密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research, 2022, 9(6): 1002–1013

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

Agr17 函数加密 (FE) 方案的 P/poly 无效性*

胡予濮[1], 刘 君[1], 王保仓[1], 董星廷[1], 潘彦斌[2]

1. 西安电子科技大学 综合业务网理论与关键技术国家重点实验室, 西安 710071
2. 中国科学院 数学与系统科学研究院 数学机械化重点实验室, 北京 100190
通信作者: 胡予濮, E-mail: yphu@mail.xidian.edu.cn

摘 要: 函数加密 (functional encryption, FE) 是密码研究领域的前沿课题, 而 Agr17 函数加密 (FE) 方案是主流 FE 方案之一. 该方案以 BGG+14 属性加密 (ABE) 为一个底层结构, 并将其改造为一个 "部分隐藏属性的谓词加密" (PHPE), 再与一个全同态加密 (FHE) 组合而成. 然而 Agr17 函数加密 (FE) 方案留下了一个问题, 即方案中的换模运算如何实现. 本文论述 Agr17 函数加密 (FE) 方案的 P/poly 无效性, Agr17 函数加密 (FE) 方案在解密阶段的换模之后无法继续运行. 指出 Agr17 函数加密 (FE) 方案的换模运算必须是双重换模, 即对全同态密文的换模和对全同态密文所寄生的属性密文的换模. 指出对全同态密文所寄生的属性密文的换模破坏了属性密文的结构, 使得其后的属性解密无法运行. 因为属性解密运算并不是普通的 LWE 解密, 而是附带条件的 LWE 解密, 换模则破坏了解密条件. 给出了一种 "自然的" 修改方案, 将小模内积换为算术内积, 由属性密文的模内积来实现算术内积. 修改方案可以正确解密, 但并不安全, 说明这种无效性并不容易通过修改方案而消失.

关键词: 带误差的学习问题 (LWE); 属性加密 (ABE); 函数加密 (FE)
中图分类号: TP309.7 **文献标识码:** A DOI: 10.13868/j.cnki.jcr.000563

中文引用格式: 胡予濮, 刘君, 王保仓, 董星廷, 潘彦斌. Agr17 函数加密 (FE) 方案的 P/poly 无效性[J]. 密码学报, 2022, 9(6): 1002–1013. [DOI: 10.13868/j.cnki.jcr.000563]

英文引用格式: HU Y P, LIU J, WANG B C, DONG X T, PAN Y B. P/poly invalidity of Agr17 functional encryption scheme[J]. Journal of Cryptologic Research, 2022, 9(6): 1002–1013. [DOI: 10.13868/j.cnki.jcr.000563]

P/poly Invalidity of Agr17 Functional Encryption Scheme

HU Yu-Pu[1], LIU Jun[1], WANG Bao-Cang[1], DONG Xing-Ting[1], PAN Yan-Bin[2]

1. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
2. Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
Corresponding author: HU Yu-Pu, E-mail: yphu@mail.xidian.edu.cn

Abstract: Functional encryption (FE) is an advanced topic in cryptography, and the Agr17 FE scheme is one of the well-known FE schemes. The Agr17 FE scheme takes a BGG+14 attribute-based

* 基金项目: 国家自然科学基金 (61972457, U19B2021); 陕西省重点研发计划 (2020ZDLGY08-04); 河南省创新型科技人才队伍建设工程

Foundation: National Natural Science Foundation of China (61972457, U19B2021); Key Research and Development Program of Shaanxi Province (2020ZDLGY08-04); Construction of Innovative Talent Team of Science and Technology, Henan Province

收稿日期: 2021-10-30 定稿日期: 2022-09-20

encryption (ABE) scheme as a base structure, which is upgraded into a "partially hiding predicate encryption" (PHPE) scheme and combined with a fully homomorphic encryption (FHE) scheme. However, the implementation of the modulus reduction is an unsolved problem in the Agr17 FE scheme. This study demonstrates that the Agr17 FE scheme is P/poly invalid. More specifically, it is shown that, in processing the P/poly function, the Agr17 FE scheme cannot be implemented any further after its modulus reduction. It is shown that the modulus reduction of the Agr17 FE scheme should be a double modulus reduction, which includes two modulus reductions for the FHE ciphertext and ABE ciphertext, respectively. It is also shown that the modulus reduction for the ABE ciphertext will destroy the structure of ABE so that the subsequent decryption cannot be executed. The reason lies in that, the decryption of ABE is an LWE decryption with conditions rather than an ordinary LWE decryption, and the modulus reduction will destroy the conditions of decryption. Moreover, a "natural revision" of the Agr17 scheme is designed, in which the small modulus inner product is changed into an arithmetic inner product, which can be obtained by the modulus inner product of the ABE ciphertext. The revised scheme can decrypt correctly, but it is insecure, which demonstrates that such invalidity cannot be easily crossed by revising the scheme.

Key words: learning with errors; attribute-based encryption; functional encryption

1 前言

1.1 问题和本文贡献

函数加密 (functional encryption, FE) 的场景是, 加密者将明文变换成为密文, 而解密者只能将密文反变换成为明文的一个函数值 (而不是明文本身). 函数加密 (FE) 是密码研究领域的前沿课题, 自从 Boneh 等人 [1] 做了形式化描述以后, 函数加密 (FE) 获得了巨大的进步 [2–20]. 其中 Agr17 函数加密 (FE) 方案 [19] 是主流 FE 方案之一. 该方案以 BGG+14 属性加密 (ABE) [21] 为一个底层结构, 并将其改造为一个 "部分隐藏属性的谓词加密" (PHPE), 再与一个全同态加密 (FHE) 组合而成. 随后的 LLW21 函数加密 (FE) 方案 [20] 是 Agr17 函数加密 (FE) 方案的改进版本, 具有更加优化的抽样特性和类似的结构. 此外, Agr17 方案也是众多工作 [22–32] 的参考文献.

然而 Agr17 函数加密 (FE) 方案留下了一个问题, 即方案中的换模运算如何实现. 以下详细叙述该问题.

首先, 换模是必须的. Agr17 函数加密 (FE) 方案需要穷举所有可能的噪声, 而噪声尺寸与模数的尺寸有相同的数量级. 因此, 只有将模数缩减为多项式级别, 才能使得穷举噪声是多项式时间可以完成的, 因而方案本身是多项式时间可计算的.

其次, 该方案 (包括会议版和完整版) 并没有详细描述换模运算的具体步骤, 仅仅指向了几个经典的参考文献 [33–35], 似乎直接引用或简单扩展这些参考文献就能得到 Agr17 函数加密 (FE) 的换模运算.

其三, 这些经典的参考文献 [33–35] 都仅仅描述了全同态加密 (FHE) 的各种换模运算, 并没有任何提及或暗示如何扩展为函数加密 (FE) 的换模运算. 我们曾经尝试了很多种方法, 试图将全同态加密 (FHE) 的各种换模运算 "自然而然地" 扩展为函数加密 (FE) 的换模运算, 始终没有成功.

最后, 全同态加密 (FHE) 的任何一种换模运算都不能直接或简单扩展成为 Agr17 函数加密 (FE) 的换模运算. 其理由为如下两个事实: 事实一, Agr17 函数加密 (FE) 方案有两个模, 一个是全同态密文的模 (我们可以称之为内模或隐模), 另一个是全同态密文所寄生的属性密文的模 (我们可以称之为外模或显模). 在应对 P/poly 函数时, 这两个模原本都应该是超多项式级别的, 因为两个模都用来处理 P/poly 函数, 故所积累的噪声尺寸都是超多项式级别的. 事实二, 这两个模 (内模和外模) 都需要缩减为多项式级别, 而且都需要缩减为相同的模, 否则方案的正确性不成立, 因为基于的 BGG+14 属性加密 (ABE) 对属性隐藏时的运算限制极为严格. 本文将详细叙述这两个事实.

