整数上全同态加密方案的重加密技术*

汤殿华，祝世雄，曹云飞
（保密通信重点实验室，四川 成都 610041）

[摘 要] 在 Gentry 的第一个全同态框架中，通过重加密技术来更新密文，是非常关键的一步，重加密技术能控制噪声不超过门限值，以免发生解密错误。文中将根据一个较快速的整数上的全同态加密方案，去除其“可忽略解密错误”技术，然后给出了基于进位加法的重加密算法来进行同态解密，并详细描述其具体步骤。

[关键词] 全同态加密；重加密；同态解密
[中图分类号] TN918.4 [文献标识码] A [文章编号] 1009-8054(2012)01-076-04

Recryption Technology of A Fully Homomorphic Encryption over Integers

TANG Dian-hua，ZHU Shi-xiong，CAO Yun-fei
(Laboratory of Science and Technology on Communication Security，Chengdu Sichuan 610041，China)

[Abstract] In the Gentry's first frame for constructing a fully homomorphic encryption scheme，the refreshing of ciphertexts with re-cryption technology is a very key step. The recryption technology can control the noise from exceeding the threshold，and thus avoid decryption errors. In accordance with the faster fully homomorphic encryption scheme over integers and the technique for removing “negligible decryption errors”，this paper gives a re-cryption algorithm based on addition with carry for decrypting homomorphically，including its detailed implementation steps.

[Keywords] fully homomorphic encryption；re-cryption；decrypting homomorphically

0 引言

1978 年由 Rivest、Adleman 和 Dertouzos[1] 提出“隐私同态”概念之后，全同态加密一直是密码学者梦寐解决的问题。全同态加密能够在没有解密密钥的条件下，对加密数据进行任意复杂的操作，以实现相应的明文操作。直到2009年，Gentry提出了第一个全同态加密方案[2]，首次解决了这个困扰密码学界 30 年的问题。

Gentry 发现一个具有自举性的同态加密方案可以转化为一个全同态加密方案[3]。所谓的自举性，是指该同态方案能够同态处理自己的解密电路以及扩展解密电路。由于加密中噪声的存在，同态加和同态乘会使噪声增加，因此方案的同态处理能力是有限的。如果方案具有自举性，就可以通过同态解密来降低密文的噪声，扩大其同态处理能力，以致能够处理任何复杂的布尔电路。由于“同态解密”可以把在 (pk, sk) 下加密的一个密文转化为在 (pk', sk') 下加密的一个新密文，且保持消息不变，所以叫做重加密技术。为了取得自举性，Gentry 在其博士论文[3]中引入了“压缩解密电路”技术，降低解密算法的计算复杂度。该技术产生了辅助解密计算的预处理密文信息，在处理这些信息时，Gentry 使用了一种非常精巧的“Three-for-two”技术，但是他并没有给出具体的步骤。

2010 年，Smart 和 Vercauteren 提出了具有相对小的密钥和密文尺寸的全同态方案[4]，并第一次尝试对全同态加密方案的实现，但由于取得自举性对维数要求过高，约为 2^{27} 维，因此只测试了 Somewhat 同态方案，并依照 Gentry 的重加密思想，在文中给出了其方案的重加密过程以及噪声分析。在 2011 年欧密会上，Gentry 和 Halevi 给出了一个全同态加密方案的实施[5]。他们对 Smart 和 Vercauteren 的方案进行了许多优化，其中对于重加密技术，用“进位加法”取代了“Three-for-two”技术，使得解密计算的复杂度更低，更容易理解和实现。文中将基于进位加法的重加密技术应用于一个整数上的全同态加密方案[6]。结合文中的重加密算法与参考文献 [6] 所提的方案，可更加清楚地理解 Gentry 的第一个全同态加密框架。

1 准备知识

引理 1[7] 设 $\vec{x}=(x_1, x_2, \cdots, x_t)$ 是一个 t 维 {0，1} 向量，令该向量的汉明重量表示为 $W=W(\vec{x})$，并记 W 的二进制表达为 $W=(W_n, W_{n-1}, \cdots, W_1, W_0)_2$（即 $W=\sum_{i=0}^{n} 2^i \times W_i$），则 i 位比特 W_i 可以表示为关于 $x_1, x_2, \cdots, x_t$ 的一个次

