一种基于智能合约的全同态加密方法

仝秦玮，李洁，王洁，胡心森，胡凯
（北京航空航天大学计算机学院，北京 100191）

摘 要： 大数据的发展使得人们越来越注重数据的价值，传统数据交易中数据不加密或采用对称方式加密。这使得数据保护、用户隐私与使用效率不可兼得。文章设计了一种基于DGHV适应智能合约的同态加密方法，使得密文可以直接进行计算，从而保护交易双方的隐私与安全。该合约可以实现同态布尔逻辑值计算、同态数值比较计算、同态长整数计算。之后给出了合同使用场景。经证明，该方法可以有效的保护数据的安全，并且可以高效的交易布尔型或整型数据。

关键词： 区块链；智能合约；全同态加密；DGHV
中图分类号： TP311 **文献标识码：** A

A full homomorphic encryption method based on smart contract

Tong Qinwei, Li Jie, Wang Jie, Hu Xinsen, Hu Kai
(*School of Computer Science and Engineering, Beihang University, Beijing 100191*)

Abstract: The development of big data makes people pay more attention to the value of data. Data in traditional data transactions is not encrypted or in symmetric encryption way. This makes data protection, user privacy, and efficiency incompatible. This paper designs a homomorphic encryption method based on DGHV using on smart contract, so that the cipher text can be directly calculated, thereby protecting the privacy and security of both parties to the transaction. The contract can implement homomorphic Boolean logic value calculation, homomorphic value comparison calculation, and homomorphic long integer calculation. The contract usage scenario is given after this article. It has been proved that this method can effectively protect the security of data, and can efficiently deal with Boolean or integer data.

Key words: block chain; smart contrast; full homomorphic encryption; DGHV

1 引言

当前大数据和云计算的发展使得人们越来越认识到数据的价值。大数据计算可以从数量庞大的低信息密度的数据中提取有效信息，而区块链为数据的交易提供了便利，使得数据的交易可追溯。但是数据交易的过程中会出现新的问题，如数据泄露与个人权限管理等安全问题，或是数据交易中的隐私问题。

数据泄漏主要分为两种，一是由于区块链本身的特点，全节点中存储着所有的数据，使得任何一个全节点都有可能泄露数据；二是数据传输的过程中存在着泄露的风险。个人权限管理是指如果用户持有一个系统无法知晓的黑名单，如何使得名单中的用户无法使用数据。还有买方不愿公开自己的交易内容等隐私问题。

这些问题都可以通过同态加密的方式得以解决。全同态加密（又叫隐私同态）的概念，最早由Rivest在1978年提出，他设想了一种加密方式，使得密文可以相互直接计算再解密，得到的

结果与明文直接计算相同。之后全球的学者们陆续提出多种方法，但都有缺陷，如不能无限深度的计算、速度缓慢、明文空间有限等。

用户在使用数据的时候往往会使用数据分析的结果，并不直接使用原始数据。所以采用同态加密的方式，仅仅交易数据的运算结果，而不交易原始数据，既保护了数据所有方的数据资产，从根本上杜绝了数据泄露的可能，也使得用户的隐私得以保护。

因此本文在DGHV的基础上，设计了一种有同态加密功能的智能合约。智能合约具有保证加密规则安全、预设规则提供效率、便于监管和便于追踪等特点，这些特点可以更好的实现布尔值的同态运算、长整数的比较运算、长整数的加法与乘法运算。本文之后又给出了典型场景，并分析了本加密方法的安全性。

2 相关工作

2.1 数据上链加密

比特币中最早的数据是不加密的，仅仅是通过哈希算法进行签名，所以全节点可以得到任意账户的全部信息。这种设计有利于系统的去中心化、可追溯的性质，但也存在无法保护账户隐私的问题。而关于链上数据的保护，更多的是采用传统加密算法，即哈希算法、对称加密算法和非对称加密算法等。类似CA系统，用非对称加密技术包裹对称密钥，制成证书，用户通过对称密钥访问数据，这类方式在处理大量的数据，或是文字数据时是有效的，但在处理高频使用的数字时就不如同态加密的方式高效。