本文论述 Agr17 函数加密 (FE) 方案的 P/poly 无效性, 即该方案对于 P/poly 函数无法运行. 具体说就是, Agr17 函数加密 (FE) 方案在解密阶段的换模之后无法继续运行. 本文分为两个阶段来论述.

第一阶段, 本文指出一次换模 (即只进行内模换模) 是行不通的, 它使得其后的模内积无法实现, 因此 Agr17 函数加密 (FE) 方案的换模运算必须是双重换模, 即对全同态密文的换模和对全同态密文所寄生的属性密文的换模. 双重换模的要点如下: (1) 对全同态密文进行的换模运算, 要理解为一系列布尔运算, 并且要转化为 "属性拟同态运算". (2) 对全同态密文所寄生的属性密文进行的换模运算, 是普通的 LWE 结构下的换模. (3) 两次换模运算都要换为同一个模, 否则无法继续运行.

第二阶段, 本文指出, 对全同态密文所寄生的属性密文的换模破坏了属性密文的结构, 使得其后的属性解密无法运行. 我们知道, 普通的 LWE 解密之前是可以对密文换模的 (虽然有一些限制条件, 比如某些量必须是小量). 但是 (BGG+14 方案的) 属性解密运算并不是普通的 LWE 解密, 而是附带条件的 LWE 解密, 这些条件被嵌入在属性密文的结构中. 换模则破坏了属性密文的结构, 因而破坏了解密条件.

此外, 为了说明这种无效性并不容易通过修改方案而消失, 我们给出了一种 "自然的" 修改方案. 方案的要点是将小模内积换为算术内积, 而特殊的算术内积是可以用属性密文的拟同态运算来实现的. 修改方案克服了无效性, 即修改方案可以正确解密. 但修改方案是不安全的, 因为解密者获得了多得多的秘密, 用代价小得多的共谋攻击就可以攻破修改方案.

1.2 内容安排

本文内容安排如下. 第 2 节回顾 Agr17 函数加密 (FE) 方案的底层结构: BGG+14 属性加密 (ABE) 方案 [21]. 我们在该节中做了大量的注解, 其中包括 BGG+14 方案的各种细节, 特别是解释了为何 BGG+14 方案应对 P/poly 函数时必须将模设定为超多项式大, 以及部分隐藏属性时对运算的限制. 第 3 节回顾 Agr17 函数加密 (FE) 方案 [19]. 我们在该节中也有大量的注解, 其中包括换模运算的特殊处理思路. 第 4 节叙述本文的实质贡献, 即论述 Agr17 函数加密 (FE) 方案的 P/poly 无效性. 首先说明应对 P/poly 函数时, 换模运算必须是双重换模, 并给出了双重换模 (不得不采用的) 技术细节, 然后指出对外模的换模运算破坏了属性加密 (ABE) 的条件结构, 使得后续运算无法进行. 第 5 节给出一种 "自然的" 修改方案, 说明修改方案能够正确解密, 但重点说明修改方案是不安全的.

2 Agr17 函数加密 (FE) 方案的底层结构: BGG+14 属性加密 (ABE)

2.1 基本符号和基本运算

设三个正整数 (m,n,q) 满足 $q=n^{\Theta(d_{\max})}$, q 为素数, $m=n\lceil\log_2 q\rceil$, 其中 $d_{\max}$ 是一个安全参数. 设 $\mathbb{Z}$ 为整数集合. 对于两个正整数 (m',m''), $(\mathbb{Z}^{m'},\mathbb{Z}^{m'\times m''},\mathbb{Z}_q^{m'},\mathbb{Z}_q^{m'\times m''})$ 的含义是清晰的. 此处 "$\bmod q$" 运算的结果属于 $\{\lceil -\frac{q}{2}\rceil+1,\lceil -\frac{q}{2}\rceil+2,\cdots,\lceil\frac{q}{2}\rceil-1\}$, 而不是属于 $\{0,1,\cdots,q-1\}$. 对于 $a\in\mathbb{Z}_q$, $\boldsymbol{A}\in\mathbb{Z}_q^{m'\times m''}$, $a\boldsymbol{A}\in\mathbb{Z}_q^{m'\times m''}$ 的含义是清晰的. 设 $\boldsymbol{G}$ 是如下的特殊矩阵

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

对任何 $\alpha\in\mathbb{Z}_q$, 存在唯一的布尔矩阵 $\boldsymbol{G}^{(\alpha)}\in\mathbb{Z}^{m\times m}$, 满足

$$\alpha\boldsymbol{G}=\boldsymbol{G}\boldsymbol{G}^{(\alpha)}(\bmod\ q).$$

对任何 $\boldsymbol{B}\in\mathbb{Z}_q^{n\times m}$, 存在唯一的布尔矩阵 $\boldsymbol{G}^{(\boldsymbol{B})}\in\mathbb{Z}^{m\times m}$, 满足

$$\boldsymbol{B}=\boldsymbol{G}\boldsymbol{G}^{(\boldsymbol{B})}(\bmod\ q).$$

以下三个算法是成熟的公共算法.

- $\mathsf{TrapGen}(n,m,q)$: 输入 (n,m,q), 输出 $(\boldsymbol{A},\boldsymbol{T})$, 其中 $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$ 是均匀矩阵, $\boldsymbol{T}\in\mathbb{Z}^{m\times m}$ 是小尺

寸的高斯矩阵, $\boldsymbol{AT}=\boldsymbol{0}\in\mathbb{Z}_q^{n\times m}$, $\boldsymbol{T}$ 是满秩的. 当然 $\boldsymbol{T}$ 关于模 q 并不是满秩的. $\boldsymbol{T}$ 称为 $\boldsymbol{A}$ 的陷门.

- $\mathsf{Encode}(\boldsymbol{A},\boldsymbol{s})$: 输入 $(\boldsymbol{A},\boldsymbol{s})\in\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^n$, 输出 $\boldsymbol{\psi}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}\in\mathbb{Z}_q^m$, 其中 $\boldsymbol{e}\in\mathbb{Z}^m$ 是小尺寸的高斯向量. $\boldsymbol{s}$ 称为被编码的原始向量, $\boldsymbol{\psi}$ 为 $\boldsymbol{s}$ 的码字, $\boldsymbol{e}$ 称为噪声向量. 我们称 $\boldsymbol{\psi}=\mathsf{Encode}(\boldsymbol{A},\boldsymbol{s})$.
- $\mathsf{ReKeyGen}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{T},\boldsymbol{D})$: 输入 $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{T},\boldsymbol{D})$, 输出 $\boldsymbol{R}$, 其中 $\boldsymbol{A},\boldsymbol{B}\in\mathbb{Z}_q^{n\times m}$ 是均匀矩阵, $\boldsymbol{T}\in\mathbb{Z}^{m\times m}$ 是 $\boldsymbol{A}$ 的陷门, $\boldsymbol{R}\in\mathbb{Z}^{2m\times m}$ 是小尺寸的高斯矩阵, $\boldsymbol{D}=[\boldsymbol{A},\boldsymbol{B}]\boldsymbol{R}\in\mathbb{Z}_q^{n\times m}$. 实际上,

$$\boldsymbol{R}=\begin{bmatrix}\boldsymbol{R}_0\\ \boldsymbol{R}_1\end{bmatrix},\boldsymbol{R}_i\in\mathbb{Z}^{m\times m},i=0,1,$$

则 $\boldsymbol{R}_1$ 是预先抽样矩阵, $\boldsymbol{R}_0$ 是对应抽样矩阵, 使用陷门矩阵 $\boldsymbol{T}$ 满足 $\boldsymbol{AR}_0=\boldsymbol{D}-\boldsymbol{BR}_1$.

