整数上的全同态加密方案的改进*

林如磊，王 箭，杜 贺
（南京航空航天大学 计算机科学与技术学院，南京 210016）

摘 要：目前的全同态加密方案的效率还很低，与实际的应用还有很大的距离，提高全同态加密方案的效率和安全性是全同态加密技术研究的重点与难点。为了提高效率，在 Dijk 等人的全同态加密方案的基础上，将模 2 运算改为模 4 运算，并使用 Gentry 的全同态思想，提出了一种更快速的全同态加密方案，改进之后的方案一次可以加密 2 bit 的数据，且公钥尺寸降低到 $\widetilde{O}(\lambda^7)$，从而比 Dijk 等人的方案具有更高的效率和更小的公钥尺寸。新方案的安全性基于近似最大公因子问题和稀疏子集和问题。

关键词：全同态加密；近似最大公因子问题；稀疏子集和问题；公钥尺寸
中图分类号：TP309 **文献标志码**：A **文章编号**：1001-3695(2013)05-1515-05
doi:10.3969/j.issn.1001-3695.2013.05.060

Improved fully homomorphic encryption over integers

LIN Ru-lei, WANG Jian, DU He
(*College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing* 210016, *China*)

Abstract: The efficiency of the present fully homomorphic encryption scheme is extra low and far from practical application. How to improve the efficiency and security of fully homomorphic encryption become the focus and pitfall of academic research. In order to improve the efficiency, this paper put forward an improved scheme of the base of Dijk's scheme using Gentry's fully homomorhpic technology. The improved scheme could encrypt 2 bit plaintext each time and reduce the public key size to $\widetilde{O}(\lambda^7)$, accordingly the improved scheme was more efficient than Dijk's scheme and had smaller public key size. The security of the proposed scheme was based on both the approximate GCD problem and the sparse-subset sum problem.

Key words: fully homomorphic encryption; approximate GCD; sparse subset sum problem; public key size

0 引言

1978 年，在 RSA[1] 密码体制刚刚提出不久，Rivest 等人[2] 提出了全同态加密的概念（又叫隐私同态，privacy homomorphisms），并认为全同态加密技术是可以实现的，同时提出了几种候选的同态加密方案。他们希望在不对密文解密的条件下，对密文进行任何运算，得到的结果解密后与对明文进行相应运算的结果相同。全同态加密思想提出之后，国内外的研究人员以及学者对全同态加密进行了大量的研究，然而他们提出的方案都只能对密文进行有限次同态运算，没有达到真正的全同态。

第一个全同态加密（fully homomorphism encryption）方案是 IBM 研究员 Gentry[3] 在 2009 年利用理想格（ideal lattice）构造的，并在其博士论文[4] 中对全同态加密方案进行了更加深入的论述。这两篇文章给全同态加密技术的研究与发展带来了强大的动力。国内外的学者提出了许多改进的全同态方案。Dijk 等人[5,6] 提出了整数上的基于模运算的全同态方案，但是方案的执行效率比较低，加密 1 bit 的明文对应 λ^5 bit 的密文（λ 是安全参数），且公钥尺寸太大。Smart 等人[7] 使用 Gentry 的全同态加密思想，提出了具有相对较小的密钥和密文尺寸的方案，提高了效率。Stehle 等人[8] 对 Gentry 的方案进行了优化，以允许可忽略概率解密错误这一弱化条件，提出了较快速的全同态加密方案，降低了比特计算复杂度。目前，国内关于全同态加密方案的论文还比较少，汤殿华等人[9] 提出一种较快速的整数上的全同态加密方案，相对文献[5]中方案具有更短的公钥尺寸、更小的解密复杂度，并在文献[10]中对 Gentry 的全同态加密技术进行了总结。虽然现在已经有很多改进的全同态加密方案被提出来，但是这些方案的效率都很低。Dijk 等人[5] 提出的整数上的加密方案，一次仅能加密 1 bit 明文，且对密文进行处理时，需要进行重加密，这将导致原本的 1 bit 密文变为一个大整数。显然，改进全同态加密方案的执行效率是未来全同态加密研究的重点与难点。

本文使用 Gentry 的全同态加密思想，对 Dijk 等人提出的整数上的全同态加密方案进行了改进。将 Dijk 等人的方案中的模 2 运算变为模 4 运算，并限制了部分参数的选取；在给出新方案的噪声增长分析之后，使用 Gentry 的全同态加密技术得到一个全同态加密方案。改进之后的方案一次可以加密 2 bit 明文，并使用文献[11]中的技术将公钥尺寸缩小到 $\widetilde{O}(\lambda^7)$，显然改进之后的方案具有更高的效率和更小的公钥尺寸。方案的安全性基于近似最大公约数问题（approximate greatest common divisor）和稀疏子集和问题。

收稿日期：2012-09-04；**修回日期**：2012-10-28 **基金项目**：国家“863”计划高技术研究发展项目(2009AA044601)；江苏省普通高校研究生科研创新计划资助项目(CXZZ12_0161)；中央高校基本科研业务费专项资金资助项目

作者简介：林如磊(1986-)，男，安徽合肥人，硕士，主要研究方向为全同态加密（linrulei@139.com）；王箭，男，教授，博士，主要研究方向为密钥管理、密码协议、隐私保护、恶意追踪等；杜贺，男，博士，主要研究方向为密钥管理、数字签名等.

