结构化加密的 PSI 协议

杨佳辉[1] 陈兰香[1),2)] 穆 怡[3)] 曾令仿[2)] 薛玉洁[1)]

[1)](福建师范大学计算机与网络空间安全学院,福建省网络安全与密码技术重点实验室 福州 350117)

[2)](之江实验室 杭州 311121)

[3)](澳门城市大学数据科学学院 澳门 999078)

摘 要 随着信息技术的快速发展,本地的存储资源和计算能力不能支持海量的数据.云计算的出现使得数据外包成为一种新的趋势,然而,数据外包使得数据的安全隐私得不到保证,可搜索对称加密(Searchable Symmetric Encryption,SSE)应运而生.传统的 SSE 方案主要针对文本类数据类型,为了实现更广泛数据类型的可搜索加密,Kamara 等人于 2010 年提出结构化加密(Structured Encryption,STE)的概念.结构化加密技术可以实现复杂数据结构的密文检索,可以用于加密社交网络图,但结构化加密方案只能实现加密数据的检索,不能对加密社交网络图作计算与统计分析.因此,我们设计隐私保护集合求交协议 max-PSI,并将其应用于结构化加密的社交网络图数据,提出结构化加密的 PSI 方案 STE_max-PSI,实现加密图中任意多个节点的邻居节点集合的最大交集大小的计算.该方案在加密的社交网络图中,可以查询任意节点亲密度最大的节点(我们认为拥有共同邻居节点最多的两个节点关系最亲密).方案通过引入混乱布隆过滤器(Garbled Bloom Filter,GBF)实现更复杂数据结构的更加丰富的查询功能,同时能够保护数据的隐私.混乱布隆过滤器在查询元素时具有可忽略的假阳性,因此,与已有的 PSI 方案相比,本方案可以极大地提高查询准确率.通过在真实的数据集上进行测试,与已有方案相比,本方案的查询结果准确率最佳.

关键词 云计算;结构化加密;加密社交网络;隐私保护集合求交;混乱布隆过滤器

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2022.02652

PSI Protocol with Structured Encryption

YANG Jia-Hui[1)] CHEN Lan-Xiang[1),2)] MU Yi[3)] ZENG Ling-Fang[2)] XUE Yu-Jie[1)]

[1)](*Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Computer and Cyber Security, Fujian Normal University, Fuzhou* 350117)

[2)](*Zhejiang Lab, Hangzhou* 311121)

[3)](*Faculty of Data Science, City University of Macau, Macao* 999078)

Abstract With the advancement of information technology, local storage and computing capacity cannot well support massive data. The emergence of cloud computing makes data outsourcing a new trend. However, data outsourcing compromises data security and privacy. Therefore, searchable symmetric encryption (SSE) was introduced as a potential solution. The traditional SSE scheme is mainly for text data types. In order to achieve searchable encryption of a wider range of data types, Kamara et al. proposed the concept of structured encryption (STE) in 2010. Structured encryption can realize searchable encryption of complex data structures. It can be used to encrypt social network graphs to achieve retrieval of encrypted data, but cannot be applied to

收稿日期:2022-02-23;在线发布日期:2022-09-26.本课题得到国家自然科学基金(62072105,61872087)、国家自然科学基金海峡联合基金重点项目(U1805263)、福建省自然科学基金资助项目(2019J01274)、浙江省"万人计划"项目(2021R52007)、之江实验室中心自设科研项目(2021DA0AM01)资助.**杨佳辉**,硕士研究生,主要研究方向为结构化加密与 PSI 协议.E-mail:1746843433@qq.com.**陈兰香**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为密码学及其应用、隐私计算.E-mail:lxiangchen@fjnu.edu.cn.**穆 怡**,博士,教授,博士生导师,主要研究领域为公钥密码学.**曾令仿**,博士,研究员,博士生导师,中国计算机学会(CCF)会员,主要研究领域为分布计算与并行处理.**薛玉洁**,硕士研究生,主要研究方向为结构化加密与知识工程.

the computation and statistical analysis of social network graphs. Therefore, we design the private set intersection protocol max-PSI, and apply it to the structured encrypted social network graph data, and propose a structured encrypted PSI scheme, STE_max-PSI, to realize any number of nodes in the encrypted graph and the calculation of the maximum intersection size of the set of neighbor nodes. In the encrypted social network graph, it can query the node with the highest affinity of any node (we think two nodes with the most common neighbors have the highest affinity). The proposed scheme utilized a garbled Bloom filter (GBF) to achieve richer query functions for more complex data structures, while protecting the privacy of data. The GBF has negligible false positives when querying elements. Compared with the existing PSI protocols, our scheme can greatly improve the query accuracy. Experiments on the real dataset show that the query accuracy of our scheme is better than the existing schemes.

Keywords　cloud computing; structured encryption; encrypted social network; private set intersection; garbled Bloom filter

1　引　言

随着互联网的发展，为了节约大规模数据的存储成本，越来越多的用户将数据外包到第三方的云存储服务器上[1]. 云存储的快速发展，使得数据加密以及密文搜索得到了极大的发展，取得了大量的研究成果. 近年来，研究者提出了各种可搜索对称加密方案，如单关键词搜索[2-5]、多关键词搜索[6-11]及动态关键词搜索[12-17].

但是，大多数密文搜索方案主要针对文本数据. 2010 年，Kamara 等人[18]提出结构化加密(Structured Encryption，STE)的概念. 将 SSE 一般化使其适用于更丰富类型的数据. 但是当前的结构化加密方案主要针对关键词检索，而没有实现统计分析等功能. 现实应用中用户可能不仅希望对云存储服务器中的数据进行简单的关键词查询，可能还希望能够对数据进行一定的计算与分析.

安全多方计算技术[19]是解决这个问题的有效手段之一. 安全多方计算允许互不信任的双方或者多方，在不揭示各自输入的前提下，共同来计算某个既定函数. 本文通过在结构化加密方案的基础上引入安全多方计算，从而实现更复杂的统计分析功能. 隐私保护集合求交(Private Set Intersection，PSI)[20]是安全多方计算的一个重要应用，可以在不泄漏非交集元素的情况下，实现对双方分别持有的集合 X 和 Y 计算其交集. 2013 年，Dong 等人[21]提出基于混乱布隆过滤器(Garbled Bloom Filter，GBF)的 PSI 协议，在处理大规模数据方面具备较好的准确率与计算效率.

在可搜索对称加密方案中，如果我们仅使用带标签数据的方案来加密社交网络图，那么我们只能执行关键词检索；类似地，如果我们仅使用支持邻居查询的图形加密方案，那么我们只能检索某个特定节点的邻居节点. 为了既能够实现关键词检索，又实现邻居查询，我们采用引入关联性的结构化加密方案，实现复杂数据结构的查询. 进一步地，为了在结构化加密的数据上作进一步的统计分析，我们利用混乱布隆过滤器，设计 max-PSI 协议，并将其应用到结构化加密数据，提出结构化加密的 PSI 协议 STE_max-PSI，实现了以下功能：

(1) 利用布隆过滤器(Bloom Filter，BF)与混乱布隆过滤器设计隐私保护集合求交协议 max-PSI，实现隐私保护的集合求交运算，进一步地，将设计的 max-PSI 应用于结构化加密的社交网络图数据，实现加密图中任意多个节点的邻居节点集合的最大交集大小的计算.

(2) 在加密的社交网络图中，可以查询任意节点亲密度最大的节点(我们认为拥有最多相同邻居节点的两个节点关系最亲密).

(3) 混乱布隆过滤器在查询元素时具有可忽略的假阳性，因此，与已有的 PSI 方案相比，本方案可以极大地提高查询准确率.

(4) 通过在真实的数据集上进行测试，与已有方案相比，本方案的查询结果准确率是最优的.

本文第 2 节介绍可搜索对称加密、结构化加密及隐私保护集合求交的相关工作；第 3 节定义符号，介绍结构化加密的系统模型及安全模型；第 4 节介

绍本方案采用的布隆过滤器与混乱布隆过滤器;第 5 节详细阐述本文提出的 STE_max-PSI 方案;第 6 节对方案的安全性进行分析;第 7 节通过实验对本文方案作性能分析;最后,对全文进行总结并指出下一步的研究工作.