收稿日期：2011-10-27
作者简介：汤殿华，1986 年生，男，硕士，研究方向：全同态加密方案；祝世雄，1965 年生，男，研究员，研究方向：密码学；曹云飞，1971 年生，男，高级工程师，研究方向：密码学。
* 基金项目：保密通信重点实验室基金资助项目（编号：9140C1103031002）

数为 2^i 的齐次对称布尔多项式，如下：

$$W(\vec{x})=e_{2^i}(\vec{x})\bmod 2=(\sum_{|S|=2^i}\prod_{j\in S}x_j)\bmod 2$$

其中 S 是变量 $\vec{x}$ 指标集的子集，即 $S\subseteq\{1,2,\cdots,t\}$。

引理 2 对于只有一个整数位的有理小数 $e=(e_0e_{-1}e_{-2}e_{-3}\cdots)_2$，即 $e=e_0+2^{-1}e_{-1}+2^{-2}e_{-2}\cdots$，则 $\lfloor e\rceil\bmod 2=(e_0+e_{-1})\bmod 2$，其中符号 $\lfloor e\rceil$ 表示取最近整数，即 $\lfloor e\rceil\in(e-1/2,\ e+1/2]$。

证明 根据 e_0，e_{-1}，e_{-2} 的取值与 $\lfloor e\rceil\bmod 2$ 的关系，得到如表 1 所示的真值表。

表 1 最近整数的真值表

e_{-1}	e_{-2}	$\lfloor e\rceil\bmod 2$
0	0	e_0
1	0	$(e_0+1)\bmod 2$
0	1	e_0
1	1	$(e_0+1)\bmod 2$

根据表 1，很容易得到 $\lfloor e\rceil\bmod 2=(e_0+e_1)\bmod 2$。

2 同态加密方案

2.1 Somewhat 同态加密方案

在 Gentry 框架中，Somewhat 同态加密方案是全同态加密方案的基础，方案 SHE=(KeyGen′, Enc′, Dec′, Evaluate′) 描述如下：

参数选取：$\rho=\lambda$，$\rho'=2\lambda$，$\eta=\tilde{O}(\lambda^2)$，$\theta=\tilde{O}(\lambda^4)$，$\gamma=\tilde{O}(\lambda^5)$，其中 λ 为安全参数。

KeyGen′(λ)：随机选择 η 比特的奇素数 $p\leftarrow(2\mathbb{Z}+1)\cap[2^{\eta-1},2^{\eta})$ 作为私钥 $sk=p$，公钥含有两个整数，首先选取 θ 比特的奇素数 q，令 $N=pq$，然后选取两个随机整数 $l\in[0,2^{\lambda}/p]$，$h\in[-2^{\rho},2^{\rho}]$，计算 $x=pl+2h$，令公钥 $pk=(N,x)$。

Enc′(pk, m)：消息为 $m\in(0,1)$，选择两个随机整数 $r_1\in[-2^{\rho'},2^{\rho'}]$ 和 $r_2\in[-2^{\rho},2^{\rho}]$，计算密文 $c=m+2r_1+r_2x \bmod N$。

Dec′(sk, c)：输出 $m'=(c \bmod p)\bmod 2$。

Evaluate′(pk, C, c_1, c_2, …, c_t)：给定一个具有 t 输入的布尔电路 C 和 t 个密文 c_i，将电路中的模 2 加法门和乘法门替换成整数上模 N 的加法门和乘法门。将 t 个密文输入到扩展的电路中执行其所有的操作，输出电路的结果。

同时定义评估算法中加法门与乘法门的运算如下：

$Add(c_0,\ c_1)$：$c_+=(c_0+c_1)\bmod N$

$Mult(c_0,\ c_1)$：$c_\times=(c_0\times c_1)\bmod N$

2.2 全同态加密方案

利用 Gentry 的压缩方法，可以把 Somewhat 同态加密方案转化为全同态加密方案。为了描述方便，本节的全同态加密方案将去除原方案[6]中的“可忽略解密错误”技术，因此对方案做了小的修改，具体方案 FHE=(KeyGen, Enc, Dec, Evaluate) 描述如下：

