On the P/poly Validity of the Agr17 FE Scheme

No Author Given

No Institute Given

Abstract. Functional encryption (FE) is a cutting-edge research topic in cryptography. The Agr17 FE scheme is a major scheme of FE area. This scheme had the novelty of "being applied for the group of general functions (that is, P/poly functions) without IO". It took the BGG+14 ABE scheme as a bottom structure, which was upgraded into a "partially hiding attribute" scheme, and combined with a fully homomorphic encryp-tion (FHE) scheme. However, the Agr17 FE scheme had a strange operation. For noise cancellation of FHE decryption stage, it used bulky "searching noise" rather than elegant "filtering". It searched total modulus interval, so that the FHE modulus should be polynomially large. In this paper we discuss the P/poly validity of the Agr17 FE scheme. First, we obtain the result that the Agr17 FE scheme is P/poly invalid. More detailedly, when the Agr17 FE scheme is applied for the group of randomly chosen P/poly Boolean functions, FHE modulus at the "searching" stage cannot be polynomially large. Our analysis is based on three restrictions of the BGG+14 ABE scheme: (1) The modulus of the BGG+14 ABE should be adapted to being super-polynomially large, if it is applied for the group of randomly chosen P/poly functions. (2) The modulus of the BGG+14 ABE cannot be switched. (3) If the BGG+14 ABE is upgraded into a "partially hiding attribute" scheme, permitted operations about hidden part of the attribute can only be affine operations. Then, to check whether the P/poly validity can be obtained by modifying the scheme, we consider two modified versions. The first modified version is controlling the FHE noise by repeatedly applying bootstrapping, and replacing a modular inner product with an arithmetic inner product. The second modified version is replacing the search for the modulus interval with the search for a public noise interval, hoping such noise interval polynomially large and tolerating the modulus which may be super-polynomially large. The first modified version may be P/poly valid, but it is weaker. There is no evidence to support the P/poly validity of the second modified version. We also present an additional conclusion that there is no evidence to support the P/poly validity of the GVW15 PE scheme. Finally, we present our response to an argument that our work is unnecessary, and show that our work is quite valuable for any interpretation.

Keywords: learning with errors · attribute-based encryption · functional encryption.

1 Introduction

There is a famous route in cryptography research: identity-based encryption (IBE) [1–5] → attribute-based encryption (ABE) [6–11] → predicate encryption (PE) [12–14] → functional encryption (FE) [15–26]. On this route, FE is a cutting-edge research topic. An FE scene is described as such: an encryptor transforms a plaintext into a ciphertext; the ciphertext is received by a group of decryptors who have respective functions; each of them can only transform the ciphertext into the corresponding function value of the plaintext (rather than the plaintext). FE is a level higher than IBE/ABE/PE, because it is much more clearly focused on the security against collusion attack, which can be simply stated as such: suppose the ith decryptor obtains y_i, the value of his function $f^{(i)}$ of the plaintext $\mathbf{m}, i \in \{1, 2, \cdots, I\}$, then the collusion of these I decryptors should obtain the knowledge of $\mathbf{m}$ never more than solving the equation group $\{f^{(i)}(\mathbf{m}) = y_i, i = 1, \cdots, I\}$. The most challenging and valuable task of FE area is to construct a scheme applied for the group of general functions (that is, $\mathsf{P/poly}$ functions) without indistinguishability obfuscation (IO). It is known [23] that such scheme is easy to be constructed by IO, and that huge size and unclear security of IO make people tend to choose other constructions.

The Agr17 FE scheme [15] is a major scheme of FE area. It had the novelty "being applied for the group of general functions (that is, $\mathsf{P/poly}$ functions) without IO". It took the BGG+14 ABE scheme [6] as a bottom structure, which was upgraded into a "partially hiding attribute" scheme, and combined with a fully homomorphic encryption (FHE) scheme. As a limitation of the security, the Agr17 FE scheme only permitted the collusion of one decryptor with the function value 1 and unbounded decryptors with function values 0, called (1, poly) collusion.

However, the Agr17 FE scheme had a strange operation. For noise cancellation of FHE decryption stage, it used bulky "searching noise" rather than elegant "filtering". By applying "lazy OR trick" [12], it searched total modulus interval, so that the modulus (we will call it "inner modulus") should be polynomially large.

Why such? In fact, it comes from a restriction of the BGG+14 ABE scheme, when it is upgraded into a "partially hiding attribute" scheme. We believe that the Agr17 FE scheme didn't carefully check more restrictions of the BGG+14 ABE scheme, and we will.

In this paper we discuss the $\mathsf{P/poly}$ validity of the Agr17 FE scheme. First, we obtain the following result: the Agr17 FE scheme is $\mathsf{P/poly}$ invalid. More detailedly, when the Agr17 FE scheme is applied for the group of randomly chosen $\mathsf{P/poly}$ Boolean functions, FHE modulus (inner modulus) at the "searching" stage cannot be polynomially large.

Our analysis is based on three restrictions of the BGG+14 ABE scheme, as follows.

The first restriction: The modulus of the BGG+14 ABE (we will call it "outer modulus") should be adapted to being super-polynomially large, if it is applied for the group of randomly chosen $\mathsf{P/poly}$ functions.

The second restriction: The modulus of the BGG+14 ABE (outer modulus) cannot be switched.

The third restriction: If the BGG+14 ABE is upgraded into a "partially hiding attribute" scheme, permitted operations about hidden part of the attribute can only be affine operations (about outer modulus). By the way, "filtering" doesn't belong to such affine operations, so that noise cancellation can only use "searching noise" rather than "filtering".

Then, to check whether the P/poly validity can be obtained by modifying the scheme, we consider two modified versions. The first modified version is controlling the FHE noise by repeatedly applying bootstrapping, hoping such noise polynomially large, and replacing a modular inner product with an arithmetic inner product. The second modified version is replacing the search for the modulus interval with the search for a public noise interval, hoping such noise interval polynomially large and tolerating the modulus (inner modulus) which may be super-polynomially large. We can only say that the first modified version may be P/poly valid, but it is weaker. There is no evidence to support the P/poly validity of the second modified version. Because the searching method of the second modified version is just the same as that of the GVW15 PE scheme [12], we obtain an additional conclusion that there is no evidence to support the P/poly validity of the GVW15 PE scheme [12].

Finally, we present our response to an argument that our work is unnecessary, and show that our work is quite valuable for any interpretation.

2 The BGG+14 ABE Scheme: Detailed Description and Our Analysis

An ABE (more detailedly, KP-ABE) scene is described as such: an encryptor transforms a plaintext into a ciphertext, and the ciphertext is related to an attribute; the ciphertext and the attribute are received by a group of decryptors who have respective functions; if his function value of the attribute equals y_0, the decryptor can transform the ciphertext back into the plaintext, otherwise the decryptor can only transform the ciphertext into gibberish.

The BGG+14 ABE scheme is a bottom structure of the Agr17 FE scheme. In this section, we present a detailed description of the BGG+14 ABE scheme, and discuss its three restrictions.

2.1 Notations and Operations

Let (m, n, q) denote three positive integers such that $q = n^{\Theta(d_{max})}$ where q is prime, $m = n\lceil \log_2 q \rceil$, and d_{max} has been well explained. There are two notes: (1) When n is fixed, q belongs to a very large region. (2) When q increases, m increases with much slower speed, such difference of speeds makes that the BGG+14 ABE scheme can be applied for complicated functions.

