软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1549−1565 [doi: 10.13328/j.cnki.jos.004996]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

云环境下集合隐私计算*

李顺东[1], 周素芳[1], 郭奕旻[1], 窦家维[2], 王道顺[3]

[1](陕西师范大学 计算机科学学院,陕西 西安 710062)

[2](陕西师范大学 数学与信息科学学院,陕西 西安 710062)

[3](清华大学 计算机科学与技术系,北京 100084)

通讯作者: 李顺东, E-mail: shundong@snnu.edu.cn

摘 要: 多方保密计算是网络空间安全与隐私保护的关键技术,基于同态加密算法的多方保密计算协议是解决云计算安全的一个重要工具.集合隐私计算是多方保密计算的基本问题,具有广泛的应用.现有的集合隐私计算方案多是基于两方的情况,基于多方的方案较少,效率较低,且这些方案都不能扩展到云计算平台.首先设计了一种编码方案,根据该编码方案和同态加密算法,在云计算环境下构造了一个具有普遍适用性且抗合谋的保密计算集合并集问题解决方案.该方案中的同态加密算法既可以是加法同态,又可以是乘法同态的加密算法.进一步利用哥德尔编码和ElGamal 公钥加密算法构造了一种适用于云计算的高效集合并集计算方案.这些方案还可以对多个集合中的所有数据进行保密排序,并证明这些方案在半诚实模型下是安全的.所提方案经过简单改造,也可以保密地计算多个集合的交集.

关键词: 云安全;密码学;多方保密计算;保密计算集合并集;保密计算集合交集;保密排序

中图法分类号: TP309

中文引用格式: 李顺东,周素芳,郭奕旻,窦家维,王道顺.云环境下集合隐私计算.软件学报,2016,27(6):1549−1565. http://www.jos.org.cn/1000-9825/4996.htm

英文引用格式: Li SD, Zhou SF, Guo YM, Dou JW, Wang DS. Secure set computing in cloud environment. Ruan Jian Xue Bao/ Journal of Software, 2016,27(6):1549−1565 (in Chinese). http://www.jos.org.cn/1000-9825/4996.htm

Secure Set Computing in Cloud Environment

LI Shun-Dong[1], ZHOU Su-Fang[1], GUO Yi-Min[1], DOU Jia-Wei[2], WANG Dao-Shun[3]

[1](School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

[2](School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, China)

[3](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Secure multiparty computation (SMC) is a key technology of cyberspace security and privacy preservation, and it is vital to provide secure cloud computing with SMC based on homomorphic encryption schemes. Secure set computing, which has extensive applications, is a fundamental problem in SMC. Existing solutions to secure set computing are mainly constructed between two parties, but less presented on multi-parties. Those schemes are inefficient, and are hardly adequate to cloud computing. This study proposes a new coding scheme and incorporates homomorphic encryption algorithm to construct a protocol for secure set union computing in cloud environment. The proposed scheme is universal and secure against the collusion of participants. The homomorphic encryption adopted can be either additive or multiplicative. The paper also proposes an efficient secure set union computing scheme, incorporating the Gödel

* 基金项目: 国家自然科学基金(61272435, 61373020)
Foundation item: National Natural Science Foundation of China (61272435, 61373020)
收稿时间: 2015-08-14; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21
CNKI 网络优先出版: 2016-01-22 10:14:48, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1014.004.html

numbering and ElGamal public key encryption. The proposed schemes can be used to sort multiple sets, and are proved to be secure in the semi-honest model. In addition, with few modifications, the protocol can also securely compute the intersection of multiple sets.

Key words: secure cloud; cryptography; secure multi-party computation; secure set union; secure set intersection; secure sorting

云计算通过网络将大量的存储资源、计算资源及软件资源链接在一起,形成资源池,给我们带来了超强的计算能力、无限的存储能力和巨大的经济效益,为广大计算机用户提供高效、便捷的服务.云计算已经成为信息领域关注的热点问题,各国都给予了高度的关注.但利用云进行计算或存储需要把数据交给云,这对参与者计算的数据安全性与隐私构成了巨大的威胁,使得云计算的安全问题成为学术界及工业界关注的焦点[1],也是广大用户是否使用云计算的决定性因素.

多方保密计算(secure multipaty computation)是网络空间安全与隐私保护的关键技术,具有广泛的应用,但往往需要进行计算量比较大的公钥加密运算,数据较多的时候对很多用户都是一个难题.研究利用云计算平台完成这样的计算,同时又保护用户的隐私,对云计算的推广普及和云计算中的隐私保护都有重要的理论与实际意义.本文是这方面工作的有益探索.

多方保密计算最早由姚期智教授[2]提出,是指多个参与者联合进行的秘密计算,计算结束后,每个参与者除了计算结果外,其输入信息没有任何泄露.随后,Goldreich 等人[3,4]从理论上证明了任意的多方保密计算问题都是可解的,并给出了通用的解决方案.但同时指出,从计算效率方面考虑对具体的问题应该研究具体的解决方案.这使得人们纷纷研究各种各样多方保密计算问题的解决方案,如百万富翁问题[5,6]、保密计算几何[7]、保密的信息比较[8]、保密的集合问题[9]、保密的远程访问[10]、保密拍卖[11]、保密的数据挖掘[12]等.Goldreich 等人还设计了一个编译器,借助于该编译器,可以用一个对于半诚实(semi-honest)参与者安全的多方保密计算协议生成一个对于恶意(malicious)参与者也安全的多方保密计算协议.这一成果也体现了研究半诚实模型下的多方保密计算协议的价值,Goldreich 在文献[4]中专门论述了研究半诚实模型下多方保密计算的理论与实际意义.

集合是一个非常重要的概念,现实中的很多问题都可以用集合表示,集合问题的研究是多方保密计算研究的一个重要方面.现有的关于保密计算集合问题的研究包括保密计算集合的交集[13−16]、保密计算集合的并集[17−21]以及保密计算集合交集与并集的势[22,23]等,但这些研究多是基于两个参与者的情况,对于多个参与者的研究还比较少,且计算效率较低.

Freedman[13]首先给出了多个参与者集合交集的保密计算方案,该方案借助于 Paillier 公钥加密算法,同时利用多项式不经意求值的方法给出了解决方案,但该方案需要对每个参与者的多项式系数加密,然后通过对加密后的多项式计算来实现集合交集的保密计算,计算复杂性与通信复杂性都很高,效率很低,且所有集合的元素都来自一个无限大的范围.Kissner[17],Frikken[18]给出的多个集合并集的保密计算方案,将每个参与者的集合表示成多项式的形式,然后利用多项式的数学性质与同态加密算法给出了具体的解决方案.但他们需要借助混合网或将元素混淆来计算所有集合的并集,计算复杂性较高,并且计算的结果不是真正的交集或者并集,而是一个多重集,泄露了较多的信息.文献[19]通过对所有参与者集合中的元素进行不经意排序,根据排序结果得到多个集合并集的计算结果,但该方法中的不经意排序主要是借助于文献[24]中电路的方法实现的.

基于电路的方法虽然高效,但也存在电路相关的一些缺点.

- 首先,一个简单的算法用电路实现也很困难,转换成的电路的规模也是惊人的,存储这样的电路会很快消耗完计算机的所有内存资源[25];
- 此外,电路方法的可移植性与普遍适用性较差,需要预先估计要计算集合元素的范围,从而推算出可能需要的比较门(gate)的数量及电路的深度,超出预定范围的对象将不能进行计算,需要根据具体的应用场景设计相应的电路,不具有普遍适用性.

文献[20]将参与者的集合表示成多项式的形式,并利用翻转罗朗级数(reversed Laurent series)和秘密共享的方法给出了集合并集的保密计算方案.文献[21]借助全同态加密算法和多项式求值的方法给出的方案只能计算集合的并集,方案的通信轮数是参与者个数的线性函数,但方案的计算复杂性仍然很高,效率较低.以上的这些方案均不能借助于半可信的云服务器完成集合的保密计算,参与者的计算量都是很大的.

本文研究的集合隐私计算问题是指多个参与者在云计算环境下的集合并集与交集的保密计算问题,由于云平台中运行的各类云应用没有固定不变的基础设施和安全边界,难以实现参与者的数据安全与隐私保护,因此需要参与者利用密码学的方法来实现安全的云计算.根据多方保密云计算[26]的思想,将云看做是由多个云服务器组成的集合,云上的计算看作多个云服务器间的计算.借助于新的编码方法,将每个参与者的集合用一个相应的向量替代,根据新向量的计算结果可以得到原有集合的计算结果.该方案可以用乘法同态加密算法构造,也可以用加法同态加密算法构造,具有普遍的适用性.同时,该方案也可以借助于一个半可信的云服务器实现隐私的集合计算.

利用哥德尔编码(Gödel numbering)与 ElGamal 公钥加密算法,本文进一步提出了一种高效的云集合计算方案,该方案只需要 n 个参与者进行 n 次加密与一次解密,加密和解密的次数与参与者拥有集合元素的个数无关,具有较高的效率.为了方便说明,本文给出的集合隐私计算方案是在保密计算集合并集的基础上给出的,这些方案经过简单改造,也可以用来保密计算集合的交集.本文设计的解决方案均可以抵抗合谋攻击且具有广泛的适用范围,对于任何已知全集的集合计算方案都适用.

本文贡献如下:

(1) 为高效地解决保密的集合并集计算问题,设计了新的编码方法——1-r(0-r)编码方法.通过该编码方法将参与者拥有的集合编码成一个向量,避免了两两计算集合并集的习惯思维和两两计算可能导致的信息泄露;

(2) 为了抵抗合谋攻击,利用模运算的性质提出了密文分割方案,该方案可以把密文随机地分割成任意个份额,而且这些份额的乘积等于原来的密文.这种分割不需要对密文进行因子分解,具有较高的效率;

(3) 利用上述两种工具,提出了保密计算集合并集的协议,该协议具有很好的性质.可以用加法同态加密算法,也可以用乘法同态加密算法实现;可以借助半可信的云服务器实现保密计算集合的并集;经过简单改造,还可以保密计算集合的交集.这种方案的一个直接结果就是可以对多个集合中的数据进行保密排序.用模拟范例证明了协议的安全性;

(4) 利用哥德尔编码将一组数编码成一个数,对于一组数,只需要进行一次加密就可以进行集合并集的计算,进一步降低了协议的计算复杂性.