而智能合约最初定义为一套以数字形式指定的承诺，包括合约参与方可以在上面执行这些承诺的协议。但是由于计算机的限制，很长时间没有发展。在区块链出现后改善了这一局面，智能合约借助区块链的可信环境与共识基础可以有效的执行其功能。以太坊为智能合约提供了条件，它的EVM虚拟机是一个完全独立的沙盒，使得所有节点完全按照顺序执行合约代码，产生相同的结果以达成共识。虚拟机提供256位的栈，为256位的哈希算法与椭圆曲线算法提供方便[1]。智能合约方便了数据自定义上的使用。

2.2 同态加密

同态加密就是在不访问数据的情况下委托对数据的处理，而现有的同态加密的技术特指对密文直接运算结果等价于对明文计算再加密。最早的同态加密算法由Sander和Tschudin在1998年提出，他们提出了整数域上的加法和乘法的同态加密机制；更重要的是，他们给出了同态加密的详细要求，即整数域上的全同态加密等价于同时满足加法全同态和乘法全同态。第一个真正意义上的全同态加密方法是Gentry在2009年基于理想格提出的[2]。Dijk等人在此基础上，提出了更简单、更易理解，但公钥尺寸大的整数同态加密方法（DGHV）[3]。Coron进一步改进了这个方法，使得加密公钥尺寸变小[4]。Zuowei Wu改造了传统的DGHV方法，有限的减少了加密解密时间，但增加了噪音与复杂度[5]。林如磊等人在传统DGHV的基础上作修改，使得原本一次只能加解密1bit的方法增加到2bit[6]，李子臣等人又在这个基础上改进为加解密k bit[7]，但这些改进都以牺牲原算法的抗噪声能力作为代价。

同态加密技术可用公式表示。$f(m_1,m_2,m_3,...)$表示特定的函数，$E(m),D(c)$表示特定的加密、解密函数，则同态性质表示为：

$$f(m_1,m_2,m_3,...) = D(f(\mathrm{E}(m_1),\mathrm{E}(m_2),\mathrm{E}(m_3),...))$$

若$f(m_1,m_2,m_3,...)$存在有效的加法运算或乘法运算，称该加密算法支持半同态加密；若存在有效的加法运算和乘法运算，称该加密算法支持全同态加密。

DGHV是一种全同态加密方法，本文首先介绍一种对称的加密方法，然后介绍根据上述方法改进的一种公钥方法。

对称方法定义：

（1）$\mathrm{KeyGen}(\lambda)$根据安全参数λ生成一个大奇数p密钥，η(bit)为生成的密钥p的位数。

（2）$\mathrm{Encrypt}(pk,m)$表示加密过程，pk是公钥，m是明文，根据Dijk中的规定，$m \in \{0,1\}$，即m可以表示一个bit位，很多研究者对此处进行研究，将m扩展至2位甚至k位。r和q都为正随机数，长度分别为ρbit和γbit。加密过程可表示为：

$$\text{Encrypt}(pk,m): c = m + 2r + pq$$

（3）$\text{Decrypt}(sk,c)$表示解密过程，sk表示私钥，c表示密文解密过程表示为：

$$\text{Decrypt}(sk,c): m = (c \bmod p) \bmod 2$$

很多学者对此进行研究，可将加密解密公式中的2替换为2^k，可以将明文扩展至k位。

值得注意的是，此时公钥和私钥均为P，所以这是一种对称方法。有研究可以通过使用私钥生成一系列零密文的方式来生成公钥。即：

$$pk_i = \text{Encrypt}(sk,0) = 2r_i + pq_i$$

$$\text{Encrypt}(pk,m_1) = c_1 = m_1 + 2r_1 + pk$$

此时可以通过多组公钥pk，来达到非对称加密的目的。

该方法的同态性证明，假设密钥为p，1位的明文为m_1和m_2，则有：

$$\text{Encrypt}(sk,m_1) = c_1 = m_1 + 2r_1 + pq_1$$

$$\text{Encrypt}(sk,m_2) = c_2 = m_2 + 2r_2 + pq_2$$

将密文直接运算可得到：

$$c_1 + c_2 = (m_1 + m_2) + 2(r_1 + r_2) + p(q_1 + q_2)$$

$$c_1c_2 = (m_1 + 2r_1 + pq_1)(m_2 + 2r_2 + pq_2)$$

$$= m_1m_2 + 2(m_1r_2 + m_2r_1 + 2r_1r_2) + p(m_1q_2 + m_2q_1 + 2r_1q_2 + 2r_2q_1 + pq_1q_2)$$