2 相关工作

云计算作为一种全新的计算模式,拥有虚拟化、超大规模和易扩张等特点,并且由于其灵活的服务提供方式成为研究热点.云服务器可以向用户提供计算和存储等 IT 资源服务,用户可以租赁云平台部分存储资源,将自己的数据外包到云服务器,在节约成本的同时可以方便地共享数据.然而,当用户将数据外包给云服务器时,不可避免地会面临数据安全与隐私问题.如何保障云服务器上的数据安全是云存储服务面临的重要挑战.因此,为了保护数据安全,用户通常把数据加密后再存储到云服务器.但是,数据加密后,如何对密态数据进行查询及处理成为一个亟待解决的问题.

可搜索对称加密. 2000 年,Song 等人[2]提出第一个可搜索对称加密(Searchable Symmetric Encryption, SSE)方案 SWP,该方案采用特殊的两层加密结构来加密数据.其核心思想是分别加密每个单词,然后将一个具有特殊格式的哈希值嵌入到密文中,使用顺序扫描来搜索密文.但是其计算复杂度与每个文档的单词总数呈线性增长. 2003 年,Goh 等人[3]最早提出安全索引结构,并为索引制定了一个安全模型,称为针对自适应选择查询攻击的语义安全性(Semantic Security against Adaptive Chosen Keyword Attack, IND-CKA).他提出通过布隆过滤器来为每一个文件构造索引的方法,查询时通过对陷门关键词进行多次哈希,就能判定某个密文是否包含该陷门关键词,该方案提升了查询效率,但产生了查询误判率. 2005 年,Chang 等人[4]提出数据字典的设计以实现对关键词的精确查找,但无法实现主动攻击下可证明的语义安全性. 2006 年,Curtmola 等人[5]首次提出满足非适应性安全的 SSE 方案和满足适应性安全的 SSE 方案,该方案的查询效率与文件的最大数量呈子线性增长. 2017 年,Kim 等人[9]提出具有高效更新的前向安全动态可搜索对称加密方案,通过设计一种双字典数据结构,可以同时利用正向索引和倒排索引,从而提高了搜索效率. 2020 年,Du 等人[12]提出支持布尔查询的动态多客户端可搜索对称加密方案,该方案允许数据拥有者授权多个客户端对加密数据库执行布尔查询. Liu 等人[22]提出面向云存储的多用户可验证可搜索对称加密方案,允许多个用户执行搜索. 2021 年,Patranabis 等人[13]提出前向安全与后向安全的可搜索对称加密方案,不仅支持添加新文档,还支持删除旧文档,同时能够保证最大限度地减少向云服务器透露的信息. He 等人[14]提出具有常量客户端存储成本的安全动态可搜索对称加密方案,将更新带来的开销控制在常量范围内. Cao 等人[15]针对现有可搜索对称加密方案中的访问模式和搜索模式,提出一种高效的可搜索对称加密方案,并实现对访问模式和搜索模式的保护. Li 等人[16]提出具有前向搜索隐私保护的可搜索对称加密方案,并提出了前向搜索隐私保护的概念,即对新添加的文档进行搜索操作,不会泄露有关过去查询的任何信息. Huang 等人[23]针对非自适应的文件注入攻击,提出一种针对 SGX 文件注入攻击的高效可搜索对称加密方案.

上述方案大部分针对文本数据,然而现实生活中数据类型多种多样,如图形图像、社交网络及位置信息等,因此需要针对这些更丰富的数据类型提出更加一般化的可搜索对称加密方案.

结构化加密. 2010 年,Kamara 等人[18]最早提出结构化加密(Structured Encryption, STE)的概念,他们提出将可搜索对称加密方案一般化,使其可应用于更丰富类型的数据,如社交网络等.他们提出将数据视为一个数据结构 δ 和数据项序列 $m=\{m_i \mid i\in n\}$.设 δ 是一个具有 n 个节点的无向图,则 m 的第 i 个数据项与无向图上的第 i 个节点相关联.用户使用私钥,可以为任何查询构造令牌 τ,利用 τ 和加密的 δ 可恢复指向加密数据项 m_i 的指针.他们在文中提出关联性的概念,通过关联将文本数据部分与图结构部分建立关联,从而可以表达复杂的数据结构. 2011 年,Cao 等人[24]提出在加密图上实现隐私保护的查询方案,他们的方案允许在加密域中计算内积,但会导致误报. 2017 年,Liu 等人[25]在加密图上提出 top-k 最近邻关键字查询. Wang 等人[26]在加密图上进一步提出用于精确最短距离查询的方案,该方案支持动态更新. 2019 年,Kamara 等人[27]设计支持差分隐私统计查询的加密数据库,并提出一个支持保密直方图查询的数据加密方案. 2021 年,Kamara 等人[28]为了提高查询效率,进一步提出高效的最短路径查询的图加密方案.

隐私保护集合求交.对明文下的集合计算交集

是很容易的事情，为了保护数据隐私，将数据加密后，计算密文集合的交集则非常困难. 隐私保护集合交集(Private Set Intersection，PSI)计算是安全多方计算的一个重要方面，目的是在保护通信双方数据隐私的前提下完成数据集的交集运算. 2004 年，Freedman 等人[20]提出利用不经意多项式计算[29]，并结合同态加密，实现在半诚实和恶意模型下的高效的 PSI 协议. 2008 年，Hazay 等人[30]提出标准模型下基于不经意伪随机函数的 PSI 协议，并以此为基础设计适用于恶意敌手模型下单边模拟安全的 PSI 协议，以及适用于威慑因子为 1/2 的隐蔽敌手模型下的 PSI 协议. 2010 年，Cristofaro 等人[31]提出一种基于 RSA 盲签名的 PSI 协议. 2013 年，Dong 等人[21]提出可处理 TB 级集合元素的 PSI 协议，该协议基于混乱布隆过滤器(Garbled Bloom Filter，GBF)、秘密分享及不经意传输协议，因为没有使用开销较大的公钥密码，从而实现了较高的计算效率. 2016 年，Freedman 等人[32]提出使用不同的哈希结构来表示集合元素，以降低计算复杂度. 2017 年，Hallgren 等人[33]引入门限(threshold) PSI，为研究 PSI 提供了新的途径[34-37]. 2018 年，Chen 等人[38]提出针对恶意云服务器的完全同态加密下的 PSI 协议. PSI 计算有很广泛的应用，如有隐私保护下的人类基因检测[39]、近邻检测[40]及婚恋推荐配对等.

本文在结构化加密数据的基础上提出 max-PSI 隐私保护集合交集计算协议，针对多个集合，返回其最大交集的大小. 该方案通过引入混乱布隆过滤器 GBF，使其具有查询元素时可忽略的假阳性，从而提高查询准确率. 在社交网络图中找到满足某种关键词属性且与自己关系更加密切的节点时(我们认为两者拥有更多交集元素则表示两者关系更加密切)，本方案具有明显的优势.

3　符号定义与系统模型

3.1　符号定义

本文使用的符号如表 1 所示，下文中使用的几个函数定义如下：

(1) 伪随机函数 $F:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{\log(n)+\lambda}$.

(2) 伪随机置换 $P:\{0,1\}^{\lambda}\times\{0,1\}^{*}\rightarrow\{0,1\}^{*}$.

(3) 随机排列 $\pi:[n]\rightarrow[n]$.

(4) 伪随机函数 $f:\{0,1\}^{\eta}\times\{0,1\}^{\lambda}\rightarrow\{0,1\}^{\lambda}$.