1 预备知识

1.1 隐私的集合计算问题

设 n 个参与者 $P_1,\ldots,P_n$ 分别拥有集合 $X_1=\{x_{11},...,x_{1l_1}\},...,X_n=\{x_{n1},...,x_{nl_n}\}\subset U$ (其中,$U=\{u_1,\ldots,u_m\}$且 $u_1<\ldots<u_m$),组成二元组集合$\{(P_1,X_1),\ldots,(P_n,X_n)\}$.他们想要在不泄露$\{(P_1,X_1),\ldots,(P_n,X_n)\}$的前提下,计算这 n 个集合的交集或并集.

隐私计算集合的并集是指 n 个参与者计算这 n 个集合的并集 $X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$,每个参与者可以知道所有集合的并集,而对其他参与者集合中的元素没有任何信息.

n 个参与者计算这 n 个集合的交集 $X_1\cap\ldots\cap X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$,每个参与者可以知道所有集合的交集,而对其他参与者集合中的元素没有任何信息,这就是隐私计算集合的交集问题.

1.2 理想模型(ideal model)

假设有一个可信的第三方(trusted third party,简称 TTP),他在任何情况下都不会撒谎,也绝不会泄露任何不该泄露的信息.理想的多方保密计算是指多个参与者借助可信的第三方进行保密计算.理想的多方保密计算模型可以描述为:有 n 个参与者 $P_1,\ldots,P_n$,他们分别将各自的秘密 $X_1,\ldots,X_n$ 告诉可信的第三方,由可信的第三方单独计算函数 f:

$$f(X_1,\ldots,X_n)=(f_1(X_1,\ldots,X_n),\ldots,f_n(X_1,\ldots,X_n)).$$

然后,将计算结果 $f_1(X_1,\ldots,X_n),\ldots,f_n(X_1,\ldots,X_n)$分别告诉 $P_1,\ldots,P_n$.在计算过程中,$P_1,\ldots,P_n$除了从协议得到可信

第三方发送给自己的计算结果外得不到其他任何信息.理想的多方保密计算模型虽然简单,但具有最高的安全性,任何一个计算 $f(X_1,\ldots,X_n)$的实际多方保密计算协议的安全性都不可能超过这个协议,其他保密计算协议可以通过与理想模型进行比较来检验其安全性.理想多方保密计算方案存在的问题是:在遍布世界的网络环境中,要找到一个可信的第三方是极不现实的.

1.3 半诚实模型的安全性

本文假设所有的参与者都是半诚实参与者.半诚实参与者在协议的执行过程中会按照协议要求忠实地履行协议,执行协议后,除了协议的执行结果外没有任何信息泄露.但他们可能会记录下协议执行过程中收集到的所有信息,并试图根据收集到的信息(多个参与者时,我们应该考虑多个参与者在协议执行过程中收集到的信息,而不是一个参与者收集到的信息[4])推算出其他参与者的输入.所以,半诚实模型又称为诚实但好奇模型或被动模型.

密码学不同领域中采用的安全性证明方法不同.基于可证明安全理论的证明适用于密码学中的加密与数字签名领域,而模拟范例是目前多方保密计算研究中广泛接受、普遍采用的证明方法.该方法的原理是将一个实际的多方保密计算协议与一个理想的多方保密计算协议的安全性进行对比,如果实际的多方保密计算协议不比理想的多方保密计算协议泄露更多信息,则实际的多方保密计算协议是安全的.

由于多数多方保密计算协议都是利用公钥加密算法构造的,因此多方保密计算协议的安全性证明一般是在假设相关公钥加密算法安全的条件下,具体的多方保密计算协议与理想的多方保密计算协议相比不会泄露更多的信息,即,将多方保密计算协议的安全性归约到公钥加密算法的安全性.假设公钥加密算法是安全的,那么一个参与者利用理想的多方保密计算协议得到的信息可以通过构造一个模拟器来模拟实际多方保密计算协议的执行过程.如果模拟的过程与实际执行过程是计算不可区分的,则说明实际协议是安全的.本文采用模拟范例对协议进行安全性证明.

假设 n 个参与者 $P_1,\ldots,P_n$ 分别拥有集合 $X_1,\ldots,X_n$,他们想要借助协议 π保密地计算 $f(X_1,\ldots,X_n)$.

令 $\bar{X}=(X_1,\ldots,X_n)$.在协议执行过程中,$P_i(i=1,\ldots,n)$得到的信息序列记为 $view_i^{\pi}(\bar{X})=(X_i,r_i,M_i^1,\ldots,M_i^t)$,其中,$M_i^j(j=1,\ldots,t)$ 表示 P_i 第 j 次得到的信息.对于 $I=\{i_1,\ldots,i_s\}\subseteq\{1,\ldots,n\}$,我们令:

$$view_I^{\pi}(\bar{X})=(view_{i_1}^{\pi}(\bar{X}),\ldots,view_{i_s}^{\pi}(\bar{X})).$$

定义 1. 在参与者都是半诚实的情况下,如果存在概率多项式时间算法 S 对于每个 $I\subseteq\{1,\ldots,n\}$都使得式(1)成立,则协议 π保密地计算 n 元函数 f.

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n} \tag{1}$$

其中,$X_I=(X_{i_1},\ldots,X_{i_s})$,$\overset{c}{\equiv}$表示计算上不可区分.

1.4 哥德尔编码

哥德尔在证明哥德尔不完备定理时,首创了哥德尔编码.哥德尔编码将非负整数序列与自然数建立起一一对应关系.有穷序列$(a_1,\ldots,a_m)$借助素数序列$(p_1,\ldots,p_m)$(其中,$p_1,\ldots,p_m$ 可以是任意不同的 m 个素数,为了使计算简单,一般取从 2 开始的 m 个连续素数)建立如下的对应关系:

$$[a_1,\ldots,a_m]=\prod_{j=1}^{m}p_j^{a_j} \tag{2}$$

$[a_1,\ldots,a_m]$称作有穷序列$(a_1,\ldots,a_m)$的哥德尔数.根据算术基本定理,任意的自然数可以唯一地分解成多个素数的乘积,而构成哥德尔数的素数序列$(p_1,\ldots,p_m)$是已知的,因此,由$[a_1,\ldots,a_m]$可以很容易得到序列$(a_1,\ldots,a_m)$.

在本文中,我们用哥德尔编码建立起向量和自然数之间的一一对应关系.向量编码成的哥德尔数是一个非常大的数,这在日常的计算中将极大地增加计算量,但在密码学中,这样的增加是完全可以接受的.因为公钥密码学的计算都是在一个非常大的群中进行的,为了取得安全性,即使非常小的自然数加密也需要在一个很大的群中进行运算,这是依靠公钥加密保证安全必须付出的代价.

1.5 同态加密算法

一个传统的公钥加密方案ε包括3种算法:KeyGen$_\varepsilon$,Encrypt$_\varepsilon$,Decrypt$_\varepsilon$.对于KeyGen$_\varepsilon$算法,给定一个安全参数λ,输出一个私钥sk和对应的公钥pk,并给定明文空间$\mathcal{P}$和密文空间$\mathcal{C}$,即:

- KeyGen$_\varepsilon(\lambda)\leftarrow(sk,pk,\mathcal{P},\mathcal{C})$;
- Encrypt$_\varepsilon$算法对于给定的公钥pk和明文$M\in\mathcal{P}$输出相应的密文$C\in\mathcal{C}$,即:

$$\text{Encrypt}_\varepsilon(pk,M)\leftarrow(C)(M\in\mathcal{P}).$$

- 根据密文C和私钥sk,Decrypt$_\varepsilon$算法输出相应的明文M:

$$\text{Decrypt}_\varepsilon(sk,C)\leftarrow(M)(C\in\mathcal{C}).$$

一个同态加密算法ε除了以上3种算法外,还包括一个多项式时间的Evaluate$_\varepsilon$算法.如果C_i是M_i用公钥pk加密的密文,为算法Evaluate$_\varepsilon$输入公钥pk、一种操作S和密文组$\mathcal{C}=\langle C_1,\ldots,C_m\rangle$,算法Evaluate$_\varepsilon$输出$S(M_1,\ldots,M_m)$的密文,即:

$$\text{Evaluate}_\varepsilon(pk,S,\mathcal{C})\leftarrow\text{Evaluate}_\varepsilon(pk,S(M_1,\ldots,M_m)).$$

ElGamal公钥加密算法是具有乘法同态性质的加密算法.对于消息M,其密文可以表示成如下形式:

$$E(M)=(c_1,c_2)=(g^r \bmod p, MH^r \bmod p).$$

其中,r是加密者选择的一个随机数,$h=g^x \bmod p$是私钥x对应的公钥.

对消息M_1,M_2的密文做乘法运算:

$$E(M_1)\times E(M_2)=(g^{r_1},M_1h^{r_1})\times(g^{r_2},M_2h^{r_2})=(g^{r_1+r_2},M_1\times M_2h^{r_1+r_2})=E(M_1\times M_2).$$

即:对M_1,M_2的密文做乘法运算,相当于对M_1,M_2做乘法运算之后再加密.如果$M_1=1$,对M_1的密文与M_2的密文做Evaluate运算后,得到M_2不同的密文.

Paillier公钥加密算法[27]是一种具有加法同态性质的加密算法.消息M的密文可以表示为

$$E(M)=g^Mr^N \bmod N^2.$$

其中,$M\in Z_N$,N是RSA算法中的模数;$r\in Z_N^*$是加密者选择的随机数.对消息M_1,M_2的密文做乘法运算:

$$E(M_1)\times E(M_2)=(g^{M_1}r_1^N \bmod N^2)\times(g^{M_2}r_2^N \bmod N^2)=g^{M_1+M_2}(r_1\cdot r_2)^N \bmod N^2=E(M_1+M_2).$$

因此,对消息M_1,M_2的密文做乘法运算,相当于对其明文做加法运算之后再加密.与ElGamal公钥加密算法类似,如果$M_1=0$,对M_1的密文与M_2的密文运算得到M_2新的密文.