1 基本符号与定义

1.1 符号说明

对于实数 z 用$\lceil z\rceil \in [z, z+1)$、$\lfloor z\rfloor \in (z-1, z]$和$[z] \in (z-\frac{1}{2}, z+\frac{1}{2}]$表示向上取整、近似取整以及向下取整。对于实数 z 和整数 p,用 $q_p(z)$表示 z 除以 p 的商(近似取整),$r_p(z)$或者$[z]_p$ 表示 $z \bmod p$。本文中模 p 运算的取值范围是 $(-\frac{p}{2}, \frac{p}{2}]$,从而有 $q_p(z) = [\frac{z}{p}]$,$r_p(z) = z - q_p(z)p$。

本文中用希腊字母表示方案中参数的取值范围,下面是方案中使用到的希腊字母的含义:

λ:方案的安全参数;

ρ:噪声的长度,为了阻止暴力攻击(brute-force attacks),噪声长度应该取 $\rho = \omega(\log \lambda)$;

η:私钥的二进制长度,为了让压缩的解密电路属于允许电路中,私钥长度应该满足 $\eta \geqslant \rho\Theta(\lambda \log^2 \lambda)$;

γ:公钥的二进制长度,为了防止基于格的对近似最大公钥数问题的破解,公钥长度应该满足 $\gamma = \omega(\eta^2 \log \lambda)$;

τ:需要的公钥个数,应该满足 $\tau \geqslant \gamma + \omega(\log \lambda)$,本文使用文献[11]中的方法,可以得到一个公钥尺寸较小的方案,实际的公钥个数为 $2\sqrt{\tau}$。

1.2 相关定义

定义 1 一个加密方案 E 如果满足:对密文做任意操作之后再解密,结果与对明文进行相应操作结果相同,则称该加密方案是全同态加密方案。也可以描述为,若 Decrypt_E 是方案 E 的解密算法,sk 是私钥,pk 是公钥,$f(x_1, x_2, \cdots, x_t)$是一个 t 元函数,则 $\text{Decrypt}_E(f(c_1, c_2, \cdots, c_t), sk) = f(m_1, m_2, \cdots, m_t)$。一般而言,公钥体制的全同态加密方案由如下四个算法组成:

KeyGen(λ):根据安全参数 λ 产生公钥 pk 和私钥 sk。

Encrypt(pk, m):用公钥 pk 将明文 m 加密为密文 c。

Decrypt(sk, c):用私钥 sk 解密密文 c,得到明文 m。

Evaluate($pk, f, c_1, c_2, \cdots, c_t$):输入公钥 pk,具有 t 个输入的函数 f,以及 t 个密文 $c_1, c_2, \cdots, c_t$,其中 $c_i = \text{Encrypt}(pk, m_i)$ $(i = 1, 2, \cdots, t)$。输出 $c^* = \text{Evaluate}(pk, f, c_1, c_2, \cdots, c_t)$,且满足 $\text{Decrypt}(sk, c^*) = f(m_1, m_2, \cdots, m_t)$。Evaluate 算法的含义是:对密文进行任意操作再解密,结果与对明文进行相应的操作结果相同。此处的函数 f 也可以换成函数对应的电路 C 作为输入。

定义 2 Permitted function(允许函数)。方案 $E = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt}, \text{Evaluate})$ 是一个同态加密方案,f 是一个 t 元函数,若对 KeyGen(λ)产生的任意公钥对(pk, sk)、明文序列 $m_1, m_2, \cdots, m_t$ 和密文序列 $c_1, c_2, \cdots, c_t$($c_i = \text{Encrypt}(pk, m_i)$),都有 $\text{Decrypt}(sk, \text{Evaluate}(pk, f, c_1, c_2, \cdots, c_t)) = f(m_1, m_2, \cdots, m_t)$ 成立,则称函数 f 是 E 的允许函数。函数 f 对应的电路 C 称为方案 E 的允许电路(permitted circuit),方案 E 的所有允许电路的集合记为 C_E。若方案 E 的允许电路的集合 C_E 可以是任意的电路,则称 E 是全同态加密方案(fully homomorphic encryption scheme);否则,称 E 是部分同态方案(somewhat homomorphic encryption scheme)。

定义 3 Augmented decryption circuits(扩展的解密电路)。一个同态加密方案 E 的扩展解密电路是由一个加法或乘法电路将两个解密电路连接起来得到的,即

$$\text{Dec}_{\text{Add}}(sk, c_1, c_2) = \text{Decrypt}(sk, c_1) + \text{Decrypt}(sk, c_2)$$

$$\text{Dec}_{\text{Mult}}(sk, c_1, c_2) = \text{Decrypt}(sk, c_1) \times \text{Decrypt}(sk, c_2)$$

