软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(7):1893−1908 [doi: 10.13328/j.cnki.jos.005358]

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

基于字符串排序的高效保密数据库查询*

李顺东[1], 亢 佳[1], 杨晓艺[1], 窦家维[2]

[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)

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

摘 要: 安全多方计算是近年来国际密码学界研究的热点问题之一,是信息社会隐私保护的核心技术.保密地将字符串按照字典序排序问题是一个全新的安全多方计算问题,在信息安全领域有重要的实际意义和广泛的应用前景.它不仅可以提高保密数据库查询的效率,还可以解决大数据情况下的百万富翁问题.为了保密地判断两个字符串按照字典序排序的位置关系,首先设计了一种新的编码方法和一种基于 ElGamal 加密算法的云外包计算下的同态加密方案,在此基础上提出了一个高效、简单的协议,并对协议进行了正确性和安全性分析,同时给出了协议计算复杂性和通信复杂性的理论分析与实验验证.最后将保密的字符串排序问题协议应用于解决百万富翁问题,从根本上解决了大数据情况下的百万富翁问题.

关键词: 密码学;安全多方计算;字符串排序;数据库保密查询;同态加密;百万富翁问题

中图法分类号: TP309

中文引用格式: 李顺东,亢佳,杨晓艺,窦家维.基于字符串排序的高效保密数据库查询.软件学报,2018,29(7):1893−1908. http://www.jos.org.cn/1000-9825/5358.htm

英文引用格式: Li SD, Kang J, Yang XY, Dou JW. String sorting based efficient secure database query. Ruan Jian Xue Bao/ Journal of Software, 2018,28(7):1893−1908 (in Chinese). http://www.jos.org.cn/1000-9825/5358.htm

String Sorting Based Efficient Secure Database Query

LI Shun-Dong[1], KANG Jia[1], YANG Xiao-Yi[1], DOU Jia-Wei[2]

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

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

Abstract: Secure multiparty computation is a research focus in the international cryptographic community and a pivotal privacy preserving technology in cyberspaces. Privacy-Preserving lexicographical string sorting, as a completely new problem of secure multi-party computation, has important practical significance and broad application prospects in information security. It can not only improve the efficiency of secure database query but also solve the millionaires' problem in the case that the numbers to be compared are very large. In this study, to privately determine the lexicographical order of two private strings, an encoding scheme is first proposed to encode private numbers, then based on a homomorphic encryption algorithm supported by cloud computing outsourcing, a simple and efficient protocol is developed. Furthermore, a proof is provided to shoe that this protocol is secure in the semi-honest model, and its correctness is also analyzed. At the same time, the computational complexities and communication complexities of the protocol are analyzed, and the efficiency of the protocol on a PC is demonstrated. Finally, as a fundamental solution, the scheme is applied to solve the millionaires' problem in the case that the numbers to be compared are very large.

* 基金项目: 国家自然科学基金(61272435)
Foundation item: National Natural Science Foundation of China (61272435)
本文由"面向隐私保护的新型技术与密码算法"专题特约编辑彭长根教授推荐.
收稿时间: 2017-05-29; 修改时间: 2017-07-13; 采用时间: 2017-08-22; jos 在线出版时间: 2017-10-17
CNKI 网络优先出版: 2017-10-17 13:38:03, http://kns.cnki.net/kcms/detail/11.2560.TP.20171017.1338.005.html

Key words: cryptography; secure multi-party computation; string sorting; secure database query; homomorphic encryption; millionaires' problem

在现代社会,信息与人们的生活密切相关,无处不在.人们的衣食住行,人与人之间的沟通交往,企业的经营管理...都需要信息,社会的发展也依赖于信息.数据库作为现代信息系统的核心部分,存储和管理大量的重要信息,通过数据库查询的方式能够帮助人们快捷、准确、全面地获取所需信息.数据库包含了大量敏感信息,特别是政府部门和个人的敏感信息,如用户个人信息、政府要员人事资料和其他关键信息资源,这些重要信息在进行数据库查询的时候很容易泄露.所以,如何保密地查询数据库并获得所需要的信息非常关键.

所谓保密的数据库查询就是在数据库查询过程中,数据库不知道用户查询的是哪一条信息,同时用户只能得到自己需要的查询结果,而不知道数据库中的其他信息.例如:情报工作者为了得到某一政府要员的个人资料,想要查询政府机关所拥有的数据库,要查的政府要员属于敏感信息,不能泄露给政府机关,同时,对于情报工作者不查询的记录数据库应该尽可能防止信息的泄露.保密的数据库查询问题是安全多方计算在数据库中的重要应用,在安全多方计算领域中有着广阔的应用前景和实用价值.

然而,现有的数据库保密查询方法虽然能够有效地保护用户和数据库的隐私信息,但却存在花费高、效率低、查准率低的情况.因为现在的保密查询基本原理是将要查询的记录与数据库中的所有记录都进行加密,然后利用查询记录的密文与数据库中所有记录的密文进行有关的查询操作.所以,当数据库的记录很多时,保密查询的计算复杂性很高,查询效率很低.如何能从数据库中更加快速、准确地查询到保密数据呢?在提到的例子中:情报工作者获得一份名单(名单保密,只有自己知道),需要到保密数据库中查询名单上人物的详细资料,数据库管理员选择数据库中间的一条保密记录 m 与名单上的一个名字 n 进行比对.如果名单上名字 n 排序在保密记录 m 之前,那么直接让这个名字 n 与数据库中 m 之前的记录进行比较;如果名单上名字 n 排序在保密记录 m 之后,那么直接让名字 n 与数据库中 m 之后的记录进行比较.以此类推,直到查询到的记录很少时,调用现有的保密查询协议进行查询,可以大大提高效率,而且不泄露私密信息.用这种"二分法"来进行保密数据的查询,效率将会大大提高.所以,如何确定名单上名字 n 和数据库中记录 m 的排序关系将是提高效率的关键.上述保密高效查询问题的关键是设计出高效的字符串保密排序协议.本文的主要工作就是设计这样的协议.

安全多方计算(secure multi-party computation,简称 SMC)作为隐私保护和信息安全的关键技术,已成为国内外密码学界的研究热点之一[1−8].安全多方计算保证在不泄露参与者私有数据的前提下,使这些私有数据能够被参与者利用,并进行保密的合作计算,从而使人们能够最大限度地利用私有数据,而不破坏数据的保密性.安全多方计算问题由 Yao[9]首先提出,Goldreich[10]等人对其进行了深入的研究,形成了安全多方计算问题的理论基础[11−13],推动了安全多方计算的发展.

安全多方计算研究在计算科学中占有重要的地位.很多学者致力于安全多方计算问题的研究,提出了各种安全多方计算问题及其解决方案.所研究问题的方向可以总结为以下几类:(1) 保密的科学计算问题[14−23];(2) 保密的计算几何问题[24−27];(3) 保密的求集合交集、并集问题;(4) 保密的数据挖掘问题[28];(5) 保密的数据库查询问题;(6) 其他安全多方计算问题.

保密的排序问题是保密比较大小问题的自然推广,与保密比较大小问题一样都是保密的科学计算中的重要问题,已经得到了很多研究,但是目前对保密的排序问题方面的研究仍然处于初始阶段,仅限于数据的比较,例如保密地判断两个字符串按照字典序排序的位置问题还未得以研究.字符串保密排序问题是一个新的安全多方计算问题,具体可以描述为将两个参与者分别拥有的字符串按照字典序保密排序,而不泄露关于双方字符串的任何信息.保密判断两个字符串的排序关系对日常生活中问题的解决有着很大的实用意义,不仅可以提高数据库查询的效率,更是在保密的基因测序、基因序列匹配、网络安全、拍卖、招标等方面有着广泛的应用前景.根据字符串保密排序的含义可知,字符串保密排序问题是百万富翁问题的推广.理论上比较两个数的大小和字符串排序本质相同,因此可以用百万富翁问题协议解决字符串排序问题,但也存在一些问题需要解决.例如,假设用 01,02,…,26 表示字母 $a,b,\ldots,z$,如果按照字典排序,字符串 d 应该排在 ab 的后面,但如果用百万富翁协议

来比较 04 和 0102,显然,04 应该排在前面.只有两个字符串的长度相同时才能比较,或者把两个字符串填充成同样长度,但这样会增加计算量,也会泄露一些信息.目前百万富翁问题的研究情况如下.