2.2 布尔函数的算术表示和大模表示

为了使 BGG+14 属性加密方案成立, 需要将布尔函数表示为 mod q 函数 (即大模函数). 这是容易的, 只要将每个布尔运算先转化为算术运算, 再转化为大模运算. 对于两个比特变量 x_1 和 x_2,

$$\begin{aligned}&x_1\cdot x_2(\mathrm{mod}\ 2)=x_1\cdot x_2=x_1\cdot x_2(\mathrm{mod}\ q),\\&x_1+x_2(\mathrm{mod}\ 2)=x_1+x_2-2x_1\cdot x_2=x_1+x_2-2x_1\cdot x_2(\mathrm{mod}\ q).\end{aligned}$$

推而广之, 布尔函数的每一步运算都可以转化为大模运算. 因此布尔函数就表示为 mod q 函数 (即大模函数), 唯一不同的是 "自变量" 的变化范围是比特域而不是大模域.

2.3 BGG+14 属性加密 (ABE) 的拟同态运算

设 l 维属性变量为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_l)$, 其中每个 x_i 是比特变量. 取 l 个矩阵 $\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_l\in\mathbb{Z}_q^{n\times m}$. 再取另外 l 个矩阵 $x_1\boldsymbol{G}+\boldsymbol{B}_1,x_2\boldsymbol{G}+\boldsymbol{B}_2,\cdots,x_l\boldsymbol{G}+\boldsymbol{B}_l\in\mathbb{Z}_q^{n\times m}$. 以下说明, 对任何 P/poly 布尔函数 $f(\boldsymbol{x})$, 都有对以上矩阵的 "小尺寸的线性组合运算", 得到以下的新矩阵

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

其中 $\boldsymbol{B}_f$ 与 $\boldsymbol{x}$ 的值无关. 由第 2.2 节注意到, 任何布尔运算都可看作 $\mathbb{Z}_q$ 上的运算组合, 任何布尔函数都可看作 $\mathbb{Z}_q$ 上的函数. 不仅如此, 这个特殊的 $\mathbb{Z}_q$ 上的函数的每一步运算结果都在区间 $[-2,2]$ 之内. 首先考虑如下四种简单情形.

情形一, 如 $f(\boldsymbol{x})=\alpha x_i$, 其中 α 为常数, 则 "小尺寸的线性组合运算" 为

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

其中 $\boldsymbol{B}_f=\boldsymbol{B}_i\boldsymbol{G}^{(\alpha)}$.

情形二, 如 $f(\boldsymbol{x})=x_i+x_j$, 则 "小尺寸的线性组合运算" 为

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

其中 $\boldsymbol{B}_f=\boldsymbol{B}_i+\boldsymbol{B}_j$.

情形三, 如 $f(\boldsymbol{x})=x_i\cdot x_j$, 其中 $i\le j$, 则 "小尺寸的线性组合运算" 为

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

其中 $\boldsymbol{B}_f=-\boldsymbol{B}_j\boldsymbol{G}^{(\boldsymbol{B}_i)}$.

情形四, 如 $f(\boldsymbol{x})=\alpha\cdot x_{j_1}\cdot x_{j_2}\cdot\cdots x_{j_k}, j_1\le j_2\le\cdots j_k$, α 是常数, 则 "小尺寸的线性组合运算" 为

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

其中 $\boldsymbol{G}_1,\boldsymbol{G}_2,\cdots,\boldsymbol{G}_k$ 都是 $\mathbb{Z}^{m\times m}$ 上的布尔矩阵, 递归定义如下:

$$\begin{aligned}\boldsymbol{G}_1&=\boldsymbol{G}^{\alpha},\\ \boldsymbol{G}_i&=\boldsymbol{G}^{(-\boldsymbol{B}_{j_{i-1}}\boldsymbol{G}_{i-1})},\end{aligned}$$

其中, $i=2,3,\cdots,k$, $\boldsymbol{B}_f=-\boldsymbol{B}_{j_k}\cdot\boldsymbol{G}$ 也与 $\boldsymbol{x}$ 无关.

最后, 我们认定, 只要反复的次数是多项式级别, 反复进行 "小尺寸的线性组合" 仍然是 "小尺寸的线性组合". 于是, 由以上四种运算重复进行, 就得到以下结论: 任何 P/poly 布尔函数 f 都可以对以上矩阵做 "小尺寸的线性组合", 得到 $f(\boldsymbol{x})\boldsymbol{G}+\boldsymbol{B}_f$.

然后做如下编码:

$$\begin{aligned}\boldsymbol{c}_1&=\mathsf{Encode}(x_1\boldsymbol{G}+\boldsymbol{B}_1,\boldsymbol{s}),\\ \boldsymbol{c}_2&=\mathsf{Encode}(x_2\boldsymbol{G}+\boldsymbol{B}_2,\boldsymbol{s}),\\ &\vdots\\ \boldsymbol{c}_l&=\mathsf{Encode}(x_l\boldsymbol{G}+\boldsymbol{B}_l,\boldsymbol{s}).\end{aligned}$$

对码字 $(\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_l)$ 做相同的 "小尺寸的线性组合" (仅仅多一个转置), 将得到

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

我们称对 $(\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_l)$ 的这种 "小尺寸的线性组合" 为关于布尔函数 f 的拟同态运算. 需要特别提及 BGG+14 方案的两个设计技巧. 技巧一: 上述的情形三表明, 在做乘法运算的拟同态运算时, 噪声的扩张形式大约为一个原噪声乘一个比特阵, 而不是两个原噪声的某种乘积. 这极大地降低了噪声的扩张速度. 技巧二: 上述的情形三是情形四的特例, 然而情形四并不是情形三的重复操作. 这个设计的意思是, 在做连乘运算的拟同态运算时, 用一次情形四比重复用情形三得到的噪声小得多.

一个原噪声乘一个随机的比特阵, 新噪声尺寸是原噪声尺寸的约 $\sqrt{\frac{m}{2}}$ 倍. 因此在做乘法运算的拟同态运算时, 得到的新噪声尺寸至少是其中一个原噪声尺寸的 $\sqrt{\frac{m}{2}}$ 倍; 在做连乘运算的拟同态运算 (即上述的情形四) 时, 得到的新噪声尺寸也至少是其中一个原噪声尺寸的 $\sqrt{\frac{m}{2}}$ 倍. 在一个 P/poly 函数的各步运算中, 连乘运算并不多见, 而且相邻两步运算即使都是乘法运算, 也不见得可以合并为连乘运算. 这就是说, 连乘运算的拟同态运算 (即上述的情形四) 对噪声扩张的抑制效果是有限的. 综上所述, BGG+14 方案对 P/poly 函数的拟同态运算使得最终的噪声尺寸可以达到超多项式. 因此, 应对 P/poly 函数的 BGG+14 方案必须将模 q 设定为超多项式大.

