计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20170443
56(3):496-507, 2019

抵抗自适应密钥恢复攻击的层级全同态加密

李增鹏[1,2] 马春光[1] 赵明昊[3]
[1](哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)
[2](青岛大学计算机科学技术学院 山东青岛 266071)
[3](清华大学软件学院 北京 100084)
(lizengpeng@hrbeu.edu.cn)

Leveled Fully Homomorphic Encryption Against Adaptive Key Recovery Attacks

Li Zengpeng[1,2], Ma Chunguang[1], and Zhao Minghao[3]
[1](*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)
[2](*College of Computer Science and Technology, Qingdao University, Qingdao, Shandong* 266071)
[3](*School of Software, Tsinghua University, Beijing* 100084)

Abstract A major open problem is to protect leveled homomorphic encryption from adaptive attacks that allow an adversary to learn the private key. In order to achieve the goal of preventing key recovery attacks on fully homomorphic encryption (FHE), Li Zengpeng et al (PROVSEC'16) proposed an multiple secret keys fully homomorphic encryption scheme under the learning with errors (LWE) assumption to prevent key recovery attacks on FHE, which did not use the notion of "valid ciphertexts" of Loftus et al (SAC'11). However, utilizing the information of noise, the attacks can still recover the information of the secret key. Li Zengpeng et al.'s scheme cannot provide an efficient method to protect the secret key. In this paper, Inspired by the work of Li Zengpeng et al (EPRINT'16), we first give a new method of key recovery attacks to Li Zengpeng et al.'s scheme; then, we propose a new FHE scheme with multiple secret keys which differs from EPRINT'16, and prove our new scheme against key recovery attacks. Our main idea is to adopt the dual version of encryption algorithm and generate a "one-time" secret key every time, so that even if an attacker can learn some bits of the one-time private key from each decryption query and cannot obtain some bits of noise, the scheme still does not allow them to compute a valid private key.

Key words adaptive key recovery attacks; lattice-based cryptography; learning with errors; fully homomorphic encryption; multiple secret keys

摘 要 由于攻击者可以通过自适应攻击的方式获得私钥,因此目前一个重要的公开问题是如何保护层级全同态加密(fully homomorphic encryption, FHE)免受自适应攻击.在该问题的研究中,为实现抵抗自适应密钥恢复攻击的目标,近来,李增鹏等人(PROVSEC'16)在容错学习(learning with errors, LWE)假设下,提出了一个多私钥的全同态加密方案,该方案并不依赖于 Loftus 等人(SAC'11)利用的"有效密文"概念.然而尽管该方案能抵抗私钥信息的泄漏,但利用噪声信息仍然能够实现自适应的密钥恢复攻击.基于李增鹏等人(EPRINT'16)的工作,给出对李增鹏等人方案的一种新的自适应攻击方法,并提出了一种不同于 EPRINT'16 的对偶的多私钥全同态加密方案以抵抗该自适应攻击.其核心思想是采用对偶的加密方式,并在每次解密算法运行时,都生成一个"一次"私钥,因此即使攻击者能够从每个解

收稿日期:2017-06-12;**修回日期**:2018-12-26
基金项目:国家自然科学基金项目(61472097,61802214)
This work was supported by the National Natural Science Foundation of China (61472097, 61802214).
通信作者:马春光(machunguang@hrbeu.edu.cn)

密查询中得到该一次私钥的某些比特,但却无法获得噪声的某些比特,因此,攻击者仍不能计算出一个有效的私钥.

关键词 自适应密钥恢复攻击;格基密码学;容错学习;全同态加密;多私钥

中图法分类号 TP391

众所周知,若敌手访问一个解密预言机,则能引起对基本的 Regev[1] 或 GPV(Gentry, Peikert, Vaikuntanathan)[2]加密以及各种全同态加密(fully homomorphic encryption, FHE)方案[3-6]的攻击.这些攻击能够让敌手直接获得私钥.因此,与那些获得并利用消息的某些比特信息而进行的攻击相比,上述攻击形式更为严重.实际上,如果不要求方案具有同态加密的功能,那么存在一些基于格的 IND-CCA2 加密方案(如文献[2]),但是这些方法与同态加密并不兼容.因此,本文关心的是如何获得这些方案 CCA1 安全的变体.此外,Brakerski 等人,如文献[7-9]利用密钥交换、自举等方法获得的一系列全同态加密方案,但这些方案并不能实现 IND-CCA1 安全,因为在这些方案中,其公钥包含了对私钥信息的加密.随后,GSW(Gentry-Sahai-Waters)方案[10]能够实现层级全同态加密而不需要任何密钥交换过程,因此其方案为实现 FHE 方案的 IND-CCA1 安全性提供了一种可能,因为对于那些使用密钥交换、自举或者其他方法的全同态加密方案来说,该方案避免了在公钥中包含对私钥加密的信息.

LMSV(Loftus, May, Smart, Vercauteren)[6]研究了 Gentry[11]基于理想格的同态加密方案(及其变体[12])在自适应攻击下私钥的安全性,同时,他们证明了如果敌手能够访问解密预言机,那么就能确定私钥.此外,Loftus 等人[12]给出了 Smart-Vercauteren 密码系统的一个变体,在该方案中,即使敌手有一个解密预言机,私钥仍旧安全;该结果是基于"有效密文"的概念由解密算法所保证,并且其安全性依赖于一个非常强的知识假设.随后,由于构造 Smart-Vercauteren 密码系统所依赖的计算假设,即短主理想(the short principal ideal)问题被攻破[13-16],因此 LMSV 方案不再被认为是安全的.近来,Cannitte 等人[17]提出了第 1 个真正意义上的 IND-CCA1 安全的 FHE 方案,但他们并没有完全解决密文长度仍然依赖于电路输入长度的问题.因此 IND-CCA1 安全的层级全同态加密问题仍旧没有得到完全解决.值得注意的是,Loftus 等人[12]着重解释了密文有效性攻击(ciphertext validity attacks, CVA)的相关性,即该模型允许敌手访问一个能够决定密文是否有效的预言机.同时,他们证明了敌手在 CVA 预言机的帮助下解密挑战密文的可能性(至少私钥仍是安全的.然而该攻击不是 CCA1 攻击,而是 CCA2 攻击).Loftu 等人[6]认为对同态加密方案的 CCA1 和 CVA 攻击在实践中是可实现的(他们在文献[6]第 6 节中写道"在真实世界中,通常是敌手将其选择的密文先发给某一参与方,然后通过观察该参与方的行为获得该预言机").例如考虑一个场景,给出了一个 CVA 预言机的精确描述.如果使用者正在将一个加密的数据库储存在云端并且进行查询,那么攻击者能够发送其选择的密文作为回应.如果这些密文是无效的,那么使用者可能重新发送相同的查询直到收到一个有效的密文作为回应.Bleichenbacher[18]利用 CVA 预言机来攻击 RSA 的某些变体就是一个典型的例子.由此可以认为,提出新的技术来让同态加密方案的私钥变得安全,从而抵抗 CVA 攻击、密钥恢复攻击(key recovery attacks, KRA)是非常必要且紧迫的.

为解决这个问题,李增鹏等人[19]提出一个不同于 Loftus 等人[6]的方案.他们给出了 GSW 方案[11]的一个变体,即多私钥 GSW 方案(MGSW),该变体允许有一个公钥对应着多个私钥.即通过使用"一次"一私钥的方法来避免私钥完全暴露的风险,从而不再依赖于"有效密文"的概念.其核心思想是,即使敌手能够从每个解密查询中得到一次私钥的某些比特,但他们也不可能将从多个解密查询中得来的信息结合起来计算出真正的私钥.但是,该方案也存在风险,即当解密查询时,利用噪声信息可能恢复出私钥信息.在本文 3.1 节给出该攻击方案的具体描述.此外,不禁要问:是否存在这样一种全同态加密方案,它可有效抵抗一些已知的密钥恢复攻击?

为了解决上述问题,本文提出了 MGSW 方案的一种"对偶"版本,记为 DMGSW 方案,并着重解释了该 DMGSW 方案如何能够结合多个私钥有效抵抗一些已知的自适应密钥恢复攻击.

此外,为证明方案能够抵抗特定类(或某些已知攻击)的自适应攻击,本文利用剩余 Hash 引理和向量空间投影论证给出理论框架.遗憾的是,不能证明该方案的 IND-CCA1 安全性.但希望本文的方案对

于完全实现可证明安全的 IND-CCA1 全同态加密方案来说是一块垫脚石.

诚然,证明全同态加密的 IND-CCA1 安全性是一个非常有挑战性的工作,最接近的解决方案是 Loftus 等人[6]的结果,他们使用了一个非常强的知识假设,但是该结果现在已经被攻破了. Cannitte 等人[17]基于 multi-key 的全同态加密方案[20-21],实现了 IND-CCA1 安全,但他们的方案依赖复杂的 multi-key 技术,且密文长度仍依赖于电路输入长度,使得该问题并未完全解决.

1 预备知识

本节着重介绍本文需用到的数学符号.

1.1 符 号

令 $n\in\mathbb{N}$,$[n]$表示集合$\{1,2,\cdots,n\}$. 令 X 为平均值 μ:$E[X]=\mu$ 的一个随机变量,这里 E 表示 X 的平均值或期望值. X 的标准差是 $\delta=\sqrt{E[(X-\mu)^2]}$. 本文中,列向量以粗体小写字母表示,如 $\boldsymbol{x}$. 用转置来表示行向量 $\boldsymbol{x}^{\mathrm{T}}$. 用粗体大写字母,如 $\boldsymbol{R}$ 来表示矩阵. 此外,本文用到的内积和范数定义如下:$\langle \boldsymbol{x},\boldsymbol{y}\rangle=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$ 表示 2 个向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的标准欧几里得内积. 对于向量 $\boldsymbol{v}=(v_1,v_2\cdots,v_n)^{\mathrm{T}}$,$\ell_\infty$ 范数是 $\|\boldsymbol{v}\|_\infty=\max\{|v_1|,|v_2|,\cdots,|v_n|\}$,$\ell_1$ 范数是 $\|\boldsymbol{v}\|_1=\sum_{i=1}^{n}|v_i|$,欧几里得范数是 $\|\boldsymbol{v}\|_2=\sqrt{\langle \boldsymbol{v},\boldsymbol{v}\rangle}=\sqrt{\sum_{i=1}^{n}|v_i|^2}$. 对于一个向量 $\boldsymbol{v}$,用 $\|\boldsymbol{v}\|$ 表示其 ℓ_2 范数.

引理 1[10]. 矩阵向量剩余 Hash 引理. 令 $\kappa\in\mathbb{N}$,$n\in\mathbb{N}$,$q\in\mathbb{N}$,且 $m\geqslant n\,\mathrm{lb}\,q+2\kappa$. 令 $\boldsymbol{A}\xleftarrow{R}\mathbb{Z}_q^{m\times n}$ 是一个均匀随机矩阵(注意 R 是随机选取),令 $\boldsymbol{r}\xleftarrow{R}\{0,1\}^m$ 且 $\boldsymbol{y}\xleftarrow{R}\mathbb{Z}_q^{n\times 1}$,那么:

$$\Delta((\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{r}),(\boldsymbol{A},\boldsymbol{y}))\leqslant 2^{-\kappa},\tag{1}$$

其中,$\Delta(\boldsymbol{A},\boldsymbol{B})$表示分布 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的统计距离.