表 1　符号定义

符号	说明	符号	说明
G	无向图	λ	安全参数，生成密钥的长度
$m=\{m_1,\cdots,m_n\}$	n 个保密数据项集合	$v=\{v_1,\cdots,v_n\}$	n 个半保密数据项集合
$M=\{(m_1,v_1),\cdots,(m_n,v_n)\}$	要加密的明文对集合	c	加密数据项后的密文集
K	密钥集合	$L=\{L_1,\cdots,L_n\}$	数组集合，L_i 表示节点 i 的所有邻居节点组成的数组
I	指针集合，数据项在集合里的标识号	J	指针集合，指针集合 I 随机置换后的结果，即 $J=\pi[I]$
$\|W\|$	关键词集合所包含的关键词个数	λ_1	构造 GBF 时添加的随机字符串的长度
l	BF 和 GBF 的数组长度	$\|L_i\|$	表示数组 L_i 的长度
$R(w)$	查询关键词 w 返回的结果集	$\|R(w)\|$	$R(w)$ 中的元素个数
$BF=\{BF_i,i\in R(w)\}$	根据关键词返回的结果集 $R(w)$ 生成的布隆过滤器集合	k	BF 与 GBF 中哈希函数个数

3.2　系统模型

系统模型如图 1 所示，主要包括三个实体：数据拥有者(Data Owner，DO)、云服务器(Cloud Server，CS)和用户(User，U). 图中 v_i 表示数据项 m_i 的半保密数据，T 表示根据关键词 w 生成的陷门 $P_{K_1}(w)$，P 是一个伪随机置换函数，D_i 表示 $[\langle\pi[i],v_i\rangle\oplus F_{K_2}(w)]$，$\pi_i$ 表示 $\pi[i]$. 结构化加密的 max-PSI 方案的整个执行过程定义为一个六元组 $STE\_max\text{-}PSI=(Gen,BuildIndex,Enc,Token,Search,Dec)$ 系统中每个算法的描述如下：

(1) $Gen(1^{\lambda})\rightarrow K$：生成密钥. 数据拥有者输入安全参数，输出 $K=\{K_1,K_2,K_3\}$，其中 K_1 和 K_2 用于加密关键词索引，K_3 用于加密数据项.

(2) $BuildIndex(K,L,G,M,W)\rightarrow(Index,L^{*})$：数据拥有者构建关键词倒排索引，对倒排索引使用 K_1 加密得到 $Index$，构建邻居节点数组集合并使用 K_2 加密得到 L^{*}.

(3) $Enc(K_3,M)\rightarrow c$：对明文对集合 M 中的每一个数据项使用伪随机排列进行混淆，即 $\pi[n]\rightarrow\pi[n]$，然后使用 K_3 加密得到 c.

(4) $Token(K,w,m_a)\rightarrow(\tau,v_a)$：用户向数据拥有者发出搜索请求、待搜索关键词以及感兴趣的数据项 $m_a(a\in n)$，数据拥有者返回检索陷门 $\tau:=(P_{K_1}(w),F_{K_2}(w))$ 及一个半保密数据项 $v_a=P_{K_1}(m_a)$，

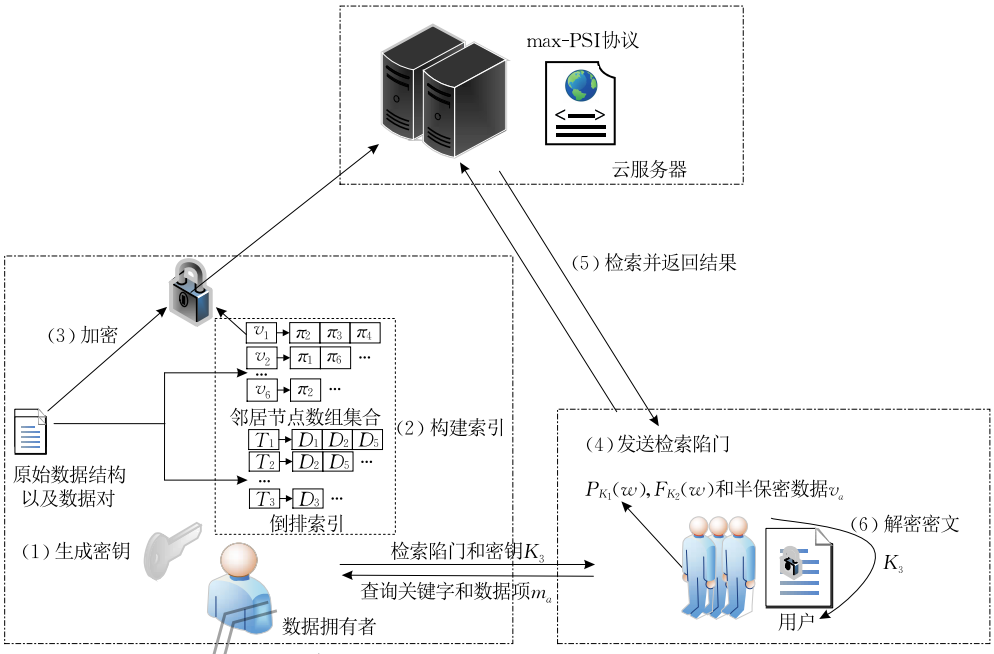


图 1 系统模型

用户发送检索陷门和半保密数据项 v_a 到云服务器上进行检索.

(5) $Search(c,\tau,v_a,L^*,Index)\rightarrow c_j$:云服务器收到检索陷门后,根据 $P_{K_1}(w)$ 找到倒排索引入口,再根据 $F_{K_2}(w)$ 得到 $\langle\pi[i],v_i\rangle$;云服务器根据半保密数据项 v_i 以及 v_a 检索邻居节点数组集合,云服务器运行 max-PSI 协议,返回检索结果给用户.

(6) $Dec(K_3,c_j)\rightarrow m_j$:用户利用从数据拥有者授权获得的密钥 K_3 将 c_j 解密得到 m_j.

3.3 安全模型

云服务器被认为是半诚实的,会正确地执行检索任务并返回正确的结果,但是云服务器会好奇数据拥有者上传的密文数据以及检索陷门,有可能在执行检索过程中记录一些信息.根据云服务器掌握的信息,有以下两种安全模型:

(1)已知密文模型.敌手知晓数据拥有者存储的各种密文信息,包括加密后的安全索引,在检索过程中提交的陷门、密文数量等.但是敌手并不知晓任何密钥,敌手只能通过上述已知密文来进行已知密文攻击.

(2)已知背景信息模型.在这种模型下,敌手不仅知晓已知密文模型下的所有信息,还会对信息进行统计分析,包括对查询结果进行记录,对比不同查询陷门之间的关系等.例如云服务器可以对多次不同的查询对比其查询部门与返回结果,这样云服务器就可知晓是否有过相同查询,并进一步进行分析.

3.3.1 泄露函数

针对已知密文模型,我们给出了泄露函数 $\mathcal{L}_1$.针对已知背景信息模型,我们给出了泄露函数 $\mathcal{L}_2$.本小节将会介绍结构化加密方案泄露函数的一般定义,其中 $Index$ 和 L^* 是加密后的索引.

(1)泄露函数 $\mathcal{L}_1$:主要由初始化阶段构成,泄露函数 $\mathcal{L}_1$ 指明了结构化加密方案在 (G,M) 的基础上泄露如密文项的个数、返回结果集合的大小等信息.即 $\mathcal{L}_1=\{n,|R(w)|\}$.

(2)泄露函数 $\mathcal{L}_2$:主要由查询模式和交集模式构成,即 $\mathcal{L}_2=QP(q_t)+IP(q_t)$.

查询模式 $QP(q_t)$:查询模式存储了当前查询操作与之前查询操作是否相同.q 是一个非空的查询陷门集合,查询模式是一个二进制向量,对于任何查询陷门 $q_t\in q$,当且仅当时 $q_t=q_i$,向量的第 i 个元素置为 1.即云服务器可以通过对比查询陷门来判断是否为重复查询.

交集模式 $IP(q_t)$:交集模式显示了何时哪些相同密文项被访问,但不显示哪些密文项被访问(因为本文会通过随机排列隐藏它们之间的位置关系).交集模式的第 t 个元素位置存放 $\pi[c_t]$,其中 $c_t:=Search(Index,L^*,M,q_t)$.

