FE for Inner Products and Its Application to Decentralized ABE

Zhedong Wang[1] *, Xiong Fan[2], and Feng-Hao Liu[3]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China; State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. wangzhedong@iie.ac.cn.
[2] Cornell University, Ithaca, NY, USA. xfan@cs.cornell.edu.
[3] Florida Atlantic University, Boca Raton, FL, USA. fenghao.liu@fau.edu.

Abstract. In this work, we revisit the primitive functional encryption (FE) for inner products and show its application to decentralized attribute-based encryption (ABE). Particularly, we derive an FE for inner products that satisfies a stronger notion, and show how to use such an FE to construct decentralized ABE for the class $\{0,1\}$-LSSS against bounded collusions in the plain model. We formalize the FE notion and show how to achieve such an FE under the LWE or DDH assumption. Therefore, our resulting decentralized ABE can be constructed under the same standard assumptions, improving the prior construction by Lewko and Waters (Eurocrypt 2011). Finally, we also point out challenges to construct decentralized ABE for general functions by establishing a relation between such an ABE and witness encryption for general NP statements.

1 Introduction

In this work, we revisit the functional encryption (FE) for inner products [6] and show its application to decentralized attribute-based encryption [22]. In particular, we identify a stronger notion for FE required in this application, and show how to build such a scheme under the LWE or DDH assumption. Our new analysis improves the parameters of the LWE-based scheme (over [6]) substantially. Next, we show how to build a decentralized ABE against bounded collusion from FE for inner products that satisfies the stronger notion. By combining the instantiation of the FE, we can derive a decentralized ABE against bounded collusion from LWE or DDH, improving the prior work [42] in the perspective of weaker assumptions. Below, we briefly review the contexts and motivation of our study.

1.1 A Brief History and Motivation

We start with the application of decentralized ABE, and then discuss our central tool – FE for inner products.

* This work was done when the author was visiting Florida Atlantic University.

Attribute-based Encryption. Attribute-based Encryption (ABE) [11, 34] generalizes public key encryption to allow fine grained access control on encrypted data. In (ciphertext-policy) attribute-based encryption, a ciphertext ct for message μ is associated with a policy function f, and a secret key sk corresponds to an attribute x. The functionality requires that decryptor learns the underlying message μ if attribute x satisfies the policy function f, and if not, security guarantees that nothing about μ can be learned. In the past decade, significant progress has been made towards constructing attribute-based encryption for advanced functionalities based on various assumptions [5, 8, 14, 17, 18, 20, 26, 29, 32, 33, 35, 39, 43, 50, 52, 55, 56].

In 2007, Chase [22] considered an extension called multi-authority (or decentralized) ABE. In almost all ABE proposals listed above, the secret keys are generated by one central authority, who has the ability to verify all attributes for the secret keys it generated. These systems can be utilized to share information according to a policy within a domain or organization, where the central authority is in charge of issuing secret keys. However, in many applications, we wish to share some confidential information associated with a policy function across several different trusted organizations. For instance, in a joint project by two corporations, like Google and Facebook, they both may issue secret keys associated with attributes within their own organizations. This setting is outside the scope of the single authority ABE, as the single authority is not capable to verify attributes from different organizations. In [42], the authors show how to construct a decentralized ABE that captures the desired properties, for a large class of functions. Their solutions are secure against unbounded collusion in the random oracle model, or against bounded collusion in the standard model. For both cases, their proofs of security however, rely on several new computational assumptions on bilinear groups. Moreover, their security model only captures a static corruption where the adversary must commit to a set of corrupted parties at the beginning of the security game. To our knowledge, there is no construction that is based on better studied computational assumptions, such as DDH or LWE, even for the setting of bounded collusion. Thus, we ask:

Can we build a decentralized ABE under standard assumptions, even for some restricted class of functions and against bounded collusion?

Along the way to answer this question, we identify an interesting connection between decentralized ABE and functional encryption for inner products [6]. We first review the context of functional encryption (for inner products), and then elaborate on the connection in the technical overview section below.

Functional Encryption (for Inner Products). In a Functional Encryption (FE) scheme [16, 49], a secret key sk_g is associated with a function g, and a ciphertext ct_x is associated with some input x from the domain of g. The functionality of FE requires that the decryption procedure outputs $g(x)$, while security guarantees that nothing more than than $g(x)$ can be learned. Constructing FE for general functions is quite challenging – the only known solutions (supporting unbounded key queries) either rely on indistinguishability obfuscation [25]

or multilinear maps [27]. On the other hand, researchers have identified some special classes of functions that already suffice for many interesting applications [3, 7]. One of them is the *inner products*, where a ciphertext ct encrypts a vector $\boldsymbol{y} \in D^\ell$ for some domain D, and a secret key for vector $\boldsymbol{x} \in D^\ell$ allows computing $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ but nothing else. Abdalla et al. [1] constructed a scheme and prove security against *selective* adversaries, who have to declare the challenge messages $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ at the beginning before seeing the master public key mpk. More recently, Agrawal et al. [6] constructed an adaptively secure FE for inner products that removes this restriction, and in particular, their scheme guarantees the indistinguishability-based (IND) security for key queries both before and after the challenge ciphertext. Moreover, Agrawal et al. [6] pointed out that the IND-based security achieves "almost" the best possible security, as it implies the simulation-based (SIM) security (for the case of inner products) without post-challenge-ciphertext key queries. On the other hand, the SIM-based security is in general impossible to achieve even for one post-challenge-ciphertext key query [6, 16].

In this work, we observe that the IND-based security does not suffice for our task of constructing decentralized ABE with a stronger security guarantee. Furthermore, the efficiency of the currently best known lattice-based construction (FE for inner products) [6] degrades exponentially with the dimension of the vector $\boldsymbol{y}$. A subsequent work [4] improved the parameters significantly, yet with a tradeoff of weaker security where the adversary can only receive $\mathsf{sk}_{\boldsymbol{x}}$ for random $\boldsymbol{x}$'s before the challenge ciphertext and cannot issue more key queries afterwards. Their scheme [4] is useful in the setting of designing trace-and-revoke systems, but cannot be applied to the decentralized ABE where the adversary can obtain keys of his own choice, both before and after the challenge ciphertext. Thus, the applicability of currently known FE for inner products is still limited. Therefore, we ask the following question:

Can we further generalize the security framework and construct more efficient schemes of FE for inner products?

1.2 Our Results

Below we summarize our answers to the two questions in three folds as below.

1. For the question related to decentralized ABE, we first generalize the security notion of [42] by considering adaptive corruption of parties (in addition to making adaptive key queries). Then we construct a new scheme for $\{0,1\}$-LSSS (a class that captures monotone boolean formula) with the building block – functional encryption for inner products (with a stronger security requirement). Our scheme is in the plain model and the security holds against bounded collusion.
2. We formalize this requirement and instantiate two schemes – one by LWE, and the other by DDH. Our constructions make *essential modifications* of the constructions by [6], and we improve the analysis significantly (especially for the LWE-based construction), resulting in more efficient schemes with stronger security.

3. We show that decentralized ABE for general access structures is somewhat equivalent to witness encryption (WE) for general NP relations. This can be viewed as a challenge in achieving decentralized ABE based on standard assumptions, as we are not aware of any construction of WE for NP relations based on standard assumptions.

By putting (1) and (2) together, we achieve the following informal theorem:

Theorem 1.1 (Informally Stated) *Assume the DDH or LWE assumption. For the function class of* $\{0,1\}$-LSSS*, there exists a decentralized ABE scheme that is secure against adversary who can make adaptive key queries and adaptively corrupt parties, with bounded collusions.*

Our LWE-based construction provides another path to construct decentralized ABE that is potentially secure against quantum computers as long as LWE is quantum hard. Next we compare our DDH construction with that of the prior work [42]. First, both schemes achieve the function class $\{0,1\}$-LSSS. Second, our scheme achieves stronger security against adaptive corruptions of parties, yet the work [42] achieves security against static security where the adversary needs to commit to a subset of corrupted parties at the beginning of the security experiment. Third, our scheme only relies on the DDH assumption without the need of pairings, yet the work [42] requires three new assumptions on bilinear groups. Finally, the work [42] can support an unbounded number of collusions by using random oracle, yet in the plain model, their scheme can only support a bounded number of collusions. On the other hand, our scheme works natively in the plain model and supports a bounded number of collusions. We leave it as an interesting open question whether our scheme can be upgraded in the random oracle model.

1.3 Our Techniques

Decentralized ABE. In this work, one focus is to construct decentralized ABE, following the direction of the prior work [22, 42]. We first briefly review the setting. In a decentralized ABE system, anyone can become an authority by simply creating a public key and issuing secret keys to different users for their attributes, without coordinating with (or even being aware of) other authorities. Similarly as in [22, 42], we use the concept of global identifiers (GID) to link together secret keys issued by different authorities. Ciphertexts are generated corresponding to some access structure over attributes. The decryptor can recover the message if he holds secret keys from different attributes that have the same GID and satisfy the access structure specified by the ciphertext. In the security model, the adversary can corrupt authorities and query authorities for attributes associated with GID adaptively, with the restrictions that the information learned from these collusion cannot help adversary decrypt challenge ciphertext. For the bounded collusion setting, the number of GID queried by adversary is fixed according to the scheme parameter.

To present our intuition, we first consider a very simple case and then present how we can generalize the idea. Let us assume that there is only one known GID,

and there are three parties P_1, P_2, P_3, where each P_i holds only one attribute i. In this case, constructing a decentralized ABE is simple. Each P_i just samples $(\mathsf{pk}_i, \mathsf{sk}_i)$ from a regular encryption scheme, and outputs pk_i as the master public key and keeps sk_i as the master secret key. To issue a key for the attribute i, the party P_i just simply outputs sk_i. To encrypt a message m, the encryptor first secret share $(w_1, w_2, w_3) \leftarrow \mathsf{Share}(m)$ (according to its access structure), and outputs $\mathsf{Enc}_{\mathsf{pk}_1}(w_1), \mathsf{Enc}_{\mathsf{pk}_2}(w_2), \mathsf{Enc}_{\mathsf{pk}_3}(w_3)$ as the ciphertext. Intuitively, if the decryptor holds a set of keys $\{\mathsf{sk}_j\}_{j\in S}$ where S satisfies the access structure, then the decryptor can obtain $\{w_j\}_{j\in S}$ and recover the original message m. On the other hand, if S does not satisfy the structure, then by the security of the secret sharing scheme, the adversary cannot learn any information about m.

To generalize the idea to a larger GID domain, we consider a *new* secret sharing that takes shares over polynomials.[4] Particularly, we let $p_0(x) = m$ be a constant degree polynomial with the coefficient m. The encryptor now shares $(p_1(x), p_2(x), p_3(x)) \leftarrow \mathsf{Share}(p_0(x))$, and outputs the ciphertext as $\Big(\mathsf{Enc}_{\mathsf{pk}_1}(p_1(x)), \mathsf{Enc}_{\mathsf{pk}_2}(p_2(x)), \mathsf{Enc}_{\mathsf{pk}_3}(p_3(x))\Big)$. Suppose we can achieve the following properties:

1. P_i can issue a secret key $\mathsf{sk}_{i,\mathsf{GID}}$ such that whoever holds the key can learn $p_i(\mathsf{GID})$.
2. Suppose $\Big(\{p_i(\mathsf{GID}_1)\}_{i\in S_1}, \{p_i(\mathsf{GID}_2)\}_{i\in S_2}, \ldots, \{p_i(\mathsf{GID}_t)\}_{i\in S_t}\Big)$ is given for distinct $\mathsf{GID}_1, \ldots, \mathsf{GID}_t$.
   (a) If there exists some S_j that satisfies the access structure, then one can recover $p_0(\mathsf{GID}_j) = m$.
   (b) If no such S_j exists, then $p_0(x) = m$ remains hidden.

Then it is not hard to see that the scheme also achieves the decentralized ABE, as P_i can issue $\mathsf{sk}_{i,\mathsf{GID}}$ as Property 1, the decryption works by Property 2(a), and intuitively, security is guaranteed by Property 2(b).

Now we elaborate on how to implement the properties in more details. First, we can see that functional encryption (FE) is exactly what we need for Property 1, and in fact, FE for inner products suffices for the functionality. The encryption algorithm can encrypt $\boldsymbol{y} = (c_0, c_1, \ldots, c_k)$ that represents the polynomial $p(x) = \sum_{i=0}^{k} c_i x^i$, and a key for GID can be set as the FE key $\mathsf{sk}_{\boldsymbol{x}}$ for $\boldsymbol{x} = (1, \mathsf{GID}, \mathsf{GID}^2, \ldots, \mathsf{GID}^k)$. By using the FE decryption with the secret key $\mathsf{sk}_{\boldsymbol{x}}$, one can learn $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = p(\mathsf{GID})$. To implement Property 2, we find that we can apply the known $\{0,1\}$-LSSS sharing scheme [13] over the coefficients of the polynomial, and prove Properties 2(a) and 2(b). If the shares of polynomials are of degree k, then we can tolerate up to $t = k - 1$ distinct GID queries. See our Section 4 for further details.