1.2 离散高斯

实际上,格密码方案[22]主要依赖于格上的高斯概率分布. 在本文中,依然需要分析高斯分布上错误元素的行为. 首先给出高斯分布的相关定义.

定义 1[23]. 令 L 为 $\mathbb{Z}^m$ 的一个子集. 对于向量 $\boldsymbol{c}\in\mathbb{R}^m$ 和参数 $\sigma\in\mathbb{R}$,定义:

$$\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})=\exp\left(-\pi\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{\sigma^2}\right),$$

$$\rho_{\sigma,\boldsymbol{c}}(L)=\sum_{\boldsymbol{x}\in L}\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}).$$

以 $\boldsymbol{c}$ 为均值、以 σ 为参数 L 的离散高斯分布是:

$$\forall\,\boldsymbol{y}\in L,D_{L,\sigma,\boldsymbol{c}}=\frac{\rho_{\sigma,\boldsymbol{c}}(y)}{\rho_{\sigma,\boldsymbol{c}}(L)}.$$

为了方便表示,将 $\rho_{\sigma,0}$ 和 $D_{L,\sigma,0}$ 简写为 ρ_σ 和 D_σ,将 $D_{\mathbb{Z}^m,\sigma,0}$ 简写为 D_σ^m.

定义 2[9]. B-界分布. $\{\chi_n\}_{n\in\mathbf{N}}$是整数上的高斯分布,如果满足:

$$\Pr_{\varphi\leftarrow\chi_n}[|\varphi|\geqslant B]\leqslant 2^{-\Omega(n)},$$

则称之为 B-界分布.

对于整数上的一个分布 $\chi=\chi(\lambda)$,且整数界为 $B=B(\lambda)$,如果存在 $\Pr_{\varphi\leftarrow\chi(\lambda)}[|\varphi|\leqslant B(\lambda)]$,那么称 χ 是 B-界的.

引理 2[23]. 噪声边界满足 2 个条件:

1) 对于 $\forall\,k>0$,参数 $e_i(i\in[m])$满足

$$\Pr[|e_i|>k\sigma,e_i\leftarrow D_\sigma^1]\leqslant 2\exp\left(-\frac{k^2}{2}\right);$$

2) 对于 $\forall\,k>0$,向量 $\boldsymbol{e}=[e_1,e_2,\cdots,e_m]$满足

$$\Pr[\|\boldsymbol{e}\|>k\sigma\sqrt{m},\boldsymbol{e}\leftarrow D_\sigma^m]\leqslant k^m\exp\left(\frac{m}{2}(1-k^2)\right).$$

注 1. 本文中,假设 $\sigma\geqslant 2\sqrt{n}$. 因此,如果 $\boldsymbol{e}\leftarrow D_\sigma^m$,那么在平均情况下 $\|\boldsymbol{e}\|\approx\sqrt{m}\sigma$. 由引理 2 中的 2)可知,$\|\boldsymbol{e}\|\leqslant 2\sigma\sqrt{m}$.

1.3 容错学习(learning with errors, LWE)

LWE 问题是全同态加密 GSW 方案[10]及本文方案的主要计算假设. 具体如下:

定义 3. 对于一个秘密向量 $\boldsymbol{s}^i\in\mathbb{Z}_q^n$,通过均匀随机地选择向量 $\boldsymbol{a}^i\in\mathbb{Z}_q^n$,选取噪声变量 $e_i\leftarrow\chi$,并输出 $(\boldsymbol{a}^i,b_i=\langle\boldsymbol{s}^i,\boldsymbol{a}^i\rangle+e_i(\mathrm{mod}\ q))$,从而选取 $\mathbb{Z}_q^n\times\mathbb{Z}^q$ 上的 LWE 分布 $\mathcal{A}_{s,\chi}$.

LWE 问题有 2 个版本:搜索(search)版本,是已知 LWE 样本来寻找秘密向量;判定(decision)版本,是用来区分 LWE 实例和均匀随机样本.

定义 4. 对于 $i\in[m]$,任一均匀随机的 $\boldsymbol{s}^i\in\mathbb{Z}_q^n$(对于所有抽样来说是固定的),给定了从 $\mathcal{A}_{s,\chi}$ 中选取的 m 个独立样本$(\boldsymbol{a}^i,b_i)\in\mathbb{Z}_q^n\times\mathbb{Z}^q$,从而找到 $\boldsymbol{s}^i$.

定义 5. 给定 m 个独立样本$(\boldsymbol{a}^i,b_i)\in\mathbb{Z}_q^n\times\mathbb{Z}^q$,其中每个抽样取自 2 个分布:1)LWE 分布 $\mathcal{A}_{s,\chi}$,其中 $\boldsymbol{s}\in\mathbb{Z}_q^n$是一个均匀随机的秘密向量(对所有的抽样来说是固定的);2)均匀分布. 即(以不可忽略的优势)区分以上 2 个分布.

Regev[1]及文献[20-21,24]证明了对于恰当的参数,LWE 问题与格的近似最短向量(shortest vector problem, SVP)问题同样困难.

1.4 层级全同态加密

定义 6. 令 $L=L(\kappa)$为一个固定的函数. 为深度是 L 的一类电路$\{C_\kappa\}_{\kappa\in\mathbf{Z}}$构造的一个 L-层级全同态加密(leveled-FHE)方案,该方案包括 4 个 PPT 算法($KeyGen,Enc,Dec,Eval$):

1) 密钥生成算法 $KeyGen$. 是一个随机化算法,它以安全参数 1^κ 为输入,并输出一个公钥 pk 和私钥 sk.

2) 加密算法 Enc. 是一个随机化算法,它以一个公钥 pk 和一个消息 $\mu\in\{0,1\}$为输入,并输出一个密文 c.

3) 解密算法 Dec. 是一个确定性算法,它以一个私钥 sk 和一个密文 ct 为输入,并输出一个消息 $\mu\in\{0,1\}$.

4) 同态运算算法 $Eval$. 输入一个公钥 pk,一个运算电路 $C\in C_\kappa$,和一个密文列表 $ct_1,ct_2,\cdots,ct_{\ell(\kappa)}$,并输出一个密文 ct^*.

并要求 2 个正确性成立:

1) 对于任意 $\kappa,\mu\in\{0,1\}$和由 $KeyGen(1^\kappa)$输出的任意(pk,sk),有:

$$\mu=Dec(sk,(Enc(pk,\mu))).$$

2) 对于任意 $\kappa,\mu_1,\mu_2,\cdots,\mu_\ell$,和 $C\in C_\kappa$,有:

$$C(\mu_1,\mu_2,\cdots,\mu_\ell)=Dec(sk,(Eval(pk,C,Enc(pk,\mu_1),Enc(pk,\mu_2),\cdots,Enc(pk,\mu_\ell))).$$

使用选择明文攻击(chosen plaintext attacks, CPA)安全性的标准概念来定义安全性.

定义 7. 如果对于任意多项式时间敌手 $\mathcal{A}$ 来说,下面的公式在 κ 上是可忽略的,那么说一个同态加密方案是不可区分选择明文攻击安全的(也称为 IND-CPA 安全的):

$$|Pr[\mathcal{A}(pk,Enc(pk,0))=1]-Pr[\mathcal{A}(pk,Enc(pk,1))=1]|,$$

其中,$(pk,sk)\leftarrow KeyGen(1^\kappa)$.

此外,根据选择密文安全(chosen ciphertext attacks, CCA1)的概念,在其攻击的第 1 阶段中,敌手 $\mathcal{A}$ 已知一个公钥,然后访问一个解密预言机. 它可以获得任意输入的解密. 在第 2 阶段中,敌手已知一个挑战密文 $\mathrm{Enc}(pk,\mu)$并且不再需要对解密预言机进行查询. CCA1 安全性模型适用于分析敌手是否能从解密查询中得到私钥. 因此,CCA1 是很适合用来研究同态加密的一个安全模型.

1.5 基本工具

下面回顾一下 Brakerski 等人如文献[7-8,10]提出的一些基本工具. 令 $q,m\in\mathbf{Z}$. 令 $\ell=\lfloor\mathrm{lb}\,q\rfloor+1$,因此,$2^{\ell-1}\leqslant q<2^\ell$ 且 $N=m\times\ell$.

定义 8. $PowerOf2$ 的幂运算. 算法 $PowerOf2$ 输入一个 m 维向量 $\boldsymbol{v}=[v_1,v_2,\cdots,v_m]\in\mathbb{Z}_q^m$,输出 $\mathbb{Z}_q^N$ 中的一个 N 维向量:

$$(v_1,2v_1,\cdots,2^{\ell-1}v_1,v_2,2v_2,\cdots,2^{\ell-1}v_2,\cdots,v_m,2v_m,\cdots,2^{\ell-1}v_m)^{\mathrm{T}}.$$

定义 9. $BitDecomp$ 比特分解. 算法 $BitDecomp$ 输入一个向量 $\boldsymbol{v}\in\mathbb{Z}_q^m$,输出一个 N 维向量,其中 $params\leftarrow\mathrm{MGSW.Setup}(1^\kappa,1^L)$是 v_i 二进制表示中的第 j 位(按最低有效位到最高有效位排列). 换句话说:

$$v_i=\sum_{j=0}^{\ell-1}2^jv_{i,j}.$$

定义 10. $BitDecomp$ 逆比特分解. 算法 $BitDecomp^{-1}$输入一个向量

$$\boldsymbol{v}=(v_{1,0},v_{1,1},\cdots,v_{1,\ell-1},v_{2,0},v_{2,1},\cdots,v_{2,\ell-1},\cdots,v_{m,0},v_{m,1},\cdots,v_{m,\ell-1})^{\mathrm{T}}\in\mathbb{Z}_q^N,$$

输出:

$$\left(\sum_{j=0}^{\ell-1}2^jv_{1,j},\sum_{j=0}^{\ell-1}2^jv_{2,j},\cdots,\sum_{j=0}^{\ell-1}2^jv_{m,j}\right)^{\mathrm{T}}\in\mathbb{Z}_q^m.$$

这里,输入向量 $\boldsymbol{v}$ 不需要是$\{0,1\}$-向量,可以是 $\mathbb{Z}^N$ 上的任意向量.