在满足$2(m_1r_2 + m_2r_1 + 2r_1r_2) < \text{p}$，在不严格要求时可简写为$\text{r} < \frac{\sqrt{p}}{2}$，有：

$$\text{Decrypt}(sk,\text{c1} + \text{c2}) = ((c_1 + c_2) mod\, p) \bmod 2 = (m_1 + m_2) \bmod 2$$

$$\text{Decrypt}(sk,\text{c1c2}) = ((c_1c_2) mod\, p) \bmod 2 = m_1m_2$$

同时满足加法同态与乘法同态，即满足全同态的性质。

3 同态加密链上数据的智能合约

本文设计的智能合约架构如图1所示。最底层为区块链系统和该种系统所支持的合约语言，如以太坊和Solidity；本文还设计了适应该合约的随机数生成器，保障了合约的一致性和密钥的安全性；在上一层是DHGV层，该层是同态加密的核心与基础；在此之上本文设计了长整型数据与布尔数据的同态计算，包括逻辑运算、比较运算与加减乘三则运算方法；最上层本文设计了数据上链和下链的接口。

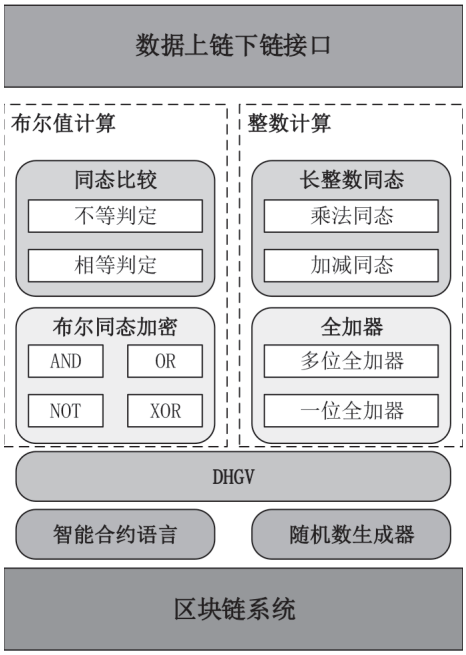

图1 合约架构图

3.1 随机数生成器

在加密的过程中，需要使用随机数，但是智能合约需要区块链上所有节点达成共识，所以如何生成全网一致的“随机数”就变得很重要。随机数的生成需要随机数种子和随机数生成算法。常用的随机数生成算法有很多，例如Solidity通常采用keccak256哈希函数作为随机数的生成器，该函数可以有良好的随机数性质，但是随机数生成的过程容易被攻击。传统的随机数生成过程需要本结点的Nonce值作为随机数种子，恶意节点会大量计算Nonce的值，直到随机事件的结果对自己有利，所以使用由用户指定的随机数种子可以杜绝类似的攻击。

本文选择线性同余发生器法与keccak256函数结合的方法，这种方法适合生成整数随机数，并且计算量小，只要选取适当的参数与随机数种子，即可全网快速的生成相同的随机数，保障了不可预知与快速共识的两个性质，其生成法则公式为：

$$\begin{cases} X_{n+1} = (aX_n + c) \bmod m,\ n \geq 0 \\ R_{n+1} = keccak256(X_{n+1}) \bmod k \end{cases}$$

使用者上传数据时需要确定X_0即随机数种子，之后即可根据如上公式产生随机数。

3.2 布尔值的同态加密

布尔运算只有两个值｛false, true｝与传统DGHV中的明文空间｛0,1｝非常契合。本文规定映射：

$$\mathrm{G(m)}=\begin{cases} false, & if\ m=0 \\ true, & if\ m=1 \end{cases}$$

$$G^{-1}(F)=\begin{cases} 0, & if\ F=false \\ 1, & if\ F=true \end{cases}$$

即可完成由1bit整数到布尔值的相互转换。常用的布尔运算有与、或、非三种，其他运算均可以通过上述三种相互计算得到。DGHV支持加法和乘法同态，其运算可表示为表1和表2。

表1 同态加运算和异或运算对比

$m_1(\mathrm{G}(m_1))$	$m_2(\mathrm{G}(m_2))$	m_1+m_2	$F_1\oplus F_2$
0(false)	0(false)	0	false
0(false)	1(true)	1	true
1(true)	0(false)	1	true
1(true)	1(true)	0	false