2.4 BGG+14 属性加密 (ABE) 方案 [21]

- 生成主钥 $(\mathbf{mpk},\mathbf{msk})$: 运行 $\mathsf{TrapGen}(n,m,q)$ 得到 $(\boldsymbol{A},\boldsymbol{T})$. 随机选 $\boldsymbol{B}_i\in\mathbb{Z}_q^{n\times m}$, $i=1,2,\cdots,l$, $\boldsymbol{D}\in\mathbb{Z}_q^{n\times m}$. 输出
$$\mathbf{mpk}=(\boldsymbol{A},\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{D}),\ \mathbf{msk}=\boldsymbol{T}.$$
- 生成布尔函数 f 的解密钥 $\mathbf{sk}_f$: 首先生成 $\boldsymbol{B}_f$ (请注意, $\boldsymbol{B}_f$ 由 2.3 节的方法生成, 其中属性值任意选取, 所得到的 $\boldsymbol{B}_f$ 与属性值无关), 然后运行 $\mathsf{ReKeyGen}(\boldsymbol{A},y_0\boldsymbol{G}+\boldsymbol{B}_f,\boldsymbol{T},\boldsymbol{D})$ 得到 $\boldsymbol{R}\in\mathbb{Z}^{2m\times m}$.

令

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

- 加密: 明文 $\boldsymbol{m}$ 是 m-维布尔向量. 属性 $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$ 被发送给加密者. 加密者随机选 $\boldsymbol{s} \in \mathbb{Z}_q^n$, 计算 (Encode$(\boldsymbol{A}, \boldsymbol{s})$, Encode$(x_1\boldsymbol{G} + \boldsymbol{B}_1, \boldsymbol{s})$, Encode$(x_2\boldsymbol{G} + \boldsymbol{B}_2, \boldsymbol{s}), \cdots,$ Encode$(x_l\boldsymbol{G} + \boldsymbol{B}_l, \boldsymbol{s})$, Encode$(\boldsymbol{D}, \boldsymbol{s})$). 于是密文为

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

- 解密: 解密者用自己的 f 和属性值 $\boldsymbol{x} = (x_1, x_2, \cdots, x_l)$ 对 $(\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_l)$ 做拟同态运算得到

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

然后解密者用解密钥 $\mathbf{sk}_f = \boldsymbol{R}$ 计算

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

当 $f(\boldsymbol{x}) = y_0$ 时, 使用 "Rounding" 即得明文 $\boldsymbol{m}$; 当 $f(\boldsymbol{x}) \neq y_0$ 时, 得乱码. 通常取 $y_0 = 1$.

2.5 BGG+14 属性加密 (ABE) 方案中隐藏属性值

所谓 "隐藏属性值" 指的是只有加密者知道属性值, 解密者不知道. 关键问题是解密者在不知道一部分属性值的情况下能否做 f 的拟同态运算.

容易看出, 当 f 是 mod q 线性函数时, 解密者完全不需要知道属性值 $\boldsymbol{x}$ 就能完成拟同态运算. 然而 2.2 节告诉我们, 任何布尔函数都不是 mod q 线性函数. 换句话说, 任何布尔函数作为一个 mod q 函数, 不但含有 mod q 加法运算, 也含有 mod q 乘法运算. 而当做乘法运算时 (即做乘法运算的拟同态运算时), 一个属性比特可以隐藏, 另一个属性比特必须告诉解密者.

为了支持本文的分析, 我们将函数 f 做了扩展, 给出以下两种情形.

情形五, 设解密者知道 $t'\boldsymbol{G} + \boldsymbol{B}'$ 和 $v\boldsymbol{G} + \boldsymbol{B}''$, 知道比特值 v, 不知道比特值 t', $f(t', v) = t' \cdot v \cdot 2^{k'} \pmod p$, 其中模 $p < q$. 则 "小尺寸的线性组合运算" 为

$$v(t'\boldsymbol{G} + \boldsymbol{B}') \cdot \boldsymbol{G}^{(2^{k'} \pmod p)} - (v\boldsymbol{G} + \boldsymbol{B}'') \cdot \boldsymbol{G}^{(\boldsymbol{B}'\boldsymbol{G}^{(2^{k'} \pmod p)})} = f(t', v) \cdot \boldsymbol{G} + \boldsymbol{B}_f,$$

其中 $\boldsymbol{B}_f = -\boldsymbol{B}'' \cdot \boldsymbol{G}^{(\boldsymbol{B}'\boldsymbol{G}^{(2^{k'} \pmod p)})}$ 与 (t', v) 无关. 虽然解密者可以进行这个 $f(t', v)$ 的拟同态运算, 他并不知道 $f(t', v)$ 的值 (当 $v = 0$ 时他知道 $f(t', v) = 0$, 当 $v = 1$ 时他知道 $f(t', v)$ 取值为 0 和 $2^{k'} \pmod p$ 两者之一).

情形六, 设解密者知道 $a\boldsymbol{G} + \boldsymbol{B}'$ 和 $b\boldsymbol{G} + \boldsymbol{B}''$, 但不知道在 $(-\frac{p}{2}, \frac{p}{2})$ 之间的两个算术值 a 和 b, $f(a, b) = a + b \pmod p$, 其中模 $p < q$. 则无法找到 $a\boldsymbol{G} + \boldsymbol{B}'$ 和 $b\boldsymbol{G} + \boldsymbol{B}''$ 的 "小尺寸的线性组合运算" 使得结果为 $f(a, b) \cdot \boldsymbol{G} + \boldsymbol{B}_f$, 其中 $\boldsymbol{B}_f$ 与 (a, b) 无关. 这就是说, BGG+14 方案的扩展结构不支持这个 $f(a, b)$ 的拟同态运算. 为什么呢? 因为 $a + b \pmod p$ 要转化为一系列的模 q 运算才能进行拟同态. 这一系列模 q 运算中不可能仅仅包含模 q 加法而不包含模 q 乘法. 而对于模 q 乘法, 两边的值都未知时是无法进行拟

同态的.

3 Agr17 函数加密 (FE) 方案

3.1 Agr17 函数加密 (FE) 方案的概述 [19]

对于明文 u, 加密者的加密过程为以下两步. 第一步, 使用一个全同态加密 (FHE) 方案的加密程序, 将 u 加密成为全同态密文 u^*. 第二步, 将全同态密文 u^* 作为属性的公开部分, 将全同态加密 (FHE) 方案的解密钥 t 作为属性的隐藏部分, 对属性 (u^*,t) 使用 BGG+14 属性加密 (ABE) 方案, 将一个公开的 "形式明文" $\boldsymbol{m}$ 加密, 所得到的 "属性密文" $\boldsymbol{C}$ 就是 Agr17 函数加密 (FE) 方案的密文.