最早由 Yao 在文献[9]中应用电路解决了百万富翁问题,但该方案的计算复杂性和通信复杂性都很高.当要比较的两个数范围很大时,需要重复调用协议,效率极低.文献[14,15]中设计了基于不经意传输的百万富翁协议,不仅需要并行调用多次不经意传输,而且不能一次性解决大范围数据情况下的百万富翁问题,需要多次重复调用基本协议,计算复杂性高.文献[16]设计新的 0-1 编码方案,将百万富翁问题归约到向量的部分标量积的计算,然后利用 Paillier 加法同态加密体制,构造了一个高效的百万富翁协议.在假定保密输入且 $|U|=m$ 的情况下,该协议的计算复杂性为 $O(m)$.但是该协议也必须通过多次调用基本协议来分段解决大数据情况下的百万富翁问题,而且该协议只能判断两个保密数的关系是大于还是小于等于,不能完全区分小于和等于的关系.文献[17–20]没有考虑算法的简洁性和通用性,同样也存在重复多次调用协议来解决大范围数据的比较大小问题.Li 等人[21,22]设计了基于对称密码的不采用模指数运算的百万富翁协议,将百万富翁问题转化成一个集合包含问题,极大地降低了协议的时间复杂度,但是该协议仅仅适用于所比较的两个数相差不是很大(都属于同一个范围较小的全集内)的情况.当两个数范围比较大时,无法进行有效的比较.文献[23]虽然可以解决大数据情况下的百万富翁问题,但是协议是将所要比较的保密数统一成相同位数的二进制数,然后再利用新的 0-1 编码,借助 ElGamal 加密算法,将百万富翁问题归约到一个集合相交问题,从而使其得到解决.整个过程计算复杂性仍然较高,而且无法完全区分小于和等于的关系.

综上所述,现有的协议在保密比较两个大数据大小时,可能会出现以下情形:(1) 不能直接比较大范围数据,需要重复多次调用协议,导致计算复杂性很高.(2) 即使没有数据长度的限制,也不能完全区分数据小于等于的关系.所以,目前尚未有保密比较两个大数据大小问题的高效解决方案.字符串的排序与两个数大小的比较不完全等价,如果将两个要进行比较的字符串转化成数字,一般都是很大的数字,而且在比较时必须把转化成的数字通过在低位补零使两个数字位数相同(否则将无法比较),然后才能调用百万富翁问题协议来解决.这个过程进一步增加了计算复杂性而且泄露了一些信息.如果能够直接解决字符串的排序问题,也就可以解决大数字的百万富翁问题.所以研究字符串保密排序问题显得很有意义.

本文主要研究两个字符串保密排序的问题,首先设计了一种新的编码方式,利用这种编码可以将字符串的排序问题转化成集合元素判定问题(集合中是否存在元素 1,2,随机数),提出了一个高效的保密判断字符串排序的协议,这个协议实现过程的主要计算可以借助于云服务器来完成.并且在此基础上设计了可以一次性解决大范围数据情况下的百万富翁问题的协议,该协议不需要多次重复调用基本协议,同时也能够完全区分两个数的 3 种关系.

本文贡献主要如下:

(1) 针对字符串排序问题设计了一种全新的编码方法,该编码方法将字符串的排序问题转化成集合元素判定问题(集合中是否存在元素 1,2,随机数),也为解决大数据情况下的百万富翁问题提供了新思路,达到了简化问题、降低计算复杂性的目的.

(2) 针对我们的问题提出了将主要 ElGamal 加密运算外包给云服务器的方案,此加密方案在预处理阶段由云服务器计算 $R_i=(g^{r_i} \bmod p, h^{r_i} \bmod p)$,加密时通过对 R_i 执行几次简单的模乘运算,避免了原 ElGamal 加密算法加密过程中复杂的模指数运算.

(3) 基于云外包计算下的同态加密方案和创新编码方法,设计了保密地判断两个字符串位置关系的协议,并对其协议的安全性和正确性进行证明.

(4) 在保密地判断两个字符串按照字典序排序位置关系协议的基础上,提出了保密解决大数据情况下的百万富翁问题的协议.同时,利用字符串保密排序协议提出的将保密的字符串构造为编码表的方法,也解决了字符串模式匹配的问题,扩大了安全多方计算的研究领域和实际应用范围.

本文第 1 节介绍我们提出的协议中需要用到的预备知识和安全性定义.第 2 节基于一种新的编码方法和一种将 ElGamal 加密运算外包给云服务器的方法设计具体的协议解决字符串保密排序问题,并对协议的正确性

和安全性进行分析,同时利用模拟实验对协议的复杂性和效率进行分析.第 3 节提出保密解决大数据情况下的百万富翁问题的协议和保密地判断字符串模式匹配问题的协议,并对协议进行效率分析.第 4 节对文章进行总结.

1 预备知识

1.1 ElGamal同态加密方案

同态加密的概念在文献[29]中被首次提出,其保证在不影响明文数据机密性的情况下,直接对密文进行某些运算来代替对明文的运算取得同样的效果.简单来说,对密文的计算等价于明文计算之后再加密.

ElGamal 公钥密码体制[30]是 1985 年 7 月由 ElGamal 发明的,它是建立在有限乘法群上的离散对数问题的困难性假设基础之上的一种公钥密码体制.该密码体制至今仍被认为是一个安全性能较好的公钥密码体制,目前它被广泛应用于许多密码协议中.

ElGamal 算法[30]描述如下.

公开参数.首先选取大素数 p,并取 g 是乘法群 $Z_p^*=\{1,...,p-1\}$ 的一个生成元.

密钥生成.随机选取整数 d: $0<d<p-1$, 并计算 $h=g^d \bmod p$. 这里,p 和 g 是公开参数,h 是公钥,d 是私钥.

加密.对明文 $m\in Z_p^*$, 随机选取整数 $r:0<r<p-1$, 计算

$$c_1=g^r \bmod p,\ c_2=mh^r \bmod p,$$

得到密文 $c=(c_1,c_2)$.

解密.对密文 $c=(c_1,c_2)\in Z_p^*\times Z_p^*$, 用私钥 d 解密得到明文为

$$m=c_2(c_1^d)^{-1} \bmod p.$$

本文所设计的安全多方计算协议采用了 ElGamal 乘法同态加密算法,该算法是概率加密算法,具有语义安全性.假设密文

$$E(m_1)=(g^{r_1},m_1h^{r_1}),$$
$$E(m_2)=(g^{r_2},m_2h^{r_2}),$$

那么,

$$E(m_1m_2)=(g^{r_3},m_1m_2h^{r_3}),$$
$$E(m_1)\otimes E(m_2)=(g^{r_1+r_2},m_1m_2h^{r_1+r_2})=(g^{r_3},m_1m_2h^{r_3})=E(m_1m_2).$$

该算法满足如下性质:

$$E(m_1)\otimes E(m_2)=E(m_1m_2).$$

1.2 安全性定义

双方计算.双方计算是一个将随机输入对映射为输出对的随机计算过程,可表示成如下的函数形式:

$$f:\{0,1\}^*\times\{0,1\}^*\to\{0,1\}^*\times\{0,1\}^*,$$

其中,$f=(f_1,f_2)$.函数 f 意为输入对(x,y)和输出对$(f_1(x,y),f_2(x,y))$一一对应$((f_1(x,y),f_2(x,y))$为随机变量,其变化范围为一对字符串),于是函数又可记作:

$$f(x,y)=(f_1(x,y),f_2(x,y)).$$

理想保密计算协议.理想保密计算协议是假设有一个值得信赖的第三方(trusted third party,简称 TTP),无论什么情况,都绝不会泄露信息和恶意传递虚假信息.因此在可信第三方存在的情况下,Alice 和 Bob 分别将自己的私有数据 x 和 y 告诉 TTP,TTP 计算 $f(x,y)$,然后将结果分别告诉 Alice 和 Bob,而不泄露任何其他私有信息.由于 Alice 和 Bob 无法从协议中获得除 $f(x,y)$ 以外的其他信息,所以该协议是安全性最高的双方保密计算协议,被称为“理想协议”.任何计算 $f(x,y)$ 的实际双方保密计算协议的安全性都无法超出这个理想协议.

半诚实参与者[31].所谓半诚实参与者就是各参与者执行协议的每一步都是完全严格按照协议的规定,并且

其在协议执行过程当中不会中途退出协议,也不会欺骗和泄露信息,但是它们可能会试图通过分析和利用协议执行过程中自己所得到的信息来推断额外的信息.