Let $\mathbb{Z}$ denote the set of integers. For two positive integers (m', m''), $(\mathbb{Z}^{m'}, \mathbb{Z}^{m' \times m''}, \mathbb{Z}_q^{m'}, \mathbb{Z}_q^{m' \times m''})$ have been well defined. Note that the output of the operation '$\mathrm{mod}\, q$'

is within $\{\lceil -\frac{q}{2} \rceil, \lceil -\frac{q}{2} \rceil + 1, \ldots, \lceil \frac{q}{2} \rceil\}$, rather than $\{0, 1, \ldots, q-1\}$. For $a \in \mathbb{Z}_q$, the meanings of $\mathbf{A} \in \mathbb{Z}_q^{m' \times m''}$ and $a\mathbf{A} \in \mathbb{Z}_q^{m' \times m''}$ are clear. Let $\mathbf{G}$ denote the following special matrix

$$\mathbf{G} = \begin{pmatrix} 1\ 2 \cdots\ 2^{\lceil \log_2 q \rceil - 1} & & & \\ & 1\ 2 \cdots\ 2^{\lceil \log_2 q \rceil - 1} & & \\ & & \ddots & \\ & & & 1\ 2 \cdots\ 2^{\lceil \log_2 q \rceil - 1} \end{pmatrix} \in \mathbb{Z}_q^{n \times m}.$$

For any $\alpha \in \mathbb{Z}_q$, there is a unique Boolean matrix $\mathbf{G}^{(\alpha)} \in \mathbb{Z}^{m \times m}$ such that

$$\alpha \mathbf{G} = \mathbf{G}\mathbf{G}^{(\alpha)} (\mathrm{mod} q).$$

For any $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, there is a unique Boolean matrix $\mathbf{G}^{(\mathbf{B})} \in \mathbb{Z}^{m \times m}$ such that

$$\mathbf{B} = \mathbf{G}\mathbf{G}^{(\mathbf{B})} (\mathrm{mod} q).$$

Then, the following three algorithms are well known.

- $\mathsf{TrapGen}(n, m, q)$: Input (n, m, q) and output $(\mathbf{A}, \mathbf{T})$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is a uniform matrix, $\mathbf{T} \in \mathbb{Z}^{m \times m}$ is a small Gaussian matrix, $\mathbf{A}\mathbf{T} = \mathbf{0} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{T}$ is of full rank (We know that $\mathbf{T}$ is not of full rank regarding the modulus q). $\mathbf{T}$ is called a trapdoor of $\mathbf{A}$.
- $\mathsf{Encode}(\mathbf{A}, \mathbf{s})$: Input $(\mathbf{A}, \mathbf{s}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$ and output $\boldsymbol{\psi} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$, where $\mathbf{e} \in \mathbb{Z}^m$ is a small Gaussian vector. $\mathbf{s}$ is the encoded vector, $\boldsymbol{\psi}$ is the encoding of $\mathbf{s}$, and $\mathbf{e}$ is the noise vector. We say $\boldsymbol{\psi} = \mathsf{Encode}(\mathbf{A}, \mathbf{s})$.
- $\mathsf{ReKeyGen}(\mathbf{A}, \mathbf{B}, \mathbf{T}, \mathbf{D})$: Input $(\mathbf{A}, \mathbf{B}, \mathbf{T}, \mathbf{D})$ and output $\mathbf{R}$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, both $\mathbf{A}$ and $\mathbf{B}$ are uniform matrices. $\mathbf{T} \in \mathbb{Z}^{m \times m}$ is a trapdoor of $\mathbf{A}$, $\mathbf{R} \in \mathbb{Z}^{2m \times m}$ is a small Gaussian matrix, and $\mathbf{D} = [\mathbf{A}, \mathbf{B}]\mathbf{R} \in \mathbb{Z}_q^{n \times m}$. In fact,
$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix}, \mathbf{R}_0 \in \mathbb{Z}^{m \times m}, \mathbf{R}_1 \in \mathbb{Z}^{m \times m},$$
then $\mathbf{R}_1$ is the presampled matrix, and $\mathbf{R}_0$ is the co-sampled matrix. In other words, for any small matrix $\mathbf{R}_1$, the trapdoor matrix $\mathbf{T}$ guarantees to generate a small matrix $\mathbf{R}_0$ such that $\mathbf{A}\mathbf{R}_0 = \mathbf{D} - \mathbf{B}\mathbf{R}_1$.

2.2 Arithmetic Representation and Big Modulus Representation of Boolean Functions

In order to make the BGG+14 ABE scheme available, Boolean functions need to be expressed as mod$q$ functions, i.e., big modulus functions. This can be easily achieved by firstly transforming each Boolean operation into several arithmetic operations and then transforming each arithmetic operation into a big modulus operation. For example, for two bit variables x_1 and x_2,

$$x_1 \cdot x_2 (\mathrm{mod} 2) = x_1 \cdot x_2 = x_1 \cdot x_2 (\mathrm{mod} q),$$

$$x_1 + x_2 (\mathrm{mod} 2) = x_1 + x_2 - 2x_1 \cdot x_2 = x_1 + x_2 - 2x_1 \cdot x_2 (\mathrm{mod} q).$$

Then, by generalizing these transformations, each operation of a Boolean function can be converted into several operations under a big modulus. Therefore, Boolean functions are described as mod$q$ functions, except that the independent variables are in $\mathbb{F}_2$ rather than $\mathbb{Z}_q$. Another interesting feature is that, if a Boolean function is expressed as an arithmetic function step by step, the absolute value of each intermediate variable does not exceed 2. More detailedly, an intermediate variable with the absolute value 2 will not be the input of any later multiplication operation, and only be the input of some later addition operation to decrease the absolute value.

2.3 Quasi-homomorphic Operations of the BGG+14 ABE Scheme

Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$ denote an l-dimensional attribute, where each x_i is a bit variable. Take l matrices $\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_l \in \mathbb{Z}_q^{n \times m}$. Take another l matrices $x_1\mathbf{G} + \mathbf{B}_1, x_2\mathbf{G} + \mathbf{B}_2, \cdots, x_l\mathbf{G} + \mathbf{B}_l \in \mathbb{Z}_q^{n \times m}$. In the follow we will show that, for any P/poly Boolean function $f(\boldsymbol{x})$, there are some 'small-size linear combination operations' for the matrices $\{x_1\mathbf{G} + \mathbf{B}_1, x_2\mathbf{G} + \mathbf{B}_2, \cdots, x_l\mathbf{G} + \mathbf{B}_l\}$, resulting in a new matrix

$$f(\boldsymbol{x}) \cdot \mathbf{G} + \mathbf{B}_f \in \mathbb{Z}_q^{n \times m},$$

where $\mathbf{B}_f$ is independent of $\boldsymbol{x}$. Recalling subsection 2.2, any Boolean operation can be viewed as operations in $\mathbb{Z}_q$, and any Boolean function can be viewed as a function in $\mathbb{Z}_q$. Furthermore, for this special function in $\mathbb{Z}_q$, the result of each operation belongs to $[-2, 2]$. First, we consider the following four simple cases of arithmetic functions.

Case I. If $f(\boldsymbol{x}) = \alpha x_i$ where α is a constant, then the 'small-size linear combination operation' is

$$(x_i\mathbf{G} + \mathbf{B}_i)\mathbf{G}^{(\alpha)} = \alpha x_i \mathbf{G} + \mathbf{B}_i\mathbf{G}^{(\alpha)} (\mathrm{mod} q),$$

where $\mathbf{B}_f = \mathbf{B}_i\mathbf{G}^{(\alpha)}$.

Case II. If $f(\boldsymbol{x}) = x_i + x_j$, then the 'small-size linear combination operation' is

$$(x_i\mathbf{G} + \mathbf{B}_i) + (x_j\mathbf{G} + \mathbf{B}_j) (\mathrm{mod} q) = (x_i + x_j)\mathbf{G} + (\mathbf{B}_i + \mathbf{B}_j) (\mathrm{mod} q),$$

where $\mathbf{B}_f = \mathbf{B}_i + \mathbf{B}_j$.

Case III. If $f(\boldsymbol{x}) = x_i \cdot x_j$ where $i \leq j$, then the 'small-size linear combination operation' is

$$x_j(x_i\mathbf{G} + \mathbf{B}_i) - (x_j\mathbf{G} + \mathbf{B}_j)\mathbf{G}^{(\mathbf{B}_i)} = x_i x_j \mathbf{G} + (-\mathbf{B}_j\mathbf{G}^{(\mathbf{B}_i)}) (\mathrm{mod} q),$$

where $\mathbf{B}_f = -\mathbf{B}_j\mathbf{G}^{(\mathbf{B}_i)}$.