2 保密计算集合并集的解决方案

保密计算集合并集是隐私保护、保密的数据挖掘、保密的图算法等许多协议的基本模块,具有重要的现实意义与理论价值.例如在隐私保护的分布式网络监控中,有n个监控点$P_1,\ldots,P_n$,每个监控点$P_i(i=1,\ldots,n)$记录下的不良网络行为记为集合X_i,为了更好地维护网络的安全,这n个监控点想要知道在这个网络中都有哪些不良的网络行为,以便更快地检测到不良行为,降低网络的风险.但是为了隐私不能泄露每个结点检测到的具体的不良行为,这需要n个监控结点保密地计算这n个结点检测到的不良行为的并集.

2.1 基本原理

1-r 编码方法. 集合$X_i\subset U=\{u_1,\ldots,u_m\}$(其中,$u_1<\ldots<u_m$)编码为1-$r$向量$U_i=(u_{i1},\ldots,u_{im})$的方法如下:

For j=1 to m
  If $u_j\in X_i$
    $u_{ij}\leftarrow r_{ij}$
  Else $u_{ij}\leftarrow 1$
End

假设有n个集合$X_1,\ldots,X_n\subset U=\{u_1,\ldots,u_m\}$(其中,$u_1<\ldots<u_m$),将每个集合$X_i(i=1,\ldots,n)$借助于1-$r$编码方法编码

成向量 $U_i=(u_{i1},\ldots,u_{im})$,然后将各个向量的对应分量相乘得到新的向量 U'.

$$U'=U_1..U_n=(u_{11}...u_{n1},...,u_{1m}...u_{nm})=(u'_1,...,u'_m).$$

如果向量 U'中的某个分量 $u'_j\neq 1$,则说明这 n 个向量的第 j 个分量不全为 1,即,n 个集合中至少有一个集合含有元素 u_j.根据向量 U'中不为 1 的分量,可以得到 $X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$.因此,保密计算集合并集问题可以归约到保密地计算向量 U'.

以下是利用 1-r 编码方法计算集合并集的实例(实例 1).为了简单说明,实例 1 中给出的是计算 3 个集合 $X_1,X_2,X_3\subset U$ 的并集.

Instance 1 Fundamental of computing sets union

实例 1 计算集合并集的基本原理

集合/向量	1	2	3	4	5	6	7	8	9	10
U	101	102	103	104	105	106	107	108	109	110
X_1	101	–	–	–	105	–	107	–	–	–
U_1	r_{11}	1	1	1	r_{15}	1	r_{17}	1	1	1
X_2	–	–	103	–	105	–	–	108	–	–
U_2	1	1	r_{23}	1	r_{25}	1	1	r_{28}	1	1
X_3	–	–	–	104	–	106	–	–	109	–
U_3	1	1	1	r_{34}	1	r_{36}	1	1	r_{39}	1
$U'=U_1\cdot U_2\cdot U_3$	r_{11}	1	r_{23}	r_{34}	$r_{15}\cdot r_{23}$	r_{36}	r_{17}	r_{28}	r_{39}	1
σ	101	–	103	104	105	106	107	108	109	–

2.2 具体方案

在云计算环境中,云服务器之间是不完全可信的,不能直接将明文数据在云服务器之间传递.因此,需要云服务器借助加密算法将私有的数据先进行加密,然后再将加密数据在云服务器之间传递,让云服务器对密文做处理来实现数据的隐私保护.

云服务器 P_n 输入选定的安全参数λ,运行 ElGamal 公钥加密算法的 KeyGen 算法,输出相应的系统参数 $params=(p,g)$、私钥 $sk=(x)$和对应的公钥 $pk=(h)$.其中,

- p 是由安全参数λ决定的大素数,此外,为了确保 ElGamal 公钥加密算法在乘法群 Z_p^* 上计算离散对数是困难的,需要$\phi(p)=p-1$ 中至少含有一个大的素因子;
- g 是 Z_p^* 的一个生成元;
- 公钥 h 是由私钥生成的,即 $h=g^x \bmod p$.

P_n 保存私钥 sk 并将公钥 pk 与系统参数 $params$ 公布,供其他云服务器使用.

每个云服务器 $P_i(i=1,\ldots,n)$将其秘密集合 $X_i(X_i\subset U=\{u_1,\ldots,u_m\}$,其中,$u_1<\ldots<u_m)$根据 1-$r$ 编码方法编码成向量 $U_i=(u_{i1},\ldots,u_{im})$.$P_i$ 用 P_n 公布的公钥 pk 与系统参数 $params$ 将自己构造的向量 U_i 加密,即,将向量 U_i 中的每个分量分别加密.

$$E(U_i)=(E(u_{i1}),\ldots,E(u_{im})).$$

如果直接将 $E(U_i)$发送给 P_n,P_n 可以将密文解密,从而知道其他云服务器的秘密集合,即使 P_n 是完全可信的,但由于所有的服务器都将加密的向量发送给 P_n,P_n 将成为敌手攻击的众矢之的,因此需要借助于其他的方法防止 P_n 解密得到其他云服务器的信息.本文根据同态加密算法的性质,采用将 P_i 拥有的明文与 P_i 发送给 P_n 的密文混淆的方法来保证 U_i 的安全.P_i 将 $E(u_{ij})(j=1,\ldots,m)$分成非零的 $k_i(k_i\leqslant n)$份(其中,k_i 可以是不确定的,其他云服务器得不到任何有关 k_i 的信息),即 $E(u_{ij})_1,...,E(u_{ij})_{k_i}$.由于 ElGamal 公钥加密算法是具有乘法同态性的,为了保证正常解密,需要使:

$$E(u_{ij})_1..E(u_{ij})_{k_i}=E(u_{ij}).$$

P_i 每次从每个密文分量中取出一份不同的份额,共取 k_i 次,构成 U_i 的 k_i 份密文 $E(U_i)_1,...,E(U_i)_{k_i}$,然后将这 k_i 份密文发送给 n 个云服务器中的 k_i 个.这 k_i 个云服务器可以包括 P_i 自己,也可以不包括,其他云服务器得不到

这 k_i 个云服务器的任何信息.

在 ElGamal 公钥加密算法中,由于密文属于一个非常大的群 Z_p^*,将密文如 $E(u_{ij})(1\leqslant i\leqslant n,1\leqslant j\leqslant m)$分成 k_i 份是困难的, Z_p^* 中也包括一些不能进行因子分解的整数,或一些因子少于 k_i 个的整数,因此,我们不能直接利用因子分解的方法将密文分成 k_i 份.

在本方案中,由于云服务器在做 Evaluate 运算时只需要保证等式 $E(u_{ij})_1...E(u_{ij})_{k_i}=E(u_{ij})$ 成立,因此不需要真的将密文按照因子分解的方法分成 k_i 份,根据模运算的性质,只需要选择 k_i 个随机数 $r_{i1},...,r_{ik_i}\in Z_p^*$,同时保证这些随机数满足 $r_{i1}...r_{ik_i}=1 \bmod p$.从这 k_i 个随机数中随机地选择一个随机数如 r_{i1},将其与 $E(u_{ij})$相乘构成 $E(u_{ij})$的一份份额如 $E(u_{ij})_1=E(u_{ij})\cdot r_{i1}$,然后将 $E(u_{ij})_2,...,E(u_{ij})_{k_i}$ 分别用 $r_{i2},...,r_{ik_i}$ 表示,则:

$$E(u_{ij})_1...E(u_{ij})_{k_i}=r_{i1}\cdot E(u_{ij})\cdot r_{i2}...r_{ik_i}=E(u_{ij}).$$

该密文拆分方法只需要将消息加密 1 次,具有较高的效率.

在本方案中,云服务器 $P_i(i=1,\ldots,n)$将其密文向量 $E(U_i)$分成 k_i 份并发送给 n 个云服务器中的 k_i 个,其他任何一个云服务器得不到全部 k_i 份密文,即使拥有私钥 sk 也得不到关于 U_i 的任何信息.不妨设敌手得到关于 $E(U_i)$ 的 k_i-1 份密文 $E(U_i)_1,...,E(U_i)_{k_i-1}$,因为:

$$E(U_i)_1...E(U_i)_{k_i-1}\cdot E(U_i)_{k_i}=E(U_i).$$

在这个方程中,敌手不知道 $E(U_i)_{k_i}$ 与 $E(U_i)$,敌手得到的是不定方程,即使敌手拥有私钥 sk,可以将 $E(U_i)_1,...,E(U_i)_{k_i-1}$ 解密,也得不到关于 U_i 的任何信息.因此,每个云服务器将其密文分成 k_i 份发送给 n 个云服务器中的 k_i 个是安全的.

每个云服务器 $P_i(i=1,\ldots,n)$将密文分发后,也会收到其他云服务器发送过来的密文向量.P_i 将收到的密文向量的每个相应分量分别相乘,构成新的密文向量 $E(U_i')=(E(u_{i1}'),...,E(u_{im}'))$ 并发送给云服务器 P_n,从而实现云服务器发送的密文与自身明文关系的混淆.

云服务器 P_n 将收到所有密文向量的对应分量分别相乘,得到 $E(U')$,即:

$$E(U')=E(U_1')...E(U_n')=(E(u_{11}')...E(u_{n1}'),...,E(u_{1m}')...E(u_{nm}')).$$

由于每个云服务器都是半诚实参与者,在协议的执行过程中会严格按照协议的要求执行协议,且不会加入其他信息,所以 P_1 构造的密文向量 $E(U')$是所有参与者密文向量各个分量的乘积,即:

$$E(U')=E(U_1)...E(U_n)=(E(u_{11})...E(u_{n1}),...,E(u_{1m})...E(u_{nm}))=(E(u_1'),...,E(u_m')).$$

P_n 用自己的私钥将 $E(U')=(E(u_1'),...,E(u_m'))$ 解密,得到向量 $U'=(u_1',...,u_m')$.P_n 根据 U'中不为 1 的元素与集合 U 的关系,得到 $X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$并将其公布.从而,每个云服务器得到了 n 个云服务器拥有集合的并集.在此基础上给出了保密计算集合并集的解决方案,具体协议如下:

协议 1. n 个云服务器保密计算集合的并集.

