key which is a trapdoor for $[\mathbf{A}_0||\mathbf{A}_f]$.

We will not provide an analysis of the parameters of the scheme here (we will do that in Section 4 for our CP-ABE scheme instead) and we will not provide the security reduction for the [BGG+14] scheme (we refer the reader to the original paper.)

4 Our CP-ABE Scheme and Correctness

We describe our CP-ABE scheme below.

- $\mathsf{Setup}(1^\lambda)$: Let k be the length of the attributes in the CP-ABE scheme and d be the maximum depth of circuits that will be encrypted. Choose $2k$ uniformly random matrices $\mathbf{B}_{i,b} \in \mathbb{Z}_q^{m\times n}$ in the public parameters (for $i \in [k]$ and $b \in \{0,1\}$), and let the master secret key be a "joint trapdoor" for all the $\mathbf{B}_{i,b}$.

  More precisely, run $\mathsf{TrapGen}(1^{2kn}, 1^m, q)$ to generate

$$(\mathbf{B}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^{2kn}, 1^m, q)$$

  where $\mathbf{B} \in \mathbb{Z}_q^{m\times 2kn}$. Let $\{\mathbf{B}_{i,b}\}_{i\in[k],b\in\{0,1\}}$ be blocks of columns of $\mathbf{B}$; thus,

$$\mathbf{B} = [\mathbf{B}_{1,0}||\mathbf{B}_{1,1}||\ldots||\mathbf{B}_{k,0}||\mathbf{B}_{k,1}]$$

In particular, by the properties of the trapdoor, we know that $\mathbf{T}$ has small entries, and that for all i, b:

$$\mathbf{T}\mathbf{B}_{i,b} = \mathbf{0} \pmod{q}$$

Output

$$\mathsf{MPK} = \{\mathbf{B}_{i,b}\}_{i,b} \text{ and } \mathsf{MSK} = \mathbf{T}$$

- $\mathsf{Enc}(\mathsf{MPK}, f, \mu)$: To generate a ciphertext for a function f encrypting a message $\mu \in \{0, 1\}$, do the following.
  - Pick fresh ABE public parameters for the BGG+ ABE scheme.

    $$(\mathsf{BGG.MPK}, \mathsf{BGG.MSK}) \leftarrow \mathsf{BGG.Setup}(1^\lambda, 1^k, 1^d)$$

    We recall that $\mathsf{BGG.MPK}$ consists of $k+1$ matrices $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_k \in \mathbb{Z}_q^{n \times \ell}$ and $\mathsf{BGG.MSK}$ is the trapdoor for $\mathbf{A}_0$.
  - Generate an BGG+ ABE *secret key* sk_f for the circuit f.

    $$\mathsf{BGG.sk}_f \leftarrow \mathsf{BGG.KeyGen}(\mathsf{BGG.MSK}, f)$$

    We recall that $\mathsf{BGG.sk}_f$ is a pair of short vectors $(\mathbf{r}_0, \mathbf{r}_f)$ such that

    $$[\mathbf{A}_0 || \mathbf{A}_f] \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_f \end{bmatrix} = \mathbf{0} \pmod{q}$$
  - Run the offline phase of the matrix-BGG+ encryption algorithm. That is, let

    $$(\mathbf{C}_0, \{\mathbf{C}_{i,b}\}) \leftarrow \mathsf{BGG.OfflineEnc}(\mathsf{BGG.MPK})$$

    We recall that $\mathbf{C}_{i,b} = \mathbf{S}(\mathbf{A}_i + b\mathbf{G}) + \mathbf{E}_{i,b}$ and $\mathbf{C}_0 = \mathbf{S}\mathbf{A}_0 + \mathbf{E}_0$ where $\mathbf{S} \in \mathbb{Z}_q^{m \times n}$ is a uniformly random matrix and $\mathbf{E}_0, \{\mathbf{E}_{i,b}\}$ are short error matrices of the appropriate dimension.