Case IV. If $f(\boldsymbol{x}) = \alpha \cdot x_{j_1} \cdot x_{j_2} \cdot \cdots \cdot x_{j_k}, j_1 \leq j_2 \leq \cdots j_k$ and α is a constant, then the 'small-size linear combination operation' is

$$\sum_{i=1}^{k} \left(\prod_{h=i+1}^{k} x_{j_h} \right) \cdot (x_{j_i}\mathbf{G} + \mathbf{B}_{j_i}) \cdot \mathbf{G}_i = \alpha \cdot x_{j_1} \cdot x_{j_2} \cdot \cdots \cdot x_{j_k} \cdot \mathbf{G} + (-\mathbf{B}_{j_k}\mathbf{G}_k),$$

where $\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_k$ are Boolean matrices in $\mathbb{Z}^{m\times m}$ and are defined recursively as below:

$$\mathbf{G}_1 = \mathbf{G}^{\alpha},$$

$$\mathbf{G}_i = \mathbf{G}^{(-\mathbf{B}_{j_{i-1}}\mathbf{G}_{i-1})}, i = 2, 3, \cdots, k,$$

where $\mathbf{B}_f = -\mathbf{B}_{j_k} \cdot \mathbf{G}_k$ is also independent of $\boldsymbol{x}$.

Then, we affirm that iterations of 'small-size linear combination operations' are still 'small-size linear combination operations', provided the time of iterations is at the polynomial level. Thus, we draw the conlusion by repeating the aforementioned four operations: any $\mathsf{P/poly}$ Boolean function f can execute 'small-size linear combination operations' on the above matrices, resulting in $f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f$.

Next, we do the following encoding:

$$\mathbf{c}_1 = \mathsf{Encode}(x_1\mathbf{G} + \mathbf{B}_1, \mathbf{s}),$$

$$\mathbf{c}_2 = \mathsf{Encode}(x_2\mathbf{G} + \mathbf{B}_2, \mathbf{s}),$$

$$\cdots,$$

$$\mathbf{c}_l = \mathsf{Encode}(x_l\mathbf{G} + \mathbf{B}_l, \mathbf{s}).$$

By executing the same 'small-size linear combination operation' (only plus a transpose) on the codeword $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l)$, we will obtain

$$\mathbf{c}_f = \mathsf{Encode}(f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f, \mathbf{s}).$$

We call such 'small-size linear combination operations' on $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l)$ quasi-homomorphic operation of the Boolean function f.

2.4 The BGG+14 ABE Scheme [6]

- Generating master key $(\mathbf{mpk}, \mathbf{msk})$: The key generator runs $\mathsf{TrapGen}(n, m, q)$ to obtain $(\mathbf{A}, \mathbf{T})$, then he randomly picks $\mathbf{B}_i \in \mathbb{Z}_q^{n\times m}, i = 1, 2, \cdots, l, \mathbf{D} \in \mathbb{Z}_q^{n\times m}$. The output is

$$\mathbf{mpk} = (\mathbf{A}, \mathbf{B}_1, \cdots, \mathbf{B}_l, \mathbf{D}), \mathbf{msk} = \mathbf{T}.$$

- Generating secret key $\mathbf{sk}_f$ for the Boolean function f from a group of Boolean functions: The key generator firstly generates $\mathbf{B}_f$. Note that $\mathbf{B}_f$ is generated by the method in subsection 2.3. The attribute is randomly chosen, and the resulting $\mathbf{B}_f$ is independent of this attribute. Then, he runs $\mathsf{ReKeyGen}(\mathbf{A}, y_0\mathbf{G} + \mathbf{B}_f, \mathbf{T}, \mathbf{D})$ to obtain $\mathbf{R} \in \mathbb{Z}^{2m\times m}$. The output is

$$\mathbf{sk}_f = \mathbf{R}.$$

  Each $\mathbf{sk}_f$ is sent to corresponding decryptor who owns f.

- Encryption: The plaintext $\mathbf{m}$ is an m-dimensional Boolean vector. The attribute $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$ is sent to the encryptor. The encryptor randomly picks $\mathbf{s} \in \mathbb{Z}_q^n$, and computes ($\mathsf{Encode}(\mathbf{A}, \mathbf{s})$, $\mathsf{Encode}(x_1\mathbf{G} + \mathbf{B}_1, \mathbf{s})$, $\mathsf{Encode}(x_2\mathbf{G} + \mathbf{B}_2, \mathbf{s}), \cdots, \mathsf{Encode}(x_l\mathbf{G} + \mathbf{B}_l, \mathbf{s})$, $\mathsf{Encode}(\mathbf{D}, \mathbf{s})$). The ciphertext is

$$\begin{aligned}\mathbf{C} &= (\mathbf{c}_{in}, \mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l, \mathbf{c}_{out}) \\ &= (\mathsf{Encode}(\mathbf{A}, \mathbf{s}), \mathsf{Encode}(x_1\mathbf{G} + \mathbf{B}_1, \mathbf{s}), \cdots, \mathsf{Encode}(x_l\mathbf{G} + \mathbf{B}_l, \mathbf{s}), \mathsf{Encode}(\mathbf{D}, \mathbf{s}) + \lceil \frac{q}{2} \rceil \mathbf{m}) \\ &= (\mathbf{A}^T\mathbf{s} + \mathbf{e}_{in}, (x_1\mathbf{G} + \mathbf{B}_1)^T\mathbf{s} + \mathbf{e}_1, \cdots, (x_l\mathbf{G} + \mathbf{B}_l)^T\mathbf{s} + \mathbf{e}_l, \mathbf{D}^T\mathbf{s} + \mathbf{e}_{out} + \lceil \frac{q}{2} \rceil \mathbf{m}),\end{aligned}$$

  $\{\mathbf{C}, \boldsymbol{x}\}$ are sent to the group of decryptors who have respective funcitons.
- Decryption: By using his own function f and the attribute $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$, one of the decryptors executes the quasi-homomorphic operation on $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l\}$ to obtain

$$\begin{aligned}\mathbf{c}_f &= \mathsf{Encode}(f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f, \mathbf{s}) \\ &= (f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f)^T\mathbf{s} + \mathbf{e}_f(\boldsymbol{x}).\end{aligned}$$

  Then, by using $\mathbf{sk}_f = \mathbf{R}$, the decryptor computes

$$\begin{aligned}\mathbf{c}_{out} - \mathbf{R}^T \begin{pmatrix} \mathbf{c}_{in} \\ \mathbf{c}_f \end{pmatrix} &= \mathbf{D}^T\mathbf{s} - \mathbf{D}^T\mathbf{s} + \lceil \frac{q}{2} \rceil \mathbf{m} + ((y_0 - f(\boldsymbol{x}))\mathbf{G})^T \mathbf{s} + \mathbf{e}' \\ &= \lceil \frac{q}{2} \rceil \mathbf{m} + ((y_0 - f(\boldsymbol{x}))\mathbf{G})^T\mathbf{s} + \mathbf{e}'.\end{aligned}$$

  When $f(\boldsymbol{x}) = y_0$, the plaintext $\mathbf{m}$ can be obtained by using "*Rounding*"; When $f(\boldsymbol{x}) \neq y_0$, gibberish is returned. Generally, $y_0 = 1$.

We emphasize that the modulus q should be a prime, and the scheme with composite modulus is much weaker. More specifically, the scheme with the modulus which has a polynomially large factor is insecure.

2.5 Hiding a Part of Attribute in the BGG+14 ABE Scheme

The so-called 'hiding a part of attribute' means that the encryptor knows the entity of the attribute and the decryptors only receive a part of it, with another part hidden from them. The key issue is whether a decryptor can execute the quasi-homomorphic operation of his own function f when he does not know some part of the attribute.

2.6 Applied for F_q Functions: An Assumption

The BGG+14 ABE scheme can be applied for any group of Boolean functions, or for any group of arithmetic functions (also called "false" modular functions). More detailedly, for any group of Boolean functions or arithmetic functions, there

are sufficiently large $\{m, q\}$ such that the noise expansion of quasi-homomorphic operations does not exceed the modulus interval $(-q/2, q/2)$.