3.3.2 结构化加密的安全性

结构化加密方案的安全性可以通过真实游戏 $Real_{\Sigma,\mathcal{A}}(\lambda)$ 和模拟游戏 $Ideal_{\Sigma,\mathcal{A},\mathcal{S}}(\lambda)$ 来证明.其中 $\mathcal{A}$ 是敌手,$\mathcal{S}$ 是模拟器,$\mathcal{L}_1$ 和 $\mathcal{L}_2$ 是泄露函数,γ 是生成的加密索引结构.下面给出两个游戏的具体定义:

(1) $Real_{\Sigma,\mathcal{A}}(\lambda)$：挑战者先运行 $K\leftarrow Gen(1^{\lambda})$ 生成一个保密密钥 K，敌手 $\mathcal{A}$ 输出要加密的数据结构以及数据项 (G,M)，并接受由挑战者输出的 $(Index,L^{*},c)\leftarrow(K,G,M)$. 敌手进行多项式次数的自适应查询，并且对于每个查询 q，从挑战者接收令牌 $\tau\leftarrow(K,q)$. 最后，敌手 $\mathcal{A}$ 输出一个比特位 b.

(2) $Ideal_{\Sigma,\mathcal{A},\mathcal{S}}(\lambda)$：敌手 $\mathcal{A}$ 输出要加密的数据结构以及数据项 (G,M)，并接受由模拟器 $\mathcal{S}$ 通过运行泄露函数 $\mathcal{L}_1(G,M)$ 生成的 $(Index,L^{*},c)$. 敌手 $\mathcal{A}$ 进行多项式次数的自适应查询，并且对于每个查询 q，模拟器 $\mathcal{S}$ 通过运行 $(\mathcal{L}_2(G,q),V_I)$ 返回一个令牌 τ 给敌手 $\mathcal{A}$，其中 $I=Query(G,q)$. 最后，敌手 $\mathcal{A}$ 输出一个比特位.

我们称方案 Σ 对抗自适应选择查询攻击(IND-CKA)是 $(\mathcal{L}_1,\mathcal{L}_2)$ 安全的，如果对于所有的多项式时间敌手 $\mathcal{A}$，总存在一个多项式时间的模拟器 $\mathcal{S}$，使得：$|\Pr[Real_{\Sigma,\mathcal{A}}(\lambda)=1]-\Pr[Ideal_{\Sigma,\mathcal{A},\mathcal{S}}(\lambda)=1]|\leq negl(\lambda)$.

4　预备知识

4.1　布隆过滤器

布隆过滤器(Bloom Filter，BF)由一个二进制向量以及一系列的随机散列函数组成. 它可以用于检索一个关键词是否在某个关键词集合当中，其查询的存储开销与效率都优于一般的结构. 布隆过滤器可以看作是一个拥有 x 位的向量. 若存在一个集合 m 和 k 个相互独立的散列函数 $\{h_1,\cdots,h_k\}$，通过散列函数，可以将集合元素映射到布隆过滤器中. 布隆过滤器的初始值都为 0，将经过散列函数映射的地址 $\{BF[h_1(m_i)],\cdots,BF[h_k(m_i)]\}$ 的相应位置为 1. 其结构如图 2(a)所示.

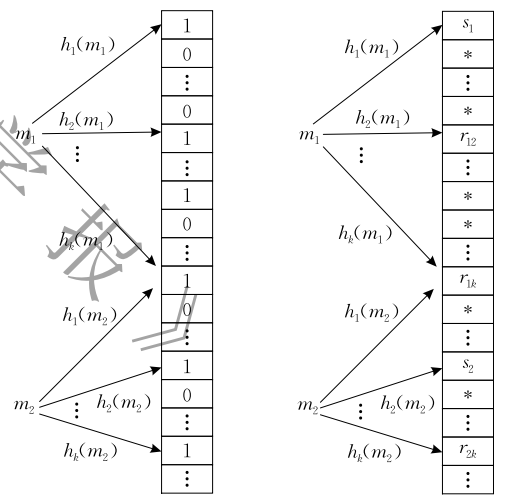
4.2　混乱布隆过滤器

2013 年，Dong 等人[21]为了更好地表示集合元素，提出改进布隆过滤器的混乱布隆过滤器(Garbled Bloom Filter，GBF). GBF 利用秘密共享的特性，更好地解决了布隆过滤器假阳性的问题，使其具有可忽略的假阳性. 混乱布隆过滤器形式上是将布隆过滤器中的位数组转换成字符串数组，数组中的每一个字符串长度为安全参数 λ_1，我们可以通过调节这个参数来获得需要的安全性.

设存在一个集合 m 和 k 个相互独立的散列函数 $\{h_1,\cdots,h_k\}$，当插入元素时，依次用 k 个散列函数将元素 m_i 映射到字符串数组的 k 个位置上去，取第一次散列地址为 inm，对于之后的散列地址，如果为空，则写入一个长度为 λ_1 的随机字符串，否则保持不变. 然后将除了散列地址为 inm 以外的所有散列地址的字符串进行异或，并将异或结果与 m_i 异或，将得到的值写入 GBF 的 inm 位置，即将异或值赋给 $GBF[inm]$. 在插入完所有元素后，将所有未被赋值的位置写入一个随机字符串.

对于混乱布隆过滤器，如果 y 不在集合 m 中，那么 k 个字符串异或结果等于 y 的概率是关于 λ_1 的可忽略函数. 同时，发生冲突的概率也是关于哈希函数个数 k 的可忽略函数. 混乱布隆过滤器的结构如图 2(b)所示，其中 $(r_{11},r_{12},\cdots,r_{1k})$，$(r_{21},r_{22},\cdots,r_{2k})$ 及“*”为随机字符串 $\{0,1\}^{\lambda_1}$，因为 $h_1(m_1)$ 位置为空，所以 $s_1=r_{12}\oplus r_{13}\oplus\cdots\oplus r_{1k}\oplus m_1$；同时，因为 $h_1(m_2)$ 位置不为空，且位置 $h_2(m_2)$ 为空，所以 $s_2=r_{1k}\oplus r_{23}\oplus\cdots\oplus r_{2k}\oplus m_2$.

当要判断元素 m_x 是否在集合中，计算 $GBF[h_1(m_x)]\oplus GBF[h_2(m_x)]\oplus\cdots\oplus GBF[h_k(m_x)]$ 是否为 m_x，若是，则在集合中，否则，不在集合中. 利用混乱布隆过滤器 GBF 具有可忽略的假阳性优点，我们将利用其构建具有更高准确率的 max-PSI 协议.



(a) 布隆过滤器　　(b) 混乱布隆过滤器

图 2　添加两个元素后的布隆过滤器和混乱布隆过滤器

5　结构化加密的 max-PSI

本节将详细描述在结构化加密的数据上实现隐私保护集合交集计算的方案 STE_max-PSI(maximum PSI with STE).

我们考虑一个社交网络图，节点对应网络中的个体，每个个体都有自己的关键词(对应 SSE 中的关键词)，若节点 i 与节点 j 有关联，则在图中存在对应的边 (i,j). 该方案结合 SSE(针对标签数据)与

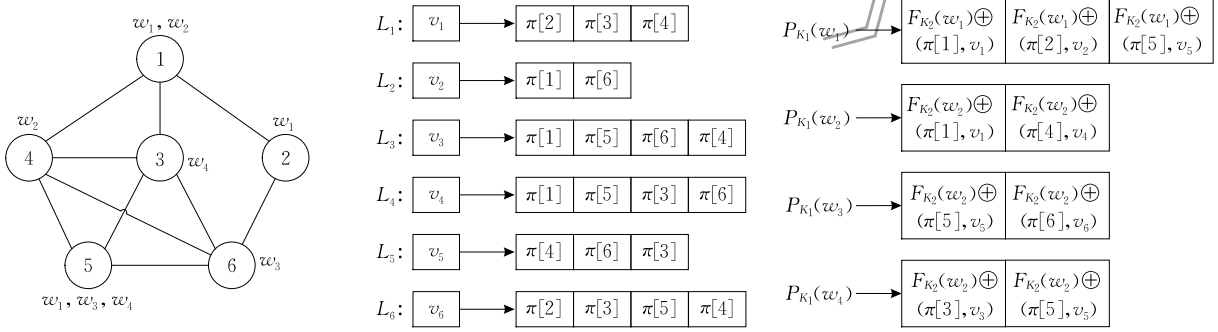
图加密(针对社交网络图),通过引入半保密数据,使检索某个关键词能返回被关键词标记的节点的邻居节点,然后通过提出的 max-PSI 协议,在加密情况下返回与某个节点拥有交集邻居节点最多的节点.社交网络图通常表现出更加丰富的结构,我们引入关联性概念,通过半保密数据 v,将针对标签数据的SSE方案与支持邻居查询的图加密方案结合起来.但是,如果仅将SSE方案与图加密方案结合,检索结果可能是大量的节点信息,即关键词检索得到的节点及其邻居节点.因此,在上述方案基础上,我们提出 max-PSI 协议,使得返回一个跟某个特定节点亲密度最大的节点(我们认为拥有相同邻居节点个数越多,则两者越亲密).