输入:$P_1,\ldots,P_n$ 各自的秘密集合 $X_1,\ldots,X_n\subset U=\{u_1,\ldots,u_m\}$;

输出:$X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$.

1. 云服务器 P_n 将自己的公钥 pk 与系统参数 $params$ 公布,并保留私钥 sk;
2. 每个云服务器 $P_i(i=1,\ldots,n)$计算如下:
   (1) 将自己的秘密集合 X_i 编码为向量 U_i;
   (2) 用 P_n 的公钥 pk 与系统参数 $params$ 加密 U_i 为 $E(U_i)$;
   (3) 将密文向量 $E(U_i)$随机地分成 k_i 份并发送给 n 个云服务器中的 k_i 个;
   (4) 把收到的所有密文向量对应的分量相乘得到新的密文向量 $E(U_i')$,并发送给 P_n;
3. 云服务器 P_n 计算如下:
   (1) 所有收到的密文向量对应的分量相乘,得到 $E(U')$;
   (2) 用自己的私钥 sk 解密 $E(U')$得到 U';
   (3) 根据 U'中不为 1 的分量得到$\{\sigma_1,\ldots,\sigma_h\}$;

(4) 公布$\{\sigma_1,\ldots,\sigma_h\}$.

为了便于理解,我们设计了一个包含 4 个云服务器的保密计算集合并集的实例.云服务器 $P_1,\ldots,P_4$(仅 P_4 拥有私钥 sk)分别拥有秘密集合 $X_1,\ldots,X_4\subset U=\{u_1,\ldots,u_m\}$,他们根据 1-$r$ 编码方法分别将 $X_1,\ldots,X_4$ 编码为向量 $U_1,\ldots,U_4$,进而根据 P_4 公布的公钥 pk 将编码的向量加密为 $E(U_1),\ldots, E(U_4)$.$P_i(i=1,\ldots,4)$密文分成的份数和密文份额发送给的云服务器都是不确定的,本例为了说明简单,假设 P_i 将其密文向量分成 3 份,自己保留 1 份,另外两份分别发送给云服务器 $P_{i+1(\mathrm{mod}\ 4)}$,$P_{i+2(\mathrm{mod}\ 4)}$(如图 1 的实线表示).在密文向量分发后,P_i 将收到的密文向量的乘积 $E(U_i')$ 发送给 P_4(如图中虚线表示).P_4 得到:

$$\begin{aligned}E(U') &= E(U_1')\ldots E(U_4')\\ &= E(U_1)_3E(U_3)_2E(U_4)_1\cdot E(U_2)_3E(U_1)_1E(U_4)_2\cdot E(U_3)_3E(U_2)_1E(U_1)_2\cdot E(U_4)_3E(U_3)_1E(U_2)_2\\ &= E(U_1)\cdot E(U_2)\cdot E(U_3)\cdot E(U_4).\end{aligned}$$

P_4 将 $E(U')$解密,得到 U',根据 U'中不为 1 的元素得到 $X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}$,然后将$\{\sigma_1,\ldots,\sigma_h\}$公布,这 4 个云服务器得到了它们集合的并集.

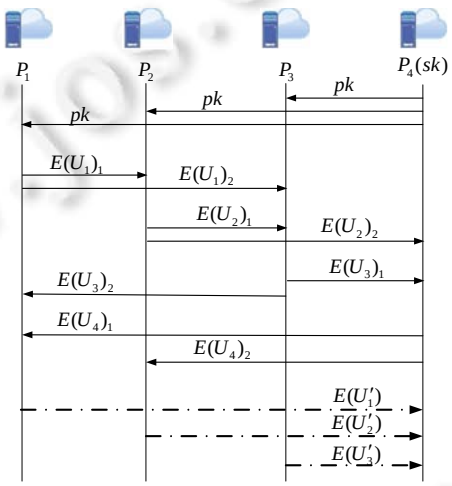

Fig.1 An instance with 4 cloud servers

图 1 4 个云服务器的实例

协议 1 描述的是有 n 个云服务器 $P_1,\ldots,P_n$ 分别拥有秘密集合 $X_1,\ldots,X_n$,他们想要知道这 n 个集合的并集而不泄露各自秘密集合的信息.该方案也可以看作是每个用户均拥有一个可信云服务器的情况.根据 1-r 编码方法,对用户借助半可信云服务器保密计算集合并集的情况,该方案经过简单改造也同样适用.

在协议 1 中,每个云服务器 $P_i(i=1,\ldots,n)$需要将秘密集合 $X_i(X_i\subset U=\{u_1,\ldots,u_m\}$,其中,$u_1<\ldots<u_m)$根据 1-$r$ 编码方法编码成向量 $U_i=(u_{i1},\ldots,u_{im})$,然后将向量 U_i 加密,在云服务器之间对密文进行计算来实现集合并集的保密计算.由于 P_i 将向量 U_i 加密,仅相当于加密 1 和随机数 $r\neq1$,加密 1 和 r 是公开的,只需要保密 1 和 r 所在的位置,因此可以将加密运算外包给半可信的云服务器$\mathcal{S}$.

将云服务器 $P_1,\ldots,P_n$ 看作云用户 $P_1,\ldots,P_n$,他们想要借助半可信云服务器$\mathcal{S}$保密地计算集合 $X_1,\ldots,X_n\subset U=\{u_1,\ldots,u_m\}$的并集.由于ElGamal加密算法是概率加密算法,同一个明文可以有多个不同的密文,每个云用户 $P_i(i=1,\ldots,n)$让云服务器$\mathcal{S}$加密一些1,生成一些随机数 r 供自己构造向量 U_i 的密文.对于向量 U_i 的每个分量 $u_{ij}(j=1,\ldots,m)$,当 $u_{ij}=1$ 时,P_i 从 $E(1)$中随机地选择 k_i 个 $E(1)$,作为 $E(u_{ij})$的 $k_i(k_i\leqslant n)$个份额;当 $u_{ij}=r_{ij}$ 时,P_i 从 $E(1)$中随机地选择 k_i-1 个 $E(1)$,再随机地选择 $r\in Z_p^*$ 作为 $E(r)$,因为随机数的密文仍是一个随机数,将 k_i-1 个 $E(1)$和 $E(r)$作为 $E(u_{ij})$的 $k_i(k_i\leqslant n)$个份额.P_i 每次从各个密文分量中取出一份不同的份额,共取 k_i 次,构成 U_i 的 k_i 份密文

$E(U_i)_1,...,E(U_i)_{k_i}$,然后将这 k_i 份密文发送给 n 个云用户中的 k_i 个.P_i 把收到的所有密文向量对应的分量相乘得到新的密文向量 $E(U_i')$,并发送给云服务器$\mathcal{S}$.$\mathcal{S}$将所有收到的密文向量对应的分量分别相乘,得到密文向量 $E(U')$,把 $E(U')$发送给 P_n.P_n用自己的私钥 sk 解密 $E(U')$得到 U',根据 U'中不为 1 的元素得到 $X_1\cup...\cup X_n=\{\sigma_1,...,\sigma_h\}$并公布.

在改造过的方案中,虽然半可信云服务器$\mathcal{S}$为每个云用户服务,但是在整个过程中,$\mathcal{S}$得到的关于用户 P_i 的信息只有一些 $E(1)$、随机数 r 与 $E(U_i')$,并不影响协议的安全性.此外,经过改造,仅需要每个云用户做少量的模乘运算,P_n 做 m 次解密,加密运算全部由$\mathcal{S}$完成,参与者 $P_1,...,P_n$ 的计算量大大降低,真正地借助于云实现了保密计算集合的并集.

以上方案也可以用加法同态加密算法(Paillier 加密算法)来构造,只需要将 $E(1)$用 $E(0)$替代.在编码时,每个云服务器将其集合编码成 0-r 的向量,不属于其集合中元素的位置用 0 替代,为属于集合中的元素选择一个不为 0 的随机数.最后,P_n 根据 U'中不为 0 的元素与集合 U 的关系得到 $X_1\cup...\cup X_n=\{\sigma_1,...,\sigma_h\}$.由于现有的乘法同态加密算法比加法同态加密算法效率高,因此本文采用的是具有乘法同态性质的 ElGamal 公钥加密算法.根据方案中的计算方法可知,得到的并集$\{\sigma_1,...,\sigma_h\}$中的元素是严格递增的,即,$\sigma_1<...<\sigma_h$.因此,该协议还可以实现保密地计算多个云服务器集合的交集.

2.3 方案分析

模拟范例是在半诚实模型下研究多方保密计算时广泛接受、普遍采用的证明方法,由于半诚实模型下的云多方保密计算属于半诚实模型下的多方保密计算,因此可以采用模拟范例的方法证明.关于协议 1 的安全性,有以下定理 1:

定理 1. n 个云服务器保密计算集合的并集协议 1(记为Π)是保密的.

证明:根据协议 1 中参与合谋的云服务器的不同,分为以下 3 种情况证明Π的安全性.

(1) 云服务器 P_n 不参与合谋,其他云服务器集合 $P_I\subseteq\{P_1,...,P_{n-1}\}$合谋想要得到云服务器 $P_i(P_i\notin P_I,i=1,...,n)$集合 X_i 中的元素,令 $X_I=(X_1,...,X_{i-1},X_{i+1},...,X_{n-1})$.通过构造使下式成立的概率多项式时间模拟器 S 来证明上述情况的安全性:

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\Pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

S 的模拟过程如下:

① 给定输入 $(X_I,f_I(\bar{X}))$,令 $(X_I,f_I(\bar{X}))=(X_1,...,X_{i-1},X_{i+1},...,X_{n-1},f_I(\bar{X}))$,$S$ 随机选择两个集合 X_n',X_i',使得 $f_I(\bar{X})=f_I(\bar{X}')$,其中,$\bar{X}'=(X_1,...,X_{i-1},X_i',X_{i+1},...,X_{n-1},X_n')$.