A question is whether the BGG+14 ABE scheme can be applied for any group of F_q functions (also called "real" modular functions). This question is essential for the Agr17 FE scheme, and the answer "no" will make it much easier to show $\mathsf{P/poly}$ invalidity of the Agr17 FE scheme. At first look, the answer to this question should be "no", because the quasi-homomorphic operation of a modular q function is not 'small-size linear combination operation'. For example, suppose u and v satisfy that $u \cdot v \bmod q \neq u \cdot v$, with $\mathbf{c}_u = \mathsf{Encode}(u\mathbf{G} + \mathbf{B}_u, \mathbf{s})$ and $\mathbf{c}_v = \mathsf{Encode}(v\mathbf{G} + \mathbf{B}_v, \mathbf{s})$. Then $\mathbf{c}_{u \cdot v} = v\mathbf{c}_u - (\mathbf{G}^{(\mathbf{B}_u)})^{\mathbf{T}} \cdot \mathbf{c}_v$, where $\mathbf{G}^{(\mathbf{B}_u)}$ is a Boolean matrix which does not make great noise expansion, while $v \cdot \mathbf{c}_u$ has a great noise expansion, quite possible to exceed the modulus interval $(-q/2, q/2)$.

Then there is a method to reduce such obstacle. The method can be called "head/tail biterization", which is to express the independent variable and the function value as bits, so that an F_q function of l-dimensional independent variable can be expressed as q' Boolean functions of $q'l$-dimensional independent variable, where $q' = \lceil \log_2 q \rceil$. The effect of such method is suspicious, because both the number of Boolean functions and the complexity of each Boolean function are not predetermined, but rather increase with the increase of q. On the other hand, we have not found an evidence to negate such method. For the sake of our clear analysis of the Agr17 FE scheme from other aspects, we take the answer "yes", that is, we take such assumption: the BGG+14 ABE scheme can be applied for any group of F_q functions.

2.7 The First Restriction

Subsection 2.1 sets $q = n^{\Theta(dmax)}$, which belongs to a very large region. In this subsection we discuss how large q should be, if the BGG+14 ABE scheme is applied for a group of randomly chosen $\mathsf{P/poly}$ Boolean functions.

Suppose two intermediate variables u and v, with $\mathbf{c}_u = \mathsf{Encode}(u\mathbf{G} + \mathbf{B}_u, \mathbf{s})$ and $\mathbf{c}_v = \mathsf{Encode}(v\mathbf{G} + \mathbf{B}_v, \mathbf{s})$, where noise vectors of $\mathbf{c}_u$ and $\mathbf{c}_v$ are respectively $\mathbf{e}_u$ and $\mathbf{e}_v$. Then $\mathbf{c}_{u \cdot v} = v\mathbf{c}_u - (\mathbf{G}^{(\mathbf{B}_u)})^{\mathbf{T}} \cdot \mathbf{c}_v$, where the noise vector of $\mathbf{c}_{u \cdot v}$ is $\mathbf{e}_{u \cdot v} = v\mathbf{e}_u - (\mathbf{G}^{(\mathbf{B}_u)})^{\mathbf{T}} \cdot \mathbf{e}_v$. We know $|v| \leq 2$ and $\mathbf{G}^{(\mathbf{B}_u)}$ is a Boolean matrix, so that averagely $|\mathbf{e}_{u \cdot v}| \approx \sqrt{\frac{n}{2}}\,|\mathbf{e}_v|$. This is the key fact for our analysis, which implies that we can find a large number of $\mathsf{P/poly}$ Boolean functions whose noise expansions are super-polynomially large.

An excuse is *Case IV* in subsection 2.3, which greatly saves the noise expansion of the quasi-homomorphic operations of continuous multiplications. More detailedly, $|\mathbf{e}_{u_1 \cdot u_2 \cdots u_k}|$ is only about $\sqrt{\frac{n}{2}} \cdot |\mathbf{e}_{u_1} + \mathbf{e}_{u_2} + \cdots + \mathbf{e}_{u_k}|$, and much smaller than $(\sqrt{\frac{n}{2}})^{k-1} \cdot |\mathbf{e}_{u_k}|$. Our answer is that, for a randomly chosen Boolean function, continuous multiplication is an event with very small probability. Notice that, even two adjacent operations are both multiplications, they form a continuous multiplication with very small probability. By considering the loss of noise expansion of *Case IV*, we can still find a large number of $\mathsf{P/poly}$ Boolean functions whose noise expansions are super-polynomially large.

Another excuse may be considering an "optimized quasi-homomorphic operation" for the multiplication operation, if $|\mathbf{e}_u| < |\mathbf{e}_v|$, take $\mathbf{c}_{u\cdot v} = u \cdot \mathbf{c}_v - (\mathbf{G}^{(\mathbf{B}_v)})^{\mathbf{T}} \cdot \mathbf{c}_u$, otherwise, take $\mathbf{c}_{u\cdot v} = v \cdot \mathbf{c}_u - (\mathbf{G}^{(\mathbf{B}_u)})^{\mathbf{T}} \cdot \mathbf{c}_v$. By such "optimized operation", $|\mathbf{e}_{u\cdot v}| \approx \sqrt{\frac{n}{2}} \min\{|\mathbf{e}_u|, |\mathbf{e}_v|\}$. Our answer is that comparing $|\mathbf{e}_u|$ and $|\mathbf{e}_v|$ is not only complicated, but also usually impossible for decryptors.

We take such a random experiment: Applying the BGG+14 ABE scheme for a randomly chosen P/poly Boolean function, and observing the noise expansion, whether polynomially large or super-polynomially large. Suppose the probability of the former is p, and that of the latter $1-p$. There is no evidence to say that $1-p$ is clearly smaller than p, so that such an opinion is reasonable: $1-p$ at least belongs to the same size grade of p. Then we take a group of k P/poly Boolean functions, the event "the noise expansions are always polynomially large" has the probability p^k, which quickly tends to 0 when k tends larger.

Finally, we obtain the first restriction: when the BGG+14 ABE scheme is applied for a group of randomly chosen P/poly Boolean functions, the modulus q should be adapted to being super-polynomially large.

2.8 The Second Restriction

We know that an LWE-based encryption scheme can apply modular switching to change the modulus, so as to simplify the computation. The BGG+14 ABE scheme is an LWE-based encryption scheme, but it is a very special one. The key information is inserted into related matrices which provides conditions of whether correct decryption or decryption failure. Modular switching will destroy the key information, so that destroys the conditions.

From the above discussion, we obtain the second restriction: the modulus of the BGG+14 ABE scheme cannot be switched.

2.9 The Third Restriction

It is clear that if a part of the attribute is hidden from the decryptors, the multiplication operation of two unknown numbers (modular q) cannot be quasi-homomorphically operated (see subsection 2.3, *Case III* and *Case IV*). So we immediately obtain the third restriction: if the BGG+14 ABE scheme is upgraded into a "partially hiding attribute" scheme, permitted operations about the hidden part of the attribute can only be affine operations (modular q).

3 The Agr17 FE Scheme [15]

3.1 An Attack on the BGG+14 ABE Scheme

The Agr17 FE scheme [15] presented an attack on the GVW15 predicate encryption (PE) scheme [12]. In fact, such attack is on the BGG+14 ABE scheme: a decryptor repeatedly asks for the decryption keys of the BGG+14 ABE scheme for the same function f. This is the common worry of all lattice-based cryptosystems which the decryptor may obtain a modified trapdoor by such repeat.

3.2 Improved BGG+14 ABE Scheme Against the Attack [15]

The improved BGG+14 ABE scheme is almost the original the BGG+14 ABE scheme (see subsection 2.4), with only an additional group of uniform matrices $\{\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_k\}$. The scheme is described as follows.