定义 1. 关联性(Associativity).关联性是结构化加密中一个很重要的概念,因为复杂结构类型的数据可能既有文本数据又有相关图结构,为了实现文本数据与图结构的关联,Kamara 等人[18]提出关联性的概念,用于将多种类型数据之间建立关联.如果一个方案将半保密数据项 v_i 和保密数据项 m_i 相关联,即查询操作除了返回指针 $j=\pi(i)$,还会返回半保密数据 v_i,那么我们就认为这个方案引入了关联性.我们将通过定义加密算法的消息空间来实现,使得消息空间除了可以获取结构和保密数据项 m_i 以外,还可以获取半保密数据项 v_i,通过半保密数据项 v_i 将复杂数据类型的两种结构建立关联.

定义 2. 诱导置换(Induced Permutation).诱导置换是结构化加密中另一个很重要的概念,考虑的是明文数据项序列 m 中数据项的位置与它们在密文 c 中位置的相关性.设 π 是 $[n]$ 上的一个置换,对于所有的数据项标识号 i,有 $m_i:=Dec_{K_3}(c_{\pi[i]})$,则称 π 为明文 m 和密文 c 上的诱导置换.诱导置换的引入使结构化加密方案部分地隐藏访问模式,因为诱导置换通过在 m 和 c 之间引入随机置换破坏了明文和密文位置的相关性.

5.1 max-PSI 协议设计

在 max-PSI 协议中,我们使用了布隆过滤器、混乱布隆过滤器以及数组数据结构.其中,布隆过滤器和混乱布隆过滤器是根据社交网络中节点的邻居节点生成.对于数组集合 L,我们使用 L_i 存放社交网络图中节点 i 的所有邻居节点.我们在图 3 中给出一个具体的示例,图 3(a)是一个有 6 个节点的社交网络图,图 3(b)是该图对应的数组集合 L,图 3(c)为我们构造的关键词倒排索引.若我们要检索关键词 w_1,则计算 $P_{K_1}(w_1)$,通过图 3(c)找到被 w_1 标记的加密数据对,然后计算 $F_{K_2}(w_1)$,与加密数据对异或得到半保密数据 $\{v_1,v_2,v_5\}$.利用图 3(b)得到 $\{L_1,L_2,L_5\}$.若我们需要查询与节点 4 最亲密的节点,则使用 L_1,L_2,L_5 生成的 BF_1,BF_2,BF_5 与 GBF_4 运行 max-PSI 隐私计算协议,可以得到 v_5,因此与节点 4 亲密度最高的是节点 5,并将对应的密文 $c_{\pi[5]}$ 返回给用户,用户使用密钥 K_3 对其解密得到明文 m_5.



(a) 社交网络图 (b) 图对应用的链表 (c) 关键词索引

图 3 方案示例图

5.1.1 Bloom Filter 构造算法

设 BF 是根据节点 i 的邻居节点数组 L_i 产生的布隆过滤器,我们用 l 表示 BF 的长度,随机选择 k 个互相独立的哈希函数 $\{h_1,\cdots,h_k\}:\{0,1\}^*\rightarrow\{0,1\}^l$ 作为 BF 的哈希函数集.Bloom Filter 的构造包括以下几个阶段,详细算法如算法 1 所示.

(1)初始化阶段:将 BF 的所有位都置 0.

(2)插入阶段:对于数组 L_i 的每个元素 $\pi[j]$,计算出其 k 个哈希值,并将 BF 里的 $\{BF[h_1(\pi[j])],\cdots,BF[h_k(\pi[j])]\}$ 位置 1.重复此操作,直到所有元素全部插入.

(3)查询阶段:判断元素 y 是否在数组集合 L_i 中,计算 y 的 k 个哈希值,检查 $\{BF[h_1(y)],\cdots,BF[h_k(y)]\}$ 是否都为 1.若是,则 $y\in L_i$,返回 true,反之,$y\notin L_i$,

返回 false.

算法 1.　BF 构造算法.

输入：$L,\{h_1,\cdots,h_k\}$

输出：BF

1. 初始化：FOR $i=1$ to l
2. $BF[i]=0$
3. END FOR
4. 插入：FOR $j=1$ to $|L_i|$
5. FOR $i=1$ to k
6. $BF[h_i(\pi[j])]=1$
7. END FOR
8. END FOR
9. 查询：FOR $i=1$ to $k-1$
10. $tmp=BF[h_i(y)]+BF[h_{i+1}(y)]$
11. END FOR
12. IF $tmp==k$ THEN
13. RETURN true
14. ELSE
15. RETURN false

5.1.2　Garbled Bloom Filter 构造算法

设 GBF 是根据节点 i 的邻居节点数组 L_i 产生的混乱布隆过滤器，l 表示 GBF 的长度，利用 BF 中互相独立的 k 个哈希函数 $\{h_1,\cdots,h_k\}:\{0,1\}^*\rightarrow\{0,1\}^l$ 作为 GBF 的哈希函数集. Garbled Bloom Filter 的构造包括以下几个阶段，详细算法如算法 2 所示.

(1) 初始化阶段：将 GBF 的所有位都置空.

(2) 插入阶段：对于数组 L_i 的每个元素 $\pi[j]$，计算出其 k 个哈希值. 取第一次空值哈希地址为 inx，对于之后的哈希地址，如果为空，则将其置为一个长度为 λ_1 的随机字符串，否则保持不变. 将除了哈希地址为 inx 以外的所有哈希地址的字符串进行异或，并将异或结果与 $\pi[j]$ 进行异或，将得到的值赋给 $GBF[inx]$. 重复此操作，直到所有元素全部插入. 将所有未被赋值的位置写入一个随机字符串.

(3) 查询阶段：判断元素 y 是否在数组 L_i 集合中. 首先计算 y 的 k 个哈希值，检查 $GBF[h_1(y)]\oplus\cdots\oplus GBF[h_k(y)]$ 是否为 y，若是，则 $y\in L_i$，返回 true，反之，$y\notin L_i$，返回 false.

算法 2.　GBF 构造算法.