We notice that any FE for inner products can achieve the functionality of Property 1 as stated above, yet connecting security of the FE and security of the decentralized ABE is not obvious. First, the (challenge) messages used in

[4] We will discuss the secret sharing in more details, but let us focus on the high level concepts at this point.

the decentralized ABE come from a distribution (i.e., shares from $\mathsf{Share}(p_0(x))$ form a distribution), yet the IND-based security considered by prior work [1, 6, 16] focuses on two fixed challenge messages $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$. It is not clear how to define two fixed $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ to capture two distributions of $\mathsf{Share}(p_0(x))$ and $\mathsf{Share}(q_0(x))$ in the decentralized ABE setting, and furthermore, is not clear how to define *admissible* key queries in our setting. SIM-based FE might be helpful, but it is impossible to achieve the notion (for challenge ciphertexts that encrypt a vector of messages) if the adversary is allowed to make post-challenge ciphertext key queries, as pointed out by [6]. We also note that the scheme of Gorbunov, Vaikuntanathan, and Wee [31] cannot be applied to our setting directly, as their SIM adaptive security only holds for challenge ciphertexts that encrypt a single-message.[5] Second, in our decentralized ABE, the adversary is allowed to corrupt parties and make key queries adaptively, i.e., at any time of the game. It is not clear whether the currently functional encryption schemes are secure under adaptive corruption if several ciphertexts under different public keys are given first and then several master secret keys are compromised. Consider the following example: suppose $\mathsf{Enc}_{\mathsf{pk}_1}(\boldsymbol{y}_1), \mathsf{Enc}_{\mathsf{pk}_2}(\boldsymbol{y}_2), \mathsf{Enc}_{\mathsf{pk}_3}(\boldsymbol{y}_3)$ are given first, and then the adversary can corrupt any subset, say sk_1 and sk_2, and make key query to sk_3, what security can we guarantee for the remaining message $\boldsymbol{y}_3$?

Functional Encryption for Inner Products. To handle the issues above, we consider a more generalized security notion of functional encryption. Intuitively, our framework considers encryption over a set of messages from two (challenge) distributions, say $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell) \leftarrow \mathcal{D}_b^\ell$ for $b \in \{0, 1\}$, under different public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$, and the ciphertexts $\big(\mathsf{Enc}_{\mathsf{pk}_1}(\boldsymbol{y}_1), \ldots, \mathsf{Enc}_{\mathsf{pk}_\ell}(\boldsymbol{y}_\ell)\big)$ are given to the adversary. The adversary can make (1) a corruption query to any sk_i and (2) a key query $\boldsymbol{x}$ to an uncorrupted sk_j, multiple times before or after the challenge phase. Our security requires that the adversary cannot guess b correctly, as long as the distributions $\mathcal{D}_0^\ell$ and $\mathcal{D}_1^\ell$ remain indistinguishable under the functionalities from (1) and (2). This security notion lies in between the IND-based security and the SIM-based security. We prove that any functional encryption that satisfies the notion can be used to implement the idea above to build a secure decentralized ABE. We present the details in Section 4.

Next we turn to the question how to build such a functional encryption scheme. We make *essential* modifications of the DDH and LWE-based constructions from the work [6], and prove that the modified schemes achieve our new security definition. Conceptually, we develop two new techniques: (1) we use a complexity leverage argument (or random guessing [36]) in a clever way that does not affect of the underlying LWE or DDH assumption *at all*. (2) Our LWE-based construction uses a re-randomized technique proposed in the work [37] to avoid the use of multi-hint extended LWE as required by the work [6]. The reduction from multi-hint LWE to LWE incurs a significant security loss, and

[5] The work [31] can derive an IND adaptively secure scheme for challenge ciphertexts that encrypt a vector of messages, but as we discussed above, IND security seems not sufficient for our application.

the standard deviation required by the discrete Gaussian distribution is large. By using our new analysis, we are able to have a direct security proof of the scheme without multi-hint LWE, resulting in an exponential improvement over the parameters. Below we elaborate on more details. As we improve over some subtle but important points of the work [6], for the next paragraph we assume some familiarity of the work [6].

We briefly review the approach of [6]. The master public/secret keys have the form $\mathsf{mpk} = (\mathbf{A}, \mathbf{U})$ and $\mathsf{msk} = \mathbf{Z}$ such that $\mathbf{U} = \mathbf{ZA}$. The ciphertext has the form $\mathsf{Enc}(\boldsymbol{y}) = (c_0, c_1)$ where $c_0 = \mathbf{A}\boldsymbol{s} + \boldsymbol{e}_0$, and $c_1 = \mathbf{U}\boldsymbol{s} + \boldsymbol{e}_1 + K\boldsymbol{y}$ for some appropriate number K. The security proof of [6] proceeds as follows:

- Let H_0 be the original game.
- Hybrid H_1: c_0 remains the same, and $c_1 = \mathbf{Z}(c_0 - \boldsymbol{e}_0) + \boldsymbol{e}_1 + K\boldsymbol{y}$.
- Hybrid H_2: c_0 is switched to the uniform vector, and c_1 remains the same as H_1.

It is quite easy to see that H_1 is just a rephrase of H_0, so the two hybrids are identical. The difference between H_1 and H_2 relies on the multi-hint extended LWE, as we need the hint of $\mathbf{Z}\boldsymbol{e}_0$ in order to simulate c_1 given c_0. Then Agrawal et al. [6] showed that in H_2, the adversary has a negligible winning probability with an information-theoretic argument. This is to say, in H_2, even a computationally unbounded adversary cannot win the game with better than a negligible probability.

To get rid of the use of multi-hint extended LWE, we modify the hybrid 1:

- New H_1': c_0 remains the same as H_0, and $c_1 = \mathsf{ReRand}(\mathbf{Z}, \boldsymbol{c}_0, \alpha q, \sigma^*) + K\boldsymbol{y}$ for some $\alpha q, \sigma^*$.

The algorithm ReRand was proposed in the work by Katsumata et al. [37]. We show that this technique can be used to improve analysis in our setting: by setting $\alpha q, \sigma^*$ appropriately, the output distribution of the ReRand will be statistically close to $\mathbf{U}\boldsymbol{s} + \boldsymbol{e}^*$ where $\boldsymbol{e}^*$ has the same distribution of $\boldsymbol{e}_1$. Consequently, c_1 can be generated independent of $\boldsymbol{e}_0$, and thus we can get rid of the need of the multi-hint LWE.

To prove security of our setting, we need to analyze the success probability of an adversary in H_2. We observe that the proof technique by the work [6] cannot be applied to our setting. Intuitively, the crux of their security proof (in H_2) relies on the following facts: (1) once the adversary submits the challenge messages $\boldsymbol{y}_0, \boldsymbol{y}_1$, the space of key queries in the remaining game is fixed to $\Lambda^\perp(\boldsymbol{y}_0 - \boldsymbol{y}_1)$. (2) The adversary cannot distinguish $\boldsymbol{y}_0$ from $\boldsymbol{y}_1$ even if he is given the ciphertext and a set of keys in each dimension of $\Lambda^\perp(\boldsymbol{y}_0 - \boldsymbol{y}_1)$ at the same time. (This is captured as $\mathbf{X}_{top}\mathbf{Z}$ in their proof.) (3) Any post-challenge key queries can be derived by a linear combination of the keys obtained in (2), i.e, $\mathbf{X}_{top}\mathbf{Z}$. In our setting however, the fact (1) no longer holds. Given two message distributions, it is not clear whether the space for the remaining key queries is even fixed or not. Therefore, their argument cannot be used in our setting.

Another possible way to handle adaptive queries is to use a complexity leveraging argument (or random guessing according to the work [36]). However, by

folklore we know that naively applying the argument will result in an exponential security loss, i.e., $\epsilon_{\mathsf{scheme}} \leq 2^{\lambda} \cdot \epsilon_{\mathsf{LWE}}$. Our new insight to tackle this problem is to apply it cleverly: we only apply the argument in the hybrid H_2 where all the analysis is information-theoretic. In more details, we show that in H_2 the advantage of any adversary who only makes pre-challenge ciphertext key queries is bounded by some ϵ_2, and by the complexity leveraging argument, the advantage of a full adversary is bounded by $2^{\lambda} \cdot \epsilon_2$. By setting the parameters appropriately, we can afford the loss without affecting the hardness of the underlying LWE or DDH assumption. Our overall advantage of the adversary would be $\epsilon_{\mathsf{scheme}} \leq \Delta(H_0, H_1) + \Delta(H_1, H_2) + \mathbf{Adv}(H_2)$. By the property of ReRand, $\Delta(H_0, H_1)$ is negligible; by the security of LWE, $\Delta(H_1, H_2) \leq \epsilon_{\mathsf{LWE}}$; by the above argument $\mathbf{Adv}(H_2) = 2^{\lambda} \cdot \epsilon_2$ can also be set to negligible. Therefore, we have $\epsilon_{\mathsf{scheme}} \leq \epsilon_{\mathsf{LWE}} + \mathsf{negl}(\lambda)$. Details can be found in the full version of the paper.

Can We Achieve Decentralized ABE for General Functions? After achieving decentralized ABE for $\{0,1\}$-LSSS, it is natural to ask whether we can do more. Here we show that any decentralized ABE for general functions implies a witness encryption (WE) for general NP statements. On the other hand, an extractable witness encryption for general NP statements plus signatures implies decentralized ABE, following the argument of the work [29]. The result provides a challenge to construct decentralized ABE for general functions under standard assumptions, as we are not aware of any construction of WE from standard assumptions. We leave it as an interesting open question whether there exists a decentralized ABE for a class between $\{0,1\}$-LSSS and general functions.

1.4 Additional Related Work

Decentralized ABE. The problem of building ABE with multiple authorities was proposed by Sahai and Waters [54], and first considered by Chase [22]. In [22], Chase introduced the concept of using a global identifier to link secret keys together. However, her system relies on a central authority and is limited to express a strict AND policy over a *pre-determined* set of authorities. Müller et al. [47,48] proposed another construction with a centralized authority for any LSSS structure, based on [55], but their construction is secure only against non-adaptive queries. Lin et al. [44] showed a threshold based scheme (somewhat decentralized) against bounded collusions. In their system, the set of authorities is fixed ahead of time, and they must interact during system setup. Chase and Chow [23] showed how to remove the central authority using a distributed PRF, but the restriction of an AND policy over a pre-determined set of authorities remained. In [42], Lewko and Waters proposed a decentralized ABE system for any LSSS structure from bilinear groups. Their system is secure against adaptive secret key queries and selective authority corruption in random oracle model. Liu et al. [45] proposed a fully secure decentralized ABE scheme in standard model, but there exists multiple central authorities issuing identity-related keys to users.

Functional Encryption for Inner Products. The problem of FE for inner products was first considered by Abdalla et al. [1], where they show constructions against selective adversaries based on standard assumptions, like DDH and LWE. Later on, Bishop et al. [12] consider the same functionality in the secret-key setting with the motivation of achieving function privacy and security against adaptive adversaries. Recently, in work by Agrawal et al. [6], they provide constructions in public key setting based on several standard assumptions that are secure against more realistic adaptive adversaries, where challenge messages are declared in the challenge phase, based on previously collected information. Benhamouda et al. [10] show a CCA-Secure Inner-Product Functional Encryption generically from projective hash functions with homomorphic properties.

For the multi-input version of the inner product functionality, more recently, Abdalla et al. [2] show a construction of multi-input functional encryption scheme (MIFE) for the inner products functionality based on the k-linear assumption in prime-order bilinear groups, which is secure against adaptive adversaries. In [24], Datta et al. describe two non-generic constructions based on bilinear groups of prime order, where one construction can withstand an arbitrary number of encryption slots.

Witness Encryption. Recently, Brakerski et al. [19] proposed a new framework to construct WE via ABE. We note that a result similar to our construction can be obtained from their work.

1.5 Roadmap

The notations and some preliminaries are described in Section 2. In Section 3, we present our new security definition for FE, and propose an LWE based construction satisfying our new security. Due to the space limitation, we defer the full security proof to the full version of the paper. In Section 4, we give a stronger security definition for decentralized ABE and present our construction. Furthermore, we explore the relationship between decentralized ABE and (extratable) witness encryption in Section 5. The DDH-based constructions for FE and decentralized ABE can be found in the full version of the paper.