参数选取：$\kappa=\gamma+2$，$r_{set}=\omega(\kappa\log\lambda)$，$r_{sub}=\lambda$

KeyGen(λ)：调用 KeyGen′(λ) 产生私钥 $sk=p$，公钥 $pk=(N,x)$，令 $K=\lfloor 2^{\kappa}/p\rceil$，随机选取一个 r_{set} 维 {0, 1} 向量 $\vec{s}$，并要求其汉明重量为 r_{sub}，且最后一个元素 $S_{r_{set}}=1$，设其指标集为 $S=\{i\,|\,s_i=1\}$，在 $\mathbb{Z}\cap(0,\ 2^{\kappa+1})$ 上均匀随机选取 u_i，$i=1,2,\cdots,r_{set}-1$，然后计算 $u_{r_{set}}=(K-\sum_{i\in S,\ i<r_{set}}u_i)\bmod 2^{\kappa+1}$，令 $y_i=u_i/2^{\kappa}$，$i=1,2,\cdots,r_{set}$，则得到向量 $\vec{y}=(y_1,y_2,\cdots,y_{r_{set}})$，输出私钥 $SK=\vec{s}$，公钥 $PK=(N,x,\vec{y})$。

Enc(PK, m)：利用 SHE 方案的加密算法生成 $c^*=$Enc′(pk, m)，然后计算 $[c^*\times y_i]_2$，$i=1,2,\cdots,r_{set}$，并对其二进制小数点后保留 $n=\lfloor\log r_{sub}\rceil+3$ 位的精度，记取过精度的数为 z_i，最后输出密文 $\tilde{c}=\{c^*,(z_1,z_2,\cdots,z_{r_{set}})\}$，称 c^* 为主密文，$\vec{z}=(z_1,z_2,\cdots,z_{r_{set}})$ 为扩展密文。

Dec(SK, $\tilde{c}$)：输出 $m'=(c^*-\lfloor\sum_{i=1}^{r_{set}}s_iz_i\rceil)\bmod 2$，也可以写成 $m'=Lsb(c^*)\oplus Lsb(\lfloor\sum_{i=1}^{r_{set}}s_iz_i\rceil)$。

Evaluate′(pk, C, c_1, c_2, …, c_t)：给定带有 t 个输入的布尔电路 C，然后将电路中的模 2 加法门和乘法门替换成整数上模 N 的加法门和乘法门，除了电路门的替换，电路结构保持不变，记为 $g(C)$，将密文 c_1，c_2，…，c_t 输入到电路 $g(C)$，在每个门进行运算时，先从每个输入线所对应的密文 $\tilde{c}$ 中提取主密文 c^*，进行重加密操作来更新密文，然后将更新后的密文 $c^{*'}$ 输入到门中进行运算，将门输出作为主密文 $c^{*''}$，并对主密文 $c^{*''}$ 进行扩展得到其扩展密文 $\vec{z}''$，由此构成密文 $\tilde{c}'=(c^{*''},\vec{z}'')$，如此在电路 $g(C)$ 中进行下去，最后得到 $g(C)$ 的输出。

3 重加密算法

3.1 重加密算法简介

首先对 Gentry 博士论文中的重加密算法做简要的回顾：令 $(pk_1,\ sk_1)$ 和 $(pk_2,\ sk_2)$ 是单比特同态加密方案 ε 的两个公私钥对，消息为 $m\in\{0,1\}$，其加密为 $c=Enc_{\varepsilon}(pk_1,m)$，$D_{\varepsilon}$ 表示解密电路。运行算法 $\text{Recrypt}_{\varepsilon}$，不用解密 c，就可以把 c 转化为 m 在公钥 pk_2 下的加密 c_{new}，即 $Dnc_{\varepsilon}(c_{new},sk_2)=m$。算法执行前需要一些辅助信息：私钥 sk_1 在 sk_2 下的加密 $\overline{sk_{1j}}=Enc_{\varepsilon}(pk_2,\ sk_{1j})$，其中 sk_{1j} 表示 sk_1 的第 j 比特。算法过程如下：