输入：$L,\{h_1,\cdots,h_k\}$

输出：GBF

1. 初始化：FOR $i=1$ to l
2. $GBF[i]=$null
3. 插入：FOR $i=0$ to $|L_i|$
4. $inx=$null
5. $final=$null
6. FOR $j=1$ to k
7. $x=h_j(\pi[i])$
8. IF $GBF[x]==$null THEN
9. IF $inx==$null THEN
10. $inx=x$
11. ELSE
12. $GBF[x]\leftarrow\{0,1\}^{\lambda_1}$
13. $final=final\oplus GBF[x]$
14. END IF
15. ELSE
16. $final=final\oplus GBF[x]$
17. END IF
18. $GBF[inx]=final$
19. END FOR
20. END FOR
21. FOR $z=0$ to l
22. IF $GBF[z]==$null THEN
23. $GBF[z]\leftarrow\{0,1\}^{\lambda_1}$
24. END IF
25. END FOR
26. 查询：FOR $i=0$ to $k-1$
27. $tmp=GBF[h_i(y)]\oplus GBF[h_{i+1}(y)]$
28. END FOR
29. IF $tmp==y$ THEN
30. RETURN true
31. ELSE
32. RETURN false
33. END IF

5.1.3　max-PSI 协议

设 GBF_a 是云服务器根据我们感兴趣的节点 a 的邻居节点数组 L_a 生成的混乱布隆过滤器，对于 $|R(w)|$ 个数组 $L_x(x\in R(w))$ 均生成对应的布隆过滤器 BF_x. 设 $count_x(x\in R(w))$ 为节点 x 与节点 a 的相同邻居节点的个数，也是 BF_x 与 GBF_a 中相同元素的计数. max-PSI 协议的构造包括以下几个阶段，详细算法如算法 3 所示.

(1) 初始化：用户随机选择 l 个随机数 $r_i\leftarrow\{0,1\}^{\lambda_1}(0\le i\le l)$ 组成字符串数组 A 作为协议的元素，并将字符串数组 A 发送到云服务器. 将 $count_x(x\in R(w))$ 全部初始化为 0.

(2) 求交：对于每个 BF_x，若 $BF_x[i]=1$，选择 $GBF_a[i]$；若 $BF_x[i]=0$，选择 $A[i]$，最后得到 $GBF_{a\cap x}$（BF_x 与 GBF_a 的交集）.

(3) 输出：对于 $L_x(x\in R(w))$ 里存储的每个元素 b，计算 $GBF_{a\cap x}[h_1(b)]\oplus\cdots\oplus GBF_{a\cap x}[h_k(b)]$，

若结果为 b,则 $count_x$ 加 1,否则 $count_x$ 不变.重复此操作,计算出所有的 $count$ 值,比较得出最大的 $count$ 值,将对应的 v_x 返回给云服务器.

算法 3. max-PSI 协议构造算法.

输入:GBF_a,$BF_x(x\in R(w))$,A

输出:$v_{\max}(count_{\max}=\max\{count_i\}_{i=1}^{l})$

1. 初始化:令 $A[i]_{i=1}^{l}\leftarrow\{0,1\}^{\lambda_1}$,max=1
2. FOR x in $R(w)$ THEN
3. $count_x=0$
4. END FOR
5. 求交:FOR x in $R(w)$ THEN
6. FOR $i=1$ to l THEN
7. IF $BF_x[i]==1$ THEN
8. $GBF_{a\cap x}[i]=GBF_a[i]$
9. ELSE
10. $GBF_{a\cap x}[i]=A[i]$
11. END IF
12. END FOR
13. END FOR
14. 输出:FOR x in $R(w)$ THEN
15. FOR $i=1$ to $|L_x|$ THEN
16. $b=L_x[i]$
17. FOR $j=1$ to $k-1$ THEN
18. $tmp=GBF_{a\cap x}[h_j(b)]\oplus GBF_{a\cap x}[h_{j+1}(b)]$
19. END FOR
20. IF $tmp==b$ THEN
21. $count_x=count_x+1$
22. END IF
23. END FOR
24. END FOR
25. FOR x in $R(w)$ THEN
26. IF $count_{\max}<count_x$
27. max=x
28. END IF
29. END FOR
30. RETURN $v_{\max}$

因为 L 中存放的是节点标识号的置换,可以保护节点信息.BF 与 GBF 用于高效地判断某元素是否属于某个集合,也不会泄露关于节点的任何信息.因此,max-PSI 协议可以在保护节点 $x(x\in R(w))$ 隐私信息的同时,求出与节点 a 拥有最多相同邻居的节点.

5.2 STE_max-PSI 方案

一个社交网络图可以被视为是一个元组(G,m),由一个无向图 G 和数据项 m(也可以称为图中节点)组成.对图中每个节点数据项 m_i,生成一个半保密数据项 v_i,使其可以索引到节点 i 的邻居节点数组 L_i.当输入一个数据项 m_a 与关键词 w,本方案可以实现:(1)返回与关键词 w 相关的节点;(2)返回与数据项 m_a 拥有相同邻居节点最多的节点.由于云服务器是诚实但好奇的,我们需要对存放在云服务器上的所有数据进行加密以保证其安全性.本方案的加密主要通过:

(1)诱导置换打乱数据项的标识号,即通过 π:$[n]\rightarrow[n]$来破坏明文数据项与密文数据项的对应关系;

(2)对构造的每个邻居节点数组 L 进行填充,使其元素个数相同;

(3)使用伪随机函数 F 扰乱每个数据对 M;

(4)使用对称密码算法加密保密数据项 m.

设$\mathcal{M}$是消息空间,所有数据对$(m,v)\in\mathcal{M}$.我们用$|m|=|v|=n$表示保密数据项和半保密数据项的个数.STE_max-PSI 方案包括六个多项式时间算法(Gen,BuildIndex,Enc,Token,Search,Dec),分别描述如下.

5.2.1 $Gen(1^\lambda)\rightarrow K$

首先由数据拥有者进行初始化,生成保密密钥.数据拥有者输入安全参数 λ,输出保密密钥 $K=(K_1,K_2,K_3)$,其描述见算法 4.

算法 4. $Gen(1^\lambda)\rightarrow K$.

输入:安全参数 λ

输出:$K=\{K_1,K_2,K_3\}$

1. FOR $i=1$ to 3
2. $K_i\leftarrow f(k_1,\lambda)$
3. END FOR
4. RETURN $K=\{K_1,K_2,K_3\}$

5.2.2 $BuildIndex(K,L,G,M,W)\rightarrow(Index,L^*)$

设数组集合 L 是每个节点的所有邻居节点数组,假设节点 i 的邻居节点有$\{m_1,m_2,m_3\}$,则数组 L_i 中存放$\{\pi[1],\pi[2],\pi[3]\}$.BuildIndex 算法主要包括以下几个步骤,其描述见算法 5.

(1)为了使所有邻居节点数组的长度一样,对数组集合 L 进行填充,令填充后的数组集合为 L^*,设 $max_L=\max(|L_1|,\cdots,|L_n|)$,则$|L_i^*|=max_L$.计算 $v_i:=P_{K_1}(m_i)$,使其链接到 L_i^*.

(2)对关键词$w\in W$,计算 $P_{K_1}(w)$,使其链接到关键词索引 $Index$.根据 $P_{K_1}(w)$,将$\langle(\pi[i],v_i)\rangle\oplus F_{K_2}(w)$存储到对应的关键词索引数组中.

(3)数据拥有者将构造的关键词索引 $Index$,L^* 发送至云服务器.

算法 5.　$BuildIndex(K,L,G,M,W)\rightarrow(Index,L^*)$.

输入：K,L,G,M,W

输出：$Index,L^*$

1. 填充 L 使其长度与$|L^*|$相同
2. FOR $i=1$ to n
3. 　$v_i=P_{K_1}(m_i)$
4. END FOR
5. 对于每个关键词 $w\in W$，根据 G 和 M 建立关键词索引 $Index$. 对于每个被关键词 w 标记的节点 i，计算 $P_{K_1}(w)$ 并将$\langle(\pi[i],v_i)\rangle\oplus F_{K_2}(w)$存储到 $Index$ 项中.
6. RETURN $Index,L^*$

5.2.3　$Enc(K_3,M)\rightarrow c$

对于数据项集合 m，令 m^* 是对 m 进行诱导排列后得到的数据项序列，则有 $c_j\leftarrow Enc_{K_3}(m_j^*)$，其中 $j=\pi[i](1\leq j\leq n)$. 数据拥有者将密文 $c=(c_1,\cdots,c_n)$发送至云服务器. 其描述如算法 6 所示.

算法 6.　$Enc(K_3,M)\rightarrow c$.