将解密电路以及扩展的解密电路记为 D_E。

定义 4 BootStrappable encryption scheme。方案 $E = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt}, \text{Evaluate})$ 是一个同态加密方案,若 E 的解密电路以及扩展的解密电路包含于 E 的允许电路的集合 C_E 中,即 $D_E \subset C_E$,则称 E 是可启动的加密方案(bootStrappable encryption)。

定义 5 Fresh ciphertexts(新鲜密文)。由加密算法得到的密文拥有最小的噪声,称为新鲜密文。

1.3 本文全同态方案的构造思路

本文按照 Gentry 的全同态加密思想提出一个一次加密 2 bit 的全同态加密方案:首先提出了一个部分的同态加密方案,只能对密文进行有限次的同态加减运算,当密文运算次数超过限定值时,密文中的噪声过大,从而无法保证解密的正确性;然后,把部分同态方案修改成 bootStrappable 部分同态加密方案;最终,得到一个全同态的加密方案。

假设 $E = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt}, \text{Evaluate})$ 是一个部分同态加密方案,由 Evaluate 的定义可知,对于公钥 pk、电路 C 以及 t 个密文 $c_1, c_2, \cdots, c_t$,其中 $c_i = \text{encrypt}(pk, m_i)$ $(i = 1, 2, \cdots, t)$,输出结果为 $c^* = \text{Evaluate}(pk, C, c_1, c_2, \cdots, c_t)$,且满足等式 $\text{Decrypt}(sk, c^*) = C(m_1, m_2, \cdots, m_t)$。如果把电路 C 换成解密电路 Dec,则 $c^* = \text{Evaluate}(pk, \text{Dec}, c', sk')$(其中,$c'$和 sk'是用 pk 对密文 c 和私钥 sk 进行加密的结果,即 $c' = \text{Encrypt}(pk, c)$、$sk' = \text{Encrypt}(pk, sk)$),而 $\text{Decrypt}(sk, c^*) = \text{Dec}(c, sk) = m$,也就是说 c^* 是 m 在 pk 下加密的密文,如图 1 所示(图中方框代表使用 pk 进行一次加密的结果)。这是一个很有用的结论,即若部分同态加密方案 E 的解密电路是允许电路,则可以使用 Evaluate 算法以及对私钥的加密得到一个新的密文,本文把这一过程称为重加密。

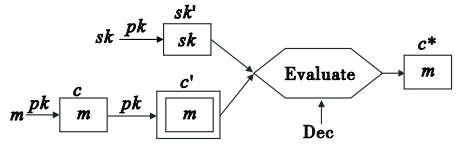

图1 Evaluate中的电路是解密电路的情况

如果把 Evaluate 中的电路换成扩展的解密电路 Dec_{Add},则有 $c^* = \text{Evaluate}(pk, \text{Dec}_{\text{Add}}, c'_1, c'_2, sk')$(其中,$c'_1$、$c'_2$ 以及 sk'是 pk 对密文 c_1、c_2 以及私钥 sk 进行加密的结果),而 $\text{Decrypt}(sk, c^*) = \text{Dec}_{\text{Add}}(c_1, c_2, sk) = m_1 + m_2$;也就是说其结果 c^* 恰好是 $m_1 + m_2$ 的密文(因为加密过程中引入随机数,所以同一明文可以有许多密文),且 c^* 是新鲜密文具有较小的噪声,如图 2 所示。同理,若 Evaluate 中的电路是 Dec_{Mult} 的时候,其结果解密可以得到 $m_1 \times m_2$,其过程如图 3 所示。

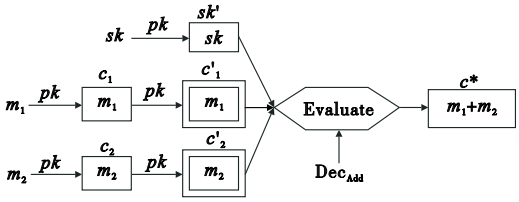

图2 Evaluate中的函数是Dec_{Add}的情况

上面就是 Gentry 构造全同态加密技术的核心思想。但是,上面的结论有一个前提:扩展的解密电路必须是允许电路。

本文给出的部分同态方案的解密电路过于复杂不在允许电路 C_E 中,所以需要对解密电路进行压缩。

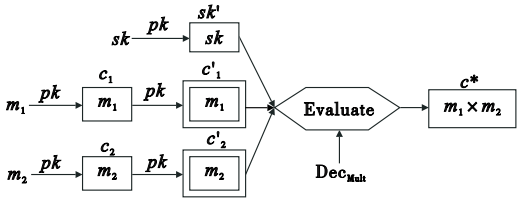

图3　Evaluate中的函数是Dec_{Mult}的情况

2　部分同态加密方案

2.1　对称方案

本节先给出一个对称的加密方案 E,然后把该对称加密方案修改成公钥体制方案。对称加密方案是对 Dijk 等人[5]的整数上的部分同态加密的改进,把方案中的模 2 运算改为模 4 运算,从而使得方案一次可以加密 2 bit 明文,且具有一定的加法和乘法同态性。