$\text{Recrypt}_{\varepsilon}(pk_2,\ D_{\varepsilon},\ <\overline{sk_{1j}}>,\ c)$

$\overline{sk_{1j}}=Enc_{\varepsilon}(pk_2,\ sk_{1j})$

$\bar{c}_j\leftarrow Enc_{\varepsilon}(pk_1,\ c_j)$，其中 c_j 表示 c 的第 j 比特

$c_{new}\leftarrow Evaluate_{\varepsilon}(pk,\ D_{\varepsilon},\ <<\overline{sk}_j>,\ <\bar{c}_j>>)$

输出 c_{new}。

在该算法中，如果方案满足环形安全 (Circular Security)，那么两对密钥可以相同。同样，该方案重加密算法需要辅助信息：私钥的加密，对私钥 $\vec{s}=(s_1,s_2,\cdots,s_{r_{set}})$ 每一比特加密得到 $\bar{s}_i\leftarrow Enc'(pk,\ s_i)$。注意，此处采用

的是 Somewhat 方案的加密算法。

整个重加密就是一个同态解密过程，分析重加密关键的是：在明文空间中解析解密算法的运算（在该方案中需要将解密算法解析成布尔电路），然后将解析之后的运算过程中的加法和乘法替换成密文空间的加法与乘法。下面将基于二进制的进位加法来解析解密算法 $m'=Lsb(c^*)\oplus Lsb(\lfloor\sum_{i=1}^{r_{set}} s_i z_i\rceil)$，然后给出重加密的详细过程。

3.2 基于进位加法的重加密算法

3.2.1 二进制上解析解密算法

步骤 1 计算 $a_i=s_i z_i$，$i=1, 2, \cdots, r_{set}$，注意 a_i 的精度仍为 n。

步骤 2 计算 $(\sum_{i=1}^{r_{set}} a_i)\bmod 2$。将每个 a_i 二进制小数按行排列，按照进位加法计算形式，从低位到高位的顺序，首先利用初等对称多项式计算最后一列之和的本位比特以及进位比特，然后计算倒数第二列，将低于此列的所有列在该列所产生的进位比特加入这一列构成一个新列，再利用初等对称多项式计算该列本位比特以及进位比特，如此计算剩余的列，去掉位权重为 2^i，$i>0$ 的进位，最后得到一个数，记为 e。以 1 个整数位，4 个小数位为例，过程如图 1 所示。

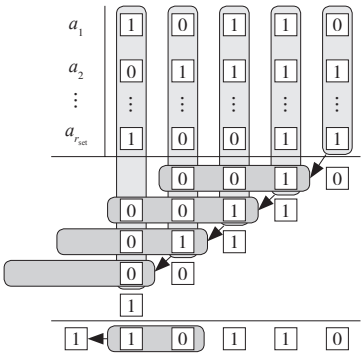

图 1 基于进位加法的解析

步骤 3 输出 $Lsb(c^*)\oplus Lsb(\lfloor e\rceil)$。

注意：很容易验证 $Lsb(\lfloor(\sum_{i=1}^{r_{set}} a_i)\bmod 2\rceil)=Lsb(\lfloor\sum_{i=1}^{r_{set}} a_i\rceil)$，所以步骤 2 中对 $(\sum_{i=1}^{r_{set}} a_i)\bmod 2$ 的计算仍然可以保证解密的正确性。

3.2.2 重加密具体步骤

按照 3.1 节的重加密算法介绍，从下面的步骤 1 至步骤 6 可以看出，步骤 1 和步骤 2 分别计算密文 $\tilde{c}=\{c, \vec{z}\}$ 中主密文 c 的加密与扩展密文 $\vec{z}$ 的加密 $(\bar{c}_j \leftarrow Enc_\varepsilon(pk_1, c_j))$；步骤 3 至步骤 6 是对解密算法布尔电路的评估 $(c_{new} \leftarrow Evaluate_\varepsilon(pk, D_\varepsilon, <<\overline{sk}>, <\bar{c}_j>>))$，其中步骤 3 计算 $a_i=s_i z_i$，$i=1, 2, \cdots, r_{set}$ 的逐比特“加密”，对应于 3.2.1 节中的步骤 1；步骤 4 计算二进制小数 $(\sum_{i=1}^{r_{set}} a_i)\bmod 2$ 的逐比特“加密”；步骤 5 计算 $Lsb(\lfloor(\sum_{i=1}^{r_{set}} a_i)\bmod 2\rceil)$ 的逐比特“加密”，步骤 4 和步骤 5 对应于 3.2.1 节中的步骤 2；步骤 6 计算 $Lsb(c^*)\oplus Lsb(\lfloor(\sum_{i=1}^{r_{set}} a_i)\bmod 2\rceil)$ 的“加密”，对应 3.2.1 节中的步骤 3。具体过程如下：