定义 11. $Flatten$ 展平运算. 算法 $Flatten$ 输入一个向量 $\boldsymbol{v}\in\mathbb{Z}_q^N$,并输出一个 N 维二元向量(即$\{0,1\}^N$ 的一个元素). 将该算法定义为

$$Flatten(\boldsymbol{v})=BitDecomp(BitDecomp^{-1}(\boldsymbol{v})).$$

注 2. 令 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{Z}_q^m$且 $\boldsymbol{x}'\in\mathbb{Z}_q^N$. 那么:

$\langle BitDecomp(\boldsymbol{x}),PowersOf2(\boldsymbol{y})\rangle=\langle\boldsymbol{x},\boldsymbol{y}\rangle$且 $\langle\boldsymbol{x}',PowersOf2(\boldsymbol{y})\rangle=\langle BitDecomp^{-1}(\boldsymbol{x}'),\boldsymbol{y}\rangle=\langle BitDecomp(BitDecomp^{-1}(\boldsymbol{x}')),PowersOf2(\boldsymbol{y})\rangle=\langle Flatten(\boldsymbol{x}'),PowersOf2(\boldsymbol{y})\rangle$.

以上算法可以从向量扩展到矩阵. 可以根据 Micciancio 和 Peikert[25] 的工具矩阵(gadget matrix) $\boldsymbol{G}$ 来解释上面的函数. 首先定义 $\boldsymbol{G}=\boldsymbol{I}_m\otimes\boldsymbol{g}\in\mathbb{Z}_q^{m\times N}$,其中,$\boldsymbol{g}=(1,2,4,\cdots,2^{\ell-1})^{\mathrm{T}}$. 对于 $\boldsymbol{v}\in\mathbb{Z}_q^m$,有 $PowersOf2(\boldsymbol{v})=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{G}$. 对于 $\boldsymbol{v}\in\mathbb{Z}_q^N$,有 $BitDecomp^{-1}(\boldsymbol{v})=\boldsymbol{G}\boldsymbol{v}$. 对于 $\boldsymbol{x}\in\mathbb{Z}_q^m$,算法 $BitDecomp(\boldsymbol{x})$可以重命名为 $\boldsymbol{G}^{-1}(\boldsymbol{x})$. 上面的定义和结果能够用 $\boldsymbol{G}$ 和 $\boldsymbol{G}^{-1}$的方式来解释,参见引理 3.

引理 3[25]. 对于任意的 $N\geqslant m\lceil\mathrm{lb}\,q\rceil$,存在一个固定的有效可计算矩阵 $\boldsymbol{G}\in\mathbb{Z}_q^{m\times N}$和一个有效可计算确定性"短原象"函数 $\boldsymbol{G}^{-1}(\cdot)$满足如下条件. 对于任意的 m',输入一个矩阵 $\boldsymbol{M}\in\mathbb{Z}_q^{m\times m'}$,逆函数 $\boldsymbol{G}^{-1}(\boldsymbol{M})$输出一个矩阵 $\boldsymbol{G}^{-1}(\boldsymbol{M})\in\{0,1\}^{N\times m'}$ 使得 $\boldsymbol{G}\boldsymbol{G}^{-1}(\boldsymbol{M})=\boldsymbol{M}$.

因此,可以将 $\boldsymbol{G}$ 看作一个具有“公共陷门”的特殊矩阵,它允许解决最小整数解(short integer solution, SIS)问题. 这里,$\boldsymbol{G}^{-1}(\cdot)$ 是一个有效可计算的函数.

2 GSW 方案

本节首先介绍全同态加密 GSW 方案[10],然后介绍李增鹏等人[19]对该方案的自适应攻击,该攻击与 Chenal 和 Tang[3] 的攻击类似.

2.1 GSW 方案

令 κ 为安全参数,L 为同态加密的层级数. leveled-FHE 方案的层级数. 给出 GSW[10] 方案的简要描述,该方案最初是根据函数 $BitDecomp$, $BitDecomp^{-1}$, $Flatten$ 定义的,但是本文采用 Alperin-Sheriff 等人[27]的简化方式,即使用工具矩阵 $\boldsymbol{G}$ 定义.

1) GSW 初始化算法 $GSW.Setup(1^{\kappa},1^{L})$:

① 选择 κ 比特的一个模 q,参数 $n=n(\kappa,L)\in\mathbb{N}$ 和 $\mathbb{Z}$ 上的错误分布 $\chi=\chi(\kappa,L)$,使得 LWE 问题对于已知攻击实现至少 2^{κ} 的安全性,选择一个参数 $m=m(\kappa,L)=O(n\lg q)$.

② 输出参数 $params=(n,q,\chi,m)$,并令 $\ell=\lfloor\lg q\rfloor+1$ 和 $N=(n+1)\ell$.

2) GSW 密钥生成算法 $GSW.KeyGen(params)$:

① 均匀选取 $\boldsymbol{t}=(t_1,t_2,\cdots,t_n)^{\mathrm{T}}\leftarrow\mathbb{Z}_q^n$,并计算:$\boldsymbol{s}\leftarrow(1,-\boldsymbol{t}^{\mathrm{T}})^{\mathrm{T}}=(1,-t_1,-t_2,\cdots,-t_n)^{\mathrm{T}}\in\mathbb{Z}_q^{(n+1)\times 1}$.

② 均匀选取随机公共矩阵 $\boldsymbol{B}\leftarrow\mathbb{Z}_q^{m\times n}$ 和一个错误向量 $\boldsymbol{e}\leftarrow\chi^m$.

③ 计算向量 $\boldsymbol{b}=\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e}\in\mathbb{Z}_q^m$ 并构造矩阵 $\boldsymbol{A}=(\boldsymbol{b}|\boldsymbol{B})\in\mathbb{Z}_q^{m\times(n+1)}$,并满足:

$$\boldsymbol{A}\boldsymbol{s}=(\boldsymbol{b}|\boldsymbol{B})\boldsymbol{s}=(\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e}|\boldsymbol{B})\begin{pmatrix}1\\-\boldsymbol{t}\end{pmatrix}=\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e}-\boldsymbol{B}\boldsymbol{t}=\boldsymbol{e}.$$

④ 返回私钥 $sk\leftarrow\boldsymbol{s}$ 和公钥 $pk\leftarrow\boldsymbol{A}$.

3) GSW 加密算法 $\boldsymbol{C}\leftarrow GSW.Enc(pk,\mu)$:

① 令 $\boldsymbol{G}$ 为上述 $(n+1)\times N$ 维的工具矩阵,并均匀选取一个随机矩阵 $\boldsymbol{R}\leftarrow\{0,1\}^{m\times N}$.

② 加密单比特消息 $\mu\in\{0,1\}$,并生成密文

$$\boldsymbol{C}=\mu\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}(\mathrm{mod}\ q)\in\mathbb{Z}_q^{(n+1)\times N},$$

需注意的是,在原 GSW 方案中,加密算法使用:

$$Flatten(\mu\mathbf{I}+BitDecomp(\boldsymbol{R}\boldsymbol{A}))\in\{0,1\}^{N\times N},$$

其中,$\mathbf{I}$ 是一个单位矩阵.

4) GSW 解密算法 $\mu'\leftarrow GSW.Dec(sk,\boldsymbol{C})$:

① 输入私钥 $sk=\boldsymbol{s}\in\mathbb{Z}_q^{n+1}$,令 k 满足 $q/4<2^{k-1}\leqslant q/2$,其中 $\boldsymbol{C}[k]$ 为 $\boldsymbol{C}$ 的第 k 列.

② 在 $(-q/2,q/2]$ 范围内计算 $x\leftarrow\langle\boldsymbol{C}[k],\boldsymbol{s}\rangle(\mathrm{mod}\ q)$,这里有 $\langle\boldsymbol{C}[k],\boldsymbol{s}\rangle=\boldsymbol{C}[k]^{\mathrm{T}}\boldsymbol{s}$,并有:

$$\boldsymbol{C}^{\mathrm{T}}\boldsymbol{s}=\mu\boldsymbol{G}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{R}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{s}=\mu(1,2,4,\cdots)^{\mathrm{T}}+\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e}.$$

由上,可以看出,计算时所选择密文矩阵 $\boldsymbol{C}$ 的第 k 列,对应着向量 $\langle\boldsymbol{C}[k],\boldsymbol{s}\rangle$ 的第 k 个坐标,即 $\mu 2^{k-1}+\boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{e}$.

③ 输出 $\mu'=\left\lfloor\lfloor x/2^{k-1}\rceil\right\rfloor$. 因此,如果 $|x|<2^{k-2}\leqslant q/4$,则返回 0;否则返回 1.

5) GSW 运算算法

$GSW.Eval(pk,(\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_l))$:

① 加法运算 $GSW.Add(\boldsymbol{C}_1,\boldsymbol{C}_2)\in\mathbb{Z}_q^{(n+1)\times N}$ 输出:

$$\boldsymbol{C}_1+\boldsymbol{C}_2=(\mu_1+\mu_2)\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{R}_1+\boldsymbol{R}_2).$$

② 乘法运算 $GSW.Mult(\boldsymbol{C}_1,\boldsymbol{C}_2)$ 计算并输出:

$$\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=(\mu_1\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1)\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=\mu_1\boldsymbol{C}_2+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=\mu_1\mu_2\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)+\mu_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_2=\mu_1\mu_2\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{R}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)+\mu_1\boldsymbol{R}_2).$$

注意 $\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)\in\mathbb{Z}_q^{(n+1)\times N}$. 此外,利用 $\boldsymbol{G}-\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$ 计算同态 NAND 门.

注 3. Mukherjee 等[20]方案中的解密算法公式是为了选择一个特定的向量 $\boldsymbol{w}$ 来计算 $\boldsymbol{s}\boldsymbol{C}\boldsymbol{G}^{-1}(\boldsymbol{w})^{\mathrm{T}}$,但相比原 GSW 方案的解密算法效率要低得多(无论是计算时间还是噪声项的大小). 因此本文仍将采用原 GSW 方案的解密算法. 此外,当 q 为 2 的幂时,还存在另外一种方式处理 $\mathbb{Z}^q$ 上的信息. 具体细节参见文献[10].

2.2 安全性

定理 1. 对于参数 $m=O(n\lg q)$,令参数 (m,n,q,χ) 使得 LWE 困难假设成立,那么该 GSW 方案是 IND-CPA 安全的.

该证明的主要步骤是证明 $(\boldsymbol{A},\boldsymbol{R}\boldsymbol{A})$ 和均匀分布计算不可区分,本文不再赘述.

2.3 密钥恢复攻击

本节着重介绍能够恢复 GSW 方案私钥的 2 种自适应攻击. 尽管该类攻击方式超出了原始 GSW 方案[10]安全模型的范围,但这种类型的攻击却是众所周知的.

1) 自适应攻击 1. 该类自适应密钥恢复攻击,类似于 Chenal 和 Tang[3] 的攻击. 敌手通过询问一些解密预言机来恢复私钥 $\boldsymbol{s}=(1,-\boldsymbol{t}^{\mathrm{T}})^{\mathrm{T}}$.