  Define the matrices

  $$\hat{\mathbf{C}}_{i,b} := \mathbf{B}_{i,b}\hat{\mathbf{S}}_{i,b} + \mathbf{C}_{i,b} = \mathbf{B}_{i,b}\hat{\mathbf{S}}_{i,b} + \mathbf{E}_{i,b} + \mathbf{S}(\mathbf{A}_i + b\mathbf{G})$$

  and

  $$\hat{\mathbf{C}}_0 := \mathbf{S}\mathbf{A}_0 + \mathbf{E}_0 + \mu\mathbf{G}$$

  The ciphertext is

  $$CT_f := \Big(\mathsf{BGG.sk}_f, \hat{\mathbf{C}}_0, \{\hat{\mathbf{C}}_{i,b}\}_{i,b}\Big)$$

- $\mathsf{KeyGen}(\mathsf{MSK}, x)$: The key sk_x for an attribute vector $x = (x_1, ..., x_k)$ is a short vector $\mathbf{t}_x$ such that for every i,

  $$\mathbf{t}_x \mathbf{B}_{i,x_i} = 0 \pmod{q}$$

  Such a vector can be generated by running $\mathsf{TrapSamp}$ with the matrix $\mathbf{B}_x := [\mathbf{B}_{1,x_1} || \ldots || \mathbf{B}_{k,x_k}]$. and the trapdoor $\mathbf{T}$.

- $\mathsf{Dec}(x, \mathsf{sk}_x, f, CT_f)$: To decrypt, compute

  $$\mathbf{t}_x \cdot [\hat{\mathbf{C}}_0 || \hat{\mathbf{C}}_{1,x_1} || \ldots || \hat{\mathbf{C}}_{k,x_k}] \cdot \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{H}_{f,x}\mathbf{r}_f \end{bmatrix} \tag{4}$$

  and output 1 if the absolute value of this number is at most $q/100$ and 1 otherwise.

Correctness. Let us rewrite the decryption expression 4. Here we will refer to the BGG+ decryption algorithm and its correctness from Section 3. We have

$$\begin{aligned}
&\mathbf{t}_x \cdot [\hat{\mathbf{C}}_0 || \hat{\mathbf{C}}_{1,x_1} || \ldots || \hat{\mathbf{C}}_{k,x_k}] \\
&\quad = \mathbf{t}_x \cdot [\mathbf{C}_0 || \mathbf{B}_{1,x_1} \hat{\mathbf{S}}_{1,x_1} + \mathbf{C}_{1,x_1} || \ldots || \mathbf{B}_{k,x_k} \hat{\mathbf{S}}_{k,x_k} + \mathbf{C}_{k,x_k}] \\
&\quad = \mathbf{t}_x \cdot [\mathbf{C}_0 || \mathbf{C}_{1,x_1} || \ldots || \mathbf{C}_{k,x_k}] \\
&\quad = \mathbf{s}_x [\mathbf{A}_0 || \mathbf{A}_1 + x_1 \mathbf{G} || \ldots || \mathbf{A}_k + x_k \mathbf{G}] + \mathbf{t}_x \cdot [\mathbf{E}_0 || \mathbf{E}_{1,x_1} || \ldots || \mathbf{E}_{k,x_k}] + [\mu \mathbf{t}_x \mathbf{G} || \mathbf{0} || \ldots || \mathbf{0}] \\
&\quad \approx \mathbf{s}_x [\mathbf{A}_0 || \mathbf{A}_1 + x_1 \mathbf{G} || \ldots || \mathbf{A}_k + x_k \mathbf{G}] + [\mu \mathbf{t}_x \mathbf{G} || \mathbf{0} || \ldots || \mathbf{0}]
\end{aligned}$$

where $\mathbf{s}_x = \mathbf{t}_x \mathbf{S}$. Now,

$$\begin{aligned}
&\mathbf{t}_x \cdot [\hat{\mathbf{C}}_0 || \hat{\mathbf{C}}_{1,x_1} || \ldots || \hat{\mathbf{C}}_{k,x_k}] \cdot \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{H}_{f,x} \mathbf{r}_f \end{bmatrix} \\
&\quad \approx \mathbf{s}_x [\mathbf{A}_0 || \mathbf{A}_1 + x_1 \mathbf{G} || \ldots || \mathbf{A}_k + x_k \mathbf{G}] \cdot \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{H}_{f,x} \mathbf{r}_f \end{bmatrix} + \mu \mathbf{t}_x \mathbf{G} \mathbf{r}_0 \\
&\quad \approx \mathbf{s}_x \cdot (\mathbf{A}_0 \mathbf{r}_0 + (\mathbf{A}_f + f(x) \mathbf{G}) \mathbf{r}_f) + \mu \cdot \mathbf{t}_x \mathbf{G} \mathbf{r}_0 \\
&\quad \approx \mu \cdot \mathbf{t}_x \mathbf{G} \mathbf{r}_0
\end{aligned}$$

where the last equation holds if $f(x) = 0$. Since $\mathbf{t}_x \mathbf{G} \mathbf{r}_0$ is large mod q w.h.p., this lets us recover μ.