表2 同态乘运算和与运算对比

$m_1(\mathrm{G}(m_1))$	$m_2(\mathrm{G}(m_2))$	m_1*m_2	$F_1\wedge F_2$
0(false)	0(false)	0	false
0(false)	1(true)	0	false
1(true)	0(false)	0	false
1(true)	1(true)	1	true

从表中可以轻易得到，一位的同态加法运算即可实现同态的逻辑异或运算，一位的同态乘运算可实现同态逻辑与运算。而或运算存在着恒等式，所以也可以实现同态的或运算。

$$F_1\vee F_2\equiv F_1\oplus F_2\oplus(F_1\wedge F_2)$$

假设存在两个布尔值F_1,F_2，其对应的整数明文为m_1,m_2，再进行加密得到密文：

$$\mathrm{Encrypt}(pk,m_1)=c_1=m_1+2r_1+pq_1$$

$$\mathrm{Encrypt}(pk,m_2)=c_2=m_2+2r_2+pq_2$$

根据如上恒等式计算c_3，c_3表示$F_1\vee F_2$对应整数的密文。

$$c_3=c_1+c_2+c_1c_2$$

$$F_1\vee F_2=\mathrm{G}(Decrypt(sk,c_3))$$

逻辑非运算是单目运算，这有些不同，本文采取特殊处理，首先容易得到，1本身可以当作1的密文，即：

$$\mathrm{Decrypt}(sk,1)=(1\bmod p)\bmod 2=1,\ p\geq 2$$

而现实中密钥$p\geq 2$这个条件显然成立，所以可规定一个常数$c1_{const}=1$用来表示1的一个密文。同理可得出，常数$c0_{const}=0$可以表示0的密文。有了这个常数密文，可规定逻辑非操作为：

$$m_1=G^{-1}(F_1)$$

$$\neg F_1=G(Decrypt(sk,c_1+c1_{const}))$$

在定义了与或非三种逻辑操作以后，其他的逻辑操作均可通过三种操作的组合获得。

3.3 同态比较器

在定义了布尔运算以后，可以在此基础上，定义一位整数比较器，并串联成为整数同态比较器。一位整数的比较运算如表3所示。

表3 一位整数比较

$m_1(\mathrm{G}(m_1))$	$m_2(\mathrm{G}(m_2))$	$m_1>m_2$	$m_1<m_2$	$m_1=m_2$
0(false)	0(false)	false	false	true
0(false)	1(true)	false	true	false
1(true)	0(false)	true	false	false
1(true)	1(true)	false	false	true

由此表可以得出，一位整数的大于、小于、等于的等价表达式：

$$m_1>m_2\equiv G(m_1)\wedge(G(m_1)\oplus G(m_2))$$

$$m_1<m_2\equiv G(m_2)\wedge(G(m_1)\oplus G(m_2))$$

$$m_1=m_2\equiv\neg(G(m_1)\oplus G(m_2))$$

根据等价表达式可以构造同态比较计算与解密方法。假设密钥对为(pk,sk)，两个一位整数为m_1,m_2，$c_1=\mathrm{Encrypt}(pk,m_1),c_2=\mathrm{Encrypt}(pk,m_2)$是对应的密文，则比较同态计算方法为：

$$m_1>m_2\equiv G(\mathrm{Decrypt}(sk,c1(c1+c2)))$$

$$m_1<m_2\equiv G(\mathrm{Decrypt}(sk,c2(c1+c2)))$$

$$m_1=m_2\equiv G(\mathrm{Decrypt}(sk,c1+c2+1))$$

多位正整数的比较器可以采用串联一位正整数比较器的方式构造。可以按照三个步骤进行同态比较运算。

（1）将比较的数字每一位进行同态加密，低位对其，高位补零（$c0_{const}$），从最高位开始比较；

（2）若大于或小于则比较结束，否则进入第

3步；

（3）从左至右比较下一位，返回第2步。

3.4 长整数同态计算

传统DGHV研究1位的整数同态加密，有学者将其改进为2位甚至k位，使其可以一次对多位进行同态计算。该方式在计算性能好，且对噪音不敏感时是一种良好的选择。但在智能合约中，计算资源比较紧张，例如在Solidity中每一步计算都需要花费Gas；另外，多位的DGHV同态加密算法在计算正整数时优势明显，但在计算负数的时候没有优势。1位DGHV则可以通过模拟二进制补码的计算原理进行整数域计算。