敌手选择密文矩阵 $\boldsymbol{C}$ 并询问解密预言机,并且该预言机将返回 $\langle\boldsymbol{C}[k],\boldsymbol{s}\rangle$ 的“最高有效位(most

signification bit, MSB)",其中 $\boldsymbol{C}[k]$ 是 $\boldsymbol{C}$ 的第 k 列并且 k 对于敌手来说是已知的. 实际计算 $\lfloor \boldsymbol{C}[k]^{\mathrm{T}}\boldsymbol{s} \pmod q / 2^{k-1} \rceil$,即 MSB.

简单来说,敌手选择适当 M 值放在指定的密文位置,即敌手选择 $\boldsymbol{C}[k]=(0,0,\cdots,0,M,0,\cdots,0)^{\mathrm{T}}$,并逐位计算得到该私钥的各位. 例如为了计算 $t_1 \in \mathbb{Z}_q$,令 $\boldsymbol{C}[k]=(0,1,0,\cdots,0)^{\mathrm{T}}$,然后对该密文矩阵进行一次解密预言机查询. 因此:

$$\langle \boldsymbol{C}[k], \boldsymbol{s} \rangle = -t_1,$$

所以敌手能够得到 t_1 的 MSB 之后,重新计算(例如通过选择密文向量 $\boldsymbol{C}[k]=(0,2,0,\cdots,0)$)获得 $2(-t_1) \pmod q$ 的 MSB,这样就产生了关于下一个 MSB 的信息(但是,如果当 $MSB=1$ 时,敌手需要考虑到规约,即进行降模处理). 而实际攻击时,为了区分正负值,敌手可以选择形如 $\boldsymbol{C}[k]=(M,1,0,\cdots,0)$ 的密文向量,因为该向量提供 $\langle \boldsymbol{C}[k], \boldsymbol{s} \rangle = M-t_1$ 的 MSB. 本文不再赘述,有关该类攻击的详细讨论可参见文献[3].

如上描述,调用解密预言机所查询矩阵的第 k 列是一个单位向量. 因此,一个通用的方法是通过禁止选择这种形式的密文来避免这种攻击. 但是由于 GSW 方案是同态的,攻击者总能够将一个随机的 0 的加密添加到密文上,使得该修改后的密文矩阵 $\boldsymbol{C}$ 与其他密文矩阵一样,不能被解密算法区分而正常解密. 当然,正如 Loftus 等人[6]所做的工作,考虑其他形式的解密算法,只需确定一个密文是否"正确地形成"即可,即是否是"有效密文". 但李增鹏等人[19]则提出"多密钥的同态加密"方案来抵抗上述自适应攻击 1,将在第 4 节给出更详细的描述.

2) 自适应攻击 2. 上述攻击 1 的主要目标是得到 $\boldsymbol{b}=\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e}$ 中的秘密值 $\boldsymbol{t}$. 然而,如果能够计算出噪声向量 $\boldsymbol{e}$,然后利用公共矩阵 $\boldsymbol{B}$ 和 LWE 实例 $\boldsymbol{b}$ 所存在的运算关系,也能够确定秘密向量 $\boldsymbol{t}$ 的值. 因此本文给出了一个方法来确定噪声向量 $\boldsymbol{e}$. 具体步骤为:

步骤 1. 对于任意的 $1 \leqslant j \leqslant m$,为了得到噪声向量 $\boldsymbol{e}$ 的第 j 个噪声项 e_j,首先考虑公钥矩阵 $\boldsymbol{A}$ 的第 j 行,并记为 $\boldsymbol{a}^j=(\boldsymbol{b}^j\boldsymbol{t}+e_j \pmod q \mid \boldsymbol{b}^j) \in \mathbb{Z}_q^{1\times(n+1)}$. 然后,将这一行插入到密文矩阵的第 k 列中,因此设 $\boldsymbol{C}[k]=(\boldsymbol{a}^j)^{\mathrm{T}}$,并令 $\boldsymbol{C}$ 的其他列为 0.

步骤 2. 在上述情况下,解密预言机计算:

$$\langle \boldsymbol{C}[k], \boldsymbol{s} \rangle = \boldsymbol{C}[k]^{\mathrm{T}}\boldsymbol{s} = \boldsymbol{a}^j\boldsymbol{s} = (\boldsymbol{b}^j\boldsymbol{t}+e_j \pmod q \mid \boldsymbol{b}^j)(1,-\boldsymbol{t}^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{b}^j\boldsymbol{t}+e_j-\boldsymbol{b}^j\boldsymbol{t}=e_j,$$

并返回 $\left| \lfloor e_j / 2^{k-1} \rceil \right|$. 因此能够得到 e_j 的 MSB(因为该位肯定为 0,因为 e_j 应该是小于 q 的).

步骤 3. 为了扩展该攻击,敌手选择一个整数 $-q/2<\mu<q/2$,并利用 $\boldsymbol{a}^j=\boldsymbol{a}^j+(\mu \mid 0,0,\cdots,0)$ 来替换 $\boldsymbol{a}^j$,并通过逐步尝试 $\mu=1,2,4,\cdots,2^d$ 值来重复进行解密计算,以此实现对方案的自适应攻击. 因此解密预言机返回 $\left| \lfloor (e_j+\mu \pmod q)/2^{k-1} \rceil \right|$,并确定该不等式 $|e_j+\mu|<2^{k-2}$ 是否成立.

步骤 4. 通过尝试 $\mu=1,2,4,\cdots,2^d$ 能够确定使得 $e_j+2^d \geqslant 2^{k-2}$ 成立的 2 的最小乘幂. 这意味着不等式 $2^{k-2}-2^d \leqslant e_j < 2^{k-2}-2^{d-1}$ 成立. 此外,利用二进制搜索类型算法,使用大约 d 次解密预言机查询,可以精确地确定噪声项 e_j.

3 MGSW 方案

第 3 节中已经概述了对 GSW 全同态加密方案的 2 种自适应攻击方法. 本节首先概述李增鹏等人[19]抵抗自适应攻击 1 的多私钥 GSW 全同态加密方案(MGSW). 然后,给出对 MGSW 方案的自适应攻击 2.

3.1 MGSW 方案

为抵抗该攻击 1,李增鹏等人[19]改进了 GSW 方案的密钥生成算法. 而 GSW 方案的密钥生成算法是选择形如 $[\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e} \mid \boldsymbol{B}]$ 的公钥 $\boldsymbol{A}$,其中私钥 $\boldsymbol{t}$ 是均匀随机选取的,$\boldsymbol{e}$ 是一个短噪声向量,而李增鹏等人[19]则不再仅使用只包含单个 LWE 实例的公钥,而是采用结构的新公钥矩阵:

$$\boldsymbol{A}'=[\boldsymbol{B}\boldsymbol{t}^1+\boldsymbol{e}^1 \mid \boldsymbol{B}\boldsymbol{t}^2+\boldsymbol{e}^2 \mid \cdots \mid \boldsymbol{B}\boldsymbol{t}^\ell+\boldsymbol{e}^\ell \mid \boldsymbol{B}],$$

其中,秘密向量 $\boldsymbol{t}^1,\boldsymbol{t}^2,\cdots,\boldsymbol{t}^\ell$ 均匀取自分布 $\mathbb{Z}_q^n$,噪声向量 $\boldsymbol{e}^1,\boldsymbol{e}^2,\cdots,\boldsymbol{e}^\ell$ 取自离散高斯分布 χ^m. 为方便,记噪声元素为 $\boldsymbol{e}^i=(e_1^i,e_2^i,\cdots,e_m^i)^{\mathrm{T}}$. 基于此,令私钥为 $\boldsymbol{s}^1=(1,0,\cdots,0,-(\boldsymbol{t}^1)^{\mathrm{T}})^{\mathrm{T}},\cdots,\boldsymbol{s}^\ell=(0,\cdots,0,1,-(\boldsymbol{t}^\ell)^{\mathrm{T}})^{\mathrm{T}}$. 因此 $\boldsymbol{A}'\boldsymbol{s}^i=\boldsymbol{e}^i \pmod q$,其中 $1 \leqslant i \leqslant \ell$. 最后,每运行一次解密算法,生成一个新的随机一次秘密:

$$\boldsymbol{s}'=\sum_{i=1}^{\ell}\lambda_i\boldsymbol{s}^i,$$

当整数 λ_i 很小(例如可以取 $\lambda_i \in \{0,1\}$ 或 $\{-1,0,1\}$ 或取自一个离散高斯分布),那么:

$$\hat{\boldsymbol{e}}=\boldsymbol{A}'\boldsymbol{s}'=\sum_{i=1}^{\ell}\lambda_i\boldsymbol{A}'\boldsymbol{s}^i=\sum_{i=1}^{\ell}\lambda_i\boldsymbol{e}^i$$

是一个短向量,且至少存在 2^{ℓ} 个可能的私钥.因此,在这种情况下,即使为了确保在实际使用过程中,方案中所有的私钥使用次数不超过一次,那么 ℓ 也并不需要很大的取值.

简要回顾李增鹏等人[19]的 MGSW 方案.

1) MGSW 初始化算法

$params \leftarrow MGSW.Setup(1^{\kappa},1^{L})$:

① 与 GSW 方案的初始化算法相同,区别在于选择参数 $\phi=O(\lg n)$(私钥的数量).

② 输出公共参数 $params=(n,q,\chi,m,\phi)$,并令 $\ell=\lfloor \lg q \rfloor+1$ 且 $N=(\phi+n)\ell$.

2) MGSW 密钥生成算法

$(pk,sk)\leftarrow MGSW.KeyGen(params)$:

① 均匀选取 $\boldsymbol{t}^i \leftarrow \mathbb{Z}_q^n, i\in[\phi]$并输出:

$$\boldsymbol{s}^i \leftarrow (\boldsymbol{I}_i \mid -(\boldsymbol{t}^i)^{\mathrm{T}})^{\mathrm{T}}=$$

$$(0,0,\cdots,1,\cdots,0,-t_1^i,-t_2^i,\cdots,-t_n^i)^{\mathrm{T}}\in\mathbb{Z}_q^{n+\phi},$$

其中,$\boldsymbol{I}_i$ 是$(\phi\times\phi)$维单位矩阵 $\boldsymbol{I}$ 的第 i 行且使得 $\boldsymbol{s}^i$ 的第 i 个坐标等于 1.

② 对于 $i\in[\phi]$,均匀选取公共矩阵 $\boldsymbol{B}\leftarrow\mathbb{Z}_q^{m\times n}$和 ϕ 个噪声向量 $\boldsymbol{e}^i\leftarrow\chi^m$.

③ 计算 $\boldsymbol{b}^i=\boldsymbol{B}\boldsymbol{t}^i+\boldsymbol{e}^i\in\mathbb{Z}_q^m$,并生成公钥矩阵$\boldsymbol{A}'=[\boldsymbol{b}^i\mid\boldsymbol{b}^2\mid\cdots\mid\boldsymbol{b}^{\phi}\mid\boldsymbol{B}]\in\mathbb{Z}_q^{m\times(n+\phi)}$.

④ 输出 $pk\leftarrow\boldsymbol{A}'$和 $sk\leftarrow\{\boldsymbol{s}^1,\boldsymbol{s}^2,\cdots,\boldsymbol{s}^{\phi}\}$.