Parameter Settings. Looking at the decryption equation, we see that the largest asymptotic growth of error happens during the BGG+ decryption depth, i.e., multiplying by the matrix $\mathbf{H}_{f,x}$ which increases the error by a multiplicative factor of $(n \log q)^{O(d)}$. Thus, setting $q \gg (n \log q)^{O(d)}$ and $m = cn \log q$ for a sufficiently large constant $c > 1$ gives us decryption correctness.

Efficiency. The key parameter of interest to us is the size of the ciphertext. This consists of the BGG+ secret key as well as the k elements of the BGG+ ciphertext. In total, the length is $\mathsf{poly}(n, \log q, k)$ which is polynomial in k and the circuit depth d, but otherwise independent of the circuit size.

4.1 Broadcast Encryption

Recall that in broadcast encryption, we have a setup algorithm that generates a master public/secret key pair for a universe of $N = 2^k$ users, an encryption algorithm that encrypts a message m to a subset $S \subseteq [N]$ of users, a key generation algorithm that gives each user $i \in [N]$ its private key sk_i, and finally a decryption algorithm that gets a user private key sk_i, a ciphertext ct as well as a description of the intended broadcast set S, and outputs μ if and only if $i \in S$.

A construction of broadcast encryption now follows immediately from the succinct CP-ABE scheme by instantiating the function f in the CP-ABE encryption by the indicator function f_S which, on input i, outputs 0 if and only if $i \in S$. The function f_S can be implemented by a circuit of size $O(|S| \cdot \mathsf{poly}(\log k))$ and depth $O(\log |S|) = O(\log N)$.

- The size of the ciphertexts in the scheme is $\mathsf{poly}(n, k, d) = \mathsf{poly}(n, \log N)$.
- The size of the keys in the scheme is $\mathsf{poly}(n, \log N)$ as well, by inspection.

In other words, this gives us a broadcast encryption scheme with polylogarithmic size ciphertexts and keys.

5 Cryptanalysis: Non-Malleable LWE

In addition to what we described in the introduction, we describe yet another avenue for cryptanalysis, and formally prove that it fails. This avenue leads to an interesting version of the LWE assumption that we call *non-malleable LWE* which we prove is hard under the standard LWE assumption.

In the CP-ABE experiment, we have $2k$ matrices $\mathbf{B}_{i,b} \in \mathbb{Z}_q^{m\times n}$, for $i \in [k]$ and $b \in \{0,1\}$. For a string $x \in \{0,1\}^k$, define a matrix $\mathbf{B}_x \in \mathbb{Z}_q^{m\times kn}$ as follows:

$$\mathbf{B}_x = [\mathbf{B}_{1,x_1}||\mathbf{B}_{2,x_2}||\ldots||\mathbf{B}_{k,x_k}]$$

We provide the adversary with trapdoors for the matrices $\mathbf{B}_x$ for polynomially many x, namely short matrices $\mathbf{T}_x$ such that $\mathbf{T}_x\mathbf{B}_x = \mathbf{0} \pmod q$, and yet expect that LWE w.r.t. $\mathbf{B}_{x^*}$ for any different x^* holds, namely that $\mathbf{B}_{x^*}\mathbf{s}+\mathbf{e}$ is pseudorandom where $\mathbf{s}$ is an LWE secret vector and $\mathbf{e}$ is an LWE error vector. The non-malleable LWE experiment captures this requirement.

The (selectively) non-malleable LWE game proceeds as follows.

- The adversary picks a string $x^* \in \{0,1\}^k$.
- The adversary is given the matrices $\{\mathbf{B}_{i,b}\}$.
- The adversary asks for and obtains a trapdoor for any matrix $\mathbf{B}_x$ where $x \neq x^*$.
- The adversary obtains either an LWE sample relative to x^*, that is $\mathbf{B}_{x^*}\mathbf{s}+\mathbf{e}$, or a uniformly random string, and is asked to distinguish between them.