假设本文协议中所有参与者都是半诚实参与者,并且提出的所有安全多方计算协议和安全性均对其适用.到目前为止,大多数研究建立在半诚实模型上,这是因为研究半诚实模型下的安全多方计算是研究恶意模型下安全多方计算的基础,只要能够设计出在半诚实参与者模型下保密计算函数 f 的协议 π,就可以通过文献[31]中利用比特承诺和零知识证明理论设计的编译器将 π 转换成恶意攻击者参与的模型下保密计算 f 的协议.在转换后的协议中,一个恶意的参与者将被迫按照半诚实的参与者行事,否则将会被发现.在研究恶意模型下安全协议时,人们总是从半诚实参与者模型入手,找到防止恶意攻击的方法后,将其添加到协议中,最终形成恶意模型下的安全协议.

模拟范例[31].模拟范例在安全性证明中被广泛使用,相对于其他安全性证明方法,它可以简便地模拟参与者执行协议的过程.

模拟范例的原理.半诚实参与者在执行协议过程中,通过自己的输入和输出进行模拟所获得的消息序列与实际过程得到的消息序列不可区分,协议就是保密的.如果一个多方计算协议可以进行这样的模拟,参与者不能从协议的执行过程中得到其他人的任何有价值的信息,那么这个多方计算协议就是保密的.

一些记号[31].假设参与保密计算的双方分别为 Alice 和 Bob.

$f(x,y)=(f_1(x,y),f_2(x,y))$代表概率多项式时间函数,$\pi$ 表示计算 f 的协议.

假设协议执行过程中输入为(x,y),Alice 得到的消息序列记为 $view_1^{\pi}(x,y)$ 是 $(x,r^1,m_1^1,...,m_t^1)$, 其中,r^1 表示 Alice 独立的硬币抛掷结果(即 Alice 选定的随机数), m_i^1 表示 Alice 第 i 次收到的信息;执行协议后,Alice 得到的输出结果记为 $output_1^{\pi}(x,y)$. Bob 得到的信息序列,输出结果也可以用类似的内容来定义.

如果有多个参与方,在执行协议的过程中任意参与方 i 得到的信息序列记为 $view_i^{\pi}(x,y)$ 是 $(x,r^i,m_1^i,...,m_t^i)$, 其中,r^i 表示第 i 个参与方独立的硬币抛掷结果; m_j^i 表示第 i 个参与方第 j 次收到的信息,执行协议以后,第 i 个参与方得到的输出结果记为 $output_i^{\pi}(x,y)$.

定义 1(半诚实参与者的保密性)[31]. 对于一个函数 f,如果存在概率多项式时间算法 S_1 与 S_2(有时称这样的多项式时间算法为模拟器),使得

$$\{S_1(x,f_1(x,y)),f_2(x,y)\}_{x,y}\overset{c}{\equiv}\{view_1^{\pi}(x,y),output_2^{\pi}(x,y)\}_{x,y} \tag{1}$$

$$\{f_1(x,y),S_2(y,f_2(x,y))\}_{x,y}\overset{c}{\equiv}\{output_1^{\pi}(x,y),view_2^{\pi}(x,y)\}_{x,y} \tag{2}$$

式中,$\overset{c}{\equiv}$ 表示计算上不可区分,则认为保密地计算 f.要证明一个多方计算方案是保密的,就必须构造满足式(1)和式(2)的模拟器 S_1 和 S_2.

2 安全多方计算方案

2.1 两个字符串排序问题的高效解决方案

问题描述:Alice 有一个字符串 $A=a_1\ldots a_n$,Bob 有一个字符串 $B=b_1\ldots b_m$.在不泄露任何 A 和 B 信息的情况下判断 A 和 B 的按照字典序排序的位置关系.这就是字符串排序的安全多方计算问题.

方案的思想:两个字符串排序的问题即为判断两个字符串按照字典序排序位置关系的问题.要判断两个字符串按照字典序排序位置关系,首先要判断两个字符串相对应每个字符的位置关系(即判断 a_i 与 b_i 的位置关系).如果同一位置的两个字符相同(即 $a_i=b_i$),那么接着比较下一个位置的字符关系.如果同一位置的字符不同,那么相应字符在前的字符串排在另一个字符串之前,相应字符在后的字符串排在另一个字符串之后.本文将保密的数据通过一种编码方式表示在一个表格中(表格中的数据由 1,2 和随机数 r_i 构成),并在乘法同态的基础上设计了一个简单、高效的协议.

通过编码方式表示保密数据:在此方案中,假设字典的字母表为 $U=\{u_1,u_2,\ldots,u_t\}$且满足 $u_1<u_2<\ldots<u_t$.在字符串 A 和 B 中均有 $a_i,b_j\in U$, 且 $a_i=u_l,b_j=u_k(1\leqslant l,k\leqslant t)$.其中,$l$ 和 k 表示字符 a_i 和 b_j 在 U 中的位置,且 $l=(a_i)_{\text{ord}}$, $k=(b_j)_{\text{ord}}$.Alice 根据 a_i 和 U 将字符串 A 表示成一个 $n\times t$ 的表格 T.表格 T 中的元素 $T[r][c]$(其中,r 代表表格行号且对应字符串 A 中字符 a_i 的下标,c 代表表格列号且 $0<c\leqslant t$)按如下规则表示:

$$T[r][c]=\begin{cases}2, & \text{如果}c<l\\ 1, & \text{如果}c=l\\ r_{rc}, & \text{如果}c>l\end{cases},$$

其中,r_{rc} 为除 1 和 2 之外的随机数,$0<c\leqslant n$,$0<c\leqslant t$.

例如:本文假设字母表 U 为 26 个英文字母表,字符串 $A=bactr$,字符串 $B=cab$.那么按照编码方式可以将字符串 A 表示为表 1(第 1 个字符 b 在字母表 U 中是第 $l=(a_1)_{\text{ord}}=(b)_{\text{ord}}=2$ 位,那么得到的表格第 1 行第 l 列的值为 1,第 l 列之前的值都为 2,第 l 列之后的值为随机数 $r_{1c}(c>l)$.其他字符按照同样的规则表示到表格 T 中).

Table 1 Coding table constructed by using bactr

表 1 利用 bactr 构造的编码表

行/列	第 1 列	第 2 列	第 3 列	…	第 18 列	第 19 列	第 20 列	第 21 列	…	第 26 列
第 1 行	2	1	r_{13}	…	$r_{1(18)}$	$r_{1(19)}$	$r_{1(20)}$	$r_{1(21)}$	…	$r_{1(26)}$
第 2 行	1	r_{22}	r_{23}	…	$r_{2(18)}$	$r_{2(19)}$	$r_{2(20)}$	$r_{2(21)}$	…	$r_{2(26)}$
第 3 行	2	2	1	…	$r_{3(18)}$	$r_{3(19)}$	$r_{3(20)}$	$r_{3(21)}$	…	$r_{3(26)}$
第 4 行	2	2	2	…	2	2	1	$r_{4(21)}$	…	$r_{4(26)}$
第 5 行	2	2	2	…	1	$r_{5(19)}$	$r_{5(20)}$	$r_{5(21)}$	…	$r_{5(26)}$

Alice 将表格 T 发送给 Bob.Bob 根据字符串 B 中每个字符在字典集中的位置对应地从表格 T 中取出 $T[1][(b_1)_{\text{ord}}]$,$T[2][(b_2)_{\text{ord}}]$,$T[3][(b_3)_{\text{ord}}]$, 并计算

$$s_1=T[1][(b_1)_{\text{ord}}],$$
$$s_2=T[1][(b_1)_{\text{ord}}]\times T[2][(b_2)_{\text{ord}}],$$
$$s_3=T[1][(b_1)_{\text{ord}}]\times T[2][(b_2)_{\text{ord}}]\times T[3][(b_3)_{\text{ord}}].$$

Bob 利用计算结果构造一个集合 $S=\{s_1,s_2,s_3\}=\{r_{13},r_{13}\times 1,r_{13}\times 1\times 2\}$, 其中, $s_j=\prod_{i=1}^{j}T[i][(b_i)_{\text{ord}}]$.

事实 1.字符串 B 排在字符串 A 之前当且仅当集合 S 中出现 2.字符串 B 排在字符串 A 之后当且仅当集合 S 中元素值没有出现 2 且不都为 1;如果集合 S 中所有元素值均为 1,那么字符串 B 为字符串 A 的子串($A=B$ 可以认为 A 是 B 的子串,也可以看作 B 是 A 的子串).

证明:先证明事实的第 1 部分.