3) MGSW 加密算法 $\boldsymbol{C}\leftarrow MGSW.Enc(pk,\mu)$:

① 加密一个消息 $\mu\in\mathbb{Z}_q$,均匀选取一个随机矩阵 $\boldsymbol{R}'\in\{0,1\}^{m\times N}$.

② 计算并输出密文 $\boldsymbol{C}=\mu\boldsymbol{G}+\boldsymbol{A}'^{\mathrm{T}}\boldsymbol{R}'\in\mathbb{Z}_q^{(n+\phi)\times N}$,其中 $\boldsymbol{G}$ 为$(n+\phi)\times N$ 维工具矩阵.

4) MGSW 解密算法 $\mu'\leftarrow MGSW.Dec(sk,\boldsymbol{C})$:

① 从$\{0,1\}$均匀选择非 0 的 $\lambda_1,\lambda_2,\cdots,\lambda_{\phi}$,并生成一个一次密钥 $\boldsymbol{s}'=\sum_{i=1}^{\phi}\lambda_i\boldsymbol{s}^i$.

② 选择一个整数 $1\leqslant i\leqslant\phi$,使得 $\lambda_i=1$,且令 $k=(i-1)\ell+j$,使得 $\boldsymbol{G}$ 的第 k 列的第 i 个数是 2^{j-1},其中 $q/4<2^{j-1}\leqslant q/2$.

③ 在$(-q/2,q/2]$范围内计算:

$$x=\langle\boldsymbol{C}[k],\boldsymbol{s}\rangle(\mathrm{mod}\ q)=\boldsymbol{C}[k]^{\mathrm{T}}\boldsymbol{s}(\mathrm{mod}\ q).$$

④ 输出 $\mu'=\left\lfloor\lfloor x/2^{j-1}\rceil\right\rfloor$.

同态运算跟原始 GSW 方案完全相同.

3.2 对 MGSW 方案的自适应攻击(自适应攻击 2)

该类攻击的主要目标是利用噪声向量 $\boldsymbol{e}$ 来恢复秘密向量 $\boldsymbol{t}$,为实现该目标,需要将 $\boldsymbol{C}$ 的第 k 列设成 $\boldsymbol{A}$ 的第 j 行 $\boldsymbol{a}^j$ 的转置,并将 $\boldsymbol{C}$ 的所有其他列设为 0.实际上,与对 MGSW 方案的自适应攻击 1 的最主要不同在于 k 值是变化的,且攻击者并不知晓.显然,该攻击方法比自适应攻击 1 要困难的多.重复使用上述方法去计算向量 $\boldsymbol{e}^1$ 的第 j 个元素 e_j^1,攻击者便可以获得向量 $\boldsymbol{e}^1$,那么敌手就能通过使用格算法来解决 LWE 的一个更简单的实例$(\boldsymbol{B},\boldsymbol{b}'=\boldsymbol{B}\boldsymbol{t}^1+\hat{\boldsymbol{e}})$从而完成密码分析.

对该攻击的具体描述如下,令:

$$\boldsymbol{a}^j=(\boldsymbol{b}^j\boldsymbol{t}^1+e_1^j,\boldsymbol{b}^j\boldsymbol{t}^2+e_2^j,\cdots,\boldsymbol{b}^j\boldsymbol{t}^{\phi}+e_{\phi}^j,\boldsymbol{b}_j),$$

且一次私钥是 $\boldsymbol{s}'=(\lambda_1,\lambda_2,\cdots,\lambda_{\phi},\sum_i\lambda_i\boldsymbol{t}^i)^{\mathrm{T}}$. 令 k'使得 $q/4<2^{k'-1}\leqslant q/2$ 成立,将 $\boldsymbol{a}_j$ 放入 $\boldsymbol{C}$ 的第 k'列,并且将其他所有列全设为 0.攻击者希望解密算法选择 $\lambda_1=1$ 和 $k=k'$.那么有 1/2 的概率使得 $\lambda_1=1$,平均情况下大约有 $\phi/2$ 的概率使得 $\lambda_i=1$.

这里需注意的是,对于攻击者为什么能够获知 $\lambda_i=1$,其原因在于,当解密者选择一些 i 并令 $\lambda_i=1$,进而计算 $\hat{\boldsymbol{e}}=\sum_{i=1}^{\phi}\lambda_i\boldsymbol{e}^i$ 和 $\boldsymbol{C}[k]^{\mathrm{T}}\cdot\hat{\boldsymbol{e}}=\mu\lambda_i 2^{j-1}+E$.而攻击者则选择一个 $\boldsymbol{C}$ 使得除第 k 列 $\boldsymbol{C}[k]$外其他所有列均为 0.那么,如果 $\lambda_i\neq1$,解密返回值为 0.而如果攻击者从解密预言机中获得一个非零值,那么他就确定 $\lambda_i=1$.

因此,解密预言机选择 $k=k'$的概率大约为 $(1/2)(2/\phi)=1/\phi$.若 $k=k'$那么解密预言机计算:

$$\boldsymbol{C}[k]^{\mathrm{T}}\boldsymbol{s}'=\boldsymbol{a}^j\boldsymbol{s}'=\sum_i\lambda_i(\boldsymbol{b}^j\boldsymbol{t}^i+e_i^j)-\boldsymbol{b}^j\sum_{i\in[\phi]}\lambda_i\boldsymbol{t}^i=$$

$$e_1^j+\sum_{i=2}^{\phi}\lambda_i e_i^j(\mathrm{mod}\ q).$$

换句话说,尽管攻击者可以“看到”e_1^j,但实际却是一个含有噪音项的 $E=\sum_{i=2}^{\phi}\lambda_i e_i^j$. 而这里,由于 e_i^j 项是固定的,噪音项 E 的分布主要在于 λ_i 的选择.然而,因为 e_i^j 最初取自离散高斯,这使得 E 的均值接近于 0 且 E 的分布类似于高斯分布.因此,很自然地希望噪音项 E 能够通过重复进行解密预言机查询而被“平均掉”,从而恢复出 e_1^j.

具体来说,攻击者选择一个恰当的整数 $-q/2<u<q/2$,并对密文矩阵 $\boldsymbol{C}$ 调用解密预言查询,这里的密文矩阵 $\boldsymbol{C}$ 的第 k'列是向量 $\bar{\boldsymbol{a}}^j=\boldsymbol{a}^j+(u\mid0,0,\cdots,0)$的转置.因此,解密预言机(假设 $k=k'$)计算 $e_1^j+\mu+E(\mathrm{mod}\ q)$,其中 E 是噪音项,并返回 $\left\lfloor\lfloor(e_1^j+\mu+E(\mathrm{mod}\ q))/2^{k-1}\rceil\right\rfloor$.

为获得 e_1^i 的 *MSB*,采取与 Chenal 和 Tang[3] 相同的方法来获得 e_1^i 的 *MSB*. 因此攻击者在得知 $\left\lfloor (e_1^i+\mu+E(\mathrm{mod}\ q))/2^{k-1} \right\rceil$ 后判定 $|e_1^i+\mu+E|<2^{k-2}$ 是否成立. 因此,重复进行相同的查询(同时要求 $k=k'$)将给出相同的计算但对应相同的噪音值 E 是不同的,直到满足上述不等式,即可获得 e_1^i 的 *MSB*.

显然该攻击方法比自适应攻击 1 要困难的多. 因此,李增鹏等人[26]的多密钥 GSW 方案仅能抵抗部分自适应攻击. 由上述描述,需要考虑一个不同的解决方案.

注 4:实际上,完全使用该方法去计算 e_1^i 可能是非常困难的,因为至少对于攻击者而言,并不知道分布 E 的均值,上述方式仅是近似获得 E 的均值. 但攻击者一旦获得噪声向量 $\boldsymbol{e}^1$ 足够多的信息,那么他们就能通过使用格算法来解决 LWE 的一个更简单的实例($\boldsymbol{B}, \boldsymbol{b}'=\boldsymbol{B}\boldsymbol{t}^1+\hat{\boldsymbol{e}}$),从而完成密码分析,其中 $\|\hat{\boldsymbol{e}}\| \ll \|\boldsymbol{e}^1\|$.

4 DMGSW 方案

在本节中,提出一个具有多私钥 DMGSW 方案,DMGSW 方案的安全性基于非齐次短整数解(inhomogeneous short integer solution, ISIS)问题,而非 LWE 问题.

定义 12. ISIS. 令 $q,n,m\in\mathbb{N}, m>n$. 令 χ 为 $\mathbb{Z}$ 上一个分布. 将 $\mathbb{Z}_q^n\times\mathbb{Z}_q^{n\times m}$ 上的 ISIS 分布定义为

$$(\boldsymbol{B}\boldsymbol{t}(\mathrm{mod}\ q), \boldsymbol{B}),$$

其中,$\boldsymbol{B}$ 为在 $\mathbb{Z}^q$ 中均匀选择的一个 $n\times m$ 矩阵,$\boldsymbol{t}\leftarrow\chi^m$ 是一个长度为 m 的整数向量. 类似于 decision-LWE 问题,判定版本 ISIS 问题(decision-ISIS)是将取自 ISIS 分布中的样本$(\boldsymbol{u},\boldsymbol{B})$区分于取自均匀分布的样本 $\mathbb{Z}_q^n\times\mathbb{Z}_q^{n\times m}$.

Ajtai[28](或 Micciancio[29])已经证明了存在一个分布 χ 使得对于恰当选取的参数来说,ISIS 问题是困难的. 实际上,分布 χ 可以取自一个离散高斯分布或取自均匀分布$\{0,1\}$.(详见文献[29]).

定理 2[19]. 令 $m>n\in\mathbb{N}, q\in\mathbb{N}$,$\chi$ 是 $\mathbb{Z}$ 上的一个离散高斯分布,它使得 ISIS 问题是困难的,ϕ 为整数且满足 $\phi=O(\mathrm{lb}\, q)$. 定义 2 个分布 $\mathcal{X}$和 $\mathcal{Y}$:

① $\mathcal{X}$是 $n\times(\phi+m)$矩阵上的分布$[\boldsymbol{u}^1|\boldsymbol{u}^2|\cdots|\boldsymbol{u}^\phi|\boldsymbol{B}]$,其中 $\boldsymbol{B}\in\mathbb{Z}_q^{n\times m}$是均匀随机选取的,并且对于所有的 $1\leqslant i\leqslant\phi$ 满足,$\boldsymbol{u}^i=\boldsymbol{B}\boldsymbol{t}^i(\mathrm{mod}\ q)$,这里 $\boldsymbol{t}^i$ 取自离散高斯分布 χ^m.

② $\mathcal{Y}$ 是 $\mathbb{Z}_q^{n\times(\phi+m)}$ 上的均匀分布,那么 2 个分布 $\mathcal{X}$ 和 $\mathcal{Y}$ 是计算不可区分的.