2 Preliminaries

Notations. We use λ to denote security parameter throughout this paper. For an integer n, we write $[n]$ to denote the set $\{1, \ldots, n\}$. We use bold lowercase letters to denote vectors (e.g. $\boldsymbol{v}$) and bold uppercase letters for matrices (e.g. $\mathbf{A}$). For a vector $\boldsymbol{v}$, we let $\|\boldsymbol{v}\|$ denote its ℓ_2 norm. The ℓ_2 norm of a matrix $\mathbf{R} = \{\boldsymbol{r}_1, ..., \boldsymbol{r}_m\}$ is denoted by $\|\mathbf{R}\| = \max_i \|\boldsymbol{r}_i\|$. The spectral norm of $\mathbf{R}$ is denoted by $\mathsf{s}_1(\mathbf{R}) = \mathsf{sup}_{\boldsymbol{x} \in \mathbb{R}^{m+1}} \|\mathbf{R} \cdot \boldsymbol{x}\|$.

We say a function $\mathsf{negl}(\cdot) : \mathbb{N} \to (0, 1)$ is negligible, if for every constant $c \in \mathbb{N}$, $\mathsf{negl}(n) < n^{-c}$ for sufficiently large n. For any set X, we denote by $\mathcal{P}(X)$ as the power set of X. For any $Y, Z \in \{0,1\}^n$, we say that $Y \subseteq Z$ if for

each index $i \in [n]$ such that $Y_i = 1$, We have $Z_i = 1$. The statistical distance between two distributions X and Y over a countable domain D is defined to be $\frac{1}{2}\sum_{d\in D} |X(d) - Y(d)|$. We say that two distributions are statistically close if their statistical distance is negligible in λ.

A family of functions $\mathcal{H} = \{h_i : D \to R\}$ from a domain D to range R is called k-wise independent, if for every pairwise distinct $x_1, ..., x_k \in D$ and every $y_1, ..., y_k \in R$,

$$\mathsf{Pr}_{h\leftarrow\mathcal{H}}[h(x_1) = y_1 \wedge ... \wedge h(x_k) = y_k] = 1/|R|^k.$$

Secret Sharing and the $\{0,1\}$-LSSS Access Structure. We briefly describe the syntax of secret sharing and the $\{0,1\}$-LSSS access structure, and refer the full version of the paper for further details. A secret sharing scheme consists of two algorithms as follow: $\mathsf{SS} = (\mathsf{SS.Share}, \mathsf{SS.Combine})$. The share algorithm $\mathsf{SS.Share}$ takes input a secret message k and an access structure $\mathbb{A}$ and output a set of shares $s_1, \ldots, s_t$. The combine algorithm $\mathsf{SS.Combine}$ takes input a subset of shares can recover the secret k if the subset satisfies the access structure $\mathbb{A}$. If not, then the secret k should remain hidden to the algorithm. Briefly speaking, if the combine algorithm just applies a linear combination over shares to recover the message, then the secret sharing scheme is called *linear*, or LSSS in brief. If the coefficients are in $\{0,1\}$, then it is called $\{0,1\}$-LSSS. It is worthwhile pointing out that the $\{0,1\}$-LSSS contains a powerful class called Monotone Boolean Formula (MBF) pointed out by the work [13, 41], who showed that any MBF can be expressed as an access structure in $\{0,1\}$-LSSS. In this work, our construction of decentralized ABE achieves the class of $\{0,1\}$-LSSS and thus supports any the class of MBF.

3 Adaptively Secure FE for Chosen Message Distributions

In this section, we define a new security notion of functional encryption, called adaptively secure functional encryption for chosen message distributions. This notion is a generalization of prior adaptively secure functional encryption [6]. We first propose the definition, and then construct an LWE-based scheme that achieves the security notion. Our construction modified the scheme of [6] in an essential way, and our security analysis provides significantly better parameters than the work [6]. The DDH-based construction and its security proof can be analyzed in a similar way as our LWE-based scheme.

3.1 New Security Definition

In functional encryption, a secret key sk_g is associated with a function g, and a ciphertext ct_x is associated with some input x from the domain of g. The functionality of FE requires that the decryption procedure outputs $g(x)$, while security guarantees that nothing more than $g(x)$ can be learned. The formal description of syntax is in the full version of the paper.

Suppose that in a functional encryption scheme, a set of messages can be chosen from two distributions, and we obtain a set of ciphertexts by encrypting each message y_i using different master public keys mpk_i. Before choosing the two message distributions, the adversary is sent a set of master public keys $\{\mathsf{mpk}_i\}_{i\in[t]}$ and can also make two kinds of queries:

- Function queries: For query (f, i), obtain secret key sk_i^f for function f from msk_i.
- Opening queries: For query i, obtain master secret key msk_i.

The natural restrictions we enforce here are (1) the distributions of queried function evaluations for the two message distributions remains indistinguishable, (2) the distributions of opening messages are also indistinguishable. Otherwise, there can be no security as the adversary can trivially distinguish the two message distributions. On the other hand, other additional information such as, the opening messages, queried keys, and the function values, may help the adversary to learn to distinguish the message distributions from the ciphertexts. Our new security notion – *adaptively secure functional encryption for chosen message distributions*, requires that the choice of the message distribution of challenger remains indistinguishable even if the adversary is given the additional information.

We formalize IND-based security definition with respect to *admissible mappings*. For ease of exposition, we first define the query mappings.

Definition 3.1 *Let $t = t(\lambda)$ be an integer and $\mathcal{M}$ be the message space. $\{x_i\}_{i\in[t]} \in \mathcal{M}^t$ is a set of messages, and $f : \mathcal{M} \to \mathcal{K}$ be a function. We define the functions $(i, f) : \mathcal{M}^t \to \mathcal{K}$ as $(i, f)(x_1, ..., x_t) = f(x_i)$, and function $(i, I) : \mathcal{M}^t \to \mathcal{M}$ as $(i, I)(x_1, ..., x_t) = x_i$.*

Definition 3.2 (Admissible mappings) *Let $t = t(\lambda)$ be an integer, $\mathcal{M}$ be the message space, and $\mathcal{M}_0, \mathcal{M}_1$ be two distributions over space $\mathcal{M}^t$. Let subsets $T_1, T_2 \subsetneq [t]$ such that $T_2 \cap T_1 = \emptyset$ and $|T_2 \cup T_1| < t$, and let $\{k_i\}_{i\in T_2}$ be a set of integers. We say that mappings $\{(i, I)\}_{i\in T_1}$ and $\{(i, f_{ij})\}_{i\in T_2, j\in[k_i]}$ are admissible if it holds that*

$$\begin{aligned}&\{\{(i, I)(\mathcal{M}_0)\}_{i\in T_1}, \{(i, f_{ij})(\mathcal{M}_0)\}_{i\in T_2, j\in[k_i]}\}\\ =&\{\{(i, I)(\mathcal{M}_1)\}_{i\in T_1}, \{(i, f_{ij})(\mathcal{M}_1)\}_{i\in T_2, j\in[k_i]}\}\end{aligned}$$

Remark 3.3 The requirement of *admissible mappings* is that the above two distributions are identical. We can also relax the definition by requiring the two distributions are statistically or computationally close.

We define the adaptive security of functional encryption for chosen message distributions through an experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{FE}}(1^\lambda, 1^t)$ between an adversary and challenger:

1. **Setup**: For $i \in [t]$, challenger first computes $(\mathsf{mpk}_i, \mathsf{msk}_i) \leftarrow \mathsf{Setup}(1^\lambda)$, then sends $\{\mathsf{mpk}_i\}_{i\in[t]}$ to adversary $\mathcal{A}$.

2. **Query Phase I**: Proceeding adaptively, adversary can make any polynomial number of queries to the oracle $\mathcal{O}(\{\mathsf{msk}_i\}_{i\in[t]}, \cdot)$ of the following two kinds:
   - Function queries (i, f_{ij}): Challenger sends back $\mathsf{sk}_{f_{ij}} \leftarrow \mathsf{KeyGen}(\mathsf{sk}_i, f_{ij})$.
   - Opening queries (i, I): Challenger sends back msk_i.
3. **Challenge Phase**: Adversary $\mathcal{A}$ sends two message distributions $\mathcal{M}_0$ and $\mathcal{M}_1$ over message space $\mathcal{M}^t$ with the restriction that any queries made in **Query Phase I** are *admissible* with respect to $(\mathcal{M}_0, \mathcal{M}_1)$ (c.f. Definition 3.2). The challenger chooses a random bit $b \in \{0, 1\}$, and sends ciphertext $\{\mathsf{ct}_i = \mathsf{Enc}(\mathsf{mpk}_i, x_i)\}_{i\in[t]}$ back to adversary, where $\{x_i\}_{i\in[t]} \leftarrow \mathcal{M}_b$.
4. **Query Phase II**: Adversary $\mathcal{A}$ can continue making queries as specified in **Query Phase I** as long as the queries are admissible.
5. **Guess**: Adversary $\mathcal{A}$ outputs his guess b'.

We define the advantage of adversary $\mathcal{A}$ in the experiment $\mathbf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 1^t)$ as

$$\mathbf{Adv}_{\mathcal{A}}(1^\lambda, 1^t) = |\mathsf{Pr}[\mathbf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 1^t) = 1] - 1/2|$$

Definition 3.4 *We say a functional encryption scheme Π is adaptively secure for chosen message distributions security if for any polynomial $t = t(\lambda)$, and any* PPT *adversary $\mathcal{A}$, we have $\mathbf{Adv}_{\mathcal{A}}(1^\lambda, 1^t) \leq \mathsf{negl}(\lambda)$.*

3.2 Functional Encryption for Inner Products Modulo p

Agrawal et al. [6] show a construction of functional encryption for inner products modulo p assuming the hardness of LWE problem. In this section, we made some *important* modifications of their construction, particularly the encryption and key generation algorithms, and then show that the modified scheme satisfies our new security definition. Our modifications and new analysis provide significantly better parameters as we will discuss below. We first present the construction:

- $\mathsf{Setup}(1^n, 1^\ell, 1^k, p)$: Set integers $m, q = p^e$ for some integer e, and real numbers $\alpha, \alpha' \in (0, 1)$. Randomly sample matrices $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ and $\mathbf{Z}_i \xleftarrow{\$} \mathbb{Z}_p^{\ell\times m}$, for $i = 1, ..., k$. Compute $\mathbf{U}_i = \mathbf{Z}_i \cdot \mathbf{A} \in \mathbb{Z}_q^{\ell\times n}$. Output $\mathsf{mpk} := (\mathbf{A}, \{\mathbf{U}_i\}_{i\in[k]})$ and $\mathsf{msk} := (\{\mathbf{Z}_i\}_{i\in[k]})$.
- $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{x})$: On input a vector $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$, where for each $i \in [k]$, $\boldsymbol{x}_i \in \mathbb{Z}_p^\ell$, compute the secret key $\boldsymbol{z}_{\boldsymbol{x}}$ as follows. As $\boldsymbol{x}$ is linearly independent from the key queries have been made so far modulo p, we can compute $\boldsymbol{z}_{\boldsymbol{x}} = \sum_{i=1}^k \boldsymbol{x}_i^T \cdot \mathbf{Z}_i$, and output secret key $\mathsf{sk}_{\boldsymbol{x}} = \boldsymbol{z}_{\boldsymbol{x}}$.
- $\mathsf{Enc}(\mathsf{mpk}, \boldsymbol{y})$: On input $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_k)$, where for each $i \in [k]$, $\boldsymbol{y}_i \in \mathbb{Z}_p^\ell$, sample $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n, \boldsymbol{e}_0 \leftarrow \mathcal{D}^m_{\mathbb{Z},\alpha q}$ and $\{\boldsymbol{e}_i\}_{i\in[k]} \leftarrow \mathcal{D}^\ell_{\mathbb{Z},\alpha' q}$ and compute

$$\boldsymbol{c}_0 = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}_0 \in \mathbb{Z}_q^m, \quad \boldsymbol{c}_i = \mathbf{U}_i \cdot \boldsymbol{s} + \boldsymbol{e}_i + p^{e-1} \cdot \boldsymbol{y}_i \in \mathbb{Z}_q^\ell, \forall i \in [k]$$

  Then, output $\mathsf{ct} = (\boldsymbol{c}_0, \{\boldsymbol{c}_i\}_{i\in[k]})$.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\boldsymbol{x}}, \mathsf{ct})$: On input $\mathsf{ct} = (\boldsymbol{c}_0, \{\boldsymbol{c}_i\}_{i\in[k]})$ and a secret key $\mathsf{sk}_{\boldsymbol{x}} = \boldsymbol{z}_{\boldsymbol{x}}$ for $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathbb{Z}_p^{k\ell}$, compute $\mu' = \sum_{i=1}^k \langle \boldsymbol{x}_i, \boldsymbol{c}_i \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle$ and output the value $\mu \in \mathbb{Z}_p$ that minimizes $|p^{e-1} \cdot \mu - \mu'|$.