1)KeyGen(λ)　根据安全参数 λ 产生 η bit 的奇数 p 作为私钥。

2)Encrypt(sk,m)　对于任意的 2 bit 的明文 $m\in\{00,01,10,11\}$,密文 $c=m+4r+pq$;其中,r 是随机选取的 ρ bit 的整数,q 是随机选择的 γ bit 的正整数。

$$\text{Decrypt}(sk,c):m=(c \bmod p) \bmod 4$$

把 $c \bmod p$ 的值称为噪声,若 $m+4r<p/2$,则噪声 $(c \bmod p)=m+4r$,从而有 $(c \bmod p) \bmod 4=(m+4r) \bmod 4=m$,即方案是正确的。可见,要保证解密的正确性必须确保 $m+4r$ 的值小于 $p/2$,由方案的参数可知,新鲜密文的 $m+4r$ 一定小于 $p/2$。方案的同态性验证:设 $c_1=m_1+4r_1+pq_1$,$c_2=m_2+4r_2+pq_2$,则有

$$\begin{aligned}&c_1+c_2=\\&[(m_1+4r_1)+(m_2+4r_2)]+p(q_1+q_2)=\\&(m_1+m_2)+4(r_1+r_2)+p(q_1+q_2)\end{aligned}\tag{1}$$

$$\begin{aligned}&c_1c_2=\\&(m_1+4r_1)(m_2+4r_2)+\\&p[(m_1+4r_1)q_2+(m_2+4r_2)q_1+pq_1q_2]=\\&m_1m_2+4(m_2r_1+m_1r_2+4r_1r_2)+\\&p(m_1q_2+m_2q_1+4r_1q_2+4r_2q_1+pq_1q_2)\end{aligned}\tag{2}$$

当 $(m_1+m_2)+4(r_1+r_2)<p/2$ 时,$((c_1+c_2) \bmod p) \bmod 4=((m_1+m_2)+4(r_1+r_2)) \bmod 4=m_1+m_2$。

当 $m_1m_2+4(m_2r_1+m_1r_2+4r_1r_2)<p/2$ 时,$(c_1\times c_2 \bmod p) \bmod 4=(m_1m_2+4(m_2r_1+m_1r_2+4r_1r_2)) \bmod 4=m_1m_2$。

可见,上述方案 E 对加法和乘法具有同态性,但是方案存在噪声,随着密文的运算噪声将会变大;当噪声大于 $p/2$ 时,$((c_1+c_2) \bmod p) \bmod 4\neq((m_1+m_2)+4(r_1+r_2)) \bmod 4=m_1+m_2$,$(c_1\times c_2 \bmod p) \bmod 4\neq(m_1m_2+4(m_2r_1+m_1r_2+4r_1r_2)) \bmod 4=m_1m_2$,显然不能正确进行解密。由上述式(1)和(2)易知,加法之后的噪声等于各自噪声之和;乘法之和的噪声等于各自噪声之积。可见乘法运算会使得噪声很快增长。在 2.3 节将会对方案的噪音进行分析,并得到方案的允许电路的范围。

2.2　公钥体制方案

为了把上述的对称方案转换成公钥体制(记为 E^*),需要添加一组“0”的密文作为公钥,并使用文献[11]中的方法减小公钥尺寸,从而得到比 Dijk 公钥尺寸小的公钥方案。

1)KeyGen(λ)　随机选择 η bit 的奇数 p 作为私钥,令 $x_0=pq_0$,当 x_0 是奇数且 $r_p(x_0)$ 能被 4 整除时继续,否则重新计算 x_0。按照上述对称方案产生 $2\sqrt{\tau}$ 个对 0 加密的密文:$b\in\{0,1\}$,$1\leqslant i\leqslant\sqrt{\tau}$,$x_{i,b}=pq_{i,b}+4r_{i,b}$。最终得到公钥尺寸为 $2\sqrt{\tau}$,公钥为 $pk=\langle x_0,x_{1,0},x_{1,1},x_{2,0},x_{2,1},\cdots,x_{\sqrt{\tau},0},x_{\sqrt{\tau},1}\rangle$。

2)Encrypt(pk,m)　产生 τ 维向量 $\boldsymbol{b}=\langle b_{i,j}\rangle$ $(1\leqslant i,j\leqslant\sqrt{\tau}$,$b_{i,j}\in\{0,1\})$,对于任意的 2 bit 的明文 $m\in\{00,01,10,11\}$,对应的密文为

$$c=(m+4r+4\sum_{1\leqslant i,j\leqslant\sqrt{\tau}}b_{i,j}x_{i,0}x_{j,1}) \bmod x_0$$

其中:r 是随机选取的 ρ bit 的整数。

$$\text{Decrypt}(sk,c):m=(c \bmod p) \bmod 4$$

在加密算法中,对 x_0 求余的目的是降低密文的大小。方案的正确性验证如下所示:

$$\begin{aligned}&(c \bmod p) \bmod 4=\\&([m+4r+4\sum_{1\leqslant i,j\leqslant\sqrt{\tau}}b_{i,j}x_{i,0}x_{j,1}]_{x_0} \bmod p) \bmod 4=\\&((m+4r+4\sum_{1\leqslant i,j\leqslant\sqrt{\tau}}b_{i,j}x_{i,0}x_{j,1}-kx_0) \bmod p) \bmod 4=\\&(m+4r+4[\sum_{1\leqslant i,j\leqslant\sqrt{\tau}}b_{i,j}x_{i,0}x_{j,1}]_p-[[k]_p[x_0]_p]_p)) \bmod 4=\\&(m+4r+4[\sum_{1\leqslant i,j\leqslant\sqrt{\tau}}b_{i,j}x_{i,0}x_{j,1}]_p) \bmod 4=m\end{aligned}$$