- Generating master key $(\mathbf{mpk}, \mathbf{msk})$: The key generator runs $\mathsf{TrapGen}(n, m, q)$ to obtain $(\mathbf{A}, \mathbf{T})$, then he randomly picks $\mathbf{B}_i \in \mathbb{Z}_q^{n\times m}, i = 1, 2, \cdots, l, \mathbf{D}_j \in \mathbb{Z}_q^{n\times m}, j = 1, 2, \cdots, k, \mathbf{D} \in \mathbb{Z}_q^{n\times m}$. The output is

$$\mathbf{mpk} = (\mathbf{A}, \mathbf{B}_1, \cdots, \mathbf{B}_l, \mathbf{D}_1, \cdots, \mathbf{D}_k, \mathbf{D}), \mathbf{msk} = \mathbf{T}.$$

- Generating secret key $\mathbf{sk}_f$ for the Boolean function f from a group of Boolean functions: The key generator firstly generates $\mathbf{B}_f$. Note that $\mathbf{B}_f$ is generated by the method in subsection 2.3. Then he chooses a random subset Δ_f of the set $\{1, \cdots, k\}$, and computes $\sum_{j\in\Delta_f} \mathbf{D}_j$. Then, he runs $\mathsf{ReKeyGen}(\mathbf{A}, y_0\mathbf{G} + \mathbf{B}_f, \mathbf{T}, \sum_{j\in\Delta_f} \mathbf{D}_j + \mathbf{D})$ to obtain $\mathbf{R} \in \mathbb{Z}^{2m\times m}$. The output is

$$\mathbf{sk}_f = \{\Delta_f, \mathbf{R}\}.$$

  Each $\mathbf{sk}_f$ is sent to corresponding decryptor who owns f.
  (A note: In the Agr17 FE scheme, it is stated on page 12 of [15], that such $\mathbf{R}$ is obtained by running $\mathsf{ReKeyGen}(\mathbf{A}, y_0\mathbf{G} + \mathbf{B}_f, \mathbf{T}, \sum_{j\in\Delta_f} \mathbf{D}_j)$ rather than $\mathsf{ReKeyGen}(\mathbf{A}, y_0\mathbf{G} + \mathbf{B}_f, \mathbf{T}, \sum_{j\in\Delta_f} \mathbf{D}_j + \mathbf{D})$. This is a minor mistake, because it cannot generate a simple ciphertext for the case of "one ciphertext and multiple decryptions". Anyway, it is not important and does not affect our results.)
- Encryption: The plaintext $\mathbf{m}$ is an m-dimensional Boolean vector. The attribute $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$ is sent to the encryptor. The encryptor randomly picks $\mathbf{s} \in \mathbb{Z}_q^n$, and computes $(\mathsf{Encode}(\mathbf{A}, \mathbf{s}), \mathsf{Encode}(x_1\mathbf{G} + \mathbf{B}_1, \mathbf{s}), \mathsf{Encode}(x_2\mathbf{G}+\mathbf{B}_2, \mathbf{s}), \cdots, \mathsf{Encode}(x_l\mathbf{G}+\mathbf{B}_l, \mathbf{s}), \mathsf{Encode}(\mathbf{D}_1, \mathbf{s}), \mathsf{Encode}(\mathbf{D}_2, \mathbf{s}), \cdots, \mathsf{Encode}(\mathbf{D}_k, \mathbf{s}), \mathsf{Encode}(\mathbf{D}, \mathbf{s}))$. The ciphertext is

$$\begin{aligned}\mathbf{C} &= (\mathbf{c}_{in}, \mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l, \mathbf{c}_{1,out}, \mathbf{c}_{2,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out}) \\ &= (\mathsf{Encode}(\mathbf{A}, \mathbf{s}), \mathsf{Encode}(x_1\mathbf{G} + \mathbf{B}_1, \mathbf{s}), \cdots, \mathsf{Encode}(x_l\mathbf{G} + \mathbf{B}_l, \mathbf{s}), \mathsf{Encode}(\mathbf{D}_1, \mathbf{s}), \\ &\quad \mathsf{Encode}(\mathbf{D}_2, \mathbf{s}), \cdots, \mathsf{Encode}(\mathbf{D}_k, \mathbf{s}), \mathsf{Encode}(\mathbf{D}, \mathbf{s}) + \lceil \frac{q}{2} \rceil \mathbf{m}) \\ &= (\mathbf{A}^T\mathbf{s} + \mathbf{e}_{in}, (x_1\mathbf{G} + \mathbf{B}_1)^T\mathbf{s} + \mathbf{e}_1, \cdots, (x_l\mathbf{G} + \mathbf{B}_l)^T\mathbf{s} + \mathbf{e}_l, \mathbf{D}_1^T\mathbf{s} + \mathbf{e}_{1,out}, \\ &\quad \mathbf{D}_2^T\mathbf{s} + \mathbf{e}_{2,out}, \cdots, \mathbf{D}_k^T\mathbf{s} + \mathbf{e}_{k,out}, \mathbf{D}^T\mathbf{s} + \mathbf{e}_{out} + \lceil \frac{q}{2} \rceil \mathbf{m}),\end{aligned}$$

  $\{\mathbf{C}, \boldsymbol{x}\}$ are sent to the group of decryptors who have respective funcitons.
- Decryption: By using his own function f and the attribute $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$, one of the decryptors executes the quasi-homomorphic operation on $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_l\}$

to obtain

$$\begin{aligned}\mathbf{c}_f &= \mathsf{Encode}(f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f, \mathbf{s}) \\ &= (f(\boldsymbol{x})\mathbf{G} + \mathbf{B}_f)^T\mathbf{s} + \mathbf{e}_f(\boldsymbol{x}).\end{aligned}$$

Then, by using $\mathbf{sk}_f = \{\Delta_f, \mathbf{R}\}$, the decryptor computes

$$\begin{aligned}\sum_{j\in\Delta_f}\mathbf{c}_{j,out} + \mathbf{c}_{out} - \mathbf{R}^T\begin{pmatrix}\mathbf{c}_{in}\\ \mathbf{c}_f\end{pmatrix} &= \left(\sum_{j\in\Delta_f}\mathbf{D}_j + \mathbf{D}\right)^T\mathbf{s} - \left(\sum_{j\in\Delta_f}\mathbf{D}_j + \mathbf{D}\right)^T\mathbf{s} + \lceil\frac{q}{2}\rceil\mathbf{m} \\ &\quad + ((y_0 - f(\boldsymbol{x}))\mathbf{G})^T\mathbf{s} + \mathbf{e}' \\ &= \lceil\frac{q}{2}\rceil\mathbf{m} + ((y_0 - f(\boldsymbol{x}))\mathbf{G})^T\mathbf{s} + \mathbf{e}'.\end{aligned}$$

When $f(\boldsymbol{x}) = y_0$, the plaintext $\mathbf{m}$ can be obtained by using "*Rounding*"; When $f(\boldsymbol{x}) \neq y_0$, gibberish is returned. Generally, $y_0 = 1$.

It is easy to see that the improved BGG+14 ABE scheme still has the three restrictions in subsection 2.7, subsection 2.8, and subsection 2.9.

3.3 Preliminaries: Inner Modulus and Outer Modulus

FHE and the BGG+14 ABE are two bottom structures of the Agr17 FE scheme. We know that {encryption, fully-homomorphic evaluation, decryption} are three computation stages of FHE. But fully-homomorphic evaluation and decryption cannot be directly implemented for the Agr17 FE scheme, instead, they can only be quasi-homomorphically operated in the BGG+14 ABE scheme. So that the modulus of FHE, Q, is called the inner modulus, and the modulus of the BGG+14 ABE, q, the outer modulus.

It is well known that the inner modulus Q should accommodate the FHE noise, and that the outer modulus q should accommodate the corresponding ABE noise.

3.4 The Encryption Process

Let $\mathbf{m}$ denote the plaintext. The encryption process is composed of the following three steps: (1) $\mathbf{m}$ is encrypted to an FHE ciphertext $\mathbf{m}^*$ by the encryption algorithm of an FHE scheme. It is worth noting that the modulus of the FHE ciphertext (the inner modulus) is Q. (2) $\mathbf{m}^*$ is taken as the public part of the attribute, and t, the decryption key of the FHE scheme, is taken as the hidden part of the attribute. Then, for such attribute $(\mathbf{m}^*, t)$, a public bit-string $\mathbf{b}$ is encrypted to an ABE ciphertext $\mathbf{C}$ by the encryption algorithm of an improved BGG+14 ABE scheme (the improved BGG+14 ABE scheme has now been upgraded into a "partially hiding attribute" scheme). It is worth noting that the modulus of the ABE ciphertext (the outer modulus) is q. (3) Finally, $\{\mathbf{C}, \mathbf{m}^*\}$ is taken as the ciphertext of the Agr17 FE scheme, to be sent to all decryptors in the group. The encryption process is shown in Fig.1

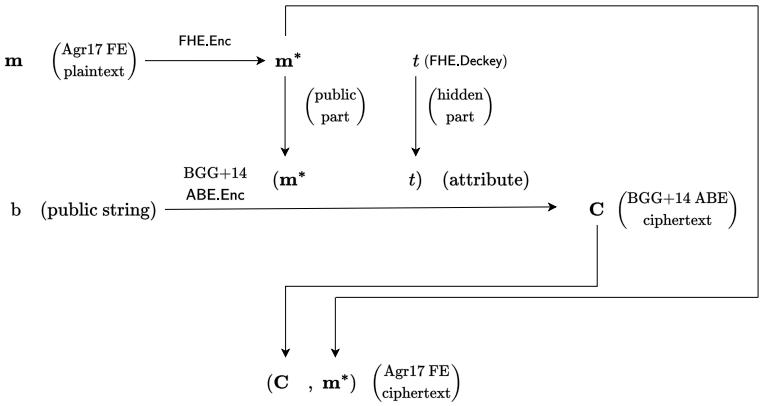

Fig. 1. The encryption process

3.5 Known and Unknown Items of the Decryptors

Now, a decryptor in the group knows the following four items.