证明. 令 $\mathcal{D}$ 是一个 PPT 敌手,他能够以不可忽略的概率区分 $\mathcal{X}$和 $\mathcal{Y}$. 对于 $1\leqslant i\leqslant\phi+1$ 来说,引入中间分布 $\mathcal{X}_i$ 如下:

$$[\boldsymbol{d}^1|\boldsymbol{d}^2|\cdots|\boldsymbol{d}^{i-1}|\boldsymbol{u}^i|\cdots|\boldsymbol{u}^\phi|\boldsymbol{B}],$$

其中,$\boldsymbol{u}^i$ 如上所述,$\boldsymbol{d}^i$ 是从 $\mathbb{Z}_q^n$中均匀选取的. 因此 $\mathcal{X}_1=\mathcal{X}$ 且 $\mathcal{X}_{\phi+1}=\mathcal{Y}$.

由上假设,$\mathcal{D}$ 能够以明显的概率 ε 区分 $\mathcal{X}_1$ 和 $\mathcal{X}_{\phi+1}$,因此,通过一系列游戏序列,存在某个 i 使得 $\mathcal{D}$ 能以至少 ε/ϕ 的概率将分布 $\mathcal{X}_i$ 和 $\mathcal{X}_{i+1}$区分开来.

显然,$\mathcal{D}$ 给出了一个 ISIS 区分器:即给出一个 ISIS 挑战($\boldsymbol{y},\boldsymbol{B}$),均匀地选取 $\boldsymbol{d}^1,\boldsymbol{d}^2,\cdots,\boldsymbol{d}^{i-1}$,从 ISIS 分布中选取 $\boldsymbol{u}^{i+1},\boldsymbol{u}^{i+2},\cdots,\boldsymbol{u}^\phi$,生成分布

$$[\boldsymbol{d}^1|\boldsymbol{d}^2|\cdots|\boldsymbol{d}^{i-1}|\boldsymbol{y}|\boldsymbol{u}^{i+1}|\boldsymbol{u}^{i+2}|\cdots|\boldsymbol{u}^\phi|\boldsymbol{B}],$$

并调用 $\mathcal{D}$ 对该分布进行区分.

通过假设,显然不存在这样的区分器. 证毕.

4.1 DMGSW 方案

本节中给出一个 DMGSW 方案. 在原 GSW 方案中,公钥是基于 LWE 实例,形如 $\boldsymbol{A}=(\boldsymbol{B}\boldsymbol{t}+\boldsymbol{e},\boldsymbol{B})$,密文是基于 ISIS 问题,形如 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}$. 如 Regev 的加密方案,其对偶方案有基于 ISIS 问题的公钥($\boldsymbol{B}^{\mathrm{T}}$,$\boldsymbol{B}^{\mathrm{T}}\boldsymbol{T}$)和基于类似 LWE 实例 $\boldsymbol{B}\boldsymbol{R}+\boldsymbol{X}$ 的密文.

1) DMGSW 初始化算法

$params\leftarrow DMGSW.Setup(1^\kappa,1^L)$:

① 选择模 $q=q(\kappa)$,格维参数 $m=m(\kappa,L)$和 n,以及分布 $\chi=\chi(\kappa,L)$,对于已知 ISIS 攻击,参数的合理选择可以实现至少 2^κ 的安全性.

② 令 $\ell=\lfloor \mathrm{lb}\, q\rfloor+1$ 且 $N=(\phi+m)\ell$,输出 $params=(m,q,\chi,n)$.

2) DMGSW 密钥生成算法

$(pk,sk)\leftarrow DMGSW.KeyGen(params)$:

① 从 χ^m 分布中选取向量$(\boldsymbol{t}^i)^{\mathrm{T}}=(t_1^i,t_2^i,\cdots,t_n^i)$,计算 $\boldsymbol{e}^i=(\boldsymbol{I}_i|-(\boldsymbol{t}^i)^{\mathrm{T}})^{\mathrm{T}}$,其中行向量 $\boldsymbol{I}_i$ 是 $\phi\times\phi$ 维单位矩阵的第 i 行.

② 均匀选取矩阵 $\boldsymbol{B}\in\mathbb{Z}_q^{n\times m}$,计算向量 $\boldsymbol{u}^i=\boldsymbol{B}\boldsymbol{t}^i$,$1\leqslant i\leqslant\phi$,令矩阵

$$\boldsymbol{A}=[\boldsymbol{u}^1|\boldsymbol{u}^2|\cdots|\boldsymbol{u}^\phi|\boldsymbol{B}]\in\mathbb{Z}_q^{n\times(\phi+m)},$$

这里 $\boldsymbol{A}\boldsymbol{e}^i=0$.

③ 输出公钥 $\boldsymbol{A}$ 和私钥$(\boldsymbol{e}^1,\boldsymbol{e}^2,\cdots,\boldsymbol{e}^\phi)$.

3) DMGSW 加密算法

$\boldsymbol{C}\leftarrow DMGSW.Enc(pk,\mu)$:

① 均匀选取矩阵 $\boldsymbol{R}\leftarrow\mathbb{Z}_q^{n\times N}$ 和 $\boldsymbol{X}\leftarrow\chi^{(\phi+m)\times N}$.

② 加密消息 $\mu\in\{0,1\}$,计算

$$\boldsymbol{C}=\mu\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}+\boldsymbol{X}(\mathrm{mod}\ q)\in\mathbb{Z}_q^{(\phi+m)\times N},$$

其中 $\boldsymbol{G}$ 为 $(\phi+m)\times N$ 维工具矩阵.

③ 输出密文 $\boldsymbol{C}$.

注 5. DMGSW 方案中密文 $\boldsymbol{C}$ 可以记为

$$Flatten(\mu\boldsymbol{I}+BitDecomp(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}+\boldsymbol{X}))=$$
$$BitDecomp(BitDecomp^{-1}(\mu\boldsymbol{I})+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}+\boldsymbol{X})=$$
$$BitDecomp(\mu\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}+\boldsymbol{X})(\mathrm{mod}\ q).$$

4) DMGSW 解密算法

$\mu'\leftarrow DMGSW.Dec(sk_i,\boldsymbol{C})$:

① 从 $\mathbb{Z}$ 选择 $\lambda_1,\lambda_2,\cdots,\lambda_\phi$ 且 $\lambda_i,i\in[\phi]$ 并不全为 0,那么生成一次私钥 $\hat{\boldsymbol{e}}=\sum_{i=1}^{\phi}\lambda_i\boldsymbol{e}^i$,使得 $\|\hat{\boldsymbol{e}}\|$ 很短. 这里,同样有 $\boldsymbol{A}\hat{\boldsymbol{e}}\equiv 0(\mathrm{mod}\ q)$.

② 确定整数 $1\leqslant k=(i-1)\ell+j\leqslant\phi\ell$,使得 $\lambda_i=1$ 且 $2^{j-1}\in(q/4,q/2]$.

③ 令 $\boldsymbol{C}[k]$ 为 $\boldsymbol{C}$ 的第 k 列,并计算 $\mu=\langle\boldsymbol{C}[k],\hat{\boldsymbol{e}}\rangle=\boldsymbol{C}[k]^{\mathrm{T}}\hat{\boldsymbol{e}}(\mathrm{mod}\ q)$. 这里:

$$\boldsymbol{C}^{\mathrm{T}}\hat{\boldsymbol{e}}=\mu\boldsymbol{G}^{\mathrm{T}}\hat{\boldsymbol{e}}+\boldsymbol{R}^{\mathrm{T}}\boldsymbol{A}'\hat{\boldsymbol{e}}+\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{e}}=\mu\boldsymbol{G}^{\mathrm{T}}\hat{\boldsymbol{e}}+\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{e}}.$$

因此:

$$\boldsymbol{C}[k]^{\mathrm{T}}\hat{\boldsymbol{e}}=\mu(0,0,\cdots,0,2^{j-1},0,\cdots,0)\hat{\boldsymbol{e}}+\boldsymbol{E}=\mu\lambda_i 2^{j-1}+E.$$

其中,$E=\boldsymbol{X}[k]^{\mathrm{T}}\hat{\boldsymbol{e}}$ 为一个小噪音项.

④ 返回 $\left|\lfloor\mu/2^{j-1}\rceil\right|\in\{0,1\}$.

注 6. 针对步骤 1),需特别强调的是,有不同的方法来选择参数 λ_i. 其中可以采用从 $\{0,1\}$ 分布中均匀选取的方法. 该方法可以得到恰当的 $\|\boldsymbol{e}'\|$ 值,因此本文分析中采用该方法. 另一种是采取从 $\mathbb{Z}$ 上离散高斯分布中选取的方法,该方法使得取值具有小的标准差且得到更高的安全性(详见第 6 节中的讨论). 此外,还可采用拒绝抽样(rejection sampling)的某些形式来获得向量 $\hat{\boldsymbol{e}}$. 为更好控制 λ_i 的大小并防止泄露关于向量 $\boldsymbol{e}^i$ 信息.

5) 同态运算 $DMGSW.Eval(pk,(\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_\ell))$:

同态操作与 GSW 方案类似,具体如下:

① 同态加法 $DMGSW.Add(\boldsymbol{C}_1,\boldsymbol{C}_2)$ 输出:

$$\boldsymbol{C}_1+\boldsymbol{C}_2=(\mu_1+\mu_2)\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{R}_1+\boldsymbol{R}_2)+(\boldsymbol{X}_1+\boldsymbol{X}_2)\in\mathbb{Z}_q^{(\phi+m)\times N}.$$

② 同态乘法 $DMGSW.Mult(\boldsymbol{C}_1,\boldsymbol{C}_2)$ 计算 $N\times N$ 的矩阵 $\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$,并输出 $\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$. 即:

$$\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=(\mu_1\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1+\boldsymbol{X}_1)\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=\mu_1\boldsymbol{C}_2+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)+\boldsymbol{X}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)=\mu_1\mu_2\boldsymbol{G}+(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)+\mu_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_2)+\boldsymbol{X}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)+\mu_1\boldsymbol{X}_2)\in\mathbb{Z}_q^{(\phi+m)\times N}.$$

③ 与非运算 $DMGSW.NAND(\boldsymbol{C}_1,\boldsymbol{C}_2)$ 计算 $\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$,并输出 $\boldsymbol{G}-\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$.

4.2 正确性

本节中分析解密和同态运算的正确性并确定参数大小. 假设 $\hat{\boldsymbol{e}}=\sum_{i=1}^{\phi}\lambda_i\boldsymbol{e}^i$ 是由均匀随机选取的 $\lambda_i\in\{0,1\}$ 所构成. 而 $\boldsymbol{e}^i$ 中的元素取自标准差为 σ 的离散高斯分布 χ,χ 是 B-界的,且 $B=6\sigma$,使得不等式 $\|\boldsymbol{e}^i\|\leqslant 2\sqrt{m}\sigma$ 成立. 因此,向量 $\hat{\boldsymbol{e}}$ 中的元素是以 $\boldsymbol{B}'=t\boldsymbol{B}$ 为界且满足 $\|\hat{\boldsymbol{e}}\|\leqslant 2\sqrt{tm}\sigma$.