2.3　噪声增长分析

上面的公钥方案 E^* 和对称方案 E 的噪声增长速度类似,下面以对称方案为例分析噪声的增长速度以及允许函数的范围。设 f 是一个 t 元 d 次函数,明文序列 $m_1,m_2,\cdots,m_t$ 对应的密文为 $c_1,c_2,\cdots,c_t$(其中 $c_i=\text{Encrypt}(sk,m_i)$)。对密文进行函数 f 运算有 $f(c_1,c_2,\cdots,c_t)=f(m_1+4r_1+pq_1,\ m_2+4r_2+pq_2,\cdots,\ m_t+4r_t+pq_t)=f(m_1+4r_1,\ m_2+4r_2,\cdots,\ m_t+4r_t)+pq$,由方案的定义可知只有当 $f(m_1+4r_1,\ m_2+4r_2,\cdots,\ m_t+4r_t)<p/2$ 时,$f(c_1,c_2,\cdots,c_t)$ 才能正确解密。一个 t 元 d 次函数 $f(x_1,x_2,\cdots,x_t)$ 的取值范围可以用一个初等对称多项式来衡量,如式(3)所示。

$$\begin{aligned}&f(x_1,x_2,\cdots,x_t)=\\&\sum_{j_1,j_2,\cdots,j_d\in U}x_{j_1}x_{j_2}\cdots x_{j_d}<C_t^dM^d<t^dM^d<p/2\end{aligned}\tag{3}$$

其中:U 是从 $\{1,2,\cdots,t\}$ 中任意选择 d 个构成的集合,且假设 $|x_i|<M$。对不等式(3)两边求对数可得

$$d<(\log p)/(\log(Mt))\tag{4}$$

若取 $x_i=m_i+4r_i$,则 $M\sim2^{\rho+2}$,$p\sim2^{\eta}$,于是有 $d<\eta/(\rho+2+\log t)$。按照2.1 节中的参数设定,上面的部分同态加密方案 E 的 permitted function 的次数最高可以是 $\alpha\lambda\log^2\lambda$。那么解密函数能否满足这个要求,或者,解密电路是不是一个允许电路(permitted circuit)?下面就对解密电路的复杂度进行分析。

$$\begin{aligned}&\text{Decrypt}(c,p)=\\&(c \bmod p) \bmod 4=\\&(c-p[c/p]) \bmod 4=\\&\text{LSB}_2(c)\oplus\text{LSB}_2(\text{LSB}_2(p)\text{LSB}_2([c/p]))\end{aligned}\tag{5}$$

解密函数可以按照式(5)中所示的方式进行计算。其中 LSB_2 表示求一个数的二进制表示的最后两位,$[c/p]$ 表示按照四舍五入的方式进行取整。式(5)中求一个数的最低两位数的运算很简单,求两个两位数的二进制乘法也很简单;导致解密函数复杂的主要原因在于 c/p 运算。下面就分析一下 c/p 的复杂性。对于两个 n 位的二进制数的相乘运算,有如下结论:

a)两个 n 位数相乘相当于加 n 个二进制的数,输出结果可以表示为输入位的 2 次多项式。

b)n 个二进制的数相加,使用 three-for-two trike 技术[12]最多 $\log_{3/2}n$ 次可以变为两个二进制数相加;输出的结果可以表

示为输入位的 $2^{\log_{3/2}n}=2^{\log n/(\log 3-\log 2)}=n^{1/(\log 3-\log 2)}\approx n^{1.71}$ 次多项式。

c)两个 n 位数相加,输出位可以表示为输入位的 n 次多项式。

从两个 n 位的二进制数相乘的次数大约为 $2n^{1.71}n=2n^{2.71}$。根据二进制数的乘法规则,c/p 要计算正确,$1/p$ 至少要取 $\log c$ 位精度。所以,计算 c/p 的多项式次数为 $2(\log p)^{2.71}\approx 2(\lambda^2)^{2.71}$,显然解密电路不属于允许电路。下一节将对解密函数进行压缩,使得解密电路属于允许电路,最终得到一个全同态的加密方案。