输入：FHE 密文 $\{c, (z_1, z_2, \cdots, z_{r_{set}})\}$，公钥 $PK=(N, x, y_1, y_2, \cdots, y_{r_{set}})$，私钥的加密 $\{\bar{s}_i\}_{i=1}^{r_{set}}$。

步骤 1 计算 $Lsb(c)$ 的加密 $c_0 \leftarrow \text{Enc}'(pk, Lsb(c))$。

步骤 2 将扩展密文 $\{z_i\}_{i=1}^{r_{set}}$ 按行写成一个 $(n+1)\times r_{set}$ 维 $\{0, 1\}$ 矩阵，每一行为 z_i 的二进制表达 $z_i=(z_{i,0}, z_{i,-1}, z_{i,-2}, \cdots, z_{i,-n})_2=\sum_{j=0}^{n} z_{i,j}\times 2^{-j}$。

$$\begin{bmatrix} z_{1,0} & z_{1,-1} & z_{1,-2} & \cdots & z_{1,-n} \\ z_{2,0} & z_{2,-1} & z_{2,-2} & \cdots & z_{2,-n} \\ z_{3,0} & z_{3,-1} & z_{3,-2} & \cdots & z_{3,-n} \\ \vdots & \vdots & \vdots & & \vdots \\ z_{r_{set},0} & z_{r_{set},-1} & z_{r_{set},-2} & \cdots & z_{r_{set},-n} \end{bmatrix}$$

将该 $\{0, 1\}$ 矩阵的每个元素进行加密：$\bar{z}_{i,j} \leftarrow Enc'(pk, z_{i,j})$，$i=1, 2, \cdots, r_{set}$，$j=0, -1, \cdots, -n$，得到 $\mathbb{Z}_N$ 上的一个 $(n+1)\times r_{set}$ 矩阵，记为 A。

步骤 3 将 $\bar{s}_i$ 乘以矩阵 A 的第 i 行的每一个元素 $(Mult(\bar{s}_i, \bar{z}_{i,j}))$，$j=0, -1, \cdots, -n$，$i=1, \cdots, r_{set}$），得到一个 $\mathbb{Z}_N$ 上的 $(n+1)\times r_{set}$ 矩阵，记为 B，$B=(b_{i,j})$，即 $b_{i,j} \leftarrow Mult(\bar{s}_i, \bar{z}_{i,j})$。

步骤 4 由步骤 3 得到一个 $\mathbb{Z}_N$ 上的矩阵 B：

$$\begin{bmatrix} b_{1,0} & b_{1,-1} & b_{1,-2} & \cdots & b_{1,-n} \\ b_{2,0} & b_{2,-1} & b_{2,-2} & \cdots & b_{2,-n} \\ b_{3,0} & b_{3,-1} & b_{3,-2} & \cdots & b_{3,-n} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{r_{set},0} & b_{r_{set},-1} & b_{r_{set},-2} & \cdots & b_{r_{set},-n} \end{bmatrix}$$

下面按照解析中的步骤 2，进行按列处理矩阵 B。

第 $-n$ 列的计算

由于私钥 $\vec{s}$ 中 1 的个数为 r_{sub}，所以矩阵 B 第 $-n$ 列所对应的 $\{0, 1\}$ 向量的汉明重量最多为 r_{sub}，则其汉明重量的二进制表达最多为 $v_n=\lfloor\log r_{sub}\rfloor+1$ 比特，由该 $\{0, 1\}$ 向量将产生 $\lfloor\log r_{sub}\rfloor$ 个进位比特和一个本位比特。利用模 N 上的初等对称多项式来计算矩阵 B 第 $-n$ 列产生的 $\lfloor\log r_{sub}\rfloor$ 个进位的密文和一个本位的密文。