输入：K_3,M

输出：c

1. 解析 M 为 m 和 v
2. FOR $i=1$ to n
3. 　$m^*[i]=m[\pi[i]]$
4. END FOR
5. FOR $i=1$ to n
6. 　$c_i=Enc_{K_3}(m^*[i])$
7. END FOR
8. RETURN $c=(c_1,c_2,\cdots,c_n)$

5.2.4　$Token(K,w,m_a)\rightarrow(\tau,v_a)$

用户向数据拥有者发送待搜索关键词以及感兴趣的数据项 $m_a(a\in n)$，数据拥有者返回检索陷门 $\tau:=(P_{K_1}(w),F_{K_2}(w))$及一个半保密数据项 $v_a=P_{K_1}(m_a)$，用户发送检索陷门和半保密数据项 v_a 到云服务器上进行检索，其描述见算法 7.

算法 7.　$Token(K,w,m_a)\rightarrow(\tau,v_a)$.

输入：K,w,m_a

输出：τ,v_a

1. 解析 K 为(K_1,K_2,K_3)
2. 计算 $P_{K_1}(w),F_{K_2}(w)$
3. 计算 $v_a:=P_{K_1}(m_a)$
4. $\tau:=(P_{K_1}(w),F_{K_2}(w))$
5. RETURN (τ,v_a)

5.2.5　$Search(c,\tau,v_a,L^*,Index)\rightarrow c_j$

云服务器将令牌 τ 解析为$(P_{K_1}(w),F_{K_2}(w))$，根据 $P_{K_1}(w)$找到对应的关键词索引，计算$\langle\pi[i],v_i\rangle=\tau\oplus F_{K_2}(w)$，输出 v_I. 云服务器根据 v_I得到邻居节点数组集合 L_I^*，根据 v_a得到单个数组 L_a^*. 云服务器对邻居节点数组集合 L_I^* 里的每个数组都生成对应的 BF，对数组 L_a^* 生成 GBF_a. 云服务器运行 max-PSI 协议，返回半保密数据 $v_{\max}$，然后将密文 c_j（$j=\pi[max]$）发送给用户，其描述见算法 8.

算法 8.　$Search(c,\tau,v_a,L^*,Index)\rightarrow c_j$.

输入：$c,\tau,v_a,L^*,Index$

输出：c_j

1. 解析 τ 为$(P_{K_1}(w),F_{K_2}(w))$
2. 对于$P_{K_1}(w)$中的每个元素 i，计算$v_{I_i}:=[\langle\pi[i],v_i\rangle\oplus F_{K_2}(w)]\oplus F_{K_2}(w)$
3. 根据 v_I,v_a找到对应的邻居节点数组$L_I^*\leftarrow v_I$与$L_a^*\leftarrow v_a$
4. 根据数组 L_I^* 生成对应的 BF，根据数组 L_a^* 生成对应的 GBF
5. 运行 $max\text{-}PSI(L_I^*,L_a^*,A)$得到 $v_{\max}$
6. 计算 $j=\pi[max]$
7. RETURN c_j

5.2.6　$Dec(K_3,c_j)\rightarrow m_j$

用户使用从数据拥有者处得到的密钥 K_3，计算 $m_j:=Dec_{K_3}(c_j)$从而得到明文，如算法 9 所示.

算法 9.　$Dec(K_3,c_j)\rightarrow m_j$.

输入：K_3,c_j

输出：m_j

1. 计算 $m_j:=Dec_{K_3}(c_j)$
2. RETURN m_j

6　安全性分析

为了使关联性（Associativity）能够起作用，我们需要两个附加属性. 第一个属性要求泄露函数 $\mathcal{L}_1$ 仅依赖于保密数据项，而不依赖于半保密数据项. 这是因为在我们的方案中，半保密数据项存放的是一个子令牌，如果允许泄露函数 $\mathcal{L}_1$ 依赖于半保密数据项，那么就有可能会泄露额外的信息. 第二个属性要求以不同的顺序执行查询不会影响泄露的信息. 方案会预先计算子令牌并保存在半保密数据项中，当我们在构造的复杂结构中执行查询操作时，这些子令牌会以不同的顺序显示出来，因此，我们要保证预先计算的子令牌造成的泄露与方案执行查询时生成的子令牌造成的泄露相同.

定义 3.　可链接（Chainability）. 一个结构化加密方案的泄露函数$(\mathcal{L}_1,\mathcal{L}_2)$针对自适应选择查询攻击是安全的，则该结构化加密方案是可链接的，并且泄露函数 $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 满足以下属性：

(1)半保密结构独立性(semi-private independence):存在泄露函数$\mathcal{L}_1$使得$\mathcal{L}_1(\delta,(m,v))=\mathcal{L}_1(\delta,m)$.

(2)顺序独立性(order independence):存在一种变换$\mathcal{Z}$,使得对于任何结构的G与任何一组查询序列$\{q_1,\cdots,q_t\}$及任何随机排列p都有:$\mathcal{Z}(\mathcal{L}_2(G,q_1),\cdots,\mathcal{L}_2(G,q_t))=(\mathcal{L}_2(G,q_{p(1)}),\cdots,\mathcal{L}_2(G,q_{p(t)}))$.

如果本文方案STE_max-PSI是一种可链接的结构化加密方案,且满足:

$\mathcal{L}_1(G,M,L^*,Index)=(n,max_L,|R(w)|)$,

$\mathcal{L}_2(G,Index,w,L^*)=(QP(w),IP(w),QP(i),IP(i))$,

其中,$max_L=\max(|L_1|,\cdots,|L_n|)$,$QP(w)$揭示了当前查询是否出现在先前的查询中,$IP(w)$揭示了何时哪些相同的密文数据项被访问,$QP(i)$揭示了$R(w)$的第$i$个节点是否出现在先前的查询中,$IP(i)$揭示了$R(w)$的第$i$个节点的邻居节点是否也是当前或者先前查询中的某个节点的邻居节点,那么我们的方案针对自适应选择查询攻击(Adaptive Chosen Query Attacks,CQA)是$(\mathcal{L}_1,\mathcal{L}_2)$安全的.

定理1. 本文提出的方案满足已知密文模型下的安全性.

证明. 已知密文模型中,敌手$\mathcal{A}$不能通过密文、加密的数据结构及构造的查询陷门来获取任何保密数据项的信息.在本文方案中,云服务器仅知道密文集合c、加密的数据结构$Index$与L^*、查询陷门τ以及在查询中的半保密数据项v.因此,此情况下云服务器可以进行唯密文攻击(Ciphertext only Attack,COA).由于本方案利用CPA(Chosen Plaintext Attacks)安全的密码算法采用密钥K_3对保密数据项进行加密,且密钥K_3仅在数据拥有者和用户之间共享,数据项m_i加密后的密文c_i存放在云服务器中,不受云服务器额外的保护.因此,概率多项式时间敌手$\mathcal{A}$不能以超过1/2的概率来区分密文c_i是对哪个明文保密数据项加密得到的结果.

敌手$\mathcal{A}$可以通过对所有密文数据项进行统计来知晓所有数据项个数为n,同时对于加密的数据结构L_i来说,敌手$\mathcal{A}$可以获取填充随机字符串后的邻居节点数组长度max_L,对关键词w进行检索,敌手$\mathcal{A}$可以获取被关键词w标记的节点个数$|R(w)|$.满足$\mathcal{L}_1(G,M,L^*,Index)=(n,max_L,|R(w)|)$.因此,本文提出的方案对自适应选择查询攻击是$\mathcal{L}_1$安全的. 证毕.

定理2. 本文提出的方案满足已知背景模型下的安全性.

证明. 已知背景模型下,我们假设云服务器除了知晓已知密文模型下的所有知识外,还可以通过统计分析获取更多信息,比如揭示当前检索关键词是否在之前的检索中出现过,何时哪些密文数据项被再次访问,某个节点是否出现在先前查询中以及某个节点的邻居节点是否出现在先前或者当前查询中其它节点的邻居节点中.我们通过挑战者游戏表明,对于所有敌手A,真实实验$Real$和理想实验$Ideal$将以可忽略的近似概率输出1.其中$\mathcal{A}$为敌手,$\mathcal{S}$为模拟器,$\mathcal{B}$为挑战者.