Decryption correctness. Correctness derives from the following equation:

$$\mu' = \sum_{i=1}^{k} \langle \boldsymbol{x}_i, \boldsymbol{c}_i \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{c}_0 \rangle = p^{e-1} \cdot \left(\sum_{i=1}^{k} \langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \bmod p \right) + \sum_{i=1}^{k} \langle \boldsymbol{x}_i, \boldsymbol{e}_i \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle \bmod q$$

If the magnitude of error term $\sum_{i=1}^{k} \langle \boldsymbol{x}_i, \boldsymbol{e}_i \rangle - \langle \boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{e}_0 \rangle$ is $\leq p^{e-1}/2$ with overwhelming probability, then the correctness holds with overwhelming probability.

Parameters setting. The parameters in Table 1 are selected in order to satisfy the following constraints. In the table below, $e, c_1, c_2, \delta_1, \delta_2, \delta$ are constants, and $\delta = \delta_1 + \delta_2$

Parameters	Description	Setting
λ	security parameter	
n	column of public matrix	λ
m	row of public matrix	$n^{1+\delta}$
p	modulus of inner products	n^{c_1}
e	power of q to p	$> 3 + (\frac{7\delta}{2} + \frac{c_2}{2} + 2)/c_1$
q	modulus of LWE	$n^{c_1 e}$
αq	Gaussian parameter of $\boldsymbol{e}_0$	$\sqrt{n^{c_2+2\delta} \cdot \log n}$
$\alpha' q$	Gaussian parameter of $\boldsymbol{e}_i$	$n^{1+\delta+c_1} \cdot \sqrt{n^{c+2\delta} \cdot \log n}$
t	number of distinct mpk's	n^{c_2}
k	number of $\boldsymbol{x}_i$ in $\boldsymbol{x}$	n^{δ_1}
ℓ	dimension of $\boldsymbol{x}_i$	n^{δ_2}
σ^*	parameter of ReRand algorithm	pm

Table 1. Parameter Description and Simple Example Setting

- To ensure correctness of decryption, we require $p^{e-1} > 2kp^2 m\ell\alpha q(2\sqrt{\ell} + \sqrt{m})$.
- To ensure the correctness of ReRand algorithm, we require $\sigma^* \geq pm$.
- By the property of ReRand algorithm, we have $\alpha' q = 2\sigma^* \alpha q$.
- To ensure small enough reduction loss for the ReRand algorithm, we require $\alpha q > \sqrt{\lambda + tk^2\ell^2 \log p}$.
- To ensure large enough entropy required by Claim (in full version), we require $m \geq 2k\ell + ek\ell(n+1) + 3\lambda$.

Comparison with the Work [4, 6]. Our analysis shows that the scheme can support a much wider range of parameters than the analysis of Agrawal et al. [6]. For their analysis, the efficiency degrade quickly when the dimension increases, and in particular, the modulus $q \geq p^{\ell}$. This is why the work [6] sets the dimension $\ell = \Omega(\log n)$. In our analysis, we can build a direct reduction to LWE (without using the intermediate extended LWE), allowing us to choose $\mathbf{Z}_i \leftarrow \mathbf{U}(Z_p^{\ell \times m})$ (instead of using a discrete Gaussian with a very large deviation). This gives us

a significant improvement over the parameters: our modulus q does not depend on ℓ in an exponential way, so we can set the dimension to any fixed polynomial, without increasing q significantly.

A subsequent work [4] improved the parameters significantly, yet with a tradeoff of weaker security where the adversary can only receive $\mathsf{sk}_{\boldsymbol{x}}$ for random $\boldsymbol{x}$'s before the challenge ciphertext and cannot issue more key queries afterwards. Their scheme [4] is useful in the setting of designing trace-and-revoke systems, but cannot be applied to the decentralized ABE where the adversary can obtain keys of his own choice, both before and after the challenge ciphertext.

Security Proof. Now we can show the following theorem that under the parameters above, the functional encryption for inner product scheme described above is adaptively secure for chosen message distributions. Due to space limit, we defer the full proof to the full version of the paper.

Theorem 3.5 *Under the* LWE *assumption, the above functional encryption for inner products is adaptively secure for chosen message distributions, assuming for each* msk_i*, the secret key queries to the* msk_i *are linearly independent.*

Remark 3.6 The functionality of the scheme described above is inner products modulo a prime p. In [6], the authors have given an attack for the case that the secret key queries are not linearly independent modulo p but linearly independent over the integers, and they proposed a stateful key generation technique to get rid of the attack. Here we can also use a stateful key generation algorithm to remove the last assumption (i.e., linearly independent queries) in the theorem.

4 Decentralized ABE: Stronger Definition and Construction

In this section, we first describe the syntax of decentralized ABE, following the work [41], and then we define a stronger security notion. Next, we present our construction and security proof, relying on the functional encryption scheme in Section 3.2. We first present a basic scheme that supports smaller GID and message spaces (Section 4.2) , and next we show an improved scheme that supports significantly larger spaces (Section 4.3).

4.1 Syntax of Decentralized ABE Scheme and Stronger Security

We first recall the syntax of decentralized ABE as defined in [41] and then present a stronger security definition. Let $\mathcal{F}$ be a function class. A decentralized ABE for $\mathcal{F}$ consists of the following algorithms:

- $\mathsf{Global.Setup}(1^\lambda) \to \mathsf{GP}$ The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system.
- $\mathsf{Authority.Setup}(\mathsf{GP}) \to (\mathsf{pk}, \mathsf{sk})$ Each authority runs the authority setup algorithm with GP as input to produce its own secret key and the public key pair $(\mathsf{sk}, \mathsf{pk})$.

- $\mathsf{Enc}(\mu, f, \mathsf{GP}, \{\mathsf{pk}\}) \to \mathsf{ct}$ Encryption algorithm takes as inputs a message μ, a function $f \in \mathcal{F}$, a set of public keys for relevant authorities, and the global parameters, and outputs a ciphertext ct.
- $\mathsf{KeyGen}(\mathsf{GID}, \mathsf{GP}, i, \mathsf{sk}) \to \mathsf{k}_{i,\mathsf{GID}}$ The key generation algorithm takes as inputs an identity GID, global parameters GP, an attribute i, and secret key sk for this authority who holds the attribute. It produces a key $\mathsf{k}_{i,\mathsf{GID}}$ for this attribute-identity pair.
- $\mathsf{Dec}(\mathsf{ct}, \mathsf{GP}, \{\mathsf{k}_{i,\mathsf{GID}}\}) \to \mu$ The decryption algorithm takes as inputs the global parameters GP, a ciphertext ct, and a set of keys $\{\mathsf{k}_{i,\mathsf{GID}}\}$ corresponding to attribute-identity pairs. It outputs a message μ, if the set of attributes i satisfies the policy specified by f and all the identities have the same GID. Otherwise, it outputs $\perp$.

Definition 4.1 (Correctness) *We say a decentralized ABE scheme is correct if for any* $\mathsf{GP} \leftarrow \mathsf{Global.Setup}(1^\lambda)$*,* $f \in \mathcal{F}$*, message* μ*, and* $\{\mathsf{k}_{i,\mathsf{GID}}\}$ *obtained from the key generation algorithm for the same identity* GID *where the attributes satisfy the policy* f*, we have*

$$\mathsf{Pr}[\mathsf{Dec}(\mathsf{Enc}(\mu, f, \mathsf{GP}, \{pk\}), \mathsf{GP}, \{\mathsf{k}_{i,\mathsf{GID}}\}) = \mu] = 1 - \mathsf{negl}(\lambda).$$

Security Definition. We define the notion of *full* security of decentralized ABE schemes. In our setting, the adversary can *adaptively* corrupt authorities, as well as making *adaptive* key queries. In a similar but weaker model defined in [41], the adversary can make adaptive key queries but only *static* corruption queries, i.e., the adversary can only corrupt parties before the global parameter is generated.

Let $t = \mathsf{poly}(\lambda)$ denote the number of authorities, and we consider parties $P_1, P_2, \ldots, P_t$, where each party P_i holds an attribute i. Then we define the security notion via an experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{dabe}}(1^\lambda, 1^t)$ between an adversary and the challenger:

1. Setup: The challenger runs $\mathsf{GP} \leftarrow \mathsf{Global\ Setup}(1^\lambda)$, and then $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Authority\ Setup}(\mathsf{GP})$ for $i \in [t]$. Then the challenger sends $\left(\mathsf{GP}, \{\mathsf{pk}_i\}_{i\in[t]}\right)$ to the adversary, and keeps $\{\mathsf{sk}_i\}_{i\in[t]}$ secretly.
2. Key Query Phase 1: Proceeding adaptively, adversary can make the two types of queries:
   (a) Secret key query (i, GID): $\mathcal{A}$ submits a pair (i, GID) to the challenger, where i is an attribute belonging to an uncorrupted authority P_i and GID is an identity. The challenger runs $\mathsf{k}_{i,\mathsf{GID}} \leftarrow \mathsf{KeyGen}(\mathsf{GID}, \mathsf{GP}, i, \mathsf{sk}_i)$ and forwards the adversary $\mathsf{k}_{i,\mathsf{GID}}$.
   (b) Corruption query (i, corr): $\mathcal{A}$ submits (i, corr) to the challenger, where i is an attribute that the adversary want to corrupt. The challenger responds by giving $\mathcal{A}$ the corresponding master secret key msk_i
3. Challenge Phase: $\mathcal{A}$ specify two messages μ_0, μ_1, and a function f, where function satisfies the following constraint. We let $\boldsymbol{\omega}_c$ denote the attributes controlled by the corrupted authorities, and for each GID we let $\boldsymbol{\omega}_{\mathsf{GID}}$ denote the attributes which the adversary has queried. We require that $f(\boldsymbol{\omega}_c, \boldsymbol{\omega}_{\mathsf{GID}}) \neq 1$

(In other words, the adversary does not hold a set of keys that allow decryption). The challenger flips a random coin $b \in \{0,1\}$ and sends the adversary an encryption of μ_b under f.

4. Key Query Phase 2: The adversary can further make corruption and key queries as the Key Query Phase 1, under the constraint of f as specified above.
5. Guess: The adversary submits a guess bit b', and wins if $b' = b$. The advantage of adversary in the experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{dabe}}(1^\lambda, 1^t)$ is defined as $\mathbf{Adv}_{\mathcal{A}}(1^\lambda) = |\mathsf{Pr}[b^{'} = b] - 1/2|$.

Definition 4.2 *A decentralized ABE scheme is fully secure if for any* PPT *adversary $\mathcal{A}$, we have $\mathbf{Adv}_{\mathcal{A}}(1^\lambda) \leq \mathsf{negl}(\lambda)$. The scheme is fully secure against k-bounded collusion if we further require that $\mathcal{A}$ can query at most k distinct* GID*'s in the experiment.*

4.2 Our Basic Construction

In the description here, we first present our basic construction of decentralized ABE for $\{0,1\}$-LSSS. The basic construction can only support $\mathsf{GID} \in \mathbf{GF}(p)$ for some fixed prime p, and the message space is also $\mathbf{GF}(p)$. Next we show how to extend the GID domain to $\mathbf{GF}(p^\ell)$ by using the field extension technique.