由加密算法对消息加密输出的密文称为第 0 层的密文 $\boldsymbol{C}$. 而层级 $i\geqslant 1$ 的密文 $\boldsymbol{C}$ 则是由密文运算算法 Eval 生成. 且至多对 0 层密文执行 i 次运算.

定义 13. 如果 $\boldsymbol{C}^{\mathrm{T}}\hat{\boldsymbol{e}}=\mu\boldsymbol{G}^{\mathrm{T}}\hat{\boldsymbol{e}}+\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{e}}$,那么称密文 $\boldsymbol{C}\in\mathbb{Z}_q^{(\phi+m)\times N}$ 是 E 噪音的,这里 $\|\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{e}}\|_\infty\leqslant E$. 如果具有 E 噪音的密文 $\boldsymbol{C}$ 满足 $E<q/8$,那么有 $\boldsymbol{C}[k]^{\mathrm{T}}\hat{\boldsymbol{e}}\equiv\mu 2^{j-1}+\xi(\mathrm{mod}\ q)$ 使得不等式 $|\xi|\leqslant E<q/8<2^{j-2}$ 成立,且

$$\frac{\boldsymbol{C}[k]^{\mathrm{T}}\hat{\boldsymbol{e}}(\mathrm{mod}\ q)}{2^{j-1}}=\mu+\frac{\xi}{2^{j-1}}=\mu+\varepsilon,$$

其中,$-\frac{1}{2}<\varepsilon<\frac{1}{2}$.

引理 4. 令 χ 为 $\mathbb{Z}$ 上的一个 B 界分布. 如果 $E\geqslant\phi B+mB^2$ 成立,那么层级为 0 的密文是 E 噪音.

证明. 对于密文 $\boldsymbol{C}=\mu\boldsymbol{G}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}+\boldsymbol{X}$ 其中 $\boldsymbol{X}$ 取自高斯分布 $\chi^{(\phi+m)\times N}$. 为方便分析,将 $\boldsymbol{X}$ 记为 $\left[\begin{matrix}\boldsymbol{X}^{\sharp}\\ \boldsymbol{X}^{*}\end{matrix}\right]$,其中噪声矩阵 $\boldsymbol{X}^{\sharp}\in\mathbb{Z}^{\phi\times N}$ 且 $\boldsymbol{X}^{*}\in\mathbb{Z}^{m\times N}$. 由于 $\hat{\boldsymbol{e}}=(\lambda_1,\lambda_2,\cdots,\lambda_\phi|-\sum_i\lambda_i\boldsymbol{t}^i)^{\mathrm{T}}$,因此有:

$$\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{e}}=(\boldsymbol{X}^{\sharp})^{\mathrm{T}}(\lambda_1,\lambda_2,\cdots,\lambda_\phi)^{\mathrm{T}}-\sum_i\lambda_i(\boldsymbol{X}^{*})^{\mathrm{T}}\boldsymbol{t}^i.$$

因为 $\|\boldsymbol{X}_I^{*}\|_2\leqslant 2\sqrt{m}\sigma$ 和 $\|\boldsymbol{t}^i\|_2\leqslant 2\sqrt{m}\sigma$,由 Cauchy-Schwarz 不等式可知:

$$|(\boldsymbol{X}_I^{*})^{\mathrm{T}}\boldsymbol{t}^i|\leqslant 4m\sigma^2\leqslant mB^2.$$

因此,有 $|(\boldsymbol{X}_I^{\sharp})^{\mathrm{T}}(\lambda_1,\lambda_2,\cdots,\lambda_\phi)^{\mathrm{T}}|\leqslant\phi B$. 综上,无穷范数的误差边界为 $\phi B+mB^2$. 证毕.

下面分析同态运算噪声规模. 具体来说,对于第 i 层的密文噪声 $(N+1)^iE$,记 E 为层级 0 的噪声. 如果满足条件 $(N+1)^LE\leqslant q/8$,那么执行 L 层同态操作后,解密正确.

引理 5. 令符号和参数如上所述(特别是,$E \geqslant \phi B + mB^2$). 令 $\boldsymbol{C}$ 为层级 $i \leqslant L$ 上的任意密文,那么 $\boldsymbol{C}$ 的噪音是 $(N+1)^i E$(无穷范数).

证明. 令 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 为 $\mu_1, \mu_2 \in \{0,1\}$ 的 2 个密文,满足 $(\boldsymbol{C}_i)^{\mathrm{T}} \hat{\boldsymbol{e}} = \mu_i \boldsymbol{G}^{\mathrm{T}} \hat{\boldsymbol{e}} + E_i, i = 1,2$.

对于密文加法,假设其 i 层密文噪声为 $(N+1)^i E$. 计算 $\boldsymbol{C}^{\mathrm{add}} = \boldsymbol{C}_1 + \boldsymbol{C}_2$. 那么:

$$(\boldsymbol{C}^{\mathrm{add}})^{\mathrm{T}} \hat{\boldsymbol{e}} = (\mu_1 + \mu_2) \boldsymbol{G}^{\mathrm{T}} \hat{\boldsymbol{e}} + (E_1 + E_2)$$

且

$$\| E_1 + E_2 \|_\infty \leqslant \| E_1 \|_\infty + \| E_2 \|_\infty \leqslant 2(N+1)^i E \leqslant (N+1)^{i+1} E.$$

对于密文乘法,$\boldsymbol{C}^{\mathrm{mult}} = \boldsymbol{C}_1 \boldsymbol{G}^{-1}(\boldsymbol{C}_2)$ 满足:

$$(\boldsymbol{C}^{\mathrm{mult}})^{\mathrm{T}} \hat{\boldsymbol{e}} = \mu_1 \mu_2 \boldsymbol{G}^{\mathrm{T}} \hat{\boldsymbol{e}} + \boldsymbol{G}^{-1}(\boldsymbol{C}_2)^{\mathrm{T}} E_1 + \mu_2 E_2,$$

使得:

$$\| \boldsymbol{G}^{-1}(\boldsymbol{C}_2)^{\mathrm{T}} E_1 + \mu_2 E_2 \|_\infty \leqslant \| \boldsymbol{G}^{-1}(\boldsymbol{C}_2)^{\mathrm{T}} E_1 \|_\infty + \| \mu_2 E_2 \|_\infty \leqslant N \| E_1 \|_\infty + \| E_2 \|_\infty.$$ 证毕.

同样的计算对 NAND 门来说同样成立. 假设 $q/8 > (N+1)^L E$,计算包含 NAND 门且电路深度为 L 的布尔电路. 输入噪声为 E 层级 0 的密文,每执行一次同态运算,噪声乘以至多为 $(N+1)$ 的一个因子. 因此,经过 L 次同态运算后,最终密文中的噪声大小为 $(N+1)^L$,且能够正确地解密.

4.3 DMGSW 方案的安全性

DMGSW 方案的安全性依赖于 ISIS 和 LWE 假设. 利用定理 2 来证明该方案 DMGSW 在 ISIS 假设下是安全的.

定理 3. 如果 ISIS 假设和 LWE 假设是困难的,那么 DMGSW 方案是 IND-CPA 安全的,对于 $\phi = O(\mathrm{lb}\, q)$ 及参数 m, n, q, χ.

证明. 安全性证明以 hybrid hop 形式,分 2 步给出. 可以用 hybrid hop 形式表述.

1) 使用一个均匀随机矩阵 $\boldsymbol{A}$ 替换公钥.

2) 使用一个均匀随机矩阵 $\boldsymbol{C}$ 替换密文. 当 $\boldsymbol{C}$ 是一个均匀随机矩阵时,敌手在 IND-CPA 游戏中不具有不可忽略的优势,因此 $\boldsymbol{C}$ 独立于消息 μ. 有必要说明的是,敌手的行为在不同的游戏步骤之间是相同的.

① Hybird. 1. 在本游戏中,用均匀矩阵代替公钥. 挑战密文与方案中的相同. 如果这 2 个游戏中,敌手成功的概率是不可忽略的,那么敌手是一个能够将含有 ISIS 实例的公钥与均匀随机矩阵区分开的算法. 然而,由 ISIS 假设和定理 2 可知,该算法不存在.

② Hybird. 2. 在本游戏中,用 $\mathbb{Z}_q^{(\phi+m)\times N}$ 上的均匀随机的矩阵来替换密文 $\boldsymbol{C}$. 如果在这个游戏中,敌手成功的概率明显区别于 Hybird. 1 中成功的概率,那么存在一个区分器来区分 LWE 分布 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{R} + \boldsymbol{X} \pmod q$ 和均匀分布.

由 LWE 假设,不存在这样的 PPT 区分器. 最后,Hybird. 2 与消息位 μ 无关,因此敌手在这个游戏中获得的优势为 0. 证毕.

5 抵抗自适应攻击的 DMGSW 方案的安全性

在 DMGSW 方案中,与 MGSW 方案的主要区别在于没有噪音项. 因此自适应攻击 2 不可能发生. 下面,证明自适应攻击 1 同样不适用于 DMGSW 方案.

该安全性分析依赖于剩余 Hash 引理(leftover hash lemma, LHL)的高斯版本,而不同于之前的平均情况. 下面,给出 Agrawal 等人[30]的定理 2 的一种特殊情况.

定理 4. 一维剩余 Hash 引理 LHL. 令 $\varepsilon, \sigma \in \mathbb{R}$,使得对于所有的绝对常数 $\varepsilon > 0$,$\sigma > C$(详见文献[30]). 令 $\phi \geqslant 10\,\mathrm{lb}(8\phi^{1.5}\sigma)$ 且 $s' \leqslant 4\phi\,\mathrm{lb}(1/\varepsilon)$. 那么统计距离为 2ε 的 2 个分布:

① 选择 1 个长度为 t 的向量 $\boldsymbol{X} \in \mathbb{Z}^\phi$,其每个元素均选自 $\mathbb{Z}^\phi$ 上参数为 σ 的离散高斯分布;选择 1 个长度为 ϕ 的向量 $\boldsymbol{z} \in \mathbb{Z}^\phi$,其每个元素选自 $\mathbb{Z}^\phi$ 上的参数为 s' 的离散高斯分布,计算并输出 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{z}$.

② 从 $\mathbb{Z}$ 上参数为 $\sigma s'$ 的离散高斯分布选择并输出一个元素. 实际上,对于一次私钥

$$\hat{\boldsymbol{e}} = \sum_{i=1}^{\phi} \lambda_i \boldsymbol{e}^i = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_\phi \\ -\sum_{i=1}^{\phi} \lambda_i \boldsymbol{t}^i \end{pmatrix} \in \mathbb{Z}_q^{\phi+m}.$$

执行解密计算 $\langle \boldsymbol{C}[k], \hat{\boldsymbol{e}} \rangle = \boldsymbol{C}[k]^{\mathrm{T}} \hat{\boldsymbol{e}} \pmod q$. 如果敌手将其选择的一个向量插入到密文矩阵 $\boldsymbol{C}$ 的第 k 列中并进行一次解密查询. 如自适应攻击 1,敌手不知道 k,但却能以很高的概率猜出该值. 实际上,敌手有 1/2 的概率获得 $\lambda_1 = 1$,当 $\lambda_1 = 1$ 时,将 $\boldsymbol{C}$ 的所有其他列设置为 0,能够确保解密预言机仅返回一个非 0 值.