此时解密者的已知项为以下 4 项:

(1) Agr17 函数加密 (FE) 方案的密文 $\boldsymbol{C}$ (实际上是 "属性密文" $\boldsymbol{C}=(\boldsymbol{c}_{\text{in}},\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_l,\boldsymbol{c}_{\text{out}})$), 而且 $(\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_l)=(\boldsymbol{C}_{u^*},\boldsymbol{C}_t)$, 其中 $\boldsymbol{C}_{u^*}$ 是对应属性的公开部分 u^* 的 "属性密文", $\boldsymbol{C}_t$ 是对应属性的隐藏部分 t 的 "属性密文";

(2) 全同态密文 u^*(实际上是属性的公开部分);

(3) 公开的 "形式明文" $\boldsymbol{m}$;

(4) 布尔函数 f 所对应的 Agr17 函数加密 (FE) 方案的解密钥 (实际上就是复合函数 Df^* 所对应的 BGG+14 属性加密 (ABE) 方案的解密钥, 其中 f^* 是 f 的同态运算, D 是全同态解密运算. 换句话说, $Df^*(u^*,t)=D(f^*(u^*),t)=f(u)$). 解密者既不知道明文 u, 也不知道全同态加密 (FHE) 方案的解密钥 t. 在此限制下, 解密者需要从中求解明文 u 的函数值 $f(u)$.

解密者的粗略解密过程如下. 他使用自己的 Agr17 函数加密 (FE) 方案的解密钥 (即 BGG+14 属性加密 (ABE) 方案的解密钥), 对密文 $\boldsymbol{C}$ 进行 "函数解密" (即进行 "属性解密"), 如果 "属性解密" 所得到的形式明文等于 $\boldsymbol{m}$, 则认为 $f(u)=1$; 否则认为 $f(u)=0$.

注意到密文 $\boldsymbol{C}$ 包含两个部分 $\boldsymbol{C}=(\boldsymbol{C}_{u^*},\boldsymbol{C}_t)$. 于是解密者的细致解密过程为以下三步.

第一步, 使用 f 的同态运算 f^*, 将 $\boldsymbol{C}_{u^*}$ 变换为新密文 $\boldsymbol{C}_{f^*(u^*)}$, 其中 $\boldsymbol{C}_{f^*(u^*)}$ 是对应新属性 $f^*(u^*)$ 的 "属性密文". 换句话说, 解密者做的运算是 "f 的同态运算的拟同态运算"(见第 2.3 节).

第二步, 将 $\boldsymbol{C}_{f^*(u^*)}$ 和 $\boldsymbol{C}_t$ 进行运算得到最终密文 $\boldsymbol{C}_{D(f^*(u^*),t)}=\boldsymbol{C}_{f(u)}$, 其中 $\boldsymbol{C}_{f(u)}$ 是对应最终属性 $f(u)$ 的 "属性密文". 换句话说, 解密者做的运算是 "同态解密运算的拟同态运算"(见第 2.3 节).

第三步, 使用自己的 BGG+14 属性加密 (ABE) 方案的解密钥, 对最终密文 $\boldsymbol{C}_{f(u)}$ 进行 "属性解密", 所得到的形式明文如果等于 $\boldsymbol{m}$, 则认为最终属性 $f(u)=1$; 否则认为最终属性 $f(u)=0$.

3.2 关于 Agr17 函数加密 (FE) 方案的同态解密运算

设同态解密运算的前期模为 Q. 根据 Agr17 函数加密 (FE) 方案的并不清晰的描述, 所使用的同态解密运算原本应该是以下的算法 1 或算法 2.

算法 1: 第一步, $f^*(u^*)$ 和 t 做模 Q 内积. 第二步, 做模 2 运算.

算法 2: 第一步, $f^*(u^*)$ 和 t 做模 Q 内积. 第二步, 检查与 0 近还是与 $Q/2$ 近, 与 0 近则取值为 0, 与 $Q/2$ 近则取值为 1.

可知, 算法 1 和算法 2 在做完第一步以后, 分别得到如下课文. 前者为: $2\times$ 小误差 + 明文比特; 后者为: 小误差 + $\lceil$模/2$\rfloor\times$ 明文比特. 容易看出, 后者乘以 2 就是前者. 因此算法 1 和算法 2 是等价的. 然而, 算法 1 或算法 2 都无法实现, 因为 Agr17 函数加密 (FE) 方案的解密者并不是直接做算法 1 或算法 2, 只是做算法 1 或算法 2 的拟同态运算. 由于解密者不知道 t, 因此当第一步的拟同态运算完成以后, 解密者并不能得到内积值, 只能得到内积值的拟同态运算结果. 在这种情况下, 第二步的拟同态运算无法完成, 因为第二步不是模 q 加法 (见第 2.3 节和第 2.5 节, 此处 q 是 BGG+14 方案的模). 基于全同态技术中的换模技术, Agr17 函数加密 (FE) 方案的同态解密运算采用了一种 "迂回战术", 即以下的算法 3.

算法 3: 第一步, 对 $f^*(u^*)$ 做换模运算, 将模 Q 降为多项式尺寸的模 p, $f^*(u^*)$ 也因此变成了新的全同态密文 $f^{**}(u^*)$. 第二步, $f^{**}(u^*)$ 与 t 做小模 p 的内积. 第三步, 穷举, 将每个 $i\in\{1,2,\cdots,p-1\}$ 都做与这个内积的模 p 乘积. 第四步, 对每个模 p 乘积都需要一个 BGG+14 方案的解密钥, 做属性解密, 查

看形式明文是否等于 $\boldsymbol{m}$. 换句话说, 查看每个模 p 乘积是否等于 1. 第五步, 到此为止, 解密者已经精确知道第二步的结果. 于是他对第二步的结果做模 2 运算, 得到 $f(u)$ 的值 (或检查与 0 近还是与 $p/2$ 近, 与 0 近则 $f(u)$ 取值为 0, 与 $p/2$ 近则 $f(u)$ 取值为 1).

按照 Agr17 函数加密 (FE) 方案的理解, 算法 3 的每一步运算都可以在 BGG+14 方案之下做拟同态运算, 而并不需要知道 t 的值. 最关键的是用 "换模 + 穷举" 替换了原来的 "模 2 运算". 当然, 这里还有很多的细节问题需要回答, 但都不是本质问题. 比如为什么 t 不需要做换模处理? 这是因为 t 用比特分量表示. 再如, $f^*(u^*)$ 也可以用比特分量表示, 为什么 $f^*(u^*)$ 必须换模呢? 这是全同态换模技术的一个基本性质, 简单说就是 t 不需做换模处理, $f^*(u^*)$ 必须做换模处理. 好在解密者知道 $f^*(u^*)$ 的值, 因此对 $f^*(u^*)$ 的换模运算的拟同态运算能够完成.

3.3 关于 Agr17 函数加密 (FE) 方案的多钥性的处理

为了使方案在安全性前提下灵活提供多个解密钥, Agr17 函数加密 (FE) 方案对 BGG+14 方案做了如下扩展. 随机选择 k 个矩阵 $\boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}, \cdots, \boldsymbol{D}^{(k)} \in \mathbb{Z}_q^{n\times m}$, 其中 k 是多项式大, 但要充分大. 所有解密者都不知道 $\{\boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}, \cdots, \boldsymbol{D}^{(k)}\}$, 但每个解密者知道自己的和矩阵 $\sum_{i\in\Delta}\boldsymbol{D}^{(i)}(\mathrm{mod}\ q)$, 其中 Δ 是 $\{1,2,\cdots,k\}$ 的子集, 不同的解密者有不同的子集 Δ. 每个解密者视自己的和矩阵 $\sum_{i\in\Delta}\boldsymbol{D}^{(i)}(\mathrm{mod}\ q)$ 为 BGG+14 方案中的矩阵 $\boldsymbol{D}$.