② S 首先按照Π中 1-r 编码方法将 $\bar{X}'$ 中的每个集合编码成向量 $U_1,...,U_{i-1},U_i^*,U_{i+1},...,U_{n-1},U_n^*$;然后,用 P_n 公布的(pk,$params$)将向量 $U_1,...,U_{i-1},U_i^*,U_{i+1},...,U_{n-1},U_n^*$中的每个分量分别加密,得到密文向量:

$$E(U_1),...,E(U_{i-1}),E(U_i^*),E(U_{i+1}),...,E(U_{n-1}),E(U_n^*);$$

随后,将 n 个密文向量分别随机地分成 $k_1,...,k_n$ 份,模拟Π中的分发方法进行分发.

③ S 将各个密文向量随机地分发,对收到的密文做 Evaluate 运算,构成新的密文向量:

$$E(U_1^*),...,E(U_{i-1}^*),E(U_i^\circ),E(U_{i+1}^*),...,E(U_{n-1}^*),E(U_n^\circ).$$

④ S 对所有密文向量对应的分量做 Evaluate 运算,得到:

$$E(U')=E(U_1^*)...E(U_{i-1}^*)\cdot E(U_i^\circ)\cdot E(U_{i+1}^*)...E(U_{n-1}^*)\cdot E(U_n^\circ).$$

⑤ 由于 $P_n\notin P_I$,ElGamal 公钥加密算法的安全性是基于 DDH 困难性假设的,DDH 困难性假设认为,概率多项式时间算法的敌手不能区分元组 $D=(g^a,g^b,g^{ab})$与 $R=(g^a,g^b,g^c)(a,b,c\in Z_p^*)$,即,不能对密文解密.$S$ 是具有概率多项式时间计算能力的模拟器,且没有 ElGamal 公钥加密算法的私钥,不能对密文进行解密,只能根据协议执行结束时得到的$\{\sigma_1,...,\sigma_h\}$推算出 U',进而计算出$\{\sigma_1',...,\sigma_h'\}$.

在Π中:

$$\begin{aligned}view_I^{\Pi}(\bar{X})&=\{view_1^{\Pi}(\bar{X}),...,view_{i-1}^{\Pi}(\bar{X}),view_{i+1}^{\Pi}(\bar{X}),...,view_{n-1}^{\Pi}(\bar{X})\}\\&=\{(X_1,...,X_{i-1},X_{i+1},...,X_{n-1}),(U_1,...,U_{i-1},U_{i+1},...,U_{n-1}),\\&\quad(E(U_1),...,E(U_{i-1}),E(U_{i+1}),...,E(U_{n-1})),\\&\quad(E(U_1'),...,E(U_{i-1}'),E(U_{i+1}'),...,E(U_{n-1}')),E(U'),(\sigma_1,...,\sigma_h)\},\end{aligned}$$

令:

$$\begin{aligned}S(X_I,f_I(\bar{X}))=\{&(X_I,...,X_{i-1},X_{i+1},...,X_{n-1}),(U_1,...,U_{i-1},U_{i+1},...,U_{n-1}),\\&(E(U_1),...,E(U_{i-1}),E(U_{i+1}),...,E(U_{n-1})),\\&(E(U_1^*),...,E(U_{i-1}^*),E(U_{i+1}^*),...,E(U_{n-1}^*)),E(U^*),(\sigma_1',...,\sigma_h')\},\end{aligned}$$

因为$(\sigma_1,...,\sigma_h)=(\sigma_1',...,\sigma_h')$,由于 ElGamal 公钥加密算法是语义安全的,用语义安全加密算法加密的数据是计算不可区分的,上述模拟过程中得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的,即:

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\Pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

所以,Π在该情况下是保密的.

(2) 云服务器 P_n 与其他云服务器集合 $P_{I'}\subseteq\{P_1,\ldots,P_{n-1}\}$合谋想要得到云服务器 $P_i\notin P_I(I=I'\cup n,i=1,\ldots,n)$集合 X_i 中的元素.P_i 将其机密集合 X_i 编码成向量 U_i 并加密为 $E(U_i)$,然后将 $E(U_i)$随机地分成 k_i 份发送给 n 个云服务器中的 k_i 个,其他云服务器对于 P_i 将这 k_i 份密文发送给了哪 k_i 个云服务器没有任何信息.云服务器集合 P_I 不知道收到的关于 $E(U_i)$的份额是否是全部份额,因此,即使 P_n 与 $P_{I'}$将收到的关于 $E(U_i)$的全部份额相应分量相乘并解密,根据解密后得到的明文也不能确定向量 U_i,即,不能得到关于集合 X_i 中的元素.在协议执行后,除 P_i 外的其他全部云服务器合谋可能会得到关于 X_i 的信息,因为 P_n 与 P_I 根据$\{\sigma_1,\ldots,\sigma_h\}$和集合 $X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n$ 的关系,得到一些关于 U_i 的信息,但是由于所有集合中的相同元素在集合$\{\sigma_1,\ldots,\sigma_h\}$中仅出现一次,因此其他云服务器只会得到集合 X_i 独有的元素,这与借助于可信的第三方的理想模型的执行结果是一样的,不会泄漏更多的信息.所以,该情况下存在概率多项式时间算法 S(S 根据情形(1)容易构造,在此省略其构造过程)对于 I 使得下式成立:

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\Pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

(3) 云服务器 P_n 与其他云服务器集合 $P_{I'}\subseteq\{P_1,\ldots,P_{n-1}\}$合谋,想要得到云服务器集合 $P_{\bar{I}}=\{P_1,...,P_{n-1}\}-P_I$ ($I=I'\cup n$)中云服务器的元素.当 $P_{\bar{I}}$ 中只有一个元素时,不妨令除 P_i 外的其他云服务器合谋想要得到集合 X_i 中的元素,这与情形(2)中的情况类似,和借助于理想模型的结果一样.当 $P_{\bar{I}}$ 中有两个元素时,不妨令除 P_i,P_{i+1} 外的云服务器合谋想要得到集合 X_i,X_{i+1} 中的元素,他们最终只能得到关于向量 U_i,U_{i+1} 的一个不定方程:

$$U'=U_1\ldots U_i\cdot U_{i+1}\ldots U_n.$$

因此得不到向量 U_i,U_{i+1} 各个分量的值,即,不能得到关于 P_i,P_{i+1} 集合 X_i,X_{i+1} 中的元素.当 $P_{\bar{I}}$ 中的元素多于两个时,情况与有两个元素类似.因此,云服务器集合 P_I 合谋得不到 $P_{\bar{I}}$ 中云服务器的元素.所以,该情况下也存在概率多项式时间算法 S(根据情形(1)容易构造 S,在此省略具体的构造过程)对于 I 使得下式成立:

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\Pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

3 保密计算集合并集问题的高效解决方案

上述方案的计算复杂性与集合 U 中元素的个数 m 线性相关,如果 m 较大,加密、解密的次数与通信量将会比较大,不便于计算.如果 m 不太大,在协议 1 的基础上,根据哥德尔编码将向量对应成一个唯一自然数的方法设计了一种高效的编码方法,使协议的计算复杂性不再与集合 U 中元素的个数 m 相关,降低了计算的复杂性.

3.1 高效编码方法基本原理

将集合 $U=\{u_1,\ldots,u_m\}$(其中,$u_1<\ldots<u_m$)对应成集合 $P=\{p_1,\ldots,p_m\}$,其中,$u_j(j=1,\ldots,m)$对应于 $p_j(j=1,\ldots,m)$,$p_1,\ldots,p_m$ 是 m 个互不相等的素数.为了便于计算,取从 2 开始的 m 个连续素数.对于集合 $X_i\subset U$,将其

按照 0-r 编码的方法编码成 0-r 向量 $U_i=(u_{i1},\ldots,u_{im})$.如果 $u_j\in X_i(j=1,\ldots,m)$,则 $u_{ij}=r_{ij}$,其中,$r_{ij}\in[1,m]$是一个随机数且 $r_{ij}\neq 0(r_{ij}\neq 0$ 是为了将集合 X_i 中的元素与集合 $\overline{X}_i=U-X_i$ 中的元素区分,但为了防止根据 $r_{1j}+\ldots+r_{nj}$ 推算出拥有元素 u_j 集合的数目,r_{ij} 必须是一个随机数;如果 $r_{ij}\in Z_p^*$,将很可能因为编码成的数过大而得不到正确的计算结果,因此令 r_{ij} 取自较小的范围[1,m]);否则,$u_{ij}=0$.然后,将向量 U_i 借助集合 P 利用哥德尔编码的方法编码成一个自然数 x_i,即:

$$[u_{i1},\ldots,u_{im}]=p_1^{u_{i1}}\ldots p_m^{u_{im}}=x_i.$$

假设有 n 个集合 $X_1,\ldots,X_n\subset U$,将每个集合 $X_i(i=1,\ldots,n)$都按照高效编码方法编码成一个自然数 x_i,将这 n 个自然数相乘得到 x,即:

$$x=x_1\ldots x_n=p_1^{u_{11}}\ldots p_m^{u_{1m}}\ldots p_1^{u_{n1}}\ldots p_m^{u_{nm}}=p_1^{u_{11}+\ldots+u_{n1}}\ldots p_m^{u_{1m}+\ldots+u_{nm}}=p_1^{u_1'}\ldots p_m^{u_m'}.$$

根据算术基本定理将 x 展开,得到集合 $U'=\{u_1',\ldots,u_m'\}$.如果 $u_j'=u_{1j}+\ldots+u_{nj}\neq 0(j=1,\ldots,m)$,则 $u_{1j},\ldots,u_{nj}$ 中至少有一个不为 0,说明 n 个集合中至少有一个集合含有 u_j.根据向量 U'中不为 0 的分量,可以得到 $X_1\cup\ldots\cup X_n=\{\sigma_1,\ldots,\sigma_h\}(1\leqslant h\leqslant m)$.因此,保密计算集合并集的问题可以归约到保密地计算 x.以下是借助于高效编码方法计算集合并集的具体实例.