(1) The ciphertext $\{\mathbf{C}, \mathbf{m}^*\}$ of the Agr17 FE scheme, where $\mathbf{C}$ is the improved BGG+14 ABE ciphertext, $\mathbf{m}^*$ is the FHE ciphertext and the public part of attribute of the improved BGG+14 ABE scheme. More detailedly, $\mathbf{C} = (\mathbf{c}_{in}, \mathbf{c}_1, \cdots, \mathbf{c}_l, \mathbf{c}_{1,out}, \mathbf{c}_{2,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$, and $(\mathbf{c}_1, \cdots, \mathbf{c}_l)$ can be restated as the follow: $(\mathbf{c}_1, \cdots, \mathbf{c}_l) = (\mathbf{C}_{\mathbf{m}^*}, \mathbf{C}_t)$, where $\mathbf{C}_{\mathbf{m}^*}$ is the ABE ciphertext corresponding to $\mathbf{m}^*$ (the public part of the attribute), $\mathbf{C}_t$ is the ABE ciphertext corresponding to t (the hidden part of the attribute).

(2) His function f and homomorphic operation f^* for FHE evaluation.

(3) Decryption key for the Agr17 FE scheme according to f, such that the FE decryption will obtain $f(\mathbf{m})$ rather than $\mathbf{m}$. In fact, it is a group of about $\lceil Q/2 \rceil$ ABE decryption keys for the improved BGG+14 ABE scheme, where the inner modulus Q was considered polynomially large. Later in subsection 3.8 we will introduce such group of decryption keys in the detail.

(4) The public bit-string $\mathbf{b}$.

The decryptors know neither the plaintext $\mathbf{m}$ nor the FHE decryption key t, which is also the hidden part of the attribute.

3.6 The First Step of the Decryption Process

For the decryptor who owns the function f, the first step of the decryption process is the transformation "$\mathbf{C}_{\mathbf{m}^*} \to \mathbf{C}_{f^*(\mathbf{m}^*)}$". Notice that it is not the transformation "$\mathbf{m}^* \to f^*(\mathbf{m}^*)$". In other words, this step is not homomorphic evaluation, but rather quasi-homomorphic operation of homomorphic evaluation. Another note is that the former is modular Q operation, while the latter is modular q operation.

We know that this step is valid for predetermined Q. More detailedly, for any predetermined Q, there are sufficiently large $\{q, m\}$ such that the ABE noise expansion of this step is within the interval $(-q/2, q/2)$. The method is to express f^* by $\lceil \log_2 Q \rceil$ Boolean functions. Can we take $Q = q$ in this step? According to the analysis of subsection 2.6, the answer seems to be "no", while we take the answer "yes". From all of the above, in this step, we take (Q, q) as one of such two cases:

(1) Q is predetermined and q is correspondingly determined.

(2) $Q = q$.

3.7 The Second Step of the Decryption Process

The second step of the decryption process is the quasi-homomorphic operation of the first stage of FHE decryption. In the following, we give the details.

FHE decryption has several versions [27–29], but basically it is composed of two stages. The major part of the first stage is computing the modular Q inner product of $f^*(\mathbf{m}^*)$ and t. The first stage includes some other affine operations which can be ignored. The second stage is called "noise cancellation".

From the statement above, the second step of the decryption process is the transformation "$\{\mathbf{C}_{f^*(\mathbf{m}^*)}, \mathbf{C}_t\} \rightarrow \mathbf{C}_{<f^*(\mathbf{m}^*),t> \bmod Q}$". If $Q \neq q$, $< f^*(\mathbf{m}^*), t > \bmod Q$ is not modular q affine operation about t, so that it cannot be quasi-homomorphically operated (see the third restriction in subsection 2.9). On the other hand, q cannot be switched (see the second restriction in subsection 2.8). Therefore, Q should either be always equal to q or be switched to q just before the second step of the decryption process.

If Q is switched to q just before the second step of the decryption process, the following four points should be noticed.

(1) Such modular switching cannot be directly implemented, but rather quasi-homomorphically operated.

(2) Such quasi-homomorphic operation is the transformation "$\mathbf{C}_{f^*(\mathbf{m}^*)} \rightarrow \mathbf{C}_{f^{**}(\mathbf{m}^*)}$", where $f^*(\mathbf{m}^*)$ is the text for the modulus Q, and $f^{**}(\mathbf{m}^*)$ is switched text for the modulus q.

(3) t cannot be switched, that is, the decryptors cannot obtain such transformation "$\mathbf{C}_t \rightarrow \mathbf{C}_{t^*}$", where t is the text for the modulus Q, t^* is switched text for the modulus q, and t^* is still FHE decryption key under the new modulus q. The reason is that the transformation "$t \rightarrow t^*$" is not modular q affine operation about hidden t (see the third restriction in subsection 2.9). In other words, t should be originally the text for the modulus q rather than the text for the modulus Q.

(4) After such modular switching, the second step of the decryption process should be the new transformation "$\{\mathbf{C}_{f^{**}(\mathbf{m}^*)}, \mathbf{C}_t\} \rightarrow \mathbf{C}_{<f^{**}(\mathbf{m}^*),t> \bmod q}$".

3.8 The Third Step of the Decryption Process

Now $Q = q$. The third step of the decryption process is the quasi-homomorphic operation of the second stage of FHE decryption, that is, the quasi-homomorphic operation of "noise cancellation".

Denote $v =< f^*(\mathbf{m}^*), t > \mod q$ (or $v =< f^{**}(\mathbf{m}^*), t > \mod q$, if the modular switching is applied). The method of noise cancellation is filtering, which is described in the following. If $v = f(\mathbf{m}) + 2e$, where e is the noise, then $v \mod 2 = f(\mathbf{m})$ (or if $v = \lceil \frac{q}{2} \rfloor f(\mathbf{m}) + e$, then $Rounding(v) = (2v \mod q) \mod 2 = f(\mathbf{m})$).

However, filtering cannot be quasi-homomorphically operated because it is not modular q affine operation about hidden v (see the third restriction in subsection 2.9). So that the decryptor can only search e under the belief that $e \in (-Q/4, Q/4)$ with $Q = q$ polynomially large. The searching method is "lazy OR trick" [12], that is, applying multiple ABE decryptions. There are two directions for "lazy OR trick", the first direction is searching the differences, while the second direction is searching the quotients. The two directions are quite similar, so we only take the first one. For each $e \in (-Q/4, Q/4)$:

(1) Take the transformation "$\mathbf{C}_v \to \mathbf{C}_{v-2e}$" (this operation is an affine operation of the hidden v).

(2) Ask for the improved BGG+14 ABE decryption key k_e for the function f_e, where $f_e(\mathbf{m}^*, t) = v - 2e =< f^*(\mathbf{m}^*), t > \mod q - 2e$ (or $< f^{**}(\mathbf{m}^*), t > \mod q - 2e$, if the modular switching is applied).

(3) Decrypt the improved BGG+14 ABE ciphertext $(\mathbf{c}_{in}, \mathbf{C}_{v-2e}, \mathbf{c}_{1,out}, \mathbf{c}_{2,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$ by applying the key k_e.

It is easy to see that:

(1) The Agr17 FE decryption key for the Boolean function f is just the set of the improved BGG+14 ABE decryption keys $\{k_e, e \in (-Q/4, Q/4)\}$.