加密时, $\boldsymbol{c}_{\mathrm{out}}$ 扩展为如下形式:

$$\boldsymbol{c}_{\mathrm{out}} = (\mathsf{Encode}(\boldsymbol{D}^{(1)}, \boldsymbol{s}) + \left\lceil\frac{q}{2}\right\rceil \boldsymbol{m}^{(1)}, \mathsf{Encode}(\boldsymbol{D}^{(2)}, \boldsymbol{s}) + \left\lceil\frac{q}{2}\right\rceil \boldsymbol{m}^{(2)}, \cdots, \mathsf{Encode}(\boldsymbol{D}^{(k)}, \boldsymbol{s}) + \left\lceil\frac{q}{2}\right\rceil \boldsymbol{m}^{(k)}).$$

加密者还公布 $\{\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \cdots, \boldsymbol{m}^{(k)}\}$. 解密时, 知道 $\boldsymbol{D} = \sum_{i\in\Delta}\boldsymbol{D}^{(i)}$ 和 Δ 的解密者同时知道 $\boldsymbol{m} = \sum_{i\in\Delta}\boldsymbol{m}^{(i)}(\mathrm{mod}\ 2)$, 而且他还知道

$$\begin{aligned}\sum_{i\in\Delta}(\mathsf{Encode}(\boldsymbol{D}^{(i)}, \boldsymbol{s}) + \left\lceil\frac{q}{2}\right\rceil \boldsymbol{m}^{(i)}) &= \mathsf{Encode}\left(\sum_{i\in\Delta}\boldsymbol{D}^{(i)}, \boldsymbol{s}\right) + \left\lceil\frac{q}{2}\right\rceil \left(\sum_{i\in\Delta}\boldsymbol{m}^{(i)}(\mathrm{mod}\ 2)\right)\\ &= \mathsf{Encode}(\boldsymbol{D}, \boldsymbol{s}) + \left\lceil\frac{q}{2}\right\rceil \boldsymbol{m}.\end{aligned}$$

4 Agr17 函数加密 (FE) 方案的无效性

4.1 我们的发现: 双重换模

现在紧盯 Agr17 函数加密 (FE) 方案的同态解密运算, 即 3.2 节中的算法 3. 首先查看算法 3 的第一步, 换模运算. 尽管有 Agr17 函数加密 (FE) 方案的描述 [19] 和本文第 3 节极尽详细的解读, 我们仍然发现了一个问题, 那就是在换模之前有两个模, 一个是全同态加密的前期模 Q, 另一个是 BGG+14 方案的前期模 q. Q 和 q 分别称为内模和外模, 内模用于全同态运算, 外模用于全同态运算的拟同态运算. 内模运算可以表示为一系列布尔运算, 因此可以对其进行拟同态运算, 并用外模运算来表示这个拟同态运算. 对于 Agr17 方案所称的换模运算, 我们当然应该理解为内模换模. 而且内模换模是需要用一系列外模运算来实现的 (即内模换模要化为拟同态运算), 因此当内模换模时外模保持不变.

然后进行算法 3 的第二步, t 与 $f^{**}(u^*)$ 的小模 p 内积. 注意到 t 和 $f^{**}(u^*)$ 都是用比特表示的, 每个比特有一个编码. 于是算法 3 的第二步又分成以下的两小步. 第一小步, t 的每个比特、$f^{**}(u^*)$ 的对应比特、2 的某个幂三者做模 p 乘积, 即第 2.5 节中的情形五. 第二小步, 将所有这些模 p 乘积再做模 p 和, 即第 2.5 节的情形六. 情形六是不可进行拟同态运算的, 因此算法 3 的第二步是无法完成的. 当然, 如果 $p = q$, 算法 3 的第二步是能完成的, BGG+14 方案支持 (见第 2.5 节).

另一个严重的障碍是, 当 f 是 P/poly 函数时, f^* 也是 P/poly 函数, 因此 Q 和 q 分别是超多项式大的 (见第 2.3 节的最后一段). 这就是说, 外模 q 不能先等于多项式大的小模 p, 而是需要经过另一种换模. 那么, 内模 Q 和外模 q 能否经过一次换模同时换为多项式大的小模 p? BGG+14 方案的结构表明这并不可能.

以上说明, 第 3.2 节的算法 3 的第一步 (换模) 必须分为两步: 内模换模 $Q \to p$ 和外模换模 $q \to p$.

4.2 内模换模 $Q \to p$

内模换模比较简单, 将内模换模视为一系列的布尔运算, 将内模 Q 换为小模 p, 同时将同态函数 $f^*(u^*)$ 换为新的同态函数 $f^{**}(u^*)$. 对这个内模换模有以下两点注解.

注解 1: 内模换模的实际操作显然是一系列布尔运算的拟同态运算, 将 "属性密文" $\boldsymbol{C}_{f^*(u^*)}$ 变为新的属性密文 $\boldsymbol{C}_{f^{**}(u^*)}$. 更具体地说, 内模换模之前为

$$\boldsymbol{C}_{f^*(u^*)} = \begin{pmatrix} \boldsymbol{c}_{f_1^*(u^*)} \\ \boldsymbol{c}_{f_2^*(u^*)} \\ \vdots \\ \boldsymbol{c}_{f_{k^*}^*(u^*)} \end{pmatrix} = \begin{pmatrix} \mathsf{Encode}(f_1^*(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_1^*}, \boldsymbol{s}) \\ \mathsf{Encode}(f_2^*(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_2^*}, \boldsymbol{s}) \\ \vdots \\ \mathsf{Encode}(f_{k^*}^*(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_{k^*}^*}, \boldsymbol{s}) \end{pmatrix}.$$

内模换模之后为

$$\boldsymbol{C}_{f^{**}(u^*)} = \begin{pmatrix} \boldsymbol{c}_{f_1^{**}(u^*)} \\ \boldsymbol{c}_{f_2^{**}(u^*)} \\ \vdots \\ \boldsymbol{c}_{f_{k^{**}}^{**}(u^*)} \end{pmatrix} = \begin{pmatrix} \mathsf{Encode}(f_1^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_1^{**}}, \boldsymbol{s}) \\ \mathsf{Encode}(f_2^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_2^{**}}, \boldsymbol{s}) \\ \vdots \\ \mathsf{Encode}(f_{k^{**}}^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_{k^{**}}^{**}}, \boldsymbol{s}) \end{pmatrix},$$

其中 $f^{**}(u^*)$ 是 $f(u)$ 的另一个同态运算, 与 $f^*(u^*)$ 不同的是, $f^{**}(u^*)$ 是小模 p 之下的课文, 满足小模 p 的解密方程

$$(< f^{**}(u^*), t > \bmod p) \bmod 2 = f(u) \text{ 或 } \mathrm{Rounding}(< f^{**}(u^*), t > \bmod p).$$

当然, 实际的小模 p 的解密过程, 在完成 $< f^{**}(u^*), t > \bmod p$ 之后是无法进行 mod 2 或 Rounding 的, 代之以穷举和属性解密尝试.

注解 2: 因为有了注解 1, 所以在内模换模时外模保持不变.

4.3 外模换模 $q \to p$ 的无效性

密文 $\boldsymbol{C} = (\boldsymbol{c}_{\mathrm{in}}, \boldsymbol{C}_{u^*}, \boldsymbol{C}_t, \boldsymbol{c}_{\mathrm{out}})$ 经过同态运算的拟同态运算变为 $(\boldsymbol{c}_{\mathrm{in}}, \boldsymbol{C}_{f^*(u^*)}, \boldsymbol{C}_t, \boldsymbol{c}_{\mathrm{out}})$, 再经过内模换模的拟同态运算变为 $(\boldsymbol{c}_{\mathrm{in}}, \boldsymbol{C}_{f^{**}(u^*)}, \boldsymbol{C}_t, \boldsymbol{c}_{\mathrm{out}})$. 现在进行外模换模.