$\Rightarrow$ 如果 B 排在 A 之前,一定存在一个 $j(1\leqslant j\leqslant m)$ 满足 $a_1=b_1,\ldots,a_{j-1}=b_{j-1},(a_j)_{\text{ord}}>(b_j)_{\text{ord}}$, 不需要考虑 $b_{j+1},\ldots,b_m$ 与 $a_{j+1},\ldots,a_m$ 的排列关系,因而,

$$s_1=1,\ldots,s_{j-1}=1,\ s_j=\prod_{i=1}^{j}T[i][(b_i)_{\text{ord}}]=2.$$

$\Leftarrow$ 如果 S 中出现一个 2,假设 $s_j=\prod_{i=1}^{j}T[i][(b_i)_{\text{ord}}]=2$ 是第 1 次出现的 2.根据表的构造原则,这意味着 $s_1=1,\ldots,s_{j-1}=1$, 即 $b_1,\ldots,b_{j-1}$ 与 $a_1,\ldots,a_{j-1}$ 相同,而 b_j 排在 a_j 的前面.因而 B 排在 A 的前面,不需要再考虑 b_j 后面的字符和 a_j 后面的字符的排列关系.

下面证明事实的第 2 部分.

$\Rightarrow$ 如果 B 排在 A 之后,一定存在一个 $j(1\leqslant j\leqslant m)$ 满足 $a_1=b_1,\ldots,a_{j-1}=b_{j-1},(a_j)_{\text{ord}}<(b_j)_{\text{ord}}$, 不需要考虑 $b_{j+1},\ldots,b_m$ 与 $a_{j+1},\ldots,a_m$ 的排列关系,因而,

$$s_1=1,\ldots,s_{j-1}=1,\ s_j=\prod_{i=1}^{j}T[i][(b_i)_{\mathrm{ord}}]=r.$$

其中,r 是一个不等于 1 和 2 的随机数.因为 $s_{j+1},\ldots,s_m$ 中都包含 s_j 这个因子,所以都是随机数.故集合 S 中没有出现 2,且都不为 1.

⇐ 如果 S 中没有出现 2,而且都不为 1.假设 $s_j=\prod_{i=1}^{j}T[i][(b_i)_{\mathrm{ord}}]=r$ 是第 1 次出现不等于 1 的元素.根据表的构造原则,这意味着 $s_1=1,\ldots,s_{j-1}=1$, 即 $b_1,\ldots,b_{j-1}$ 与 $a_1,\ldots,a_{j-1}$ 相同,而 b_j 排在 a_j 的后面.因而 B 排在 A 的后面,不需要再考虑 b_j 后面的字符和 a_j 后面的字符的排列关系.

对于第 3 种情况,因为是从第 1 个字符开始计算的,如果 B 是 A 的子串,那么 A 至少和 B 一样长,在这种情况下,B 排在 A 的前面.

根据这个事实,Alice 和 Bob 可以通过集合 S 中的元素值判断字符串 A 和 B 的按照字典序排序的位置关系,但因为用明文形式的编码表没有保密性可言,所以不能用明文的表完成这项任务,而且还有一些细节问题需要处理.因为我们需要的计算是保密的乘法运算,所以需要用 ElGamal 乘法同态加密算法,同时,为了提高算法的加密效率,我们设计了以下将加密计算外包给云来完成的方案 E.

2.2 云外包计算下的同态加密方案

密钥生成.首先选取大素数 p,并取 g 是乘法群 $Z_p^*=\{1,...,p-1\}$ 的一个生成元.随机选取整数 $d:0<d<p-1$, 并计算 $h=g^d \bmod p$.这里,p 和 g 是公开参数,h 是公钥,d 是私钥.

云外包模指数运算.云服务器随机选择 $r_1,r_2,...,r_k\in Z_p^*(k\leqslant p)$, 计算足够多的 $R_i=(R_{i1},R_{i2})$, 其中,

$$R_{i1}=g^{r_i} \bmod p,\ R_{i2}=h^{r_i} \bmod p,$$

并将它们存储在集合 R 中.上述过程可以在预处理阶段完成.

加密.从云服务器上下载 R,随机选择集合 R 中的某些元素进行一定次数的模乘运算就可以得到 $g^r \bmod p$ 和 $h^r \bmod p$.本文中以进行一次模乘运算为例来说明.例如:从 R 上随机选择 2 个元素,记作 $R_i,R_j\in R(1\leqslant i,j\leqslant k)$, 计算

$$R_\alpha=(R_{\alpha1},R_{\alpha2})=R_i\cdot R_j \bmod p,$$
$$c_1=R_{\alpha1},\ c_2=mR_{\alpha2},$$

得到密文 $c=(c_1,c_2)$.

解密.对密文 $c=(c_1,c_2)\in Z_p^*\times Z_p^*$, 用私钥 d 解密得到明文为

$$m=c_2(c_1^d)^{-1} \bmod p.$$

2.3 具体协议

为方便表达,定义如下谓词:

$$P(A,B)=\begin{cases}1, & \text{字符串}B\text{为字符串}A\text{的子串}\\ 0, & \text{字符串}B\text{按照字典序排列位于字符串}A\text{之前}.\\ 2, & \text{字符串}B\text{按照字典序排列位于字符串}A\text{之后}\end{cases}$$

协议 1.保密地判断两个字符串按照字典序排序的位置.

输入:Alice 拥有的保密的字符串 $A=a_1\ldots a_n$,Bob 拥有的保密的字符串 $B=b_1\ldots b_m$.

输出:$P(A,B)$.

假设整个字母表为 $U=\{u_1,u_2,\ldots,u_t\}$ 且满足 $u_1<u_2<\ldots<u_t$.在字符串 A 和 B 中,$a_i,b_j\in U$, 且 $a_i=u_l,b_j=u_k$ $(1\leqslant l,k\leqslant t)$.其中,$l$ 和 k 表示字符 a_i 和 b_j 在 U 中的位置,且 $l=(a_i)_{\mathrm{ord}},k=(b_i)_{\mathrm{ord}}$.

(1) Alice 利用加密方案 E 的密钥生成算法生成公私密钥对(P_K,S_K),并且将公钥 P_K 告诉 Bob,私钥 S_K 保密.

(2) 云服务器随机选择 $r_1,r_2,\ldots,r_k\in Z_p^*(k\leqslant p)$, 计算足够多的 $R_i=(R_{i1},R_{i2})$, 其中,

$$R_{i1}=g^{r_i} \bmod p,\ R_{i2}=h^{r_i} \bmod p,$$

并将它们存储在集合 R 中.Alice 从云服务器上下载 R,利用集合 R 中的某些元素通过适量的模乘运算得到

$$R_\alpha=(R_{\alpha 1},R_{\alpha 2}).$$

(3) Alice 根据 a_i 和 U 将字符串 A 加密并表示成一个 $n\times t$ 的表格 T.表格 T 表示规则如下:

$$T[r][c]=\begin{cases}E(2), & \text{如果}c<l\\ E(1), & \text{如果}c=l\\ r_{rc}, & \text{如果}c>l\end{cases},$$

其中,加密的每个 $E(1)$和 $E(2)$所需要的 R_α 都不相同(即加密每个 1,2 时,Alice 利用集合 R 中的某些元素所做的模乘运算次数不同),r_{rc} 为除 1 和 2 之外的随机数,$0<r\leqslant n$,$0<c\leqslant t$.

将字符串 A 中所有字符均按照规则放入表格 T 中后,为了不泄露字符串 A 的长度 n,在表格的末尾随机插入 $p(1<p\leqslant n)$行除 1 和 2 以外的随机数得到$(n+p)\times t$ 表格 T',并将表格 T'发送给 Bob.