We now show that it is hard for the adversary to win the *non-malleable LWE* game.

Theorem 5.1. *Non-malleable LWE is secure assuming LWE.*

Proof. We will show that given trapdoor matrices $\mathbf{T}_{\mathbf{B}_x}$ for all $x \neq x^*$,

$$\left(\sum_{i\in[k]} \mathbf{B}_{i,x_i^*}\right)\mathbf{s}+\mathbf{e}$$

is indistinguishable from random. It is easy to see that this also implies that

$$[\mathbf{B}_{1,x_1^*}||\ldots||\mathbf{B}_{k,x_k^*}] \cdot \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \ldots \\ \mathbf{s}_k \end{bmatrix} + \mathbf{e}$$

is indistinguishable from random, as desired.

Given the challenge x^*, we program the matrices $\mathbf{B}_{i,b}$ as follows. The key idea is to find matrices $\mathbf{G}_{i,b}$ where all matrices $\mathbf{G}_x$ with

$$\mathbf{G}_x = [\mathbf{G}_{1,x_1}||\mathbf{G}_{2,x_2}||\ldots||\mathbf{G}_{k,x_k}]$$

for $x \neq x^*$ have full rank with known trapdoors, and $\sum_{i\in[k]} \mathbf{G}_{i,x_i^*} = \mathbf{0} \pmod q$.

Once such matrices are found, set each

$$\mathbf{B}_{i,b} = \begin{bmatrix} \mathbf{D}_{i,b} \\ \mathbf{R}\mathbf{D}_{i,b} + \mathbf{G}_{i,b} \end{bmatrix}$$

where $\mathbf{D}_{i,b}$ are random and $\mathbf{R}$ has random small entries. The "ABB trick" (Corollary 3) now allows us to find trapdoors for all $\mathbf{B}_x$ for $x \neq x^*$. At the same time,

$$\sum_i \mathbf{B}_{i,x_i^*} = \begin{bmatrix} \sum_i \mathbf{D}_{i,x_i^*} \\ \mathbf{R} \sum_i \mathbf{D}_{i,x_i^*} \end{bmatrix}$$

so given $(\sum_i \mathbf{D}_{i,x_i^*})\mathbf{s} + \mathbf{e}$, one can simulate $(\sum_i \mathbf{B}_{i,x_i^*})\mathbf{s} + \mathbf{e}'$.

Finally, to design the matrices $\mathbf{G}_{i,b} \in \mathbb{Z}_q^{(2k-1)n \log q \times kn}$ we do the following. We will set

$$\mathbf{G}_{i,b} = \mathbf{u}_{2i-1+b} \otimes \mathbf{G}^T$$

for $i < k$, where $\mathbf{u}_j$ is the j-th unit vector (written as a column vector) and $\mathbf{G}$ is the $n \times n \log q$ gadget matrix. For the last coordinate, we will set

$$\mathbf{G}_{k,x_k^*} = \mathbf{v}_{x^*} \otimes \mathbf{G}^T \text{ and } \mathbf{G}_{k,1-x_k^*} = \mathbf{u}_{2k-1} \otimes \mathbf{G}^T$$

where $\mathbf{v}_{x^*} = -\sum_{i<k} \mathbf{u}_{2i-1+x_i^*}$.

An example with $k = 3$ is illustrated below. Let $x^* = 000$.

$$\mathbf{G}_{1,0} = \begin{bmatrix} \mathbf{G}^T \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_{2,0} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{G}^T \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_{3,0} = \begin{bmatrix} -\mathbf{G}^T \\ \mathbf{0} \\ -\mathbf{G}^T \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

$$\mathbf{G}_{1,1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{G}^T \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_{2,1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{G}^T \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}_{3,1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{G}^T \end{bmatrix},$$

□

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2010.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 298–307. IEEE Computer Society, 2003.

[AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. To appear in Eurocrypt 2020., 2020. (Personal Communication).

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 533–556, 2014.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-abe from LWE: unbounded attributes and semi-adaptive security. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 363–384, 2016.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 206–223. Springer, 2014.

[BZ17] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[GKW16] Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 361–388, 2016.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 89–98. ACM, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554. ACM, 2013.

[LOS+10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *Advances* *in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484. Springer, 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.