外模换模遵从了 BGG+14 方案的结构, 是 LWE 换模. 所谓 LWE 换模是指如下两步运算: (1) 对密文做运算 $\frac{p}{q} \times (\cdot)(\bmod p)$; (2) 对编码所用的矩阵做运算 $\frac{p}{q} \times (\cdot)(\bmod p)$. 只有这两步运算都完成, 才有可能完成其后的 LWE 解密运算. 由于全同态解密钥 t 不是解密者的已知项, 因此密文 $\boldsymbol{C}_t$ 中的编码所用矩阵不是解密者的已知项, 无法对 $\boldsymbol{C}_t$ 做外模换模. $\boldsymbol{C}_t$ 必须在加密者的加密过程中就构造成小模 p 之下的课文. 另外, 为了使 Agr17 方案最大程度上顺利进行, 不妨设 $(\boldsymbol{c}_{\mathrm{in}}, \boldsymbol{C}_t, \boldsymbol{c}_{\mathrm{out}})$ 都是小模 p 之下的课文 (而不是模 q 之下的课文). 这就是说, 连陷门 $\boldsymbol{T}$ 都是根据小模 p 构造的.

现在只对 $\boldsymbol{C}_{f^{**}(u^*)}$ 做外模换模:

$$(1)\ \frac{p}{q} \times \boldsymbol{C}_{f^{**}(u^*)} (\bmod p);$$

$$(2)\ \frac{p}{q} \times \begin{pmatrix} f_1^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_1^{**}} \\ f_2^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_2^{**}} \\ \vdots \\ f_{k^{**}}^{**}(u^*)\boldsymbol{G} + \boldsymbol{B}_{f_{k^{**}}^{**}} \end{pmatrix} (\bmod p).$$

这两步运算都能完成, 但其后的属性解密就无法进行了. 因为 BGG+14 方案的属性解密不是普通的

LWE 解密, 而是有条件的 LWE 解密. 矩阵的外模换模破坏了条件结构.

5 我们的一种 “自然的” 修改方案及其不安全性

5.1 修改方案和其效率

首先, 设 $(\boldsymbol{c}_{\text{in}}, \boldsymbol{c}_t, \boldsymbol{c}_{u^*}, \boldsymbol{c}_{\text{out}})$ 全都是大模 q 之下的课文 (而不是小模 p 之下的课文). 因此陷门 $\boldsymbol{T}$ 仍然是根据大模 q 构造的. 其次, 修改方案前边部分与原始的 Agr17 方案相同, 一直到完成了内模换模, 得到了 $f^{**}(u^*)$ 所对应的属性密文. 然后考虑将全同态解密钥 t 与全同态密文 $f^{**}(u^*)$ 的算术内积 $< t, f^{**}(u^*) >$ (即没有模作用的内积), 而不是小模 p 内积 $< t, f^{**}(u^*) > (\text{mod } p)$, 注意到这个算术内积虽然比小模 p 大得多, 但仍然是多项式大. 因此这个算术内积可以看作大模 q 的内积: $< t, f^{**}(u^*) >=< t, f^{**}(u^*) > (\text{mod } q)$, 也就可以做拟同态运算了.

下一步, 猜想该算术内积的多项式多个可能结果; 而对每个非零的可能结果 a, 用 $a^{-1}(\text{mod } q)$ 乘以对应的属性密文 $< t, f^{**}(u^*) > (\text{mod } q)$, 这一步是可以做拟同态运算的.

再下一步, 对每个非零的可能结果 a, 考虑属性 (t, u^*) 的如下函数 $a^{-1} < t, f^{**}(u^*) > (\text{mod } q)$, 索要该函数的属性解密钥, 并做属性解密.

到此为止, 解密者精确地知道算术内积 $< t, f^{**}(u^*) >$ 的值. 然后对该值取模 p, 然后取模 2 (或取 “Rounding”). 解密完成, 修改方案通过.

效率: 修改方案的属性解密钥的个数远远大于小模 p, 而原始的 Agr17 方案的属性解密钥的个数为 p. 这就是说, 修改方案的效率远低于原始的 Agr17 方案.

5.2 修改方案的不安全性

在修改方案中, 解密者知道算术内积 $< t, f^{**}(u^*) >$. 而在原始的 Agr17 方案中, 仅仅要求解密者知道 $< t, f^{**}(u^*) > (\text{mod } p)$. 这就是说, 修改方案泄露了全同态解密钥 t 的更多信息.

注意到 t 和 $f^{**}(u^*)$ 都是用比特表示的, 而算术内积 $< t, f^{**}(u^*) >$ 实际上是 t 的每个比特、$f^{**}(u^*)$ 的对应比特、2 的对应幂的乘积的和. 因此, 不妨对算术内积取模 2 运算, 得到 t 的布尔线性方程. 因此, 解密者们可以用小得多的共谋代价求解 t, 因而攻破修改方案.

参考文献

[1] BONEH D, SAHAI A, WATERS B. Functional encryption: Definitions and challenges[C]. In: Theory of Cryptography—TCC 2011. Springer Berlin Heidelberg, 2011: 253–273. [DOI: 10.1007/978-3-642-19571-6_16]

[2] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[C]. In: Advances in Cryptology—CRYPTO 2001. Springer Berlin Heidelberg, 2001: 213–229. [DOI: 10.1007/3-540-44647-8_13]

[3] COCKS C. An identity based encryption scheme based on quadratic residues[C]. In: Cryptography and Coding—Cryptography and Coding 2001. Springer Berlin Heidelberg, 2001: 360–363. [DOI: 10.1007/3-540-45325-3_32]

[4] BOYEN X, WATERS B. Anonymous hierarchical identity-based encryption (without random oracles)[C]. In: Advances in Cryptology—CRYPTO 2006. Springer Berlin Heidelberg, 2006: 290–307. [DOI: 10.1007/11818175_17]

[5] BONEH D, WATERS B. Conjunctive, subset, and range queries on encrypted data[C]. In: Theory of Cryptography—TCC 2007. Springer Berlin Heidelberg, 2007: 535–554. [DOI: 10.1007/978-3-540-70936-7_29]

[6] GENTRY C. PEIKERT C, VAIKUNTANATHAN V. Trapdoors for hard lattices and new cryptographic constructions[C]. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008). ACM, 2008: 197–206. [DOI: 10.1145/1374376.1374407]

[7] CASH D, HOFHEINZ D, KILTZ E, et al. Bonsai trees, or how to delegate a lattice basis[C]. In: Advances in Cryptology—EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 523–552. [DOI: 10.1007/978-3-642-13190-5_27]

[8] AGRAWAL S, BONEH D, BOYEN X. Efficient lattice (H)IBE in the standard model[C]. In: Advances in Cryptology—EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 553–572. [DOI: 10.1007/978-3-642-13190-5_28]

[9] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]. In: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006). ACM, 2006: 89–98. [DOI: 10.1145/1180405.1180418]

[10] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[C]. In: Proceedings of IEEE Symposium on Security and Privacy (SP 2007). IEEE, 2007: 321–334. [DOI: 10.1109/SP.2007.11]

[11] KATZ J, SAHAI A, WATERS B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]. In: Advances in Cryptology—EUROCRYPT 2008. Springer Berlin Heidelberg, 2008: 146–162. [DOI: 10.1007/978-3-540-78967-3_9]