(4) Bob 根据字符串 B 中每个字符在字典集中的位置对应地从表格 T'中取出相应的数据,并计算

$$E(s_1)=T'[1][(b_1)_{\text{ord}}],$$

$$E(s_2)=T'[1][(b_1)_{\text{ord}}]\times T'[2][(b_2)_{\text{ord}}],$$

$$\vdots$$

$$E(s_m)=T'[1][(b_1)_{\text{ord}}]\times\ldots\times T'[j][(b_j)_{\text{ord}}]\times T'[m][(b_m)_{\text{ord}}].$$

(5) Bob 利用计算结果构造一个集合:

$$S=\{E(s_1),E(s_2),\ldots,E(s_m)\}.$$

为了不泄露字符串 B 的长度 m,且当 B 排在 A 的前面时也不泄露 B 中有几个字符排在 A 的相应字符之前(比如,如果 $A=cde$,$B=abc$,Bob 计算的集合中将会出现 2,4,8.这样直接发送给 Alice 的话,Alice 将会知道 Bob 的字符串有 3 个字符,而且知道这 3 个字符都在 A 的相应字符之前.出现 2^k 就说明有 k 个字符排在 A 的相应字符之前,而这一点是应该避免的).在集合 S 中随机选取 $q(1<q<m)$个元素值(q 对于 Alice 保密),对于选取的每一个元素 $s_k(k\in 1,\ldots,m)$ 再随机选取一个数 $l\in 1,\ldots,m$, 计算 s_k^l, 利用概率加密的性质进行再随机化,并将它们随机插入到集合 S 中,得到集合

$$S'=\{E(s_1),E(s_2),\ldots,E(s_m),E(s_{m+1}),E(s_{m+2}),\ldots,E(s_{m+q})\}.$$

最后将集合 S' 中元素整体做随机置换,将得到的结果 $\Pi(S')$ 发送给 Alice.

(6) Alice 用私钥解密 $\Pi(S')$, 如果解密结果 $D(\Pi(S'))$ 中出现 2,输出 $P(A,B)=0$;如果解密结果不出现 2 且不都为 1,输出 $P(A,B)=2$;如果解密结果均为 1,输出 $P(A,B)=1$.

2.4 协议1安全性分析

加密方案 *E* 安全性分析 1.

协议 1 利用加密方案 E 加密时计算 $R_\alpha=R_i\cdot R_j \bmod p$ 不会降低协议的安全性.因为 R 虽然是公开的,但是 R 中的元素和这些元素需要进行的模乘运算都是随机的,所以 $R_\alpha=(R_{\alpha 1},R_{\alpha 2})$与$(g^r \bmod p,h^r \bmod p)$是等效的,因此任何敌手由 R 计算 R_α 的困难性与破解 ElGamal 加密算法的困难性是等价的.综上所述,加密方案 E 将计算 $R_i=(R_{i1},R_{i2})$外包给云服务器执行从语义安全角度来讲是安全的.

定理 1. 保密地判断两个字符串按照字典序排序的位置的协议 1 是安全的.

证明:通过构造使上述(1)和(2)成立的模拟器 S_1 和 S_2 证明本定理.

在协议 1 中,

$$view_1^\pi(A,B)=\{A,T',\Pi(S'),D(\Pi(S')),P(A,B)\},$$

$$f_1(A,B)=f_2(A,B)=output_1^\pi(A,B)=output_2^\pi(A,B)=P(A,B),$$

其中,A,B 是 Alice 和 Bob 的输入,T'是 Alice 用公钥 P_K 加密字符串 A 并将其表示在表格中的形式. $\Pi(S')$ 是 Bob 先根据字符串 B 从表格中抽取数据,然后将经过计算的结果构造集合,并将集合中元素置换后发送给 Alice 的结果. $D(\Pi(S'))$ 是 Alice 解密 $\Pi(S')$ 后得到的结果.

首先构造模拟器 S_1 来模拟 $view_1^{\pi}(A,B)$, S_1 的模拟过程如下.

(1) 接受输入 $(A,f_1(A,B))$, 根据 $f_1(A,B)$ 的值,选定字符串 B',使 $f_1(A,B')=f_1(A,B)$, 记 $B'=b'_1\ldots b'_m$. S_1 根据字符串 B'中每个字符在字典集中的位置对应地从表格 T'中取出相应的数据并做计算,通过添加,置换得到集合 $\Pi(S')'$.

(2) S_1用私钥解密 $\Pi(S')'$ 得到解密结果 $D(\Pi(S')')$. 通过判断解密结果 $D(\Pi(S')')$ 中是否出现2来判断字符串按照字典序排序的位置.令:

$$S_1(A,f_1(A,B))=\{A,T',\Pi(S')',D(\Pi(S')'),P(A,B)\},$$

因为

$$\Pi(S')\overset{c}{\equiv}\Pi(S')',D(\Pi(S'))\overset{c}{\equiv}D(\Pi(S')'),$$

所以

$$\{S_1(A,P(A,B)),f_2(A,B)\}\overset{c}{\equiv}\{view_1^{\pi}(A,B),output_2^{\pi}(A,B)\}.$$

类似地,还可以构造 S_2,使

$$\{f_1(A,B),S_2(B,f_2(A,B))\}\overset{c}{\equiv}\{output_1^{\pi}(A,B),view_2^{\pi}(A,B)\}.$$ □

加密方案 *E* 安全性分析 2.

在安全多方计算方案中,如何衡量协议的安全性是非常复杂的问题.我们通常借助 Goldreich 提出的模拟范例来模拟参与者执行协议的过程,从而对协议进行安全性证明.但是,通过构造模拟器的方式只能证明协议是安全的,而不能对协议的安全性水平加以区别.所以我们通过在执行协议过程中参与者私有信息的泄漏量来衡量方案的安全性水平.

一些符号的定义如下:

(1) $P(A)$表示 Bob 在没有执行任何安全多方计算协议之前获得的关于字符串 A 的先验概率.

(2) $P(B)$表示 Alice 在没有执行任何安全多方计算协议之前获得的关于字符串 B 的先验概率.

(3) $P\left(B\middle|f_1(A,B)\right)$表示 Alice 通过执行理想协议得到输出结果 $f_1(A,B)$的条件下获得的关于字符串 B 的条件概率.

(4) $P\left(A\middle|f_2(A,B)\right)$ 表示 Bob 通过执行理想协议得到输出结果 $f_2(A,B)$的条件下获得的关于字符串 A 的条件概率.

(5) $P_{\pi}\left(A\middle|f_2(A,B)\right)$ 表示 Bob 在执行协议 π 后得到输出结果 $f_2(A,B)$的条件下获得的关于字符串 A 的条件概率.

(6) $P_{\pi}\left(B\middle|f_1(A,B)\right)$ 表示 Alice 在执行协议 π 后得到输出结果 $f_1(A,B)$的条件下获得的关于字符串 B 的条件概率.

因为理想协议是最安全的协议,其他协议的安全性和信息泄露都不可能优于理想协议,一种语言中一个固定长度的字符串的先验概率是不变的,我们可以用理想协议作为标准衡量其他协议的安全性和信息泄漏量,用理想协议的条件概率和实际协议的条件概率的比值 $P\left(A\middle|f_2(A,B)\right)$ 与 $P_{\pi}\left(A\middle|f_2(A,B)\right)$ 刻画实际协议与理想协议相比是不是泄露了更多的信息,比值越接近于 1 说明实际协议泄露的信息越少,越接近于 0 说明实际协议泄露的信息越多.

定义 2. 对于计算多项式时间函数 f 的协议 π,如果满足以下条件:

$$P_{\pi}\left(A\middle|f_2(A,B)\right)=P\left(A\middle|f_2(A,B)\right),$$

$$P_{\pi}\left(B\middle|f_1(A,B)\right)=P\left(B\middle|f_1(A,B)\right),$$

则认为保密地计算 f 的协议 π 与理想协议是等价的,信息泄露最少.

也可以利用信息熵的变化来衡量信息泄露.假设字符串 A 分布在集合 $\{A_1,A_2,\ldots,A_n\}$ 上,且出现的概率为 $P(A=A_1),P(A=A_2),\ldots,P(A=A_n)$,则字符串 A 的信息熵为

$$H(A)=-\sum P(A=A_i)\times\log P(A=A_i).$$

执行理想协议并得到$f_2(A,B)$后,关于A的信息熵为

$$H\left(A\middle|f_2(A,B)\right)=-\sum P\left(A=A_i\middle|f_2(A,B)\right)\times\log P\left(A=A_i\middle|f_2(A,B)\right).$$

类似地,可以定义得到$f_1(A,B)$后,关于B的信息熵,以及执行实际协议π之后关于A,B的条件信息熵.

同样得到的关于A,B的信息越多,信息熵的减少就越多.显然,理想协议执行后信息熵的减少是最少的,其他协议信息熵的减少都不会少于理想协议.因此,我们可以用比值$H_\pi\left(A\middle|f_2(A,B)\right)\Big/H\left(A\middle|f_2(A,B)\right)$衡量协议的信息泄露量.

定义 3. 对于计算多项式时间函数f的协议π,如果满足以下条件:

$$H_\pi\left(A\middle|f_2(A,B)\right)=H\left(A\middle|f_2(A,B)\right),$$

$$H_\pi\left(B\middle|f_1(A,B)\right)=H\left(B\middle|f_1(A,B)\right),$$

则认为保密地计算f的协议π与理想协议是等价的.