3 全同态加密方案

3.1 压缩解密电路

由 2.3 节的分析可知,解密函数太复杂需要进行压缩。压缩的基本思路是:在公钥中添加一部分私钥的信息,并且在加密时使用这部分私钥信息对密文进行预处理(post process);预处理之后的密文的解密速度会更加高效,从而降低了解密电路的复杂性。预处理之后的密文会比原来的密文更长,而且上述过程引入了另外一个安全假设——稀疏子集和问题。具体步骤如下所示:

a)参数。κ、θ、Θ 是三个新增加的参数,且都关于 λ 的函数,分别为 $\kappa=\gamma\eta/\rho'$,$\Theta=\omega(\kappa\log\lambda)$ 以及 $\theta=\lambda$。

b)KeyGen(λ)。按照 2.3 节中的方案获得私钥 $sk^*=p$ 和公钥 pk^*。令 $x_p=[2^\kappa/p]$,随机选取汉明重量为 θ 的 Θ 维向量 $\boldsymbol{s}=\langle s_1,s_2,\cdots,s_\Theta\rangle$,并根据 $\boldsymbol{s}$ 向量获得集合 $S=\{i:s_i=1\}$,显然 S 中有 θ 个非零元素。随机选取整数 $u_i\in Z\cap[0,2^{\kappa+1})(i=1,2,\cdots,\Theta)$ 使得 $\sum_{i\in S}u_i=x_p\pmod{2^{\kappa+1}}$。再令 $\boldsymbol{y}=\langle y_1,y_2,\cdots,y_\Theta\rangle$($y_i=u_i/2^\kappa$ 小数点后保留 κ 位),显然,y_i 是小于 2 的正数,且满足

$$[\sum_{i\in S}y_i]_4=(1/p)-\Delta_p\quad|\Delta_p|<2^{-\kappa}\tag{6}$$

最后得到私钥 $sk=\boldsymbol{s}$、公钥 $pk=(pk^*,\boldsymbol{y})$。式(6)的证明过程如下:

$$\begin{aligned}[\sum_{i\in S}y_i]_4&=[\sum_{i\in S}u_i/2^\kappa]_4=\\&[\sum_{i\in S}u_i/2^\kappa]_4=\\&[(\sum_{i\in S}u_i)/2^\kappa]_4=\\&[(x_p\bmod 2^{\kappa+1})/2^\kappa]_4=\\&[([2^\kappa/p]\bmod 2^{\kappa+1})/2^\kappa]_4=\\&[[2^\kappa/p]/2^\kappa]_4=\\&(1/p)-|\Delta_p|\end{aligned}$$

c)Encrypt(pk,m)。先用 2.3 节中的方案进行加密,得到密文 c^*。然后,对所有的 $i\in\{1,2,\cdots,\Theta\}$ 令 $z_i=[c^*y_i]_4$ 并在小数点后保留 $n=\lceil\log\theta\rceil+3$ 位有效数字,得到的密文是 $\boldsymbol{z}=\langle z_1,z_2,\cdots,z_\Theta\rangle$ 和 c^*。

d)Decrypt(sk,c)。解密过程为

$$\begin{aligned}m&=(c^*\bmod p)\bmod 4=\\&(c^*-p[c^*/p])\bmod 4=\\&\mathrm{LSB}_2(c^*)\oplus\mathrm{LSB}_2(\mathrm{LSB}_2(p)\mathrm{LSB}_2([\sum_{i\in S}s_iz_i]))\end{aligned}$$

3.2 全同态加密方案

由文献[12]可知,对解密电路进行了一次压缩之后,解密电路的复杂性依然很高,不属于允许电路集合中。为此,需要将解密函数按照如下方式进行计算,以降低复杂性。

a)令 $a_i=s_iz_i$,$i\in\{1,2,\cdots,\Theta\}$。显然,当 $s_i=0$ 时,$a_i=0$;$s_i=1$ 时,$a_i=z_i$。a_i 是[0,4)的随机数,a_i 的小数点后面保留 $n=\lceil\log\theta\rceil+3$ 位精度。

b)由步骤 a)中的 Θ 个值 $\{a\}_{i=0}^{\Theta}$,根据等式 $\sum_{j=-1}^{n}w_j=\sum_{i=1}^{\Theta}a_i\pmod 4$ 产生 $n+2$ 个精度小于 n 的随机值 $\{w\}_{j=-1}^{n}$。

c)解密函数为 $m=\mathrm{LSB}_2(c^*)\oplus\mathrm{LSB}_2(\mathrm{LSB}_2(p)\mathrm{LSB}_2([\sum_{i\in S}w_i]))$。

步骤 a)可以使用一层乘法门的子电路完成,但是步骤 b)c)更加复杂。根据 2.3 节,将 Θ 个二进制数相加使用 three-for-two 方法至多 $\lceil\log_{3/2}\Theta\rceil+2$ 次,可以变为两个数相加。因此,步骤 b)中将求解 Θ 个数的和变为两个数的和,需要的多项式次数大约为 $d'\leqslant 2^{\lceil\log_{3/2}\Theta\rceil+2}<8\Theta^{1.71}$。其中 $\Theta=\omega(\kappa\log\lambda)$,显然已经超过了允许多项式的最高次数,需要使用其他的方法进一步降低解密函数的复杂性。