长整数计算需要考虑到进位，参考一位全加器，A、B是加数，C_{in}是低位的进位，C_o是向高位输出的进位，S是得数：

$$S = A \oplus B \oplus C_{in}$$
$$C_o = AC_{in} \vee BC_{in} \vee AB$$

将其化简为只有异或运算和与运算的形式，值得注意的是，一般情况下$\oplus$与$\vee$不等价，公式是化简的结果，不是简单的$\oplus$与$\vee$替换：

$$S = A \oplus B \oplus C_{in}$$
$$C_o = AC_{in} \oplus BC_{in} \oplus AB$$

在此基础上本文构造同态一位全加器：

$$S = Decrypt(sk, c_1 + c_2 + c_{in})$$
$$C_o = Decrypt(sk, c_1c_{in} + c_2c_{in} + c_1c_2)$$

将一位全加器串联，即高位的C_{in}接受低位的C_o作为输入，可得到长整数同态加法运算。如图2所示，链接Co0-Cin1，Co1-Cin2，Co2-Cin3，其它链接同名端口，即可通过四个一位同态加密加法器得到一个四位同态加密加法器。

观察C_o的计算过程需要两次乘法运算，在累加的过程中，低位的C_o会成为高位的，而误差主要是在密文的乘运算中累积。因此每串联一个全加器，误差就会放大一次。所以此方法的误差主要受限于串联的个数，也就是加数的位数。

同态减法的实现采用模拟补码的方式，对减数先按位取反，即每一位都加$c1_{const}$，然后将Cin0也置为$c1_{const}$，即可利用同态加密加法器实现减法。

同态乘法的实现同样依赖于大数乘法的原理。以8位整数为例，假定存在任意两个8位二进制数A和B，其中：

$\mathrm{B} = 2^7b_7 + 2^6b_6 + 2^5b_5 + 2^4b_4 + 2^3b_3 + 2^2b_2 + 2^1b_1 + 2^0b_0 = \sum_{i=0}^{7} 2^ib_i$，进而有：

$$\mathrm{A} \times \mathrm{B} = \mathrm{A}\sum_{i=0}^{7} 2^ib_i = \sum_{i=0}^{7} 2^iAb_i$$

所以大数乘法可以先进行大数（A）和每一位数（b_i）的乘法运算，之后再移位相加即可。而由于二进制乘法的特殊性，即$(2^1-1)\times(2^1-1) < 2^1$，也就是说，任意两个一位的二进制数相乘不会产生进位，用公式表示即为：

$$\mathrm{n} \times \mathrm{m} = (\mathrm{n} \times \mathrm{m}) \bmod 2,\ \forall\, \mathrm{n,m} \in \{0,1\}$$
$$Ab_i = b_i\sum_{j=0}^{7} 2^ja_j = \sum_{j=0}^{7} 2^ja_jb_i = \sum_{j=0}^{7} 2^j(a_jb_i)\ mod\ 2$$