[12] LEWKO A, OKAMOTO T, SAHAI A, et al. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption[C]. In: Advances in Cryptology—EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 62–91. [DOI: 10.1007/978-3-642-13190-5_4]

[13] AGRAWAL S, FREEMAN D.M. VAIKUNTANATHAN V. Functional encryption for inner product predicates from learning with errors[C]. In: Advances in Cryptology—ASIACRYPT 2011. Springer Berlin Heidelberg, 2011: 21–40. [DOI: 10.1007/978-3-642-25385-0_2]

[14] WATERS B. Functional encryption for regular languages[C]. In: Advances in Cryptology—CRYPTO 2012. Springer Berlin Heidelberg, 2012: 218–235. [DOI: 10.1007/978-3-642-32009-5_14]

[15] GORBUNOV S, VAIKUNTANATHAN V, WEE H. Attribute based encryption for circuits[C]. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013). ACM, 2013: 545–554. [DOI: 10.1145/2488608.2488677]

[16] GARG S, GENTRY C, HALEVI S, et al. Attribute-based encryption for circuits from multilinear maps[C]. In: Advances in Cryptology—CRYPTO 2013, Part II. Springer Berlin Heidelberg, 2013: 479–499. [DOI: 10.1007/978-3-642-40084-1_27]

[17] GARG S, GENTRY C, HALEVI S, et al. Candidate indistinguishability obfuscation and functional encryption for all circuits[C]. In: Proceedings of 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS 2013). IEEE, 2013: 40–49. [DOI: 10.1109/FOCS.2013.13]

[18] GORBUNOV S, VAIKUNTANATHAN V, WEE H. Predicate encryption for circuits from LWE[C]. In: Advances in Cryptology—CRYPTO 2015, Part II. Springer Berlin Heidelberg, 2015: 503–523. [DOI: 10.1007/978-3-662-48000-7_25]

[19] AGRAWAL S. Stronger security for reusable garbled circuits, general definitions and attacks[C]. In: Advances in Cryptology–CRYPTO 2017, Part I. Springer Cham, 2017: 3–35. [DOI: 10.1007/978-3-319-63688-7_1]

[20] LAI Q Q, LIU F H, WANG Z D. New lattice two-stage sampling technique and its applications to functional encryption—Stronger security and smaller ciphertexts[C]. In: Advances in Cryptology—EUROCRYPT 2021, Part I. Springer Cham, 2021: 498–527. [DOI: 10.1007/978-3-030-77870-5_18]

[21] BONEH D, GENTRY C, GORBUNOV S, et al. Fully key-homomorphic encryption, arithmetic circuit ABE, and compact garbled circuits[C]. In: Advances in Cryptology—EUROCRYPT 2014. Springer Berlin Heidelberg, 2014: 533–556. [DOI: 10.1007/978-3-642-55220-5_30]

[22] WANG Z D, FAN X, LIU F H. FE for inner products and its application to decentralized ABE[C]. In: Public-Key Cryptography—PKC 2019, Part II. Springer Cham, 2019: 97–127. [DOI: 10.1007/978-3-030-17259-6_4]

[23] ANANTH P, VAIKUNTANATHAN V. Optimal bounded-collusion secure functional encryption[C]. In: Theory of Cryptography—TCC 2019, Part I. Springer Cham, 2019: 174–198. [DOI: 10.1007/978-3-030-36030-6_8]

[24] GENISE N, MICCIANCIO D. Faster Gaussian sampling for trapdoor lattices with arbitrary modulus[C]. In: Advances in Cryptology—EUROCRYPT 2018, Part I. Springer Cham, 2018: 174–203. [DOI: 10.1007/978-3-319-78381-9_7]

[25] AGRAWAL S. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation[C]. In: Advances in Cryptology—EUROCRYPT 2019, Part I. Springer Cham, 2019: 191–225. [DOI: 10.1007/978-3-030-17653-2_7]

[26] AGRAWAL S, ROSEN A. Functional encryption for bounded collusions, revisited[C]. In: Theory of Cryptography—TCC 2017, Part I. Springer Cham, 2017: 173–205. [DOI: 10.1007/978-3-319-70500-2_7]

[27] DATTA P, OKAMOTO T, TAKASHIMA K. Adaptively simulation-secure attribute-hiding predicate encryption[C]. In: Advances in Cryptology—ASIACRYPT 2018, Part II. Springer Cham, 2018: 640–672. [DOI: 10.1007/978-3-030-03329-3_22]

[28] AGRAWAL S, YAMADA S. Optimal broadcast encryption from pairings and LWE[C]. In: Advances in Cryptology—EUROCRYPT 2020, Part I. Springer Cham, 2020: 13–43. [DOI: 10.1007/978-3-030-45721-1_2]

[29] AGRAWAL S, MAITRA M, YAMADA S. Attribute based encryption (and more) for nondeterministic finite automata from LWE[C]. In: Advances in Cryptology—CRYPTO 2019, Part II. Springer Cham, 2019: 765–797. [DOI: 10.1007/978-3-030-26951-7_26]

[30] CHEN Y L, VAIKUNTANATHAN V, WATERS B, et al. Traitor-tracing from LWE made simple and attribute-based[C]. In: Theory of Cryptography—TCC 2018, Part II. Springer Cham, 2018: 341–369. [DOI: 10.1007/978-

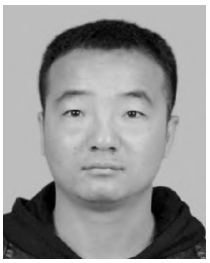
3-030-03810-6_13]
[31] AGRAWAL S, LIBERT B, MAITRA M, et al. Adaptive simulation security for inner product functional encryption[C]. In: Public-Key Cryptography—PKC 2020, Part I. Springer Cham, 2020: 34–64. [DOI: 10.1007/978-3-030-45374-9_2]
[32] ANANTH P, JAIN A, LIN H J, et al. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification[C]. In: Advances in Cryptology—CRYPTO 2019, Part III. Springer Cham, 2019: 284–332. [DOI: 10.1007/978-3-030-26954-8_10]
[33] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[C]. In: Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science (ITCS 2012). ACM, 2012: 309–325. [DOI: 10.1145/2090236.2090262]
[34] GENTRY C, SAHAI A, WATERS B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]. In: Advances in Cryptology—CRYPTO 2013, Part I. Springer Berlin Heidelberg, 2013: 75–92. [DOI: 10.1007/978-3-642-40041-4_5]
[35] BRAKERSKI Z, VAIKUNTANATHAN V. Lattice-based FHE as secure as PKE[C]. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS 2014). ACM, 2014: 1–12. [DOI: 10.1145/2554797.2554799]

作者信息

胡予濮 (1955–), 河南濮阳人, 教授. 主要研究领域为密码算法的安全性分析.
yphu@mail.xidian.edu.cn

刘君 (1993–), 陕西宝鸡人, 博士研究生. 主要研究领域为混淆和白盒密码.
jliu6@stu.xidian.edu.cn

王保仓 (1979–), 河南郸城人, 教授. 主要研究领域为公钥密码学.
bcwang@xidian.edu.cn



董星廷 (1989–), 山西临汾人, 博士研究生. 主要研究领域为属性加密.
xtdong67@163.com

潘彦斌 (1982–), 河北文安人, 副研究员. 主要研究领域为格密码学.
panyanbin@amss.ac.cn