采取与自适应攻击 1 相同的方式. 对于线性映

射 $L:\mathbb{Z}_q^{\phi+m}\to\mathbb{Z}^q$,对应着与密文 $\boldsymbol{C}[k]^{\mathrm{T}}$ 的乘法.此时,敌手可以获得:

$$L(\hat{\boldsymbol{e}})=\sum_{i=1}^{\phi}\lambda_i L(\boldsymbol{e}^i)=\sum_{i=1}^{\phi}\lambda_i L((\boldsymbol{I}_i,-(\boldsymbol{t}^i)^{\mathrm{T}})^{\mathrm{T}})$$

的1位.正如自适应攻击1,可以声明,几乎所有的向量 $(\boldsymbol{I}_i,-(\boldsymbol{t}^i)^{\mathrm{T}})^{\mathrm{T}}$ 并不在 L 的核中.依据是否含有 L 个有缺陷的向量.需要考虑2种情况:

1) 当 $L(\boldsymbol{e}^i)$ 值与在 $\mathbb{Z}_q$ 上均匀选取的元素一样时,那么在该情况下,基于在自适应攻击1中的剩余Hash引理(并假设 L 是满射且大部分私钥不会失效)足以推断 $L(\hat{\boldsymbol{e}})$ 上值与 $L(\boldsymbol{e}^i)$ 上值无关且敌手不能从该形式的查询中得到私钥.

2) 如果投影 L 将 $L((w_1,w_2,\cdots,w_{\phi+m})^{\mathrm{T}})=w_i$ 投影到坐标 $\phi<i\leqslant\phi+m$ 上.在这种情况中,$L(\boldsymbol{e}^i)$ 值取自离散高斯分布且不能使用剩余Hash引理来判定 $L(\hat{\boldsymbol{e}})$ 不携带关于 $L(\boldsymbol{e}^i)$ 的任何信息.不同的是,对于该情形,可以使用定理4来证明.在这种情况中,需假设 λ_i 是从一个参数为 σ 的离散高斯分布 χ 中选取的.

那么假设攻击者看到 $\sum_{i=1}^{\phi}\lambda_i L(\boldsymbol{e}^i)$,这里 $L(\boldsymbol{e}^i)$ 独立取自离散高斯分布 χ.那么由定理4可知,对于 $\phi=O(\mathrm{lb}\,q)$(由于 $\sigma=O(\sqrt{m})$,使得LWE假设是困难的并满足 $\phi=O(\mathrm{lb}\,\sigma)=O(\mathrm{lb}\,q)$,并且该整数 ϕ 与取自参数为 σ^2 的离散高斯中的样本不可区分.而在本文中,$L(\boldsymbol{e}^i)$ 是固定的而不是独立选取的,且敌手不可获得.

而从敌手的视角,仅可以看到 $\sum_{i=1}^{\phi}\lambda_i L(\boldsymbol{e}^i)$,$(\lambda_1,\lambda_2,\cdots,\lambda_\phi)\in\chi^\phi$,但是高斯剩余Hash引理仍能保证该值与 $L(\boldsymbol{e}^i)$ 相互独立.此外,正如前面所解释的,攻击者仅能看到一个位而不是整个值 $\sum_{i=1}^{\phi}\lambda_i L(\boldsymbol{e}^i)$,因此,本文的DMGSW方案抵抗自适应攻击.

6 总 结

自适应攻击是威胁目前主流全同态加密安全性的一种有力的攻击模式.本文给出了对李增鹏等人[19]的多私钥GSW方案的一种自适应攻击方法,并构造了一个抵抗该类自适应攻击的对偶多私钥GSW(DMGSW)方案.该方案安全性基于ISIS假设和LWE假设,同时本文给出详细的安全性分析,并证明了该方案对"一些已知"的自适应攻击的抵抗性.

参 考 文 献

[1] Regev O. On lattices, learning with errors, random linear codes, and cryptography [C] //Proc of the 37th ACM Symp on the Theory of Computing (STOC'05). New York: ACM, 2005: 84-93

[2] Gentry C, Peikert C, Vaikun-tanathan V. Trapdoors for hard lattices and new cryptographic constructions [C] //Proc of the 40th ACM Symp on the Theory of Computing (STOC'08). New York: ACM, 2008: 197-206

[3] Chenal M, Tang Qiang. On key recovery attacks against existing somewhat homomorphic encryption schemes [C] //Proc of the 3rd Cryptology and Information Security in Latin (LaintCrypt'14). Berlin: Springer, 2014: 239-258

[4] Chenal M, Tang Qiang. Key recovery attacks against NTRU-based somewhat homomorphic encryption schemes [C] //Proc of the 18th Information Security Conf (ISC'15). Berlin: Springer, 2015: 397-418

[5] Dahab R, Galbraith D S, Morais E. Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes [C] //Proc of the 8th Int Conf on Information Theoretic Security (ICITS'15). Berlin: Springer, 2015: 283-296

[6] Loftus J, May A, Smart P N, et al. On CCA-secure somewhat homomorphic encryption [C] //Proc of the 18th Selected Areas in Cryptography (SAC'11). Berlin: Springer, 2011: 55-72

[7] Zhang Zhenfei, Plantard T, Susilo W. On the CCA-1 security of somewhat homomorphic encryption over the integers [C] //Proc of the 8th Information Security Practice and Experience (ISPEC'12). Berlin: Springer, 2012: 353-368

[8] Brakerski Z. Fully homomorphic encryption without modulus switching from classical gapsvp [C] //Proc of the 32nd Int Cryptology Conf (CRYPTO'12). Berlin: Springer, 2012: 868-886

[9] Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) Fully homomorphic encryption without bootstrapping [C] //Proc of Innovations in (Theoretical) Computer Science (ITCS'12). New York: ACM, 2012: 309-325

[10] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE [C] //Proc of the 52nd IEEE Annual Symp on Foundations of Computer Science (FOCS'11). Piscataway, NJ: IEEE, 2011: 97-106

[11] Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based [C] //Proc of the 33rd Int Cryptology Conf (CRYPTO'13). Berlin: Springer, 2013: 75-92

[12] Gentry C. Fully homomorphic encryption using ideal lattices [C] //Proc of the 41st ACM Symp on the Theory of Computing (STOC'09). New York: ACM, 2009: 169-178

[13] Smart P N, Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes [C] //Proc of the 13th Int Conf on Theory and Practice of Public Key Cryptography (PKC'10). Berlin: Springer, 2010: 420-443

[14] Biasse F J, Fieker C. Subexponential class group and unit group computation in large degree number fields [J]. LMS Journal of Computation & Mathematics, 2014, 17(A): 385-403

[15] Biasse J F, Song Fang. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields [C] //Proc of the 27th ACM-SIAM Symp on Discrete Algorithms (SODA'16). New York: ACM, 2016: 893-902

[16] Cramer R, Ducas L, Peikert C, et al. Recovering short generators of principal ideals in cyclotomic rings [C] //Proc of the 35th Annual Int Conf on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'16). Berlin: Springer, 2016: 559-585

[17] Canetti R, Raghuraman S, Richelson S, et al. Chosen-ciphertext secure fully homomorphic encryption [C] //Proc of the 20th Int Conf on Theory and Practice of Public Key Cryptography (PKC'17). Berlin: Springer, 2017: 213-240

[18] Bleichenbacher D. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1 [C] //Proc of the 18th Int Cryptology Conf (CRYPTO'98). Berlin: Springer, 1998: 1-12

[19] Li Zengpeng, Galbraith D S, Ma Chunguang. Preventing adaptive key recovery attacks on the GSW levelled homomorphic encryption scheme [C] //Proc of the 10th Provable Security (ProvSec'16). Berlin: Springer, 2016: 373-383

[20] Clear M, McGoldrick C. Multi-identity and multi-key leveled FHE from learning with errors [C] //Proc of the 35th Int Cryptology Conf (CRYPTO'15). Berlin: Springer, 2015: 630-656

[21] Mukherjee P, Wichs D. Two round multiparty computation via multi-key FHE [C] //Proc of the 35th Annual Int Conf on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'16). Berlin: Springer, 2016: 735-763

[22] Peikert C. Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract [C] //Proc of the 41st ACM Symp on the Theory of Computing (STOC'09). New York: ACM, 2009: 333-342

[23] Agrawal S, Boneh D, Boyen X. Efficient lattice (H)IBE in the standard model [C] //Proc of the 29th Annual Int Conf on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10). Berlin: Springer, 2010: 553-572

[24] Lyubashevsky V. Lattice signatures without trapdoors [C] //Proc of the 31st Annual Int Conf on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'12). Berlin: Springer, 2012: 738-755

[25] Peikert C, Waters B. Lossy trapdoor functions and their applications [J]. SIAM Journal on Computing, 2011, 40(6): 1803-1844

[26] Li Zengpeng, Galbraith D S, Ma Chunguang. Preventing adaptive key recovery attacks on the gentry-sahai-waters leveled homomorphic encryption scheme, 2016/1146[R/OL]. New York: IACR Cryptology ePrint Archive, 2016 [2017-06-01]. https://eprint.iacr.org/2016/1146.pdf

[27] Alperin-Sheriff J, Peikert C. Faster bootstrapping with polynomial error [C] //Proc of the 34th Int Cryptology Conf (CRYPTO'14). Berlin: Springer, 2014: 297-314

[28] Ajtai M. Generating hard instances of lattice problems (extended abstract)[C] //Proc of the 8th ACM Symp on the Theory of Computing (STOC'96). New York: ACM, 1996: 99-108

[29] Micciancio D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions [C] //Proc of the 43rd IEEE Annual Symp on Foundations of Computer Science (FOCS'02). Los Alamitos, CA: IEEE Computer Society, 2002: 356-365

[30] Agrawal S, Gentry C, Halevi S, et al. Discrete gaussian leftover hash lemma over infinite domains [C] //Proc of the 19th Int Conf on the Theory and Application of Cryptology and Information Security (ASIACRYPT'13). Berlin: Springer, 2013: 97-116

Li Zengpeng, born in 1989. Assistant professor in the College of Computer Science and Technology of Qingdao University. Member of ACM, CCF, CACR, IEEE and IEICE. His main research interests include data security, data privacy and cryptography. In particular, his research focus are lattice-based cryptography, fully homomorphic encryption and cryptography protocol.

Ma Chunguang, born in 1974. Professor of the College of Computer Science and Technology of Harbin Engineering University. His main research interests include cryptography and information security.

Zhao Minghao, born in 1992. PhD candidate in Tsinghua University. Student member of ACM, CCF and CACR. His main research interests include cloud computing, storage system, mobile computing, privacy preserving techniques and applied cryptography.