Our construction uses the following building blocks: (1) a $\{0,1\}$-LSSS scheme SS, and (2) a fully secure functional encryption for inner product modulo p, denoted as FEIP. We can instantiate the $\{0,1\}$-LSSS as definition in [13], and the FEIP as the construction in Section 3.2. Then we define our construction $\Pi = (\mathsf{Global.Setup}, \mathsf{Authority.Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ as follows:

- $\mathsf{Global.Setup}(1^\lambda)$: On input security parameter λ, the global setup algorithm sets $k = k(\lambda)$ to denote the collusion bound of the scheme and $t = t(\lambda)$ to denote the number of associated attributes. The global setup algorithm also sets an integer $n = n(\lambda)$ and a prime number $p = p(\lambda)$. It outputs $\mathsf{GP} = (k, t, n, p)$ as the global parameter.
- $\mathsf{Authority.Setup}(\mathsf{GP})$: On input GP, for any attribute i belonged to the authority, the authority runs the algorithm $\mathsf{FEIP.Setup}(1^n, 1^\ell, 1^k, p)$ to generate $\mathsf{FEIP.mpk}_i$ and $\mathsf{FEIP.msk}_i$. Then output $\mathsf{pk} = \{\mathsf{FEIP.mpk}_i\}$ as its public key, and keep $\mathsf{sk} = \{\mathsf{FEIP.msk}_i, \forall i\}$ as its secret key. (In the basic scheme, ℓ is set to 1.)
- $\mathsf{Enc}(\mu, (\mathbf{A}, \rho), \mathsf{GP}, \{\mathsf{pk}\})$: On input a message $\mu \in \mathbb{Z}_p$, an access matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times d}$ with ρ mapping its row number x to attributes, the global parameters GP, and the public keys $\{\mathsf{FEIP.mpk}_i\}$ of the relevant authorities. The encryption algorithm invokes k times $\{0,1\}$-LSSS for secret space $\mathcal{K} = \mathbb{Z}_p$ to generate:

$$(u_{1,1}, ...u_{1,t}) \leftarrow \mathsf{SS.Share}(\mu, \mathbf{A}),$$

$$(u_{i,1}, ...u_{i,t}) \leftarrow \mathsf{SS.Share}(0, \mathbf{A}), \forall i \in [2, k].$$

  For each $(u_{1,x}, ..., u_{k,x}) \in \mathbb{Z}_p^k, x \in [t]$ it generates

$$\mathsf{FEIP.ct}_{\rho(x)} \leftarrow \mathsf{FEIP.Enc}(\mathsf{FEIP.mpk}_{\rho(x)}, (u_{1,x}, ..., u_{k,x})).$$

The ciphertext is $\mathsf{ct} = (\{\mathsf{FEIP.ct}_{\rho(x)}\}_{x=1,\ldots t})$.

– $\mathsf{KeyGen}(\mathsf{GID}, i, \mathsf{sk}, \mathsf{GP})$: On inputs attribute i, global identifier $\mathsf{GID} \in \mathbb{Z}_p$, secret key sk and global parameter GP, the algorithm sets $\mathbf{GID} = (1, \mathsf{GID}, ..., \mathsf{GID}^{k-1})$ and computes

$$\mathsf{FEIP.sk}_{i,\mathsf{GID}} \leftarrow \mathsf{FEIP.KeyGen}(\mathsf{FEIP.msk}_i, \mathbf{GID}),$$

and outputs $\mathsf{k}_{i,\mathsf{GID}} = \mathsf{FEIP.sk}_{i,\mathsf{GID}}$.

– $\mathsf{Dec}(\{\mathsf{k}_{i,\mathsf{GID}}\}, \mathbf{A}, \mathsf{ct}, \mathsf{GP})$: On input secret keys $\{\mathsf{k}_{\rho(x),\mathsf{GID}}\}$, the access matrix $\mathbf{A}$, ciphertext ct and global parameter GP, the decryptor first checks if $(1, 0, \ldots, 0)$ is in the span of the rows $\{\mathbf{A}_x\}$. If not, the algorithm outputs $\perp$. Otherwise, it computes

$$\eta_{\rho(x)} = \mathsf{FEIP.Dec}\left(\mathsf{FEIP.sk}_{\rho(x),\mathsf{GID}}, \mathsf{FEIP.ct}_{\rho(x)}\right) \text{ for each } \rho(x),$$

and outputs $\eta = \mathsf{SS.Combine}(\{\eta_{\rho(x)}\})$.

Remark 4.3 We note that for any distinct k GID's, $\mathsf{GID}_1, \ldots, \mathsf{GID}_k$, the vectors $\left\{\mathbf{GID}_i = (1, \mathsf{GID}_i, ..., \mathsf{GID}_i^{k-1})\right\}_{i\in[k]}$ are linearly independent. In our construction above, each $\mathsf{GID} \in \mathbb{Z}_p$ and the vectors can be expressed as a Vandermonde matrix $\mathbf{X} = \begin{bmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_k \\ \vdots & \ddots & \vdots \\ x_1^{k-1} & \ldots & x_k^{k-1} \end{bmatrix}$, which is full-rank if the elements $\{x_i\}_{i\in[k]}$ are distinct. Therefore, for any less than k distinct GID's, the key queries for these GID's are linearly independent.

Correctness. We show that the scheme above is correct. By correctness of the fully secure functional encryption scheme FEIP, we have that for each $\rho(x)$,

$$\eta_{\rho(x)} = u_{1,\rho(x)} + \sum_{j=1}^{k-1} u_{j+1,\rho(x)} \mathsf{GID}^j \bmod p.$$

Since $(u_{1,1}, ...u_{1,t})$ is secret sharing of message μ, and $\{(u_{i,1}, ...u_{i,t})\}_{i=2}^{k}$ is secret sharing of 0, then by correctness of $\{0,1\}$-LSSS scheme, we have that $\eta = \mu \bmod p$. This proves correctness.

Parameter Setting. We can instantiate the scheme FEIP (c.f. Section 3.2) by setting $n = \lambda$, $\ell = 1$, $k = \mathsf{poly}(\lambda)$, $t = \mathsf{poly}(\lambda)$ and $p = \mathsf{poly}(\lambda)$, and obtain a decentralized ABE with both the message space and GID space being $\mathbb{Z}_p$. In summary, our basic scheme can support any fixed $\mathsf{poly}(\lambda)$ GID and message spaces, against any fixed $\mathsf{poly}(\lambda)$ bounded collusion.

Security Proof. Next we prove security of the above scheme in the following theorem.

Theorem 4.4 *Assuming that* FEIP *is a functional encryption for inner products and* FEIP *is adaptively secure for chosen message distributions, and* SS *is a* $\{0,1\}$-LSSS*, the decentralized ABE construction Π described above is fully secure against $k-1$ bounded collusion.*

Proof. We prove the theorem by reduction. Assume that there exists an adversary $\mathcal{A}$ who breaks the scheme with some non-negligible advantage ϵ, then we can construct a reduction $\mathcal{B}$ that breaks the security of FEIP. Given an adversary $\mathcal{A}$, we define $\mathcal{B}$ as follows:

1. $\mathcal{B}$ first receives $\{\mathsf{mpk}_i\}$ from its challenger, and forwards $\{\mathsf{mpk}_i\}$ to $\mathcal{A}$.
2. $\mathcal{B}$ runs $\mathcal{A}$ to simulate the Key Query Phase 1 of the ABE security game. In each round, $\mathcal{B}$ may receive either a corruption query (i, I) or a key query (GID, i).
   - Upon receiving a query (GID, i), $\mathcal{B}$ makes a key query $\left((1, \mathsf{GID}, ..., \mathsf{GID}^k), i\right)$ to its challenger, and receives $\mathsf{k}_{i,\mathsf{GID}}$. $\mathcal{B}$ forward $\mathcal{A}$ with the key.
   - Upon receiving a query (i, corr), $\mathcal{B}$ make a query (i, I) to the challenger and receives msk_i. $\mathcal{B}$ just sends $\mathcal{A}$ msk_i.

   $\mathcal{B}$ continue to run this step until $\mathcal{A}$ makes the challenge query.
3. Upon receiving $\mathcal{A}$'s challenge query, which contains an access structure $\mathbb{A}$ and two messages $\mu_0, \mu_1 \in \mathbb{Z}_p$, $\mathcal{B}$ first checks whether all the key queries satisfy the constraint of the security game of decentralized ABE. (This can be efficiently checked in our setting). If not, $\mathcal{B}$ aborts the game and outputs a random guess. Otherwise, $\mathcal{B}$ defines two distributions $\mathcal{M}_0$ and $\mathcal{M}_1$ as follows. For $b \in \{0,1\}$, $\mathcal{M}_b$ is defined as the distribution that first samples k times of the $\{0,1\}$-LSSS procedure

$$(u_{1,1}^{(b)}, ...u_{1,t}^{(b)}) \leftarrow \mathsf{SS.Share}(\mu_b, \mathbf{A})$$

$$(u_{i,1}^{(b)}, ...u_{i,t}^{(b)}) \leftarrow \mathsf{SS.Share}(0, \mathbf{A}), \forall i \in [2, k].$$

   Then $\mathcal{M}_b$ outputs:

$$\left((u_{1,1}^{(b)}, ..., u_{k,1}^{(b)}), ..., (u_{1,t}^{(b)}, ..., u_{k,t}^{(b)})\right).$$

   $\mathcal{B}$ sends the descriptions of $\mathcal{M}_0, \mathcal{M}_1$ (which can be succinctly described, e.g., $(\mu_b, \mathbf{A})$) to the challenger, and then $\mathcal{B}$ forwards $\mathcal{A}$ the challenge ciphertext received from the external FEIP challenger.
4. $\mathcal{B}$ simulates the Key Query Phase 2 in the same way as Step 2.
5. Finally $\mathcal{B}$ outputs $\mathcal{A}$'s guess b'.

Next we are going to analyze the reduction $\mathcal{B}$. Since $\mathcal{B}$ perfectly emulates the FEIP security game for $\mathcal{A}$, $\mathcal{B}$'s advantage is the same as $\mathcal{A}$'s, assuming the queries are admissible in the FEIP security game. Therefore, it suffices to prove the theorem by showing that the queries made by $\mathcal{B}$ are admissible.

The assumption of the theorem requires that $\mathcal{A}$ can query at most $k-1$ different GID's for each msk_i. Let T_1 be the set that $\mathcal{B}$ (and also $\mathcal{A}$ as well)

makes corruption queries, and k_i be the number of secret key queries that $\mathcal{B}$ makes to msk_i. Then we need to show that the two distributions defined below are identical, i.e., $\mathcal{D}_0 = \mathcal{D}_1$, where

$$D_b = \left\{ \{(i,I)(\mathcal{M}_b)\}_{i\in T_1}, \{(i,\boldsymbol{x}_{ij})(\mathcal{M}_b)\}_{i\in T_2, j\in[k_i]} \right\},$$

where $\boldsymbol{x}_{ij} = (1, \mathsf{GID}_{ij}, ...\mathsf{GID}_{ij}^{k-1})$, T_1 is the set $\mathcal{B}$ corrupts, and T_2 is the set that $\mathcal{B}$ makes key queries but does not corrupt.

We note that, for an opening query (i,I), $(i,I)(\mathcal{M}_b) = \left(u_{1,i}^{(b)}, ..., u_{k,i}^{(b)}\right)$ can be viewed as coefficients of the polynomial $P_i(x) = u_{1,i}^{(b)} + \sum_{j=2}^{k} u_{j,i}^{(b)} \cdot x^{j-1} \mod p$). By the property of Lagrange interpolation formula, the coefficients of a polynomial $P(x)$ of degree k can be uniquely determined given $P(x_1), \ldots, P(x_k)$ for any distinct $(x_1, \ldots, x_k)$. This implies that $(i,I)(\mathcal{M}_b)$ can be simulated by $\{(i, \mathbf{GID}_{ij})(\mathcal{M}_b)\}_{j\in[k]}$ for any distinct $\{\mathsf{GID}_{ij}\}_{j\in[k]}$, where $\mathbf{GID} = (1, \mathsf{GID}, \ldots, \mathsf{GID}^{k-1})$. Therefore, it is without loss of generality to assume that D_b only contains information of the form $\{(i, \mathbf{GID}_{ij})(\mathcal{M}_b)\}$.

As we argue above, all queries are of the form $(i, \boldsymbol{x}_{ij})$, so we can re-write the queries in D_b as $\{q_1, \ldots, q_n\}$, where each q_j is of the form $(i, \boldsymbol{x})$. Denote $\{q_1, \ldots, q_n\}$ as $\overrightarrow{\boldsymbol{q}}$, and then we can further re-write D_b as $\overrightarrow{\boldsymbol{q}}(\mathcal{M}_b)$. Now it suffices to show that for every admissible $\overrightarrow{\boldsymbol{q}}$ and possible values $\boldsymbol{z}$,

$$\mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_0) = \boldsymbol{z}] = \mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_1) = \boldsymbol{z}].$$

$\mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_b) = \boldsymbol{z}]$ can be expanded as

$$\begin{aligned} &\mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_b) = \boldsymbol{z}] \\ =&\mathsf{Pr}[q_1(\mathcal{M}_b) = z_1] \cdot \mathsf{Pr}[q_2(\mathcal{M}_b) = z_2 | q_1(\mathcal{M}_b) = z_1] \cdot \cdot \\ &\mathsf{Pr}[q_n(\mathcal{M}_b) = z_n | q_1(\mathcal{M}_b) = z_1 \wedge ... \wedge q_{n-1}(\mathcal{M}_b) = z_{n-1}]. \end{aligned}$$

We first observe that the message distribution $\mathcal{M}_b = \left((u_{1,1}^{(b)}, ..., u_{k,1}^{(b)}), ..., (u_{1,t}^{(b)}, ..., u_{k,t}^{(b)})\right)$ can be viewed as coefficients of t degree-k polynomials $(P_1(x), \ldots, P_t(x))$. Since the marginal distribution of $(u_{1,i}^{(b)}, ..., u_{k,i}^{(b)})$ is uniformly random by the $\{0,1\}$-LSSS property, the marginal distribution of any polynomial P_i is uniform, i.e., a random degree-k polynomial in $\mathbb{Z}_p$. Therefore, $\mathsf{Pr}[q_1(\mathcal{M}_b) = x_1] = 1/p$, independent of b.