本文中,我们用条件概率来度量协议中 Bob 的信息泄漏量.首先以拥有单个字符的字符串的保密排序为例分析协议的信息泄露量(本文协议记为π).

假设字母表为$U=\{a,b,...,z\}$,$A=d,B=m$. A,B 在 U 上均匀分布(实际上不是均匀分布的,但分析方法相同,只是分析的计算过程稍微复杂一些).在这样的假设条件下,在保密计算前,关于 A,B 的先验概率都是 1/26,即$P(A)=P(B)=1/26$ 执行协议后 Bob 知道字符串A排在字符串B的前面,除此之外没有更多的信息,只能假设A在$\{a,b,c,d,e,f,g,h,i,j,k,l\}$上均匀分布,因此

$$P_\pi(A\mid f_2(A,B))=1/12=1/((B)_{\text{ord}}-1).$$

类似地,可以计算$P_\pi(B\mid f_1(A,B))=1/22=1/(26-(A)_{\text{ord}})$. 简单的分析计算可知, $P(A\mid f_2(A,B))=1/12$, $P(B\mid f_1(A,B))=1/22$. 所以有:

$$P_\pi(A\mid f_2(A,B))/P(A\mid f_2(A,B))=1,\ P_\pi(B\mid f_1(A,B))/P(B\mid f_1(A,B))=1.$$

其他情况下,无论是A排在字符串B的后面,或是A排在字符串B的同样位置,简单的分析都可以得出同样的结论.因此我们的协议所导致的信息泄露和理想协议的信息泄露一样,这部分信息泄露完全是函数 $f(A,B)$的信息泄露,这是无法避免的.

当字符串是多个字符而不是单个字符时,用排列组合的乘法原理也可以计算相应的条件概率,只是计算更复杂一些.经过分析可以得出如下结论,我们的协议和理想协议是等价的,泄露的信息量也和理想协议相同,都达到了最少的信息泄露.

2.5 协议1正确性分析

(1) 协议 1 在保密字符的加密过程中将计算$R_i=(R_{i1},R_{i2})$外包给云,不会影响最终的解密结果,随机变量R_α可以在解密运算中被成功消除.因为

$$R_i=(R_{i1},R_{i2})=(g^{r_i}\bmod p,h^{r_i}\bmod p),$$

$$R_j=(R_{j1},R_{j2})=(g^{r_j}\bmod p,h^{r_j}\bmod p).$$

所以可以得到

$$\begin{aligned}R_\alpha&=(R_{\alpha1},R_{\alpha2})\\&=R_i\cdot R_j\bmod p\\&=(R_{i1}\cdot R_{j1},R_{i2}\cdot R_{j2})\\&=(g^{r_i}\cdot g^{r_j}\bmod p,h^{r_i}\cdot h^{r_j}\bmod p)\\&=(g^{r_i+r_j}\bmod p,h^{r_i+r_j}\bmod p)\\&=(g^{r}\bmod p,h^{r}\bmod p).\end{aligned}$$

因此,将计算$R_i=(R_{i1},R_{i2})$外包给云服务器执行并不会影响解密结果.

(2) Alice 按照新的编码方式将字符串A表示在表格T'中,按照协议 1 将 Bob 构造的集合$\Pi(S')$解密,如果

解密的集合 $D(\Pi(S'))$ 中出现 2,则说明 Bob 按照字符串 B,从表格 T'中所取的字符至少有一个是排在字符串 A 中字符的前面,只有这样,在解密后的集合 $D(\Pi(S'))$ 中才会出现 2.同理,如果解密的集合 $D(\Pi(S'))$ 中不出现 2 且不均为 1,则说明 Bob 按照字符串 B,从表格 T'中所取的字符至少有一个是排在字符串 A 中字符的后面.如果解密的集合 $D(\Pi(S'))$ 中均为 1,那么字符串 B 为字符串 A 的子串.

因此协议 1 能够正确地判断字符串 A 和 B 按照字典序排序的位置关系.

2.6 性能分析

目前没有任何关于保密地判断两个字符串按照字典序排序位置关系的协议,因此在本节只对协议 1 进行效率分析和实验验证.本文的协议是用同态加密算法解决字符串按照字典序顺序排序的问题,基本运算都是模乘运算.

计算复杂性分析.本文在协议 1 中利用同态加密方案 E 计算 $R_i=(R_{i1},R_{i2})$是在预处理阶段由云服务器完成的,在加密过程中,只需通过对 R_i 执行一定次数的模乘运算($R_a=R_i\cdot R_j \bmod p$),就可以秘密地得到 $g^r \bmod p$ 和 $h^r \bmod p$,而不需要再做复杂的模指数运算.如果忽略预处理时间,用方案 E 加密 1 次只需要进行 1 次乘法运算和 1 次模乘运算.Alice 用加密方案 E 最多加密 nt 次,解密 m 次.加密 1 次需要 1 次模乘运算,解密 1 次需要 $\lg p$ 次模乘运算,故协议 1 的计算复杂性为 $m\lg p+nt$ 次模乘运算.

通信复杂性分析.衡量通信复杂度的指标一般用协议交换信息的比特数,或者用通信轮数,在安全多方计算研究中通常用轮数.本文中协议 1 需要进行 3 轮通信.

2.7 实验数据分析

实验测试环境.Windows 10 64 位操作系统,处理器是 Intel(R)Core(TM)i5-6600 CPU @3.30GHz,内存是 8.00GB,用 Java 语言在 MyEclipse 上运行实现.本文所做模拟实验均在此环境下进行.

实验方法.我们通过模拟实验来测试本文执行协议 1 所用时间,可通过协议执行的时间来验证方案的效率.本实验以字符串 A 和字符串 B 为例,设定字符串 A 的字符个数 n=20,字符串 B 的字符个数分别为 m=1,2,…,20,对 m 的每个设定值进行 1 000 次模拟实验测试,忽略协议中的预处理时间,统计协议执行时间的平均值.图 1 描述了判断字符串排序的执行时间随字符串字符个数增长的变化规律.

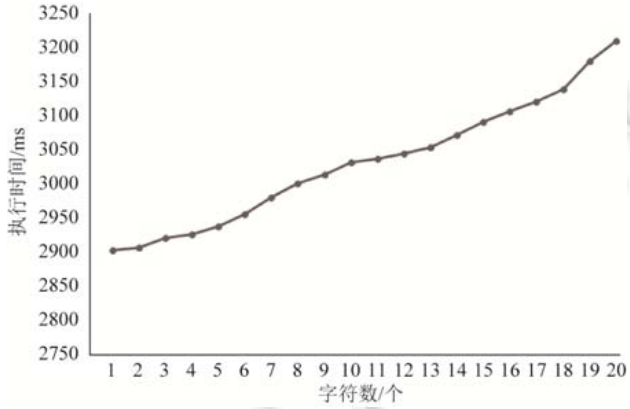

Fig.1 The execution time of the string sort increases with the number of characters in the string

图 1 字符串排序的执行时间随字符串字符个数增长的变化规律

3 应 用

本节我们将利用保密地判断两个字符串位置关系的协议来解决大数据情况下的百万富翁问题和保密地判断字符串模式匹配问题.

3.1 大数据情况下的百万富翁问题

问题描述:Alice 拥有大数据 x,Bob 拥有大数据 y.双方想在不泄露任何 x 和 y 信息的情况下知道 x 和 y 的大小关系.

我们可以将大整数 x 和 y 看成是用十进制表示的特殊类型的字符串,那么就可以将保密地判断大整数 x 和 y 的大小问题转化成保密的字符串排序问题,即通过判断字符串的位置关系来确定 x 和 y 的大小关系.假设 Alice 和 Bob 协商将大整数 x 和 y 表示成 n 位的十进制数,不足的位数分别用 0 补齐(此处为高位补 0).如:大整数 $x=x'=x_1x_2\ldots x_n$, 大整数 $y=y'=y_1y_2\ldots y_n$.

由事实 1 可知:根据集合 S 中的元素值可判断字符串 A 和字符串 B 的位置关系,于是有:如果 $x'<y'$, 那么通过调用协议 1 得到的集合 S 中的元素值没有出现 2 且不都为 1.如果 $x'=y'$, 那么集合 S 中所有元素值均为 1.如果 $x'>y'$, 那么集合 S 中出现 2.