这里使用 $d_{i\to j}$ 表示第 i 列在第 j 产生的进位，d_i 表示第 i 列所产生的本位。

进位 $d_{-n\to-(n-1)}$，$d_{-n\to-(n-2)}$，$\cdots$，$d_{-n\to-(n-v_n+1)}$ 的计算分别为：$e_{2^1}(b_{1,-n}, b_{2,-n}, \cdots, b_{r_{set},-n})\bmod N$，$e_{2^2}(b_{1,-n}, b_{2,-n}, \cdots, b_{r_{set},-n})\bmod N$，$\cdots$，$e_{2^{v_n-1}}(b_{1,-n}, b_{2,-n}, \cdots, b_{r_{set},-n})\bmod N$。

本位 d_{-n} 的计算为：$e_{2^0}(b_{1,-n}, b_{2,-n}, \cdots, b_{r_{set},-n})\bmod N$。

第 $-(n-1)$ 列的计算

由于第 $-n$ 列对第 $-(n-1)$ 列产生了一个进位 $d_{-n\to-(n-1)} \leftarrow e_{2^1}(b_{1,-n}, b_{2,-n}, \cdots, b_{r_{set},-n})\bmod N$，需要把这个进位加入第 $-(n-1)$ 列中，产生一个 $r_{set}+1$ 维的新列 $(b_{1,-(n-1)}, b_{2,-(n-1)}, \cdots, b_{r_{set},-(n-1)}, d_{-n\to-(n-1)})^T$。该列所对应的 $\{0, 1\}$

向量的汉明重量最多为 $r_{sub}+1$，同理可产生 $v_{n-1}=\lfloor \log r_{sub}+1 \rfloor$ 个进位和一个本位。

进位 $d_{-(n-1)\to-(n-2)}$，$d_{-(n-1)\to-(n-3)}$，…，$d_{-n\to-(n-v_{n-1}-1)}$ 的计算分别为：$e_{2^{v_{n-1}-1}}(b_{1,-(n-1)}, b_{2,-(n-1)}, \cdots, b_{r_{set},-(n-1)}, d_{-n\to-(n-1)}) \bmod N$，$e_{2^{v_{n-1}-2}}(b_{1,-(n-1)}, b_{2,-(n-1)}, \cdots, b_{r_{set},-(n-1)}, d_{-n\to-(n-1)}) \bmod N$，…，$e_{2^1}(b_{1,-(n-1)}, b_{2,-(n-1)}, \cdots, b_{r_{set},-(n-1)}, d_{-n\to-(n-1)}) \bmod N$。

本位 $d_{-(n-1)}$ 的计算为：$e_{2^0}(b_{1,-(n-1)}, b_{2,-(n-1)}, \cdots, b_{r_{set},-(n-1)}, d_{-n\to-(n-1)}) \bmod N$。

按照这样的方法处理剩余列，并去除位权重为 2^i，$i>0$ 的进位，最后将每一列的本位按次序组成一行数据 $(d_0, d_{-1}, \cdots, d_{-n})$。注意：在处理第 j 列时，一定要把所有第 $i=-n, -(n-1), \cdots, j-1$ 列在第 j 列产生的进位 $d_{i\to j}$ 加入到第 j 列中构成一个新列来处理。

步骤 5 计算 $(d_0, d_{-1}, \cdots, d_{-n})$ 所对应二进制小数 e 的 $\lfloor e \rceil \bmod 2$ 的密文，$o=(d_0+d_{-1}) \bmod N$。

步骤 6 计算 $c_{new}=(\bar{c}_0+o) \bmod N$。

输出：更新后的密文 c_{new}。

4 结语

Gentry 的全同态加密的最初框架中，在代入密文同态计算一个电路时，需先在电路中的每个门的每条输入线中加入一个解密电路，目的在于通过同态解密来更新密文，降低噪声，即重加密算法。文中在一个整数上的全同态加密方案基础上，分析其解密算法的比特运算过程，并将其变换成密文空间上的运算，给出了其重加密算法的详细步骤。

参考文献

[1] RIVEST R L，ADLEMAN L，DERTOUZOS M L. On Data Banks and Privacy Homomorphisms[C]//DEMILLO R A，DOBKIN D P，JONES A K，Editors. Foundations of Secure Computation. [s.l.]：Academic Press，1978：169–179.