Next we will show that for any $j \in [n]$,

$$\mathsf{Pr}\left[q_j(\mathcal{M}_b) = z_j \mid q_{j-1}(\mathcal{M}_b) = z_{j-1} \wedge \cdots \wedge q_1(\mathcal{M}_b) = z_1\right] = 1/p,$$

assuming $q_1, \ldots, q_j$ are admissible.

To prove this, we first set up some notations. We assume that $\mathsf{GID}_1, \ldots, \mathsf{GID}_{k-1}$ are the identifiers queried by the adversary. If the adversary corrupts some msk_i, then he will further learn $P_i(\mathsf{GID}_k)$ (for another GID_k) in addition to

$P_i(\mathsf{GID}_1), \ldots, P_i(\mathsf{GID}_{k-1})$. We assume that $q_j = (i, \mathbf{GID}_r)$ for some $i \in [t], r \in [k]$. Let S be an arbitrary maximal invalid set that includes all v's with $(v, \mathbf{GID}_r)$ belongs to the queries $\{q_1, \ldots, q_j\}$, according to the access structure $\mathbb{A}$, i.e. $\{v : (v, \mathbf{GID}_r) \in \{q_1, \ldots, q_j\}\} \subseteq S$. Since the queries are admissible, such a set S always exists.

By the privacy guarantee of the LSSS, we know the distributions of the polynomials $P_1(x), \ldots, P_t(x)$ generated in the encryption algorithm is identical to the following process:

- For every $v \in S$, sample $P_v(x)$ (the coefficients) uniformly and independently at random.
- For every $v \notin S$, set $P_v(x) = \mu_b - \sum_{w \in \Gamma_v} P_w(x)$, where $\Gamma_v \subseteq S$ is the reconstruction set that can be efficiently determined given $(v, \mathbb{A})$.

Next, we observe the following facts:

1. Since $P_i(x)$ is a random polynomial (the marginal distribution), we know that the (marginal) distribution $P_i(\mathsf{GID}_r)$ is uniformly random even conditioned on all $\{P_i(\mathsf{GID}_w)\}_{w \in [k] \setminus \{r\}}$ (as a random degree k polynomial is k-wise independent).
2. From the above sampling procedure, we know that $P_i(x)$ is independent of $\{P_v(x)\}_{v \in S \setminus \{i\}}$.
3. From the above two facts, we know that the (marginal) distribution $P_i(\mathsf{GID}_r)$ is still uniformly random even further conditioned on $\{P_v(x)\}_{v \in S \setminus \{i\}}$ and $\{P_i(\mathsf{GID}_w)\}_{w \in [k] \setminus \{r\}}$.
4. For every $v \notin S$, $w \in [k] \setminus \{r\}$, $P_v(\mathsf{GID}_w)$ can be deterministically obtained given the information $\{P_v(x)\}_{v \in S \setminus \{i\}}$ and $\{P_i(\mathsf{GID}_w)\}_{w \in [k] \setminus \{r\}}$. This implies that the conditional distribution $P_i(\mathsf{GID}_r)$ is uniform even further given $\{P_v(\mathsf{GID}_w)\}_{v \in [t] \setminus S, w \in [k] \setminus \{r\}}$ in addition to $\{P_v(x)\}_{v \in S \setminus \{i\}}$ and $\{P_i(\mathsf{GID}_w)\}_{w \in [k] \setminus \{r\}}$.

It is not hard to see that the information of $q_1(\mathcal{M}_b), \ldots, q_{j-1}(\mathcal{M}_b)$ can be obtained given $\{P_v(\mathsf{GID}_w)\}_{v \in [t] \setminus S, w \in [k] \setminus \{r\}}$, $\{P_v(x)\}_{v \in S \setminus \{i\}}$, and $\{P_i(\mathsf{GID}_w)\}_{w \in [k] \setminus \{r\}}$. Therefore, we have showed: for any $j \in [n]$,

$$\Pr\left[q_j(\mathcal{M}_b) = z_j \mid q_{j-1}(\mathcal{M}_b) = z_{j-1} \wedge \cdots \wedge q_1(\mathcal{M}_b) = z_1\right] = 1/p.$$

Then we can conclude that

$$\begin{aligned}
&\mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_0) = \boldsymbol{x}] \\
=&\mathsf{Pr}[q_1(\mathcal{M}_0) = x_1] \cdot \mathsf{Pr}[q_2(\mathcal{M}_0) = x_2 | q_1(\mathcal{M}_0) = x_1] \cdot \cdots \\
&\mathsf{Pr}[q_n(\mathcal{M}_0) = x_n | q_1(\mathcal{M}_0) = x_1 \wedge \ldots \wedge q_{n-1}(\mathcal{M}_0) = x_{n-1}] \\
=&\mathsf{Pr}[q_1(\mathcal{M}_1) = x_1] \cdot \mathsf{Pr}[q_2(\mathcal{M}_1) = x_2 | q_1(\mathcal{M}_1) = x_1] \cdot \cdots \\
&\mathsf{Pr}[q_n(\mathcal{M}_1) = x_n | q_1(\mathcal{M}_1) = x_1 \wedge \ldots \wedge q_{n-1}(\mathcal{M}_1) = x_{n-1}] \\
=&\mathsf{Pr}[\overrightarrow{\boldsymbol{q}}(\mathcal{M}_1) = \boldsymbol{x}].
\end{aligned}$$

This proves that all the queries $\mathcal{B}$ makes during the game are admissible. This means that $\mathcal{B}$ is a legal adversary in the security game of FEIP. Since $\mathcal{B}$ also perfect simulates the challenger of $\mathcal{A}$, the advantage of $\mathcal{B}$ is the same as the advantage of $\mathcal{A}$, a non-negligible quantity. This reaches a contraction, and completes the proof. □

4.3 An Improved Construction for Large Spaces

We can modify our basic construction so that it can support significantly lager GID and message spaces, using the technique of finite field extension to $\mathbf{GF}(p^\ell)$ for some ℓ. In more detail, we consider the embedding technique described in the work [46, 57]. Intuitively, we can compute $\mathbf{GF}(p^\ell)$ field operations via projecting $\mathbf{GF}(p^\ell)$ elements to $\mathbb{Z}_p^\ell$ (and $\mathbb{Z}_p^{\ell\times\ell}$), and thus, the field operations can be supported by our FEIP scheme.

Let $p \in \mathbb{N}$ be a prime, $\ell \in \mathbb{N}$, and let $f(x)$ be a monic irreducible polynomial in $\mathbb{Z}_p$ of degree ℓ. Then we define $R = \mathbb{Z}_p[X]/\langle f(x)\rangle$, and note that R is isomorphic to $\mathbf{GF}(p^\ell)$ as p is a prime and $f(x)$ is an irreducible polynomial of degree ℓ. We will use R as the representation of $\mathbf{GF}(p^\ell)$.

We then define two mappings $\phi : R \to \mathbb{Z}_p^\ell$ and $\mathsf{Rot} : R \to \mathbb{Z}_p^{\ell\times\ell}$ by

$$\phi : \theta = a_1 + a_2 x + ... + a_\ell x^{\ell-1} \mapsto (a_1, ..., a_\ell)^\top,$$

$$\mathsf{Rot} : \theta = a_1 + a_2 x + ... + a_\ell x^{\ell-1} \mapsto \left[\phi(\theta)\phi(\theta x)...\phi(\theta x^{\ell-1})\right].$$

We note that $\mathsf{Rot}(\theta)\cdot\phi(\vartheta) = \phi(\theta\vartheta)$, $\mathsf{Rot}(\theta)\cdot\mathsf{Rot}(\vartheta) = \mathsf{Rot}(\theta\vartheta)$, and $\mathsf{Rot}(\theta) + \mathsf{Rot}(\vartheta) = \mathsf{Rot}(\theta+\vartheta)$. This means that Rot is a ring-homomorphism from R to $\mathbb{Z}_p^{\ell\times\ell}$. If $\theta \neq \theta' \in \mathbf{GF}(p^\ell)$, then $\mathsf{Rot}(\theta) - \mathsf{Rot}(\theta') = \mathsf{Rot}(\theta - \theta') \neq 0$.

Now we present our modified construction $\Pi = (\mathsf{Global.Setup}, \mathsf{Authority.Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

- $\mathsf{Global.Setup}(1^\lambda)$: On input the security parameter λ, the global setup algorithm sets $k = k(\lambda)$ to denote the collusion bound of our scheme and $t = t(\lambda)$ to denote the number of associated attributes. The global setup algorithm also sets $n = n(\lambda)$, $\ell = \ell(\lambda)$ and $p = p(\lambda)$ to denote the input parameters of $\mathsf{FEIP.Setup}$. It outputs $\mathsf{GP} = (k, t, n, \ell, p)$ as the global parameter, and sets both the GID and message spaces as $\mathbf{GF}(p^\ell)$.
- $\mathsf{Authority.Setup}(\mathsf{GP})$: On input GP, for attribute i belonged to the authority, the authority runs the algorithm $\mathsf{FEIP.Setup}(1^n, 1^\ell, 1^k, p)$ to generate $\mathsf{FEIP.mpk}_i$ and $\mathsf{FEIP.msk}_i$. Then output $\mathsf{pk} = \{\mathsf{FEIP.mpk}_i\}$ as its public key, and keep $\mathsf{sk} = \{\mathsf{FEIP.msk}_i, \forall i\}$ as its secret key.
- $\mathsf{Enc}(\mu, (\mathbf{A}, \rho), \mathsf{GP}, \{\mathsf{pk}\})$: On input a message $\mu \in \mathbf{GF}(p^\ell)$, an access matrix $\mathbf{A} \in \mathbf{GF}(p^\ell)^{t\times d}$ with ρ mapping its row number x to attributes, the global parameters GP, and the public keys $\{\mathsf{FEIP.mpk}_i\}$ of the relevant authorities. The encryption algorithm invokes k times $\{0,1\}$-LSSS over secret space $\mathbf{GF}(p^\ell)$ to generate

  $$(u_{1,1}, ...u_{1,t}) \leftarrow \mathsf{SS.Share}(\mu, \mathbf{A}),$$

  $$(u_{i,1}, ...u_{i,t}) \leftarrow \mathsf{SS.Share}(0, \mathbf{A}), \forall i \in [2, k],$$

  For each $(u_{1,x}, ..., u_{k,x}) \in \mathbf{GF}(p^\ell)^k, x \in [t]$, the encryption algorithm first computes:

  $$(\boldsymbol{u}_{1,x}, ..., \boldsymbol{u}_{k,x}) \leftarrow \phi(u_{1,x}, ..., u_{k,x}),$$

and then generates

$$\mathsf{FEIP.ct}_{\rho(x)} \leftarrow \mathsf{FEIP.Enc}(\mathsf{FEIP.mpk}_{\rho(x)}, (\boldsymbol{u}_{1,x}, ..., \boldsymbol{u}_{k,x})).$$

The ciphertext is $\mathsf{ct} = (\{\mathsf{FEIP.ct}_{\rho(x)}\}_{x=1,...t})$.