所以Ab_i的问题又简化为了一位数相乘和移位的问题。根据这一原理，大整数的同态乘法问题也可以得到解决，其过程为：

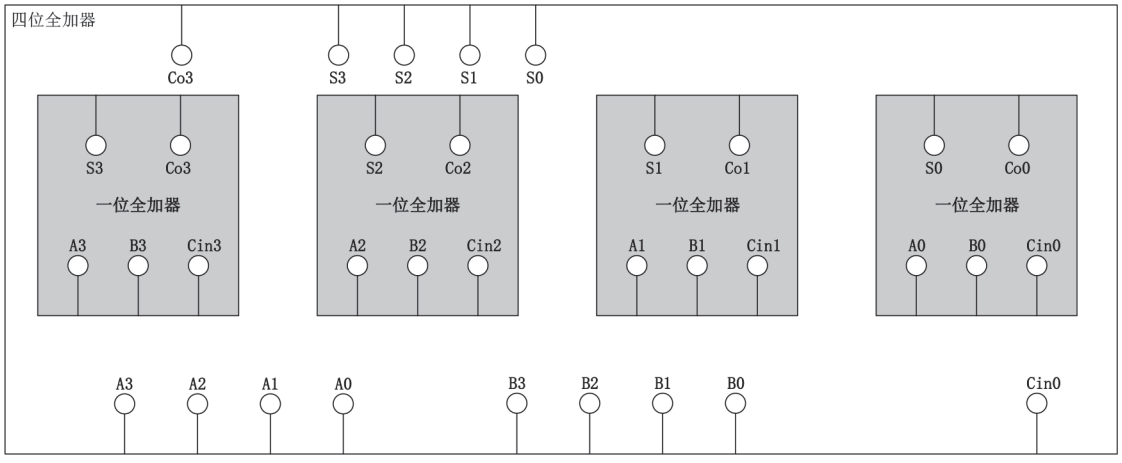

图2 串联同态加密加法器

（1）计算所有的a_ib_j，其中i$\in\{0,n_a-1\}$，j$\in\{0,n_b-1\}$，n_a和n_b是大整数A和B的位数，a_i,b_j是每一位的密文；

（2）计算所有Ab_j，其中j$\in\{0,n_b-1\}$，计算公式为（$\overline{abcd}$表示一个4位整数，每一位分别为$abcd$）：

$$Ab_j=\sum_{i=0}^{n_a-1}2^ia_ib_j=\sum_{i=0}^{n_a-1}2^ic_{i,j}$$

$$=\overline{c_{n_a-1,j}\,c_{n_a-2,j}\cdots c_{1,j}\,c_{0,j}}$$

（3）计算AB，在Ab_j的末尾补充j个$c0_{const}$，相当于左移j位，并将对其进行大整数相加，得到A、B的积C，此时C为密文；

（4）对C的每一位解密，即计算$m_k=Decrypt(sk,c_k)$，其中k$\in\{0,n_a+n_b-1\}$最后得到$\overline{m_km_{k-1}...m_1m_0}$即为乘积的明文。

在大数同态乘法中，第1、2步中的乘法是相互独立的，不会累积误差，而第三步中，进行了n_b-1次大整数的加法，每次加法最多累积n_a次误差，总计累积$n_a(n_b-1)$次1bit乘法的误差。

4 合约应用过程

该合约可以适应的场景，如图3所示。数据持有者首先将自己的数据上传到区块链网络，定制随机数种子，智能合约会将数据进行同态加密。当有人需要使用数据时，他需要向智能合约申请使用数据集X及使用函数F(X)，智能合约会拆解F(X)，直接计算出结果，返回给数据使用者。数据使用者收到的是密文数据，无法直接使用，此时向数据持有者申请解密。持有者会审查使用者的签名获得其区块链中的账号，审查该账号是否有使用数据的权限，审查结束后验证无误，就会使用私钥解密数据。值得注意的是，数据持有者并不知道X与F(X)，也就是不知道数据使用了哪些数据也不知道如何使用的这些数据。这在一定程度上也保护了数据使用者的隐私与权益。不过区块链网络获得了全部信息，仍可以将历史信息记录下来，不会影响到整个网络的可溯源性。

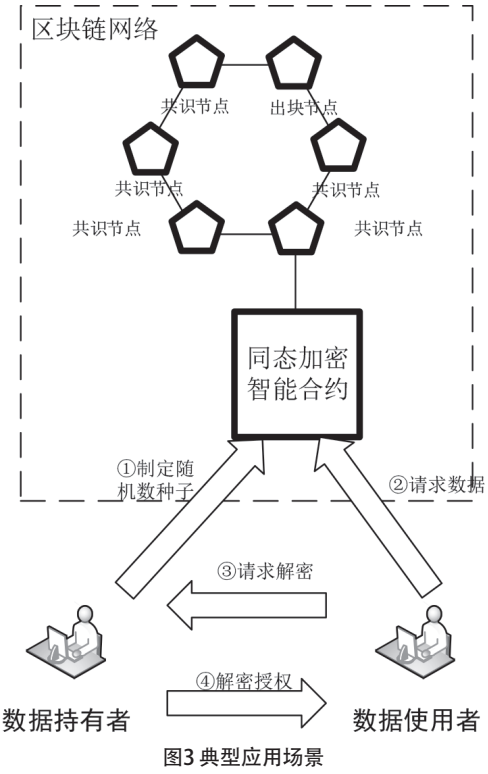

图3 典型应用场景

5 安全性分析

本文从已知密文攻击和已知明文两种方式分析。假设攻击方知道k个密文c_i，试图获得私钥p，或明文m_i。在数位较少的时候，可以直接猜想m_i，有50%的正确率，此时非常不安全。本文需要采用大整数的方式，即增加密文的数量，此时直接猜想的收益会指数型下降。另外每加密一个数据都会重新计算随机数q_ir_i，对于同样的明文，会生成完全不同的密文，也保障了安全性。