(2) If $f(\mathbf{m}) = 1$, there is only one key from $\{k_e, e \in (-Q/4, Q/4)\}$, such that corresponding BGG+14 ABE decryption is correct to obtain the public bit-string $\mathbf{b}$, while other BGG+14 ABE decryptions are fail, and obtain gibberishes. If $f(\mathbf{m}) = 0$, all BGG+14 ABE decryptions are fail. Reversely speaking, if one ABE decryption obtains $\mathbf{b}$, the FE decryptor obtains $f(\mathbf{m}) = 1$; if all ABE decryptions obtain other values, the FE decryptor obtains $f(\mathbf{m}) = 0$.

As a summarization, Table 1 illustrates the whole decryption process of the Agr17 FE scheme.

The whole decryption process is also shown in Fig.2.

4 On the P/poly Validity of the Agr17 FE Scheme

4.1 Starting Point: the Outer Modulus Should Be Super-polynomially Large

Randomly choose a group F of P/poly Boolean functions. The corresponding group of homomorphic operations of these Boolean functions is F^*. For each $f \in F$, the corresponding $f^* \in F^*$ is much large than f. Again for

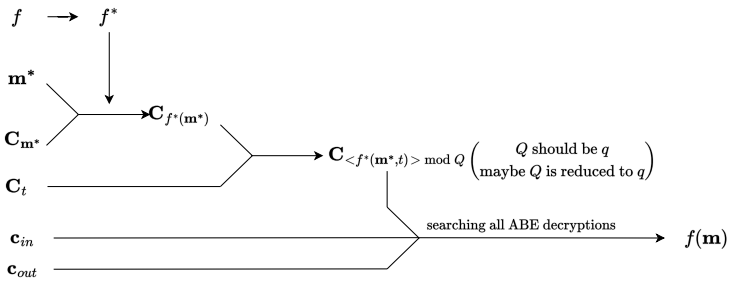

Fig. 2. The whole decryption process

Table 1. The whole decryption process of the Agr17 FE scheme

The step	The operation	The resulting text
The first step	$\mathbf{C}_{\mathbf{m}^*} \to \mathbf{C}_{f^*(\mathbf{m}^*)}$	$(\mathbf{c}_{in}, \mathbf{C}_{f^*(\mathbf{m}^*)}, \mathbf{C}_t, \mathbf{c}_{1,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$, $f^*(\mathbf{m}^*)$
Possible modular switching (if $Q \neq q$)	$\mathbf{C}_{f^*(\mathbf{m}^*)} \to \mathbf{C}_{f^{**}(\mathbf{m}^*)}$	$(\mathbf{c}_{in}, \mathbf{C}_{f^{**}(\mathbf{m}^*)}, \mathbf{C}_t, \mathbf{c}_{1,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$, $f^{**}(\mathbf{m}^*)$
The second step	$\left(\mathbf{C}_{f^*(\mathbf{m}^*)}, \mathbf{C}_t\right) \to \mathbf{C}_{<f^*(\mathbf{m}^*),t> \bmod Q}$	$(\mathbf{c}_{in}, \mathbf{C}_{<f^*(\mathbf{m}^*),t> \bmod Q}, \mathbf{c}_{1,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$, $f^*(\mathbf{m}^*)$
	$\left(\mathbf{C}_{f^{**}(\mathbf{m}^*)}, \mathbf{C}_t\right) \to \mathbf{C}_{<f^{**}(\mathbf{m}^*),t> \bmod q}$	$(\mathbf{c}_{in}, \mathbf{C}_{<f^{**}(\mathbf{m}^*),t> \bmod q}, \mathbf{c}_{1,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$, $f^{**}(\mathbf{m}^*)$
The third step	Searching by ABE decryptions	$f(\mathbf{m})$, $f^*(\mathbf{m}^*)$, maybe $< f^*(\mathbf{m}^*), t > \bmod Q$ (if $f(\mathbf{m}) = 1$)
		$f(\mathbf{m})$, $f^{**}(\mathbf{m}^*)$, maybe $< f^{**}(\mathbf{m}^*), t > \bmod q$ (if $f(\mathbf{m}) = 1$)

each $f^* \in F^*$, there is a corresponding group $\{f_e, e \in (-Q/4, Q/4)\}$, where $f_e(\mathbf{m}^*, t) =< f^*(\mathbf{m}^*), t > \mod Q - 2e$, therefore each f_e in the group is larger than such f^*. We take the group $F^{**} = \{f_e, e \in (-Q/4, Q/4), f \in F\}$, then as a bottom structure of the Agr17 FE scheme, the improved BGG+14 ABE scheme (subsection 3.2) is applied for the group F^{**}. According to the first restriction (subsection 2.7), the modulus of the improved BGG+14 ABE scheme, q (the outer modulus), should be adapted to being super-polynomially large.

4.2 P/poly Invalidity of the Agr17 FE Scheme

In the third step of the decryption process of the Agr17 FE scheme, the decryptor searches $e \in (-Q/4, Q/4)$ under the belief that Q is polynomially large (subsection 3.8). But at the beginning of this step, Q should be equal to q (subsection 3.7 and subsection 2.9), q is super-polynomially large (subsection 2.7 and subsection 4.1), and q cannot be switched (subsection 2.8).

Can the decryptor apply another modular switching to make Q polynomially large, just before the third step of the decryption process? At that moment, the BGG+14 ABE ciphertext is $(\mathbf{c}_{in}, \mathbf{C}_{<f^*(\mathbf{m}^*),t> \bmod q}, \mathbf{c}_{1,out}, \mathbf{c}_{2,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$ (or $(\mathbf{c}_{in}, \mathbf{C}_{<f^{**}(\mathbf{m}^*),t> \bmod q}, \mathbf{c}_{1,out}, \mathbf{c}_{2,out}, \cdots, \mathbf{c}_{k,out}, \mathbf{c}_{out})$), and modular switching is not affine operation of hidden $< f^*(\mathbf{m}^*), t > \mod q$ (or $< f^{**}(\mathbf{m}^*), t > \mod q$) about the outer modulus q. According to the third restriction (subsection 2.9), he cannot apply such modular switching.

From all of the above, Q cannot be polynomially large at the "searching" stage, so that the Agr17 FE scheme is $\mathsf{P/poly}$ invalid.

5 Cryptanalysis of Two Modified Versions

5.1 Cryptanalysis of the First Modified Version

The first modified version is described in the following five points.

(1) The outer modulus q is fixed to be super-polynomially large.

(2) In the first step of the decryption process (quasi-homomorphic operation of homomorphic evaluation), bootstrapping is repeatedly applied to control the FHE noise, hoping such noise polynomially large. Notice that bootstrapping should be quasi-homomorphically operated, so that the outer modulus q should be larger to accommodate both the ordinary quasi-homomorphic operation and such additional quasi-homomorphic operation.

(3) The inner modulus Q is fixed to be just able to accommodate the FHE noise, which is hoped to be polynomially large.

(4) In the second step of the decryption process (quasi-homomorphic operation of computing inner product), compute $\mathbf{C}_{<f^*(\mathbf{m}^*),t>}$ rather than $\mathbf{C}_{<f^*(\mathbf{m}^*),t>} \mod Q$, where $< f^*(\mathbf{m}^*), t >$ is arithmetic inner product. The purpose is to guarantee $< f^*(\mathbf{m}^*), t >=< f^*(\mathbf{m}^*), t > \mod q$, which is an affine operation about other modulus q.

(5) In the third step of the decryption process (searching the noise), search the interval $(-Q^\circ Q, Q^\circ Q)$ rather than $(-Q/4, Q/4)$, where Q° guarantees that the arithmetic inner product is just in the interval $(-Q^\circ Q, Q^\circ Q)$. Notice that, if Q is polynomially large, so is $Q^\circ Q$.

The effect of bootstrapping is to guarantee that the FHE noise is obtained by the homomorphic evaluation of an NC^1 function, rather than of a $\mathsf{P/poly}$ function. More detailedly, the FHE noise is obtained by the homomorphic evaluation of the FHE decryption function. However, the homomorphic evaluation of an NC^1 function does not guarantee the corresponding FHE noise polynomially large. So that we can only say that the first modified version may be $\mathsf{P/poly}$ valid.