为方便表达,定义如下谓词:

$$P(A,B)=\begin{cases}1, & x=y\\ 0, & x>y\\ 2, & x<y\end{cases}.$$

协议 2.大数据情况下的百万富翁问题.

输入:Alice 输入私有数据 x,Bob 输入私有数据 y.

输出:$P(A,B)$.

(1) 假设 Alice 和 Bob 协商将大整数 x 和 y 表示成 n 位的十进制数,不足的位数分别用 0 补齐.如:大整数 $x=x'=x_1x_2...x_n$, 大整数 $y=y'=y_1y_2...y_n$.

(2) Alice 和 Bob 调用协议 1,根据解密结果 $D(\Pi(S'))$ 中的值来判断两个大整数 x 和 y 的大小关系.如果 $D(\Pi(S'))$ 中所有的元素值均为 1,输出 $P(A,B)$=1,此时大整数 x=y;如果 $D(\Pi(S'))$ 中元素值出现 2,输出 $P(A,B)$=0,此时大整数 x>y;如果 $D(\Pi(S'))$ 中的元素值没有出现 2 且不都为 1,输出 $P(A,B)$=2,此时大整数 x>y.

协议效率分析.

在协议 2 中最多需要 $n\lg p$ 次模乘运算(n 为机密数据的长度),Alice 和 Bob 之间需要进行 3 轮通信.

文献[23]和本文的方案都可以一次性解决大数据情况下的百万富翁问题,而不需要重复调用多次基本协议,同时都对保密数据进行了编码.忽略方案中随机数选择的计算开销和双方准备阶段的计算开销,且将两个方案中的模都统一为 p 进行比较分析.

Table 2 The efficiency of the protocol 2

表 2 协议 2 性能分析与比较

	文献[23]	本文协议 2
计算复杂性	$5n\log p+4n-6$	$n\log p+2n$
通信复杂性	3	3

由表 2 可知,本文协议 2 的通信复杂性和文献[23]的通信复杂性一样,计算复杂性低于文献[23]的计算复杂性.当两个很大的数据比较大小时,p 为固定数值,$\log p$ 不会随着 n 的变化而线性增大.当 n 的取值很大时,本文中协议 2 的计算速率比文献[23]快 5 倍多.在适用范围方面,文献[23]不能完全判断两个保密数的小于和等于关系,而本文的协议 2 不仅解决了两个数比较大小的问题,也能区分两个数是否相等的问题.

实验数据分析.

实验方法.我们通过模拟实验来测试本文协议 2 和文献[23]中协议所用时间,可通过协议执行的时间来验证方案的效率.本实验假定数 A 和数 B 长度为 n,n 的变化范围为 20,21,…,40,对 n 的每个设定值进行 1 000 次模拟实验测试,忽略协议中的预处理时间,统计协议执行时间的平均值.图 2 描述了大数据情况下百万富翁协议的执行时间随机密数据长度增长的变化规律.

协议 2 的安全性依赖于保密地判断两个字符串位置关系协议的安全性,应用证明定理 1 所用的方法很容易

证明协议 2 的安全性,本文在这里省略证明过程.

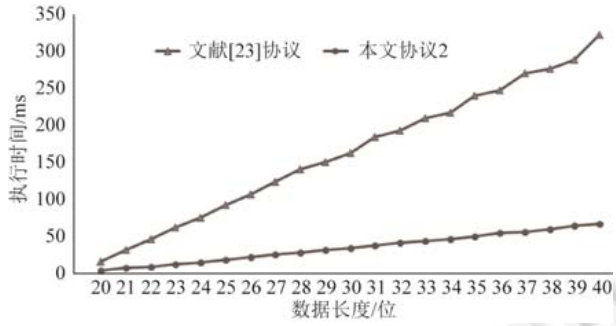

Fig.2 The execution time of the millionaires' problem increases with the number of characters in the string

图 2 大数据情况下百万富翁协议的执行时间随字符串字符个数增长的变化规律

3.2 保密地判断字符串模式匹配问题

问题描述:Alice 有一个字符串(文本串) $A=a_1\ldots a_n$, Bob 有一个字符串(模式串) $B=b_1\ldots b_m(m\leqslant n)$. 在不泄露任何 A 和 B 私有信息的情况下判断 A 中是否存在一个子串与 B 相等,也就是说,在字符串 A 中找到一个子串 $S_i=a_ia_{i+1}\ldots a_{i+m-1}$, 使 $S_i=B$.

利用协议 1 这种将保密的字符串构造为编码表的方法,适当地作处理就可以解决字符串模式匹配的问题.Alice 根据字符串 A 利用文章前面提到的编码规则构造表格 T'.根据字符串 B 中每个字符在字典集中的位置,Bob 对应地从表格 T'中第 1 行开始取值(直到第 $n-m+1$ 行为止),并将取值相乘得到结果 $E(S_i)$,记作集合 $W=\{w_1,w_2,\ldots,w_{n+m-1}\}=\{E(s_1),E(s_2),\ldots,E(s_{n+m-1})\}$. Alice 解密,如果集合 W 中出现 1,则说明字符串 A 中存在子串与字符串 B 相等.

为方便表达,定义如下谓词:

$$P(A,B)=\begin{cases}1, & \text{字符串}A\text{ 和字符串}B\text{ 匹配}\\ 0, & \text{字符串}A\text{ 和字符串}B\text{ 不匹配}\end{cases}.$$

协议 3.保密地判断两个字符串是否模式匹配.

输入:Alice 拥有保密的字符串 $A=a_1a_2\ldots a_n$,Bob 拥有保密的字符串 $B=b_1b_2\ldots b_m(m\leqslant n)$.

输出:$P(A,B)$.

(1) 令 $i=1$.

(2) Alice 调用协议 1 编码保密数据的方法构造表格 T',Bob 根据字符串 B 中每个字符在字典集中的位置对应地从表格 T'中取出相应的数据,并计算

$$E(s_i)=T'[i][(b_1)_{\text{ord}}]\times\ldots\times T'[j][(b_j)_{\text{ord}}]\times T'[m+i-1][(b_m)_{\text{ord}}].$$

(3) 令 $i=i+1$.Bob 循环执行以上第(2)步,直到 $i=n-m+1$ 为止.

(4) Bob 将循环得到的结果 $E(s_i)$记为 $W=\{w_1,w_2,\ldots,w_{n+m-1}\}=\{E(s_1),E(s_2),\ldots,E(s_{n+m-1})\}$.为了不泄露字符串 A 中有几个子串和字符串 B 相等,再在 W 中随机选取 z 个元素值,将它们随机插入到 W 中,得到集合 $W'=\{w_1,w_2,\ldots,w_{n+m-1},w_{n+m},w_{n+m+1},\ldots,w_{n+m-1+z}\}$, 最后将集合 W'中元素做置换,将得到的结果 $\psi(W')$ 发送给 Alice.

(5) Alice 根据解密结果 $D(\psi(W'))$ 中的值来判断两个字符串是否模式匹配.如果 $D(\psi(W'))$ 中元素值出现 1,则输出 $P(A,B)=1$,此时字符串 A 中存在一个子串 $S_i=a_ia_{i+1}\ldots a_{i+m-1}$, 使得 $S_i=B$;否则,输出 $P(A,B)=0$.

协议效率分析.

本节将与 2010 年的文献[32]中所提出的同样使用 ElGamal 同态加密算法和新的保密数据编码方法解决字符串匹配问题的协议作比较.忽略方案中随机数选择的计算开销和双方准备阶段的计算开销,对两个方案进行对比.

Table 3 The efficiency of the protocol 3

表 3 协议 3 性能分析与比较

	文献[32]	本文协议 3
计算复杂性	$mn(4nk+1)\lg p$	$(n+m-1)\lg p$
通信复杂性	mn^2+mn	3

在表 3 中,n 为字符串 A 的字符个数,m 为字符串 B 的字符个数,k 为字符串 A 和 B 每个字符对应的 ASCII 值的二进制位数,p 为大素数.$mn(4nk+1)\lg p$ 表示执行文献[32]中协议需要进行的模乘次数.由表 3 可知,本文协议 3 的计算复杂性远小于文献[32],而且文献[32]的通信复杂性和字符串 A,B 的长度有关,会随着字符串长度的增加而越来越大,本文协议 3 的通信复杂性为定值,远低于文献[32].

实验数据分析.