在尝试获取明文失败后，攻击方试图获取私钥p，该获取过程为：

$$\begin{cases}c_1=m_1+2r_1+pq_1\\c_2=m_2+2r_2+pq_2\\\quad\vdots\\c_k=m_k+2r_k+pq_k\end{cases}$$

可知，即使攻击方知道密文对应的明文，在随机数不被预测的情况下，c可近似的看作是p的倍数。破解私钥的过程可以看作根据一些列c_1,c_2,c_3,c_4,...,c_k近似p的倍数求p的过程。该过程就是近似最大公约数问题(approximate—GCD problem)，而此问题就目前来看是无解的，所以该方法是安全的。

6 结束语

本文根据DGHV同态加密方法，设计了随机数生成方法及可进行同态布尔逻辑值计算、同态数值比较计算、同态长整数计算的智能合约。该方法适应智能合约中计算资源相对紧张，要求全网共识的环境。同时，本文证明了方法的同态性与安全性，还提出了一种该方法应用的典型场景。

本文提出的智能合约还具备以下两个特点：采用独特的随机数生成方式，该方式更适合智能合约场景中使用，并且熵源由数据持有方产生，计算量小，可全网同步；将原本只有一位的DGHV扩展至长整数范围，使其更具备实用价值。

综上所述，该智能合约和加密方法可以适应保护交易双方的隐私与安全的要求。

基金项目：

1. 国家自然基金项目（项目编号：61672074，61672075）；

2. 教育部中国移动基金（项目编号：MCM20180104）；

3. 国家重点研发计划（项目编号：2018YFB1402702）。

参考文献

[1] 范吉立,李晓华,聂铁铮, 等.区块链系统中智能合约技术综述[J].计算机科学,2019,46(11):1-10.

[2] Gentry, Craig. Fully Homomorphic Encryption Using Ideal Lattices [A]. Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing [C].New York, NY, USA: Association for Computing Machinery,2009.169-178.

[3] M. Dijk, C. Gentry, S. Halevi, and V.Vaikuntanathan. Fully homomorphic encryption over the integers[J]. Applications of Cryptographic Techniques: Springer, Berlin, 2010, 24-43.

[4] J. S. Coron，A. Mandal，D. Naccache, M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys[A]. Proceedings of the 31st Conference on Advances[C]. 2011.

[5] Zuowei Wu and Taoshen Li. An Improved Fully Homomorphic Encryption Scheme under the Cloud Environment[A]. Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing[C]. New York, NY, USA Association for Computing Machinery, 2017, 251－252.

[6] 林如磊,王箭,杜贺.整数上的全同态加密方法的改进[J].计算机应用研究,2013,30(5):1515-1519.

[7] 李子臣,张峰娟,王培东.一种短密钥高效全同态加密方法[J].计算机应用研究,2017,34(02):487-489+494.

[8] 朱凤霞.基于区块链技术的交易数据库加密技术[J].电子设计工程,2020,28(3):93-97.

[9] 王化群,张帆,李甜,等.智能合约中的安全与隐私保护技术[J].南京邮电大学学报（自然科学版）,2019,39(4):63-71.

[10] 叶俊,于天娇,郭祯,荆兆星. 基于区块链的可验证医疗数据统计方案[J]. 网络空间安全, 2019, 10(12): 1-.

作者简介：

仝秦玮（1996-），男，汉族，内蒙古包头人，北京航空航天大学计算机学院，在读硕士；主要研究方向和关注领域：区块链与数据隐私保护。

李洁（1983-），女，汉族，北京人，北京航空航天大学计算机学院，在读博士；主要研究方向和关注领域：区块链与数据隐私保护。

王洁（1993-），男，汉族，四川资阳人，北京航空航天大学计算机学院，在读硕士；主要研究方向和关注领域：分布式机器学习与区块链。

胡心森（1996-），男，汉族，河南信阳人，北京航空航天大学计算机学院，在读硕士；主要研究方向和关注领域：分布式机器学习与区块链。

胡凯（1963-），男，汉族，湖南长沙人，北京航空航天大学，博士，北京航空航天大学计算机学院，教授；主要研究方向和关注领域：分布式计算、区块链与数字社会。