实例 2. 为了便于理解,仍以实例 1 中的数据为例进行说明.集合 U={101,102,103,104,105,106,107,108,109,110}对应于集合 P={2,3,5,7,11,13,17,19,23,31},假设要计算 3 个集合 X_1={101,105,107},X_2={103,105,108},X_3={104,106,109}$\subset U$ 的并集,根据 0-r 编码方法将集合 X_1,X_2,X_3 分别编码成 0-r 向量 U_1,U_2,U_3,即:

$$U_1=(3,0,0,0,2,0,1,0,0,0),$$
$$U_2=(0,0,2,0,1,0,0,2,0,0),$$
$$U_3=(0,0,0,2,0,1,0,0,1,0).$$

将向量 U_1,U_2,U_3 借助于集合 P 利用哥德尔编码的方法编码成自然数 x_1,x_2,x_3,即:

$$x_1=[3,0,0,0,2,0,1,0,0,0]=2^3\cdot 11^2\cdot 17^1=16456,$$
$$x_2=[0,0,2,0,1,0,0,2,0,0]=5^2\cdot 11^1\cdot 19^2=99275,$$
$$x_3=[0,0,0,2,0,1,0,0,1,0]=7^2\cdot 13^1\cdot 23^1=14651.$$

将这 3 个自然数相乘得到:

$$x=x_1\cdot x_2\cdot x_3=23934890379400=2^3\cdot 3^0\cdot 5^2\cdot 7^2\cdot 11^3\cdot 13\cdot 17\cdot 19^2\cdot 23^1\cdot 29^0.$$

即得到向量 U'=(3,0,2,2,3,1,1,2,1,0).根据向量 U'中不为 0 的分量得到:

$$X_1\cup X_2\cup X_3=\{101,103,104,105,106,107,108,109\}.$$

3.2 具体方案

云服务器 P_n 给 ElGamal 公钥加密算法的 KeyGen 算法输入安全参数λ,输出相应的系统参数 $params=(p,g)$,私钥 $sk=(x)$与对应的公钥 $pk=h(h=g^x \bmod p)$.P_n 将 pk,$params$ 公布并保留 sk.

每个云服务器 $P_i(i=1,\ldots,n)$将私有集合 X_i 借助高效编码方法编码成自然数 x_i,把 x_i 用 P_n 公布的 pk,$params$ 加密为 $E(x_i)$.如果直接将 $E(x_i)$发送给 P_n 是不安全的,P_n 可以通过解密将 x_i 恢复,因此需要借助其他方法混淆密文与云服务器的对应关系.P_i 将 $E(x_i)$随机地分成非零的 $k_i(k_i\leqslant n)$份(k_i 的值是不确定的且其他云服务器没有任何信息) $E(x_i)_1,\ldots,E(x_i)_{k_i}$,为了保证 ElGamal 公钥加密算法正常解密,需要使 $E(x_i)_1\ldots E(x_i)_{k_i}=E(x_i)$ 成立.将 $E(x_i)$分成 k_i 份,只需要选择 k_i 个随机数 $r_{i1},\ldots,r_{ik_i}$,它们满足 $r_{i1}\ldots r_{ik_i}=1 \bmod p$,然后从 k_i 个随机数中选择 1 个随机数如 r_{i1} 与 $E(x_i)$相乘构成 $E(x_i)$的一份份额如 $E(x_i)_1=r_{i1}\cdot E(x_i)$,用 $r_{i2},\ldots,r_{ik_i}$ 分别表示 $E(x_i)_2,\ldots,E(x_i)_{k_i}$,则:

$$E(x_i)_1\ldots E(x_i)_{k_i}=r_{i1}\cdot E(x_i)\cdot r_2\ldots r_{ik_i}=E(x_i).$$

P_i 将 $r_{i1}\cdot E(x_i),r_{i2},\ldots,r_{ik_i}$ 分别发送给 n 个云服务器中的 k_i 个.这 k_i 个云服务器可以包括 P_i 自己也可以不包括,具体发给了哪 k_i 个云服务器,其他云服务器没有任何信息.假设敌手已经获得关于 $E(x_i)$的 k_i-1 份份额,不妨令敌手得到的密文份额为 $E(x_i)_1,\ldots,E(x_i)_{k_i-1}$,因为 $E(x_i)_1\ldots E(x_i)_{k_i-1}\cdot E(x_i)_{k_i}=E(x_i)$,敌手不知道方程中的 $E(x_i),E(x_i)_{k_i}$,

得到的是有两个未知数的不定方程,即使敌手拥有私钥 sk 可以解密 $E(x_i)_1,...,E(x_i)_{k_i-1}$,也无法得到有关 x_i 的信息,因此,每个云服务器将密文分成 k_i 份发送给 n 个云服务器中的 k_i 个是安全的.P_i 将密文分发后,也会收到其他云服务器发送过来的密文.P_i 对收到的所有密文做 Evaluate 运算,构成新的密文 $E(x_i')$ 并发送给 P_n.

云服务器 P_n 把收到的所有密文做 Evaluate 运算,得到 $E(x)$.由于协议中的每个云服务器都是半诚实参与者,在协议执行过程中会严格按照协议要求执行协议,不会加入其他信息,所以 P_n 构造的 $E(x)$是所有云服务器秘密数的密文之积,即 $E(x)=E(x_i)\dots E(x_n)$.P_n 用私钥 sk 把 $E(x)$解密得到明文 x,然后将 x 用算术基本定理展开,得到 $x=p_1^{u_1'}...p_m^{u_m'}$,即,向量 $U'=(u_1',...,u_m')$.P_n 根据 U'中不为 0 的分量与集合 U 的关系得到集合$\{\sigma_1,\dots,\sigma_h\}\subset U(1\leqslant h\leqslant m)$,并将其公布.$n$ 个云服务器得到了保密计算集合并集的结果.在此基础上,给出了高效的保密计算集合并集协议,具体如协议 2.

协议 2. 高效的保密计算集合并集.

输入:$P_1,\dots,P_n$ 各自的秘密集合 $X_1,\dots,X_n\subset U=\{u_1,\dots,u_m\}$;

输出:$X_1\cup\dots\cup X_n=\{\sigma_1,\dots,\sigma_h\}(1\leqslant h\leqslant m)$.

1. 云服务器 P_n 保留私钥 sk 公布公钥 pk 与系统参数 $params$;
2. 每个云服务器 $P_i(i=1,\dots,n)$计算如下:
   (1) 根据高效编码方法将自己的秘密集合 X_i 编码成自然数 x_i;
   (2) 用公钥 pk 加密 x_i 为 $E(x_i)$;
   (3) 把 $E(x_i)$随机地分成 k_i 份并发送给 n 个云服务器中的 k_i 个;
   (4) 将收到的所有密文相乘得到新的密文 $E(x_i')$,并发送给 P_n;
3. 云服务器 P_n 计算如下:
   (1) 把所有收到的密文相乘,得到 $E(x)$;
   (2) 用私钥 sk 解密 $E(x)$得到 x;
   (3) 将 x 用算术基本定理展开得到向量 U';
   (4) 根据 U'中不为 0 的分量得到集合$\{\sigma_1,\dots,\sigma_h\}$并公布.

根据方案的计算方法可知:得到的并集$\{\sigma_1,\dots,\sigma_h\}$中的元素是严格递增的,即,$\sigma_1<\dots<\sigma_h$.因此,该协议还可以实现保密地计算多个云服务器集合中所有元素的排序.该协议与协议 1 经过简单的改造也适用于 w 个云用户借助 n 个半可信云服务器 $P_1,\dots,P_n$ 保密计算集合并集的问题,现以协议 2 为例作简单说明.假如有 w 个云用户 $V_1,\dots,V_w$,他们想要借助半可信的云保密计算集合的并集.每个云用户 V_i 用高效编码方法把自己的秘密集合转化成秘密数 x_i,并将 x_i 随机地分成 k_i 份发送给 n 个云服务器中的 k_i 个.这 n 个云服务器将收到的秘密份额相乘并加密,然后将密文发送给 P_n.P_n 将所有的密文相乘并解密,得到 x,从而得 w 个云用户集合的并集$\{\sigma_1,\dots,\sigma_h\}$.虽然需要 n 个云服务器合作计算 w 个云用户的并集,但是计算出云用户的秘密集合需要得到相应秘密数的全部信息,而每个云服务器没有任何一个云用户秘密向量的全部信息,因此得不到云用户的信息,从而可以得到 w 个云用户的并集.

3.3 方案分析

我们采用模拟范例的方法证明了协议 2 的安全性,具体如定理 2.

定理 2. 高效的保密计算集合并集协议 2(记为 π)是保密的.

证明:分 3 种情况证明 π的安全性:

(1) 云服务器 P_n 不参与合谋,其他云服务器集合 $P_I\subseteq\{P_1,\dots,P_{n-1}\}$合谋想要得到云服务器 $P_i(P_i\notin P_I,i=1,\dots,n)$集合 X_i 中的元素.通过构造使下式成立的概率多项式时间模拟器 S 来证明上述情况的安全性:

$$\{S(X_I,f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_I^{\pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

S 的工作过程如下:

① 给定输入 $(X_I,f_I(\bar{X}))$,令:

$$(X_I, f_I(\bar{X})) = (X_I,...,X_{i-1},X_{i+1},...,X_{n-1}, f_I(\bar{X})),,$$

S 随机选择两个集合 X'_n, X'_i,使得 $f_I(\bar{X}) = f_I(\bar{X}')$,其中, $\bar{X}' = (X_1,...,X_{i-1},X'_i,X_{i+1},...,X_{n-1},X'_n)$.

② S 按照π中高效的编码方法将 $\bar{X}'$ 中的每个集合编码成自然数 $x_1,...,x_{i-1},x_i^*,x_{i+1},...,x_{n-1},x_n^*$,然后用 P_n 公布的 ElGamal 公钥加密算法的公钥与系统参数(*pk*,*params*)将 $x_1,...,x_{i-1},x_i^*,x_{i+1},...,x_{n-1},x_n^*$ 加密,得到:

$$E(x_1),...,E(x_{i-1}),E(x_i^*),E(x_{i+1}),...,E(x_{n-1}),E(x_n^*).$$

随后,将 n 份密文随机地分成 $k_1,\ldots,k_n$ 份,模拟π中的分发方法进行分发.