实验方法.我们通过模拟实验来测试本文协议 3 和文献[32]协议所用的时间,可通过协议执行的时间来验证方案的效率.本实验以字符串 A 和字符串 B 为例,设定字符串 A 的字符个数为 26,字符串 B 的字符个数分别为 m=1,2,…,20,对 m 的每个设定值进行 1 000 次模拟实验测试,忽略协议中的预处理时间,统计协议执行时间的平均值.图 3 描述了字符串模式匹配的执行时间随模式串字符个数增长的变化规律.

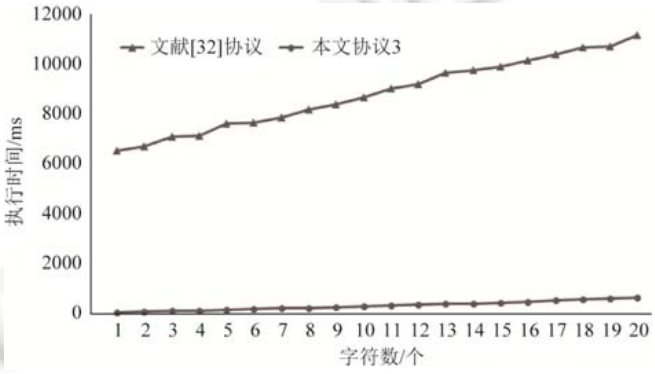

Fig.3 The execution time of the string matching problem increases with the number of characters in the string

图 3 字符串模式匹配的执行时间随模式串字符个数增长的变化规律

协议 3 的安全性可以应用证明定理 1 所用的方法,本文在这里省略证明过程.

4 结 论

字符串排序问题是安全多方计算中新的研究问题,具有重要的研究意义和应用前景.本文基于一种新的编码方式和一种云外包计算下的同态加密方案设计了两个字符串保密排序的安全多方计算协议,并利用模拟器证明了方案的安全性,同时将保密的字符串排序协议应用于解决大数据情况下的百万富翁问题和保密地判断字符串模式匹配问题.本文研究的问题都是基于半诚实模型的,对于安全多方计算的研究与应用有重要的理论意义,但恶意模型的安全性更高、更具有实际意义,所以如何实现恶意模型下的字符串保密排序问题是我们今后研究的问题.

References:

[1] Goldwasser S. Multi party computations: Past and present. In: Proc. of the 16th Annual ACM Symp. on Principles of Distributed Computing. ACM, 1997. 1–6.

[2] Goldreich O. Secure multi-party computation. Manuscript, Preliminary Version, 1998. 86–97.

[3] Freedman MJ, Nissim K, Pinkas B. Efficient private matching and set intersection. In: Proc. of the Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer-Verlag, 2004. 1–19.

[4] Lynn B, Prabhakaran M, Sahai A. Positive results and techniques for obfuscation. In: Advances in Cryptology-EUROCRYPT 2004. Berlin, Heidelberg: Springer-Verlag, 2004. 20–39.

[5] Aggarwal G, Mishra N, Pinkas B. Secure computation of the *k* th-ranked element. In: Advances in Cryptology-EUROCRYPT 2004. Berlin, Heidelberg: Springer-Verlag, 2004,3027:40–55.

[6] Fitzi M, Holenstein T, Wullschleger J. Multi-Party computation with hybrid security. In: Advances in Cryptology-EUROCRYPT 2004. Berlin, Heidelberg: Springer-Verlag, 2004,4:419–438.

[7] Ishai Y, Kushilevitz E. On the hardness of information-theoretic multiparty computation. In: Advances in Cryptology-EUROCRYPT 2004. Berlin, Heidelberg: Springer-Verlag, 2004,3027:439–455.

[8] Golle P, Juels A. Dining cryptographers revisited. In: Advances in Cryptology-Eurocrypt 2004. Berlin, Heidelberg: Springer-Verlag, 2004,3027:456–473.

[9] Yao AC. Protocols for secure computations. In: Proc. of the 23rd Annual Symp. on Foundations of Computer Science. IEEE, 1982. 160–164.

[10] Goldreich O, Micali S, Wigderson A. How to play any mental game. In: Proc. of the 19th Annual ACM Symp. on Theory of Computing. ACM, 1987. 218–229.

[11] Yasin S, Haseeb K, Qureshi RJ. Cryptography based e-commerce security: A review. Int'l Journal of Computer Science Issues, 2012,9(2):132–137.

[12] Sharma R. Review paper on cryptography. Int'l Journal of Research, 2015,2(5):141–142.

[13] Kumar SN. Review on network security and cryptography. Int'l Trans. of Electrical and Computer Engineers System, 2015,3(1): 1–11.

[14] Ioannidis I, Grama A. An efficient protocol for Yao's millionaires' problem. In: Proc. of the 36th Annual Hawaii Int'l Conf. on System Sciences. IEEE, 2003. 6–9.

[15] Li SD, Dai YQ, You QY. Efficient solution to Yao's millionaires' problem. Dianzi Xuebao (Acta Electronica Sinica), 2005,33(5): 769–773 (in Chinese with English abstract).

[16] Li SD, Wang DS. Efficient secure multiparty computation based on homomorphic encryption. Dianzi Xuebao (Acta Electronica Sinica), 2013,41(4):798–803 (in Chinese with English abstract).

[17] Sheikh R, Mishra DK, Kumar B. Secure multiparty computation: From millionaires problem to anonymizer. Information Security Journal: A Global Perspective, 2011,20(1):25–33.

[18] Grigoriev D, Shpilrain V. Yao's millionaires' problem and decoy-based public key encryption by classical physics. Int'l Journal of Foundations of Computer Science, 2014,25(4):409–417.

[19] Karimian AN. Efficient non-interactive secure two-party computation for equality and comparison [Ph.D. Thesis]. University of Calgary, 2015.

[20] Lipmaa H, Toft T. Secure equality and greater-than tests with sublinear online complexity. In: Proc. of the Int'l Colloquium on Automata, Languages, and Programming. Berlin, Heidelberg: Springer-Verlag, 2013. 645–656.

[21] Li SD, Wang DS, Dai YQ. *et al*. Symmetric cryptographic solution to Yao's millionaires' problem and an evaluation of secure multiparty computations. Information Sciences, 2008,178(1):244–255.

[22] Li SD, Wang DS, Dai YQ. Symmetric cryptographic protocols for extended millionaires' problem. Science in China Series F: Information Sciences, 2009,52(6):974–982.

[23] Lin HY, Tzeng WG. An efficient solution to the millionaires' problem based on homomorphic encryption. In: Proc. of the Int'l Conf.on Applied Cryptography and Network Security. Berlin, Heidelberg: Springer-Verlag, 2005. 456–466.

[24] Atallah MJ, Du W. Secure multi-party computational geometry. In: Proc. of the Workshop on Algorithms and Data Structures. Berlin, Heidelberg: Springer-Verlag, 2001. 165–179.

[25] Du W, Atallah MJ. Secure multi-party computation problems and their applications: A review and open problems. In: Proc. of the 2001 Workshop on New Security Paradigms. ACM, 2001. 13–22.

[26] Li SD, Wu CY, Wang DS, *et al*. Secure multiparty computation of solid geometric problems and their applications. Information Sciences, 2014,282:401–413.

[27] Li SD, Wang DS, Dai YQ. A secure multi-party computation solution to intersection problems of sets and rectangles. Progress in Natural Science, 2006,16(5):538–545.

[28] Lindell Y, Pinkas B. Privacy preserving data mining. Journal of Cryptology, 2002,15(3):177–206.

[29] Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978,4(11):169–180.

[30] Xu MZ, You L. Information Security and Cryptology. Beijing: Tsinghua University Press, 2007. 96–98 (in Chinese).

[31] Goldreich O. Foundations of Cryptography: Volume 2, Basic Applications. London: Cambridge University Press, 2004. 599–764.

[32] Luo Y, Shi L, Zhang C, *et al*. Privacy-Preserving protocols for string matching. In: Proc. of the 4th Int'l Conf. on Network and System Security (NSS). IEEE, 2010. 481–485.

附中文参考文献:

[15] 李顺东,戴一奇,游启友.姚氏百万富翁问题的高效解决方案.电子学报,2005,33(5):769–773.

[16] 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,2013,41(4):798–803.

[30] 徐茂智,游林.信息安全与密码学.北京:清华大学出版社,2007.96–98.

李顺东(1963－),男,河南平顶山人,博士,教授,主要研究领域为密码学,信息安全.

杨晓艺(1993－),女,硕士,主要研究领域为密码学,信息安全.

亢佳(1992－),女,硕士生,主要研究领域为密码学,信息安全.

窦家维(1963－),女,博士,副教授,主要研究领域为密码学,应用数学.