设 a_i 的二进制表示为 $a_{i,1}a_{i,0}.a_{i,-1}a_{i,-2}\cdots a_{i,-n}$,显然 $a_i=\sum_{j=-1}^{n}2^{-j}a_{i,-j}$。分别将 a_i 的二进制写在第 i 行中,如图 4 所示,并分别求出每一列的汉明重量(Hamming weight)用 W_j 表示,即 $W_j(j=-1,0,1,2,\cdots,\Theta)$ 是 $a_{1,-j}a_{2,-j}a_{3,-j}\cdots a_{\Theta,-j}$ 中 1 的个数。由 3.2 节可知,a_i 中顶多有 θ 个不为 0,显然 W_j 不会大于 θ,从而 W_j 的二进制顶多为 $\lceil\log(\theta+1)\rceil$ 位。

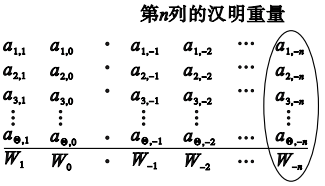

图4 计算a_i和的算法

由文献[12]中的引理 5 可知 W_j 的二进制中的每一位都可以表示为关于 $a_{1,-j}a_{2,-j}a_{3,-j}\cdots a_{\Theta,-j}$ 至多为 θ 次的多项式,且所有的这些多项式可以被一个大小为 $O(\theta\cdot\Theta)$ 的电路计算出来。只要计算出 W_j 就可以通过 $\sum_{i=1}^{n}a_i=\sum_{j=-1}^{n}W_{-j}2^{-j}$ 来计算 a_i 的和。令 $w_j=W_{-j}2^{-j}\pmod 4$($j=-1,0,1,2,\cdots,n$)精度为 $\lceil\log(\theta+1)\rceil<n$,这恰好是步骤 b)中需要的 $n+2$ 个精度小于 n 的随机数。使用 three-for-two 方法,将这 $n+2$ 个数的和求解出来所需的多项式的次数至多为

$$32(n+2)^{1/(\log(3/2))}<32\lceil\log\theta+4\rceil^{1.71}<32\log^2\theta$$

综上,计算步骤 a)所需的多项式次数为 2;步骤 b)中计算 W_j 的多项式次数为 θ,$n+2$ 个数相加需要的多项式次数为 $32\log^2\theta$,所以,解密电路的多项式次数为 $2\cdot\theta\cdot 32\log^2\theta$。其中参数 $\theta=\lambda$,所以按照这种方式计算,解密电路的复杂度至多为 $64\lambda\log^2\lambda$。从而,扩展的解密电路的深度至多为 $128\lambda\log^2\lambda$。由 2.4 节中的不等式(5)可知,扩展的解密电路属于允许电路,从而方案是全同态的。

3.3 方案的安全性分析

方案的安全性基于近似最大公约数问题,下面给出近似最大公约数问题的定义:

定义 6 近似最大公约数问题(approximate-GCD problem)随机选择 n 个大整数 p 的近似倍数 $a_1,a_2,a_3,\cdots,a_n$,根据 $a_1,a_2,a_3,\cdots,a_n$ 求出 p 的过程就称为近似最大公约数问题。

定理 假设攻击者可以用一个优势为 ε 的算法 A 攻破本文中的方案,则存在一个算法 B 以至少 $\varepsilon/2$ 的概率可以由 $x_i=pq_i+4r_i(1\leqslant i\leqslant\tau)$ 求出 p,即算法 B 可以以至少 $\varepsilon/2$ 的概率解决近似最大公约数问题。

此定理的证明过程与 Dijk 等人提出的方案类似,此处就不再赘述。

对方案的任何攻击方案都可以转换为近似最大公约数问题的解决方案,若攻击者可以破解本文中的方案,则攻击者可以由 $x_i = pq_i + 4r_i (1 \leqslant i \leqslant \tau)$ 求出 p,而实际上最大公约数问题到目前为止是不能被解决的,所以,本文方案是安全的。

在压缩的方案中,需要在公钥中添加对私钥的暗示 y,这引入了另外一个安全假设:稀疏子集和难题(sparse subset sum problem,SSSP)。在 server-aided cryptography[13] 加密方案中,给出了稀疏子集和难题的详细介绍。对于本文中的方案,只要让 θ 足够大,就可以避免对 SSSP 的暴力攻击。

4　结束语

本文在 Dijk 等人方案的基础上,将模 2 运算变为模 4 运算,从而得到了一种整数上的仅使用简单代数运算的全同态加密方案;并且本文使用了 Jean-Sebastien 等人的思想[11] 降低了方案的公钥尺寸。方案一次可以加密 2 bit 的明文,因此比 Dijk 等人的方案具有更高的效率和更短的公钥尺寸。方案的安全性基于近似最大公约数问题和稀疏子集和问题。

参考文献:

[1] RIVEST R, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM,1978,21(2):120-126.

[2] RIVEST R, SHAMIR A, DERTOUZOS M. On data banks and pri-vacy homomorphisms [J]. Foundations of Secure Computation, 1978,7(1):169-177.

[3] BONEH D, GENTR Y. A fully homomorphic encryption scheme [D]. Stanford:Stanford University,2009.

[4] GENTR Y. Fully homomorphic encryption using ideal lattices[C]// Proc of the 41st Annual ACM Symposium on Theory of Computing. New York: ACM Press,2009:169-178.