③ S 将各个密文随机地分发后对得到的密文做 Evaluate 运算,得到新的密文:

$$E(x_1^*),...,E(x_{i-1}^*),E(x_i^\circ),E(x_{i+1}^*),...,E(x_{n-1}^*),E(x_n^\circ).$$

然后,对所有密文做 Evaluate 运算,得到:

$$E(x') = E(x_1^*)...E(x_{i-1}^*)\cdot E(x_i^\circ)\cdot E(x_{i+1}^*)...E(x_{n-1}^*)\cdot E(x_n^\circ).$$

④ 由于 $P_n \notin P_I$,具有概率多项式时间计算能力的 S 没有 ElGamal 公钥加密算法的私钥,因此,S 不能对密文解密.

在π中:

$$\begin{aligned} view_I^\pi(\bar{X}) &= \{view_1^\pi(\bar{X}),...,view_{i-1}^\pi(\bar{X}),view_{i+1}^\pi(\bar{X}),...,view_{n-1}^\pi(\bar{X})\} \\ &= \{(X_1,...,X_{i-1},X_{i+1},...,X_{n-1}),(x_1,...,x_{i-1},x_{i+1},...,x_{n-1}), \\ &\quad (E(x_1),...,E(x_{i-1}),E(x_{i+1}),...,E(x_{n-1})), \\ &\quad (E(x'_1),...,E(x'_{i-1}),E(x'_{i+1}),...,E(x'_{n-1})),E(x),(\sigma_1,...,\sigma_h)\}, \end{aligned}$$

令:

$$\begin{aligned} S(X_I, f_I(\bar{X})) = \{&(X_1,...,X_{i-1},X_{i+1},...,X_{n-1}),(x_1,...,x_{i-1},x_{i+1},...,x_{n-1}), \\ &(E(x_1),...,E(x_{i-1}),E(x_{i+1}),...,E(x_{n-1})), \\ &(E(x_1^*),...,E(x_{i-1}^*),E(x_{i+1}^*),...,E(x_{n-1}^*)),E(x^*),(\sigma'_1,...,\sigma'_h)\}, \end{aligned}$$

因为 $(\sigma'_1,...,\sigma'_h) = (\sigma_1,...,\sigma_h)$. 由于 ElGamal 公钥加密算法是语义安全的,任何数据用语义安全的加密算法加密的结果是计算不可区分的,上述模拟过程中得到的消息序列与实际执行过程中得到的消息序列是计算不可区分的,即:

$$\{S(X_I, f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n} \overset{c}{\equiv} \{view_I^\pi(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

所以,π在该情况下是保密的.

(2) 云服务器 P_n 与其他云服务器集合 $P_{I'}\subseteq\{P_1,\ldots,P_{n-1}\}$合谋想要得到云服务器 $P_i(P_i\notin P_I=P_{I'}\cup P_n)$集合 X_i 中的元素.P_i 将其秘密集合 X_i 编码成 x_i 并加密为 $E(x_i)$,然后将 $E(x_i)$分成 k_i 份发送给 n 个云服务器中的 k_i 个,其他云服务器对于 P_i 将这 k_i 份密文发送给了哪 k_i 个云服务器没有任何信息.云服务器集合 P_I 不知道收到的关于 $E(x_i)$ 的份额是否是全部份额,因此,即使 P_I 将其收到的关于 $E(x_i)$的全部份额相应分量相乘并解密,根据解密后得到的明文也不能确定 x_i,从而不能得到关于 P_i 的机密集合 X_i.在协议执行后,除 P_i 外的其他全部云服务器合谋可能会得到关于 X_i 的信息,因为 P_n 与 $P_{I'}$根据$\{\sigma_1,\ldots,\sigma_h\}$与集合 $X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n$ 的关系得出关于 x_i 的信息,但是由于集合$\{\sigma_1,\ldots,\sigma_h\}$中不同元素只出现一次,因此其他云服务器只会得到集合 X_i 独有的元素,但这与借助于可信的第三方的理想模型的执行结果是一样的,不会泄露更多的信息.所以,该情况下存在概率多项式时间算法 S(S 根据情形(1)容易构造,在此省略其构造过程)对于 I 使得下式成立:

$$\{S(X_I, f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n} \overset{c}{\equiv} \{view_I^\pi(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}.$$

(3) 云服务器 P_n 与其他云服务器集合 $P_{I'}\subseteq\{P_1,\ldots,P_{n-1}\}$合谋,想要得到云服务器集合 $P_{\bar{I}}=\{P_1,...,P_n\}-P_I$ ($I=I'\cup n$)中云服务器集合的元素.当 $P_{\bar{I}}$ 中只有一个元素时,不妨令除 P_i 外的其他云服务器合谋想要得到集合 X_i 中的元素,这与情形(2)中的情况类似,与借助于理想模型的结果一样.当 $P_{\bar{I}}$ 中有两个元素时,不妨令除 P_i,P_{i+1} 外的云服务器合谋想要得到集合 X_i,X_{i+1} 的元素,最终,他们只能得到关于 x_i,x_{i+1} 的不定方程:$x=x_1,\ldots,x_i,x_{i+1},\ldots,x_n$,

因此得不到 x_i,x_{i+1} 的具体值,即,不能得到关于 P_i,P_{i+1} 的集合 X_i,X_{i+1} 中的元素.当 $P_{\bar{I}}$ 中的元素多于两个时,情况与有两个元素时类似.因此,云服务器集合 P_I 合谋得不到 $P_{\bar{I}}$ 中云服务器集合中的元素.所以,该情况下也存在概率多项式时间算法 S(根据情形(1)容易构造 S,在此省略具体的构造过程)对于 I 使得下式成立:

$$\{S(X_I, f_I(\bar{X}))\}_{\bar{X}\in(\{0,1\}^*)^n} \overset{c}{\equiv} \{view_I^{\pi}(\bar{X})\}_{\bar{X}\in(\{0,1\}^*)^n}. \quad \square$$

因为解决的问题不完全相同,要比较不同方案的计算复杂性是困难的.以上协议 1、协议 2 给出的方案均可以借助于半可信的云服务器进行保密地计算,不再需要做复杂的加密计算,效率较高.为了与现有协议进行比较,下面给出的均是在不借助半可信云服务器计算情况下的效率分析.

在协议 1 中,n 个云服务器中的每个云服务器都需要对其向量中的 m 个分量加密,共需要加密 $m\cdot n$ 次;最终,参与者 P_n 需要将经 Evaluate 运算后的密文向量 $E(U')$解密,需要解密 m 次.每个云服务器需要将自己的密文向量随机地分成 k 份发送给 k 个云服务器,需要传递 $m\cdot n\cdot k$ 个密文,每个云服务器将其收到的密文向量经 Evaluate 运算后发送给 P_n,需要传递 $m\cdot n$ 个密文,因此共需要传递 $n\cdot m\cdot(k+1)$个密文.

协议 2 中,n 个云服务器需要将其构造的机密数加密,共需要 n 次加密;最终,只需要云服务器 P_n 解密 x.每个云服务器需要将自己构造的密文分成 k 份发送给 k 个云服务器,需要传递 nk 个密文;随后,每个云服务器需要将其收到的密文做 Evaluate 运算,然后将计算后的密文发送给 P_n,需要传递 n 个密文;则共需要传递 $n(k+1)$个密文.在以往保密计算集合并集问题的解决方案中,即使不需要保密,其计算复杂度与通信复杂度也与云服务器拥有的元素个数相关.该方案的计算复杂度与通信复杂度仅与参与者的个数线性相关,而与云服务器集合中元素的个数无关,效率较高.

文献[21]保密计算集合并集的方案是借助全同态加密算法和多项式求值给出的,方案中需要计算 n^2l 次加密运算与 nl 次解密运算(l 表示 n 个参与者拥有元素的总数),参与者之间需要传递 n^2l 个密文,而现有的全同态加密算法的效率较低.

保密计算集合的并集问题协议比较见表 1,其中,E,D 分别表示本文中用到的 ElGamal 公钥加密算法加密、解密时的运算量,E_F,D_F 表示文献[21]中使用的全同态加密算法.

Table 1 Comparison of secure multiparty secure set union protocols

表 1 多个参与者保密计算集合并集协议的比较

	计算开销	通信开销
文献[21]的方案	$ln^2E_F+lnD_F$	n^2l
协议 1	$mnE+mD$	$mn(k+1)$
协议 2	$nE+D$	$n(k+1)$

4 保密计算集合的交集

$P_1,\ldots,P_n$ 分别拥有秘密集合 $X_1,\ldots,X_n\subset U=\{u_1,\ldots,u_m\}(u_1<\ldots<u_m)$,他们想要知道 $X=X_1\cap\ldots\cap X_n$ 而不想泄露 $\{(P_1,X_1),\ldots,(P_n,X_n)\}$.保密求集合的交集是隐私保护中一个非常重要的问题,在隐私保护的商业服务、隐私保护的数据挖掘等方面有着重要的应用,但现有的方案效率比较低,且不适用于云计算环境.本文给出一种高效保密计算集合交集的解决方案,该方案经过简单改造也可以应用于云安全计算环境,此处仅给出一般情况下保密求集合交集的方案,(见如协议 3).

协议 3. 保密计算集合交集.

输入:$P_1,\ldots,P_n$ 各自的秘密集合 $X_1,\ldots,X_n\subset U=\{u_1,\ldots,u_m\}$;

输出:$X=X_1\cap\ldots\cap X_n$.

1. P_n 保留私钥 sk,将公钥 pk,系统参数 $params$ 和素数集合 $P=\{p_1,\ldots,p_n\}$公布;
2. 每个参与者 $P_i(i=1,\ldots,n)$计算如下:
   (1) 将秘密集合 X_i 编码为 0-r 向量 U_i(集合 X_i 中存在的元素在向量 U_i 的对应位置编码成 0,其他位置编码成一个随机数);

(2) 借助哥德尔编码将向量 U_i 编码成自然数 x_i;

(3) 用 P_n 的公钥 pk 加密 x_i 为 $E(x_i)$,将 $E(x_i)$随机地分成 k_i 份发送给 n 个参与者中的 k_i 个;

(4) 对收到的所有密文做 Evaluate 运算得到新的密文 $E(x_i')$ 并发送给 P_n;

3. P_n 计算如下:

(1) 对所有收到的密文做 Evaluate 运算,得到 $E(x)$;

(2) 用私钥 sk 解密 $E(x)$得到 x,然后根据算术基本定理展开得到向量 U';

(3) 根据 U'中 0 的分量得到交集 $X=X_1\cap\ldots\cap X_n$;

(4) 将 X 公布.