– $\mathsf{KeyGen}(\mathsf{GID}, i, \mathsf{sk}, \mathsf{GP})$: On input attribute i, global identifier $\mathsf{GID} \in \mathbf{GF}(p^\ell)$, secret key sk and global parameters GP. To generate a key for GID for attribute i belonging to an authority, the authority first computes k elements $\mathsf{GID}^j \in \mathbf{GF}(p^\ell), \forall j \in [k-1]$, then computes $\mathsf{Rot}(\mathsf{GID}^j), \forall j \in [k-1]$, and denotes the column vectors of $\begin{bmatrix} \mathbf{I} \\ \mathsf{Rot}(\mathsf{GID}) \\ \vdots \\ \mathsf{Rot}(\mathsf{GID}^{k-1}) \end{bmatrix}$ to be $\{\boldsymbol{g}_j\}_{j\in\ell}$, where $\mathbf{I}$ is the identity matrix in $\mathbb{Z}_p^{\ell\times\ell}$, finally sets

$$\mathsf{FEIP.sk}_{i,\mathsf{GID}}^{(j)} \leftarrow \mathsf{FEIP.KeyGen}\left(\mathsf{FEIP.msk}_i, \boldsymbol{g}_j, \mathsf{rand}_i\right), \forall j \in [\ell].$$

Outputs $\mathsf{k}_{i,\mathsf{GID}} = \{\mathsf{FEIP.sk}_{i,\mathsf{GID}}^{(j)}\}_{j\in[\ell]}$.

– $\mathsf{Dec}(\{\mathsf{k}_{i,\mathsf{GID}}\}, \mathbf{A}, \mathsf{ct}, \mathsf{GP})$: On input secret keys $\{\mathsf{k}_{\rho(x),\mathsf{GID}}\}$, the access matrix $\mathbf{A}$, ciphertext ct and global parameters GP, the decryptor first checks if $(1, 0, \ldots, 0)$ is in the span of the rows $\{\mathbf{A}_x\}$ or not. If not, the algorithm outputs $\perp$. Otherwise, it computes

$$\eta_{\rho(x)}^{(j)} = \mathsf{FEIP.Dec}\left(\mathsf{FEIP.sk}_{\rho(x),\mathsf{GID}}^{(j)}, \mathsf{FEIP.ct}_{\rho(x)}\right), \forall j \in [\ell], \text{ for each } \rho(x),$$

and sets $\boldsymbol{\eta}_{\rho(x)} = \left(\eta_{\rho(x)}^{(1)}, ..., \eta_{\rho(x)}^{(\ell)}\right)$, $\theta_{\rho(x)} = \phi^{-1}(\boldsymbol{\eta}_{\rho(x)})$, then outputs $\theta = \mathsf{SS.Combine}(\{\theta_{\rho(x)}\})$.

Correctness. By correctness of the scheme FEIP, we have that for each $\rho(x)$,

$$\begin{aligned}\boldsymbol{\eta}_{\rho(x)} &= \boldsymbol{u}_{1,\rho(x)} + \sum_{j=1}^{k-1} \boldsymbol{u}_{j+1,\rho(x)} \mathsf{Rot}(\mathsf{GID}^j) (\text{mod } p) \\ &= \phi(u_{1,\rho(x)} + \sum_{j=1}^{k-1} u_{j+1,\rho(x)} \mathsf{GID}^j).\end{aligned}$$

Then $\theta_{\rho(x)} = u_{1,\rho(x)} + \sum_{j=1}^{k-1} u_{j+1,\rho(x)} \mathsf{GID}^j$. By correctness of the $\{0,1\}$-LSSS scheme over $\mathbf{GF}(p^\ell)$, we have that $\theta = \mu$. This proves correctness.

Parameters. We can instantiate the scheme FEIP (c.f. Section 3.2) by setting $n = \lambda$, $\ell = \mathsf{poly}(\lambda)$ $k = \mathsf{poly}(\lambda)$, $t = \mathsf{poly}(\lambda)$ and $p = \mathsf{poly}(\lambda)$, and obtain a decentralized ABE with both the message space and GID space being $\mathbf{GF}(p^\ell)$. In summary, our modified scheme can support *exponential-sized* GID and message spaces, against any fixed $\mathsf{poly}(\lambda)$ bounded collusion.

Security. Security of the modified scheme can be proven in the same way as our basic scheme, as the only difference is the underlying finite field. We note that the analysis that the distributions D_b are identical in the proof of Theorem 4.4 works for any underlying finite field (either $\mathbb{Z}_p$ or $\mathbf{GF}(p^\ell)$), so the analysis can be carried to the modified scheme straightforwardly. To avoid repetition, we just state the theorem as follow.

Theorem 4.5 *Assume that* FEIP *is a functional encryption for inner products that is adaptively secure for chosen message distributions, and* SS *is a* $\{0,1\}$-LSSS *over* $\mathbf{GF}(p^\ell)$*. Then the scheme* Π *above is a fully secure decentralized ABE against* $k-1$ *bounded collusion for* $\{0,1\}$-LSSS *over* $\mathbf{GF}(p^\ell)$*.*

Combining Theorem 4.5 and the instantiation by Theorem 3.5, we obtain the following corollary:

Corollary 4.6 *Assume the LWE assumption. Then there exits a decentralized ABE that is fully secure against* $k-1$ *bounded collusion for any polynomial* k*, for the function class* $\{0,1\}$-LSSS *over* $\mathbb{Z}_p$ *for any polynomial prime* p*. The scheme supports exponential-sized* GID *and message spaces.*

5 Witness Encryption and Decentralized ABE

In this section, we discuss the relation between decentralized ABE and witness encryption, which is introduced by Garg et al. [28]. We first recall the syntax of witness encryption and its security, after that we give a construction of witness construction for NP language using decentralized ABE for general circuits, and then show that extractable witness encryption implies decentralized ABE for general circuits.

5.1 Witness Encryption

We recall the syntax of WE introduced by Garg et al. [28], and also the extractability security defined by Goldwasser et al. [29]. A witness encryption scheme for an NP language L (with corresponding witness relation R) consists of algorithms $\Pi = (\mathsf{Enc}, \mathsf{Dec})$:

- $\mathsf{Enc}(1^\lambda, x, \mu)$: On input the security parameter λ, an unbounded-length string x, and a message $\mu \in \{0,1\}$, the encryption algorithm outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{ct}, w)$: On input a ciphertext and an unbounded-length string w, the decryption algorithm outputs a message μ or a special symbol $\perp$.

Definition 5.1 (Witness Encryption) *We say* Π *described above is a witness encryption, if it satisfies:*

- **Correctness**: *For any security parameter* λ*, any* $\mu \in \{0,1\}$*, and* $x \in L$ *such that* $R(x,w) = 1$*, we have that*

$$\mathsf{Pr}[\mathsf{Dec}(\mathsf{Enc}(1^\lambda, x, \mu), w) = \mu] = 1$$

– **Soundness Security**: *For any* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for any* $x \notin L$*, we have:*

$$|\mathsf{Pr}[\mathcal{A}(\mathsf{Enc}(1^\lambda, x, 0)) = 1] - \mathsf{Pr}[\mathcal{A}(\mathsf{Enc}(1^\lambda, x, 1)) = 1]| < \mathsf{negl}(\lambda)$$

Definition 5.2 (Extractable security) *A witness encryption scheme for an NP language* L *is secure if for all* PPT *adversary* $\mathcal{A}$*, and all poly* q*, there exist a* PPT *extractor* E *and a poly* p*, such that for all auxiliary inputs* z *and for all* $x \in \{0,1\}^*$*, the following holds:*

$$\begin{aligned}\Pr[b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{WE.Enc}(1^\lambda, x, b) : \mathcal{A}(x, \mathsf{ct}, z) = b] \geq 1/2 + 1/q(|x|) \\ \Rightarrow \Pr[E(x, z) = \omega : (x, \omega) \in R_L] \geq 1/p(|x|).\end{aligned}$$

5.2 Witness Encryption from Decentralized ABE for General Circuit

We first describe a transformation from witness encryption for NP languages from decentralized ABE for general circuits. Intuitively, the witness encryption can use the Decentralized ABE scheme in the following way: the general circuit f is used as the NP verifier such that that the decryptor can recover the message if he has the witness ω for the statement x satisfying $f(x, \omega) = 1$.

More specifically, given an NP language L, we present witness encryption scheme $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ for L as follows:

– $\mathsf{WE.Enc}(1^\lambda, x, \mu)$: The encryption algorithm takes input a string $x \in \{0,1\}^n$ (whose witness has length bounded by m) and message μ. Then the algorithm runs the following procedures:
  - It runs $\mathsf{Global.Setup}$ and $\mathsf{Authority.Setup}$ to generate a global parameters GP and public keys $\{\mathsf{pk}_i\}_{i\in[n+m]}$ and secret keys $\mathsf{sk} = \{\mathsf{sk}_i\}_{i\in[n+m]}$.
  - Then it invokes KeyGen to generate $\{\mathsf{k}_{i,x_i}\}_{i\in[n]}$ and $\{\mathsf{k}_{j,0}, \mathsf{k}_{j,1}\}_{j=n+1}^{n+m}$.[6]
  - It sets $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ as the NP verifier for L that on input $x \in \{0,1\}^n, \omega \in \{0,1\}^m$ outputs 1 iff ω is a valid witness of x. Then it generates $\mathsf{ct} \leftarrow \mathsf{Enc}(\mu, f, \mathsf{GP}, \{\mathsf{pk}_i\}_{i\in[n+m]})$.
  - Finally, it outputs $\mathsf{ct} = \big(x, \{\mathsf{k}_{i,x_i}\}_{i\in[n]}, \{\mathsf{k}_{j,0}, \mathsf{k}_{j,1}\}_{j=n+1}^{n+m}, \{\mathsf{pk}_i\}_{i\in[n+m]}, \mathsf{ct}\big)$.

– $\mathsf{WE.Dec}(1^\lambda, \omega, \mathsf{ct})$: The decryption algorithm takes input a witness $\omega \in \{0,1\}^m$ for the statement $x \in \{0,1\}^n$ and a ciphertex ct for x. Then the algorithm runs the following procedures:
  - It first checks if $f(x, \omega) = 1$ holds, if not, the decryption algorithm outputs $\perp$.
  - Otherwise, for $j = n+1, \ldots, n+m$, the decryption algorithm chooses k_{j,ω_t} from $\{\mathsf{k}_{j,0}, \mathsf{k}_{j,1}\}_{j=n+1}^{n+m}$ (where $\omega_i \in \{0,1\}$ is the i-th bit of ω). Then it outputs

$$\mu = \mathsf{Dec}\left(\mathsf{ct}, \mathsf{GP}, \left\{\{\mathsf{k}_{i,x_i}\}_{i\in[n]}, \{\mathsf{k}_{j,\omega_t}\}_{j=n+1}^{n+m}\right\}\right).$$

[6] In our setting, we consider a general case where there is no GID.

Correctness of the witness encryption scheme is straightforward from the correctness of decentralized ABE scheme. Now we can show the following theorem. Due to space limit, we defer the full proof to the full version of the paper.

Theorem 5.3 *Assuming that Π is a secure decentralized ABE scheme for general circuits (against 1-bounded corruption, static corruption of authorities and selective key queries), the witness encryption scheme above is secure.*

Remark 5.4 Bellare et al. [9] has introduced a stronger security of WE which is denoted as adaptive soundness security. However, our construction can not achieve the stronger adaptive soundness security, because the NP language L we defined is not (efficiently) falsifiable.

Remark 5.5 We note that a weaker notion of decentralized ABE (where the adversary makes static corruption at the beginning of security game, and key queries only once) suffices to construct the witness encryption scheme. This demonstrates the hardness to construct decentralized ABE for general circuits.

Remark 5.6 The scheme we construct above makes use of a decentralized ABE scheme for $n + m$ authorities. We can also construct a WE scheme by invoking a decentralized ABE scheme for only two authorities. Intuitively, we set the attribute space as $\{0,1\}^n$. Then the NP statement $x \in \{0,1\}^n$ is the one attribute controlled by the non-corrupt authority, and the witness $\omega \in \{0,1\}^m$ of x is the one attribute controlled by the corrupt authority. We set f as the NP verifier algorithm. And the decryptor of WE scheme can recover the message if he can find the witness ω for x such that $f(x, \omega) = 1$. Then we can obtain the scheme similarly to the scheme above.

5.3 Decentralized ABE from Extractable Witness Encryption

Next, we show how to construct a decentralized ABE for general circuits from the following two components: (1) an extractable witness encryption scheme $\mathsf{WE} = (\mathsf{WE.Enc}, \mathsf{WE.Dec})$ [29], and (2) an existentially unforgeable signature scheme $\mathsf{SIG} = (\mathsf{SIG.KeyGen}, \mathsf{SIG.Sign}, \mathsf{SIG.Verify})$ [30].