[5] Van DIJK, GENTR Y, HALEV I,*et al*. Fully homomorphic encryption over the integers[C]//Proc of the 29th Annual International Conference on Theory and Applications of Cryptograhic Techniques. Berlin:Springer-Verlag,2010:24-43.

[6] GENTRY C. Computing arbitrary function of encryped data[J]. Communications of the ACM,2010,53(3):97-105.

[7] SMART N P,VERCAUTEREN F. Fully homomorphic encryption with relatively small key and ciphertext sizes[C]//Proc of the 13th International Conference on Practice and Theory in Public Key Cryptography. Berlin:Springer-Verlag,2010:420-443.

[8] STEHLE D, STEINFELD R. Faster fully homomorphic encryption [C]//Proc of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2010: 377-394.

[9] 汤殿华,祝世雄,曹云飞. 一个较快速的整数上的全同态加密方案[J]. 计算机工程与应用,2012,48(28):117-122.

[10] 汤殿华,祝世雄,曹云飞. 整数上的全同态加密方案的重加密技术[J]. 信息安全与通信保密,2012,2012(1):76-79.

[11] JEAN-SEBASTIEN C, MANDAL A, NACACHE D, *et al*. Fully-homomorphic encryption over the integers with shorter public-keys [C]//Proc of the 31st Annual Conference on Advances in CRYPTOLOGY. Berlin:Springer-Verlag,2011:487-504.

[12] KARP R M, RAMACHANDRAN V. A survey of parallel algorithms for shared-memory machines,CSD-88-408[R]. [S. l.]:UC Berkeley,1988.

[13] NGUYEN P Q, STERN J. Adapting density attacks to low weight knapsacks[C]//Proc of Asiacrypt'05. Heidelberg: Springer-Verlag, 2005:41-58.

(上接第 1507 页)

表 1　三个基于证书签名方案的计算量比较

比较项	文献[3]方案	文献[7]方案	本文方案
签名阶段	1M + 2SM	1E + 1M	3Mul
验证阶段	3Pa	3Pa + 1E + 2M	2Mul + 5E
总计算量	3Pa + 1M + 2SM	3Pa + 2E + 3M	5Mul + 5E

由表 1 可知,由于对运算的复杂度和计算量远远高于模幂运算和其他运算,本文方案在计算效率方面明显高于其他含有对运算的基于证书签名方案。同时本方案中签名长度仅为 $\mathrm{bit}(q) + \mathrm{bit}(p)$($\mathrm{bit}(x)$ 表示 x 的二进制位数)。由于计算量小,计算效率高,证书的传递不需要安全信道,CA 和用户之间传递的数据量很少(具体见方案),且签名的长度很短是固定值,因此本文方案特别适用于移动通信等计算能力和带宽受限的应用领域。具体如何将本文方案应用于移动应用环境及其他领域,限于篇幅,这里就不详细分析了。

5　结束语

本文围绕着基于证书签名方案展开研究,提出了一个基于证书的、强安全的、不含对运算的签名方案,证明了方案在随机预言机模型下可以抵抗适应性选择消息和身份攻击,抵抗公钥替换攻击和 CA 攻击,并进行了效率分析。目前对基于证书数字签名方案的研究还很少,特别是具有特殊性质的基于证书签名及标准模型下可证安全的基于证书签名等。对这些内容的研究具有重要意义,也是下一步研究的方向。

参考文献:

[1] GENTRY C. Certificate-based encryption and the certificate revocation problem[C]//Lecture Notes in Computer Science, vol2656. Berlin: Springer-Verlag,2003:272-293.

[2] KANG B G, PARK J H, HAHN S G. A certificate-based signature scheme[C]//Lecture Notes in Computer Science, vol2964. Berlin: Springer-Verlag,2004:99-111.

[3] LI Ji-guo, HUANG Xin-yi, MU Yi, *et al*. Certificate-based signature: security model and efficient construction[C]//Lecture Notes in Computer Science, vol4582. Berlin: Springer-Verlag, 2007: 110-125.

[4] LIU J K, BAEK J, SUSILO W, *et al*. Certificate-based signature scheme without pairings and random oracles[C]//Lecture Notes in Computer Science, vol5222. Berlin: Springer-Verlag, 2008: 285-297.

[5] WU W, MU Y, SUSILO W. Certificate-based signatures: new definitions and a generic construction from certificateless signatures[C]// Lecture Notes in Computer Science,vol5379. Berlin: Springer-Verlag, 2009: 99-114.

[6] 李志敏,徐馨,李存华. 高效的基于证书数字签名设计方案[J]. 计算机应用研究,2012,29(4):1430-1433,1444.

[7] 王雯娟,黄振杰,郝艳华. 一个高效的基于证书数字签名方案[J]. 计算机工程与应用,2011,47(6):89-92.

[8] POINTCHEVAL D, STERN J. Security proofs for signatures schemes [C]//Advances in Cryptology-Eurocrypt. Berlin: Springer-Verlag, 1996:387-398.

[9] KAWAHARA Y, TAKAGI T, OKAMOTO E. Efficient imple mentation of tate pairing on a mobile phone using java[C]//Lecture Notes in Computer Science, vol4456. Berlin: Springer-Verlag, 2007: 396-405.