协议 3 中保密计算集合交集的方案是在协议 2 的基础上经过简单改造得来的,即:将原本向量 U_i 中取 0 的位置改为随机数,而将原本取随机数的位置改为 0,并根据 U'中 0 的分量得 n 个集合的交集.在协议 1 中,将原本取 1(0)的位置改为随机数,原本取随机数的位置改为 1(0).最后,根据 U'中 1(0)的分量也可以得 n 个集合的交集.

推论. 保密计算集合交集协议是保密的.

根据定理 2 的安全性证明,该推论的安全性很容易证明,在此省略证明过程.

5 结 论

集合隐私计算是许多保密计算的基础,具有广泛的应用.本文给出的云计算环境下的集合隐私计算方案包括多个集合并集的保密计算方案与多个集合交集的保密计算方案.本文首先在云计算环境下设计了基于 1-r 编码方法与同态加密算法的集合计算方案,该方案效率较高且适用于任何全序关系对象的集合计算.随后,又借助 0-r 编码方法和哥德尔编码的思想设计了一个高效编码方法,根据高效编码方法给出了一种高效的云集合并集保密计算方案.利用 1-r(0-r)编码方法,避免了进行两两计算所导致的信息泄露,也可以借助于半可信的云实现保密计算,而不泄露参与者的信息,降低了集合计算的计算复杂性,特别是可以进行预处理,大大降低了在线计算量.如果将生成的并集按照从大到小的顺序排列,这样的协议还可以保密地对多个集合的所有元素进行排序.

本文构造的云计算环境下的集合隐私计算方案均是属于有限范围内的.对于输入范围未知的集合隐私计算问题,参与者可以将拥有集合中的元素划分为不同的所属范围,然后再按照本文的方法对不同范围内的元素进行保密计算,进而得到参与者集合的隐私计算.但该方法对参与者集合中元素的信息有较多的泄露,因此,研究云计算环境下无限范围内的集合隐私计算问题是我们进一步研究的问题.

References:

[1] Feng DG, Zhang M, Zhang Y, Xu Z. Study on cloud computing security. Ruan Jian Xue Bao/Journal of Software, 2011,22(1): 71−83 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/SP.J.1001.2011. 03958]

[2] Yao AC. Protocols for secure computations. In: Shamir A, ed. Proc. of the 23rd IEEE Symp. on Foundations of Computer Science. Piscataway: IEEE Press, 1982. 160−164. [doi: 10.1109/SFCS.1982.38]

[3] Goldreich O, Micali S, Wigderson A. How to play any mental game. In: Aho AV, ed. Proc. of the 19th Annual ACM Conf. on Theory of Computing. Piscataway: IEEE Press, 1987. 218−229. [doi: 10.1145/28395.28420]

[4] Goldreich O. Foundations of Cryptography: Volume 2, Basic Applications. London: Cambridge University Press, 2004.

[5] Ioannidis I, Grama A. An efficient protocol for Yao's millionaires' problem. In: Kauffman RJ, ed. Proc. of the 36th Hawaii Int'l Conf. on System Science. Piscataway: IEEE Press, 2003. 1−6. [doi: 10.1109/HICSS.2003.1174464]

[6] Li SD, Wang DS. Efficient secure multiparty computation based on homomorphic encryption. Dian Zi Xue Bao/Chinese Journal of Electronics, 2013,41(4):798−803 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2013.04.029]

[7] Li SD, Wu CY, Wang DS, Dai YQ. Secure multiparty computation of solid geometric problems and their applications. Information Sciences, 2014,282:401−413. [doi: 10.1016/j.ins.2014.04.004]

[8] Fagin R, Naor M, Winklery P. Comparing information without leaking it. Communications of the ACM, 1996,39(5):77−85. [doi: 10.1145/229459.229469]

[9] Du WL, Atallah MJ. Privacy-Preserving cooperative statistical analysis. In: Proc. of the 17th Annual Conf. of Computer Security Applications. Piscataway: IEEE Press, 2001. 102−110. [doi: 10.1109/ACSAC.2001.991526]

[10] Du WL, Atallah MJ. Protocols for secure remote database access with approximate matching. In: Ghosh AK, ed. Proc. of the Advance of E-Commerce and Privacy. New York: Springer-Verlag, 2001. 87−111. [doi: 10.1007/978-1-4615-1467-1_6]

[11] Cachin C. Efficient private bidding and auctions with a noblivious third party. In: Motiwalla J, ed. Proc. of the 6th ACM Conf. on Computer and Communications Security. New York: ACM Press, 1999. 120−127. [doi: 10.1145/319709.319726]

[12] Koh HC, Tan G. Data mining applications in healthcare. Journal of Healthcare Information Management, 2011,19(2):64−72.

[13] Freedman MJ, Nissim K, Pinkas B. Efficient private matching and set intersection. In: Cachin C, Camenisch JL, eds. Proc. of the Advances in Cryptology (EUROCRYPT 2004). Berlin, Heidelberg: Springer-Verlag, 2004. 1−19. [doi: 10.1007/978-3-540-24676-3_1]

[14] Hazay C, Lindell Y. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. Journal of Cryptology, 2010,23(3):422−456. [doi: 10.1007/s00145-008-9034-x]

[15] Fischlin M, Pinkas B, Sadeghi AR, Schneider T, Visconti I. Secure set intersection with untrusted hardware tokens. In: Kiayias A, ed. Proc. of the 11th Int'l Conf. on Topics in Cryptology (CT-RSA 2011). Berlin, Heidelberg: Springer-Verlag, 2011. 1−16. [doi: 10.1007/978-3-642-1907 4-2_1]

[16] Shi R, Mu Y, Zhong H, Cui J, Zhang S. An efficient quantum scheme for private set intersection. Quantum Information Processing, 2016,15(1):363−371. [doi: 10.1007/s11128-015-1165-z]

[17] Kissner L, Song D. Privacy-Preserving set operations. In: Shoup V, ed. Proc. of the 25th Annual Int'l Conf. on Advances in Cryptology. Berlin, Heidelberg: Springer-Verlag, 2005. 241−257. [doi: 10.1007/11535218_15]

[18] Frikken K. Privacy-Preserving set union. In: Katz J, Yung M, eds. Proc. of the 5th Int'l Conf. on Applied Cryptography and Network Security. Berlin, Heidelberg: Springer-Verlag, 2007. 237−252. [doi: 10.1007/978-3-540-72738-5_16]

[19] Blanton M, Aguiar E. Private and oblivious set and multiset operations. In: Youm HY, Kisa YW, eds. Proc. of the 7th ACM Symp. on Information, Computer and Communications Security. New York: ACM Press, 2012. 40−41. [doi: 10.1145/2414456.2414479]

[20] Seo J, Cheon J, Katz J. Constant-Round multi-party private set union using reversed laurent series. In: Fischlin M, ed. Proc. of the 15th Int'l Conf. on Practice and Theory in Public Key Cryptography (PKC 2012). Berlin, Heidelberg: Springer-Verlag, 2012. 398−412. [doi: 10.1007/978-3-642-30057-8_24]

[21] Chun JY, Hong D, Jeong IR, Lee DH. Privacy-Preserving disjunctive normal form operations on distributed sets. Information Sciences, 2013,231:113−122. [doi: 10.1016/j.ins.2011.07.003]

[22] Tillmanns J. Privately computing set-union and set-intersection cardinality via bloom filters. In: Foo E, Stebila D, eds. Proc. of the 20th Australasian Conf. on Information Security and Privacy (ACISP 2015), Vol.9144. Berlin, Heidelberg: Springer-Verlag, 2015. 413−430. [doi: 10.1007/978-3-319-19962-724]

[23] De Cristofaro E, Gasti P, Tsudik G. Fast and private computation of cardinality of set intersection and union. In: Pieprzyk J, ed. Proc. of the Cryptology and Network Security. Berlin, Heidelberg: Springer-Verlag, 2012. 7712: 218−231. [doi: 10.1007/978-3-642-35404-5_17]

[24] Wang G, Luo T, Goodrich MT, Du W, Zhu Z. Bureaucratic protocols for secure two-party sorting, selection, and permuting. In: Feng D, ed. Proc. of the 5th ACM Symp. on Information, Computer and Communications Security. New York: ACM Press, 2010. 226−237. [doi: 10.1145/1755688.1755716]

[25] Bellare M, Hoang VT, Rogaway P. Foundations of garbled circuits. In: Yu T, ed. Proc. of the 2012 ACM Conf. on Computer and Communications Security. New York: ACM Press, 2012. 784−796. [doi: 10.1145/2382196.2382279]

[26] Maheshwari N, Kiyawat K. Structural framing of protocol for secure multiparty cloud computation. In: Proc. of the 2011 5th Asia Modelling Symp. Piscataway: IEEE Press, 2011. 187−192. [doi: 10.1109/AMS.2011.42]

[27] Paillier P. Public-Key cryptosystems based on composite degree residuosity classes. In: Stern J, ed. Proc. of the Advances in Cryptology—EUROC (RYPT'99). Berlin, Heidelberg: Springer-Verlag, 1999. 223−238. [doi: 10.1007/3-540-48910-X_16]

附中文参考文献:

[1] 冯登国,张敏,张妍,徐震.云计算安全研究.软件学报,2011,22(1):71−83. http://www.jos.org.cn/1000-9825/3958.htm [doi: 10.3724/SP.J.1001.2011.03958]

[6] 李顺东,王道顺.基于同态加密的高效多方保密计算.电子学报,2013,42(4):798−803. [doi: 10.3969/j.issn.0372-2112.2013.04.029]

李顺东(1963－),男,河南鲁山人,博士,教授,博士生导师,主要研究领域为密码学,信息安全.

周素芳(1990－),女,硕士生,主要研究领域为信息安全.

郭奕旻(1992－),女,硕士,主要研究领域为信息安全.

窦家维(1963－),女,博士,副教授,主要研究领域为应用数学,应用密码学.

王道顺(1964－),男,博士,副教授,主要研究领域为密钥管理,数字水印,多媒体安全.