In our construction, we assume that each authority P_i has a polynomial number of distinct attributes $S_i = \{x_j\}$. This is without loss of generality because we can always encode the party's ID in the first several bits of the attributes. Our construction $\Pi = (\mathsf{Authority\ Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ (we omit algorithm $\mathsf{Global\ Setup}$, as it does not affect the functionality) is described as follows:

- $\mathsf{Authority\ Setup}(1^\lambda)$ (for party P_j): On input security parameter λ, for each attribute x_i belonged to the authority P_j, i.e., $\boldsymbol{x}_i \in S_j$, the algorithm runs $\mathsf{SIG.KeyGen}(1^\lambda)$ to generate key pair $(\mathsf{svk}_{\boldsymbol{x}_i}, \mathsf{ssk}_{\boldsymbol{x}_i})$. Then it sets $\mathsf{pk} = \{\mathsf{svk}_{\boldsymbol{x}_i}\}_{\boldsymbol{x}_i \in S_j}$ as the public key, and keeps $\mathsf{sk} = \{\mathsf{ssk}_{\boldsymbol{x}_i}\}_{\boldsymbol{x}_i \in S_j}$ as its secret key.
- $\mathsf{Enc}(\{\mathsf{pk}\}, f, \mu)$: On input public key pk, a function f and message μ, the encryption algorithm sets an instance x_f as $x_f = (\{\mathsf{svk}_{\boldsymbol{x}_i}\}, f)$, and defines

NP language L such that $x_f \in L$ if and only if there exists n signature pairs $(\sigma_1, (\boldsymbol{x}_1, \mathsf{GID})), \ldots, (\sigma_n, (\boldsymbol{x}_n, \mathsf{GID}))$ such that

$$(\forall i, \mathsf{SIG.Verify}_{\mathsf{svk}_{\boldsymbol{x}_i}}(\sigma_i, (\boldsymbol{x}_i, \mathsf{GID})) = 1) \wedge (f(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) = 1)$$

Next it computes and outputs $\mathsf{ct} \leftarrow \mathsf{WE.Enc}(x_f, \mu)$.

- $\mathsf{KeyGen}(\boldsymbol{x}_j, \mathsf{GID}, \mathsf{sk}_i)$: On input attribute $\boldsymbol{x}_j$, the authority outputs $\mathsf{k}_{j,\mathsf{GID}} = \sigma_j \leftarrow \mathsf{SIG.Sign}(\mathsf{ssk}_i, (\boldsymbol{x}_j, \mathsf{GID}))$ if $\boldsymbol{x}_j \in S_i$. Otherwise, it outputs $\perp$.
- $\mathsf{Dec}(\{\mathsf{k}_{i,\mathsf{GID}}\}, \mathsf{GP}, \mathsf{ct})$: If the decryptor has a set of keys with the same GID such that $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = 1$, and all the signature verifications succeed, then $\{\mathsf{k}_{i,\mathsf{GID}}\}$ servers as witness for x_f, and it calls $\mathsf{WE.Dec}(\omega, \mathsf{ct})$ to recover the message μ. Otherwise, the decryption fails.

It is straightforward that the correctness of the scheme described above comes from the correctness of witness encryption WE and signature scheme SIG.

Next, we are going to show that the construction above achieves a ABE against static corruption. For convenience of our proof, we use the following presentation of definition for security against static corruption. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary, and T denote the set of authorities.

$\underline{\mathsf{Exp}^{\mathsf{dabe}}(1^\lambda)}$:

1. $(T', \{\mathsf{pk}_i, \mathsf{sk}_i\}_{i \in T'}) \leftarrow \mathcal{A}_1(1^\lambda)$
2. $\{\mathsf{pk}_j, \mathsf{sk}_j\}_{j \in T \setminus T'} \leftarrow \mathsf{Authority.Setup}(1^\lambda)$
3. $(f, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{KeyGen}(\mathsf{sk}_j, \cdot)}(\{\mathsf{pk}_k\}_{k \in [T]})$
4. Choose a bit b at random and let $\mathsf{ct} \leftarrow \mathsf{Enc}(\{\mathsf{pk}_k\}_{k \in [T]}, f, b)$
5. $b' \leftarrow \mathcal{A}_2^{\mathsf{KeyGen}(\mathsf{sk}_j, \cdot)}(\mathsf{state}, \mathsf{ct})$
6. If $b = b'$ and for all attributes $\boldsymbol{x}_{go}$ that $\mathcal{A}$ makes key requests to oracle $\mathsf{KeyGen}(\mathsf{sk}_j, .)$ along with the attributes $\boldsymbol{x}_{co}$ controlled by corrupted authorities ($\mathcal{A}$), we have $f(\boldsymbol{x}_{go}, \boldsymbol{x}_{co}) \neq 1$, output 1, else output 0.

We say that the scheme is secure (against static corruption of authorities) if for all PPT adversaries $\mathcal{A}$, the advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{dabe}}$ of $\mathcal{A}$ is negligible. where:

$$\mathbf{Adv}_{\mathcal{A}}^{\mathsf{dabe}} = |\Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{dabe}}(1^\lambda) = 1] - 1/2|.$$

Then we can show the following theorem. Due to space limit, we defer the full proof to the full version of the paper.

Theorem 5.7 *Assuming the existence of an extractable witness encryption scheme* WE *and an existentially unforgeable signature scheme* SIG*, then the scheme described above is secure against static corruption of authorities.*

6 Conclusion

We investigated the constructions of LWE-based and DDH-based decentralized ABE, which satisfy stronger security notion that adversary can make corruption queries of parties adaptively in addition to making adaptive key queries.

As a building block, we first introduced a functional encryption for inner product functionality with a stronger security requirement, and then we proposed the constructions of FE for inner product with the stronger security by making some modifications of the constructions by [6]. Combining the FE for inner product with the stronger security and a $\{0,1\}$-LSSS scheme, we obtained the constructions of the desired decentralized ABE. Finally, we showed that decentralized ABE for general access structures is somewhat equivalent to witness encryption (WE) for general NP relations.

Our scheme is in the plain model and the security holds against bounded collusion, the work [42] can support an unbounded number of collusions by using random oracle. We leave it as an interesting open question whether our scheme can be upgraded in the random oracle model.

Acknowledgements. We would like to thank Qiang Tang, Mingsheng Wang for their helpful discussions and suggestions. We also thank the anonymous reviewers of PKC 2019 for their insightful advices. Zhedong Wang is supported by the National Key R&D Program of China-2017YFB0802202. Xiong Fan is supported in part by IBM under Agreement 4915013672 and NSF Award CNS-1561209. Feng-Hao Liu is supported by the NSF Award CNS-1657040. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

1. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In Katz [38], pages 733–751.
2. M. Abdalla, R. Gay, M. Raykova, and H. Wee. Multi-input inner-product functional encryption from pairings. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 601–626. Springer, Heidelberg, Apr. / May 2017.
3. S. Agrawal. Stronger security for reusable garbled circuits, general definitions and attacks. In Katz and Shacham [40], pages 3–35.
4. S. Agrawal, S. Bhattacherjee, D. H. Phan, D. Stehlé, and S. Yamada. Efficient public trace and revoke from standard assumptions: Extended abstract. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 17*, pages 2277–2293. ACM Press, Oct. / Nov. 2017.
5. S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Heidelberg, Dec. 2011.
6. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Robshaw and Katz [51], pages 333–362.
7. S. Agrawal and A. Rosen. Functional encryption for bounded collusions, revisited. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 173–205. Springer, Heidelberg, Nov. 2017.
8. P. Ananth and X. Fan. Attribute based encryption for rams from lwe. Cryptology ePrint Archive, Report 2018/273, 2018. `https://eprint.iacr.org/2018/273`.

9. M. Bellare and V. T. Hoang. Adaptive witness encryption and asymmetric password-based cryptography. In Katz [38], pages 308–331.
10. F. Benhamouda, F. Bourse, and H. Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In S. Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 36–66. Springer, Heidelberg, Mar. 2017.
11. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.
12. A. Bishop, A. Jain, and L. Kowalczyk. Function-hiding inner product encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 470–491. Springer, 2015.
13. D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. Cryptology ePrint Archive, Report 2017/956, 2017. `https://eprint.iacr.org/2017/956`.
14. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.
15. D. Boneh, T. Roughgarden, and J. Feigenbaum, editors. *45th ACM STOC*. ACM Press, June 2013.
16. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, Mar. 2011.
17. X. Boyen. Attribute-based functional encryption on lattices. In A. Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Heidelberg, Mar. 2013.
18. X. Boyen and Q. Li. Turing machines with shortcuts: Efficient attribute-based encryption for bounded functions. In M. Manulis, A.-R. Sadeghi, and S. Schneider, editors, *ACNS 16*, volume 9696 of *LNCS*, pages 267–284. Springer, Heidelberg, June 2016.
19. Z. Brakerski, A. Jain, I. Komargodski, A. Passelegue, and D. Wichs. Non-trivial witness encryption and null-io from standard assumptions. Cryptology ePrint Archive, Report 2017/874, 2017. `https://eprint.iacr.org/2017/874`.
20. Z. Brakerski and V. Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Robshaw and Katz [51], pages 363–384.
21. R. Canetti and J. A. Garay, editors. *CRYPTO 2013, Part II*, volume 8043 of *LNCS*. Springer, Heidelberg, Aug. 2013.
22. M. Chase. Multi-authority attribute based encryption. In S. P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 515–534. Springer, Heidelberg, Feb. 2007.
23. M. Chase and S. S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, *ACM CCS 09*, pages 121–130. ACM Press, Nov. 2009.
24. P. Datta, T. Okamoto, and J. Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. In M. Abdalla and R. Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Heidelberg, Mar. 2018.
25. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013.

26. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In Canetti and Garay [21], pages 479–499.
27. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Functional encryption without obfuscation. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 480–511. Springer, Heidelberg, Jan. 2016.
28. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In Boneh et al. [15], pages 467–476.
29. S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In Canetti and Garay [21], pages 536–553.
30. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
31. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In Safavi-Naini and Canetti [53], pages 162–179.
32. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In Boneh et al. [15], pages 545–554.
33. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, Aug. 2015.
34. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, Oct. / Nov. 2006. Available as Cryptology ePrint Archive Report 2006/309.
35. M. Green, S. Hohenberger, and B. Waters. Outsourcing the decryption of abe ciphertexts. In *USENIX Security Symposium*, volume 2011, 2011.
36. Z. Jafargholi, C. Kamath, K. Klein, I. Komargodski, K. Pietrzak, and D. Wichs. Be adaptive, avoid overcommitting. In Katz and Shacham [40], pages 133–163.
37. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In *Proceedings, Part II, of the 22nd International Conference on Advances in Cryptology — ASIACRYPT 2016 - Volume 10032*, pages 682–712, 2016.
38. J. Katz, editor. *PKC 2015*, volume 9020 of *LNCS*. Springer, Heidelberg, Mar. / Apr. 2015.
39. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, Apr. 2008.
40. J. Katz and H. Shacham, editors. *CRYPTO 2017, Part I*, volume 10401 of *LNCS*. Springer, Heidelberg, Aug. 2017.
41. A. Lewko and B. Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology - EUROCRYPT 2011 - International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 568–588, 2010.
42. A. Lewko and B. Waters. Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 568–588. Springer, 2011.
43. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product

encryption. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Heidelberg, May / June 2010.
44. H. Lin, Z. Cao, X. Liang, and J. Shao. Secure threshold multi authority attribute based encryption without a central authority. In D. R. Chowdhury, V. Rijmen, and A. Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 426–436. Springer, Heidelberg, Dec. 2008.
45. Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In V. Atluri and C. Díaz, editors, *ESORICS 2011*, volume 6879 of *LNCS*, pages 278–297. Springer, Heidelberg, 2011.
46. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *43rd FOCS*, pages 356–365. IEEE Computer Society Press, Nov. 2002.
47. S. Müller, S. Katzenbeisser, and C. Eckert. Distributed attribute-based encryption. In *International Conference on Information Security and Cryptology*, pages 20–36. Springer, 2008.
48. S. Muller, S. Katzenbeisser, and C. Eckert. On multi-authority ciphertext-policy attribute-based encryption. *Bulletin of the Korean Mathematical Society*, 46(4):803–819, 2009.
49. A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/2010/556`.
50. R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 195–203. ACM Press, Oct. 2007.
51. M. Robshaw and J. Katz, editors. *CRYPTO 2016, Part III*, volume 9816 of *LNCS*. Springer, Heidelberg, Aug. 2016.
52. Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13*, pages 463–474. ACM Press, Nov. 2013.
53. R. Safavi-Naini and R. Canetti, editors. *CRYPTO 2012*, volume 7417 of *LNCS*. Springer, Heidelberg, Aug. 2012.
54. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.
55. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Heidelberg, Mar. 2011.
56. B. Waters. Functional encryption for regular languages. In Safavi-Naini and Canetti [53], pages 218–235.
57. K. Xagawa. Improved (hierarchical) inner-product encryption from lattices. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 235–252. Springer, Heidelberg, Feb. / Mar. 2013.