For the (1, poly) collusion, the decryptor with the function value 1 obtains his $f(\mathbf{m}) = 1$, his $f^*(\mathbf{m}^*)$, and his $< f^*(\mathbf{m}^*), t >$, where $< f^*(\mathbf{m}^*), t >$ includes more knowledge of t than $< f^*(\mathbf{m}^*), t > \mod Q$. So we say that the first modified version is weaker than the original Agr17 FE scheme, although we have not constructed a new attack on it. In other words, this modified version needs a new proof of the security.

5.2 Cryptanalysis of the Second Modified Version

The idea of the second modified version is just same as in the GVW15 PE scheme [12], which is trying to search a public noise interval $(-B, B)$ rather than searching the modulus interval $(-Q/4, Q/4)$, hoping B polynomially large

and tolerating Q super-polynomially large. Can it be possible? Such possibility can only be checked for the following three cases of (Q, q).

Case I. Q is always equal to q, and all FHE evaluations don't apply bootstrapping.

Case II. Q is always equal to q, and some FHE evaluations may apply bootstrapping.

Case III. Q is not equal to q in the first step of the decryption process, then a modular switching is applied to make $Q = q$, then the second and third steps of the decryption process are implemented.

A special note: *Case I* and *Case II* depend on the "yes" answer to the question in subsection 2.6. If the answer to this question is "no", immediately both *Case I* and *Case II* are P/poly invalid. Now we take the "yes" answer.

For a randomly chosen P/poly Boolean function f, homomorphic evaluation f^* will make the noise of FHE ciphertext super-polynomially large if bootstrapping is not applied. This is a well-known fact. So that the noise interval $(-B, B)$ cannot be polynomially large, and *Case I* is P/poly invalid.

Bootstrapping is an FHE technique, by which the noise is reduced while the modulus is kept, so that subsequent homomorphic evaluation operations can be implemented. As we know, existing bootstrapping methods are not so powerful to let the noise polynomially large while the modulus is super-polynomially large. So that there is no evidence to support the P/poly validity of *Case II*.

A misunderstanding should be corrected that, if an FHE ciphertext under old modulus is switched to another FHE ciphertext under new modulus, the noise involved in the ciphertext is switched proportionally. The practical situation is that new noise may be added. If old modulus and new modulus are coprime, new noise is certainly added. Then how about the ratio of the modulus and the size of the noise, after modular switching? As we know, it should be polynomially large, by considering the security. More specifically, such ratio should be polynomially large after the modular switching, even it is super-polynomially large before the modular switching. Yes we can make such ratio super-polynomially large after the modular switching, by making some key variable much smaller, but it is not the recommended form.

Now let us go to *Case III*. q is a prime, so Q and q are coprime. If Q is switched to q, the ratio of the modulus q and the size of the corresponding noise should be polynomially large. On the other hand, q is super-polynomially large (subsection 2.7), so that the noise interval $(-B, B)$ cannot be polynomially large. This is a natural reasoning. So that there is no evidence to support the P/poly validity of *Case III*.

Finally, there is no evidence to support the P/poly validity of the second modified version.

5.3 On the P/poly Validity of the GVW15 PE Scheme [12]

PE is described by two different scenes, one is "an ABE with the attribute hidden from decryptors", while the other is "an ABE with the attribute hidden

from those decryptors whose function values are 0". GVW15 PE scheme [12] can match these two scenes according to two different collusion forms.

The GVW15 PE scheme [12] had the similar structure with the Agr17 FE scheme, with only the following four changes: the position of the public bit-string of the Agr17 FE was taken as that of the plaintext of the GVW15 PE; the position of the plaintext of the Agr17 FE was taken as that of the attribute of the GVW15 PE; the GVW15 PE took the original BGG+14 ABE (subsection 2.4) rather than the improved BGG+14 ABE (subsection 3.2); the "searching" method is the same as that of the second modified version (subsection 5.2).

From all of the above, we obtain such additional conclusion: there is no evidence to support the $\mathsf{P/poly}$ validity of the GVW15 PE scheme.

6 Our Response to an Argument

Some people say that our work is unnecessary, because Agr17 (GVW15) didn't hope to be applied for $\mathsf{P/poly}$ functions. Their evidence is that Agr17 (GVW15) clearly declared that the scheme was applied for functions of "depth d". Yes, that is true, and the necessity of our work would be questionable if there were only one such declaration, although the size of d was not clearly specified.

However, Agr17 (GVW15) also clearly declared that the scheme can use an FHE modular switching (both Agr17 and GVW15 called it "modulus reduction", in page 10 of [15] and page 7 of [12]). How to explain such declaration? The most natural explanation is that FHE modulus may be super polynomially large if FHE modular switching is not used. In other words, this indicates that Agr17 (GVW15) was not intended to be applicable only to shallow functions, which do not require FHE modular switching during the FHE evaluation stage. This is one motive of our work. Now how do we understand functions which Agr17 (GVW15) hoped to be applied for?

The first and most natural interpretation is that Agr17 (GVW15) was intended to be applicable to $\mathsf{P/poly}$ functions, where "depth d" can be polynomially large. Under this interpretation, our work is highly valuable.

The second interpretation is that Agr17 (GVW15) did not hope to be applied for $\mathsf{P/poly}$ functions, where "depth d" can only be a small number, although the scheme can use FHE modular switching. According to our work, functions for the second interpretation are restricted in such way: FHE modulus (inner modulus) may be super polynomially large if FHE modular switching is not used, but corresponding ABE modulus (outer modulus) is certainly polynomially large without ABE modular switching. This strong restriction is clear contribution of our work. What do shapes of functions under our strong restriction look like? Agr17 (GVW15) never described them, and it seems that Agr17 (GVW15) never found such strong restriction.

References

1. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
2. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45325-3_32
3. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_17
4. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
5. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
6. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G.,Vaikuntanathan, V., and Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE, and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
7. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS 2006. pp. 89–98. ACM (2006). https://doi.org/10.1145/1180405.1180418
8. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute based encryption for circuits. In: STOC 2013. pp. 545–554. ACM (2013). https://doi.org/ [DOI: 10.1145/2488608.2488677
9. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_27
10. Agrawal, S., Maitra, M., Yamada, S.: Attribute based encryption (and more) for nondeterministic finite automata from LWE. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11693, pp. 765–797. Springer, Cham (2019) https://doi.org/10.1007/978-3-030-26951-7_26
11. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy 2007. pp. 321–334. IEEE (2007). https://doi.org/10.1109/SP.2007.11
12. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 503–523. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_25
13. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_9
14. Datta, P., Okamoto, T., Takashima, K.: Adaptively simulation-secure attribute-hiding predicate encryption. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018. LNCS, vol. 11273, pp. 640–672. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_22

15. Agrawal, S.: Stronger security for reusable garbled circuits, general definitions and attacks. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 3–35. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_1
16. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_16
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206. ACM (2008) https://doi.org/10.1145/1374376.1374407
18. Agrawal, S., Rosen, A.: Functional encryption for bounded collusions, revisited. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 173–205. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_7
19. Agrawal, S., Libert, B., Maitra, M., Titiu, R.: Adaptive simulation security for inner product functional encryption. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020. LNCS, vol. 12110, pp. 34–64. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45374-9_2
20. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
21. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_2
22. Waters, B.: Functional encryption for regular languages. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 218–235. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_14
23. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS 2013. pp. 40–49. IEEE (2013). https://doi.org/10.1109/FOCS.2013.13
24. Lai, Q., Liu, FH., Wang, Z.: New lattice two-stage sampling technique and its applications to functional encryption – stronger security and smaller ciphertexts. In: Canteaut, A., Standaert, FX. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 498–527. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_18
25. Wang, Z., Fan X., Liu FH.: FE for inner products and its application to decentralized ABE. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11443, pp. 97–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17259-6_4
26. Ananth, P., Vaikuntanathan, V.: Optimal bounded-collusion secure functional encryption. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11891, pp. 174–198. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_8
27. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: ITCS 2012. pp. 309–325. ACM (2012). https://doi.org/10.1145/2090236.2090262
28. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
29. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: ITCS 2014. pp. 1–12. ACM (2014). https://doi.org/10.1145/2554797.2554799