[2] GENTRY C. Fully Homomorphic Encryption Using Ideal Lattices[C]//STOC '09. [s.l.]：ACM，2009：178.

[3] GENTRY C. A Fully Homomorphic Encryption Scheme[D]. Stanford：Stanford University，2009.

[4] SMART N P，VERCAUTEREN F. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes[C]//Public Key Cryptography–PKC'10. [s.l.]：Springer，2010：420–430.

[5] GENTRY Craig，HALEVI Shai. Implementing Gentry's Fully–homomorphic Encryption Scheme[C]//EUROCRYPT. [s.l.]：Springer，2011：129–148.

[6] 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 [DB/OL]. (2011–08–04)[2011–09–10]. http：//www.cnki.net/kcms/detail/11.2127.TP.20110804.1604.040.html.

[7] van DIJK M，GENTRY C，HALEVI S，et al. Fully Homomorphic Encryption over the Integers[C]//Proc. of Eurocrypt. [s.l.]：Springer，2010：24–43.

（上接第 75 页）

RTO 时间。RTO1=7，RTO2=5，RTO3=10，总的 RTO=22。最后再恢复对 RTO 无要求的业务 E，完成整个恢复分层。

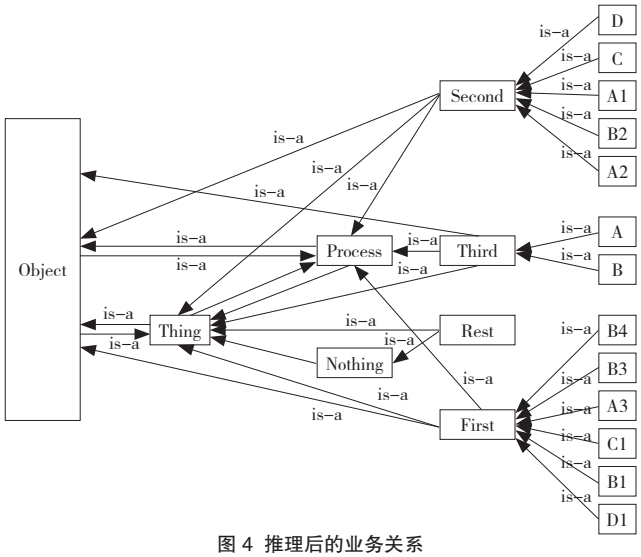

图 4 推理后的业务关系

4 结语

将本体的语言运用到业务排序领域，并通过具体实验验证了本体语言的使用，可使整个系统更为直观地对业务进行分层。实验中，首先对业务对 RTO 有无要求进行划分，无 RTO 要求的一般是对关键业务无太大影响的，所以可优先恢复有 RTO 要求的业务，在这些业务中对其他业务依赖程度越小的越先恢复，这样进行的分层可在同一时间内恢复多个业务，充分节约了时间。当然，这个实验还有两个附加条件：不能存在相互依赖关系以及不存在环。其实，实际情况要复杂的多，今后可在这个角度进行更多的研究。

参考文献

[1] 邓志鸿，唐世渭，张铭，等 . Ontology 研究综述 [J]. 北京大学学报：自然科学版，2002，38(5)：16–18.

[2] 肖文，张保稳，陈晓桦 . 一种基于 RTO 的业务排序算法 [J]. 信息安全与通信保密，2010(1)：104–106.

[3] 王琨 . 基于最优化理论的灾难恢复计划的量化数学模型 [J]. 吉林大学学报：工学版，2007(1)：146–150.

[4] 李勉 . 业务连续性与灾难备份技术的研究 [D]. 上海：交通大学，2009.

[5] 许敦 . 一种基于遗传算法的业务排序方法 [J]. 信息安全与通信保密，2011(1)：107–110.