(1)$Game_0$:执行这个游戏相当于执行真实实验$Real_{\Sigma,\mathcal{A}}(\lambda)$.挑战者$\mathcal{B}$通过运行$Gen(1^\lambda)$产生密钥$K=(K_1,K_2,K_3)$,敌手$\mathcal{A}$输出$(G,M,L)$并且接收挑战者$\mathcal{B}$通过运行$BuildIndex(K,L,G,M)$产生的$(Index,L^*)$和$Enc(K,M)$产生的$c$,敌手$\mathcal{A}$自适应的输出一组查询,对每个查询,敌手$\mathcal{A}$接收一个令牌$\tau\leftarrow Token(K,w)$.最后,敌手$\mathcal{A}$输出一个比特位$b$.

(2)$Game_1$:$Game_1$与$Game_0$相似,除了在构造方案的第二步,敌手$\mathcal{A}$接收的$(Index,L^*)$被替换成$\mathcal{S}(\mathcal{L}_1(G,M,L^*,Index))$,并且对于每个关键词$w$的令牌$\tau$计算如下.用$R(w)$表示关键词$w$标记的节点集合,$v_w:=(v_i)_{i\in R(w)}$表示被关键词$w$标记的节点的半保密数据项集合,$\mathcal{A}$计算$\tau\leftarrow\mathcal{S}(\mathcal{L}_2(G,Index,w,L^*),v_w)$.最后,敌手$\mathcal{A}$输出一个比特位$b$.

文献[18]已证明会存在一个多项式时间的敌手$\mathcal{A}$,破坏了方案的CQA安全性.

假设存在多项式时间的敌手$\mathcal{A}$,使得$Game_0$,$Game_1$的概率差异不可忽略.证明存在一个多项式时间的敌手$\mathcal{A}$破坏了方案的CQA安全性.

$\mathcal{A}$生成密钥$K_1\leftarrow Gen(1^\lambda)$和模拟器$\mathcal{S}$,从模拟器$\mathcal{S}$接收到$(G,M,L)$后,对于每个$1\leq i\leq n$,计算$v_i=P_{K_1}(i)$.$\mathcal{A}$输出$(G,M,L)$和$v=(v_1,\cdots v_n)$,并且接收$(Index,L^*)$和$Enc(K,M)$产生的$c$.敌手$\mathcal{A}$通过发送关键词$w$查询并返回它接收到的令牌$\tau$来回答$\mathcal{S}$对关键词$w$的查询,$\mathcal{S}$与敌手$\mathcal{A}$输出相同.

敌手$\mathcal{A}$运行在真实游戏中,则模拟器$\mathcal{S}$才与$Game_0$中的输出相同.如果敌手$\mathcal{A}$运行在理想游戏中,则模拟器$\mathcal{S}$才与$Game_1$中的输出相同.

(3)$Game_2$与$Game_1$相似,只是不事先计算$v_i=P_{K_1}(m_i)$,只在需要时才计算.即对每个关键词w,τ的计算如下:

① 对于所有的$i\in R(w)$,$v_i=P_{K_1}(m_i)$.

② $\tau\leftarrow(\mathcal{L}_2(G,Index,w,L^*),v_w)$,$v_w:=(v_i)_{i\in R(w)}$.

因为令牌的生成是无状态的,所以$Game_2$等价于$Game_1$.

(4)$Game_3$与$Game_2$相似,只是$\mathcal{L}_1(G,M,L^*,Index)$

与$\mathcal{L}_2((G,Index,w,L^*),v_w)$是直接提供的而不是通过$(G,M,w)$计算得到.

观察得知，游戏 $Game_3$ 相当于模拟器$\mathcal{S}$执行的理想实验 $Ideal_{\Sigma,\mathcal{A},\mathcal{S}}(\lambda)$. 显然，$Game_2$ 和 $Game_3$ 的输出分布是相同的，它们是不可区分的.　　证毕.

通过以上分析，我们得到在已知背景模型下，除了会揭示当前检索关键词是否在之前的检索中出现过、何时哪些密文数据项被再次访问、某个节点是否出现在先前查询中以及某个节点的邻居节点是否出现在先前或者当前查询中其他节点的邻居节点中，没有其他任何额外泄露，即满足$\mathcal{L}_2(G,Index,w,L^*)=(QP(w),IP(w),QP(i),IP(i))$，因此，本文提出的方案对自适应选择查询攻击是$\mathcal{L}_2$安全的.

综上所述，我们的方案 STE_max-PSI 针对自适应选择查询攻击 CQA 是安全的.

7　性能分析

本小节对所提方案进行性能分析. 实验机器配置为 Intel(R) Core(TM) i5-7200U CPU 2.70 GHz、8 GB 内存与 64 bits Win 10 操作系统. 实验利用本地虚拟机 VMware 加载开源项目 OpenStack 进行性能测试，使用 python 语言. 实验使用 Pajek 数据集下的子集 CS phd①.

我们将从 max-PSI 协议的通信与时间开销、max-PSI 协议的准确率及整体方案的时间开销等方面对本方案的性能进行评估. 我们取安全参数 λ 为 256 bits，使用 SHA-256 算法实现.

7.1　max-PSI 协议通信与时间复杂度

本文提出的 max-PSI 协议与其它具有代表性的 PSI 协议的通信与时间开销比较如表 2 所示，其中 N 表示集合大小，λ 表示安全参数，k 表示哈希函数的个数，l 表示布隆过滤器以及混乱布隆过滤器的长度.

表 2　方案比较

方案	通信复杂度	时间复杂度
文献[33]	$O(N\times\lambda)$	$O(N^2)$
文献[35]	$O(N\times\lambda\times\log(k+l))$	$O(\log(\lambda)\times N)$
文献[37]	$O(N\times\lambda)$	$O(N\times\log^2 N)$
本文方案	$O(N\times\lambda)$	$O(N\times k)$

本文方案的通信复杂度优于文献[35]，与文献[33,37]的通信复杂度相同. 相比而言，本文方案的时间复杂度优于其它三个方案.

7.2　准确率

本文提出的 max-PSI 协议与文献[36]及文献[37]的准确率比较如图 4 所示，其中 N 表示集合中元素个数，k 为哈希函数个数，l 为所选布隆过滤器的长度，随机字符串长度为 λ_1.

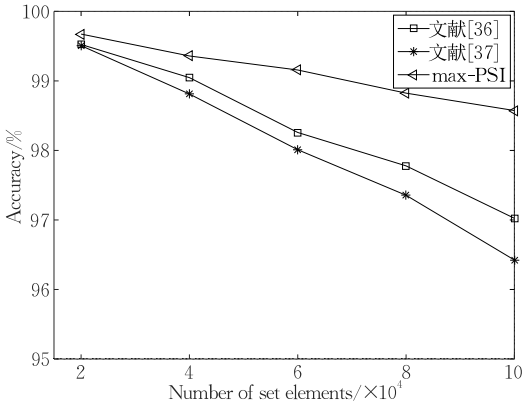


图 4　不同 PSI 协议的准确率比较

文献[36]的误判率为$(1-e^{-k\times N/l})^{\lambda_1}$，文献[37]的误判率为 $k\times N^2\times 2^{-\lambda_1}$，max-PSI 协议的误判率为 $2^{-\lambda_1}$. 从图 4 中可以看出，本文提出的 max-PSI 协议的准确率最高，集合元素个数在 20 000 至 100 000 之间时，max-PSI 协议的准确率在 98.5%～99.6%之间，文献[36]的准确率在 97%～99.5%之间，文献[37]的准确率在 96.4%～99.5%之间.

7.3　时间开销

当λ取 256 bits 时，我们测试了方案生成 GBF 与 BF，求集合交集及关键词查询的时间开销. 为了最大程度地防止布隆过滤器产生冲突，我们取 BF 和 GBF 长度为 $l=2\times max_L$，哈希函数个数 k 为 50 个，构建 GBF 生成的随机字符串长度 λ_1 为 128 bits，时间开销如图 5 所示.

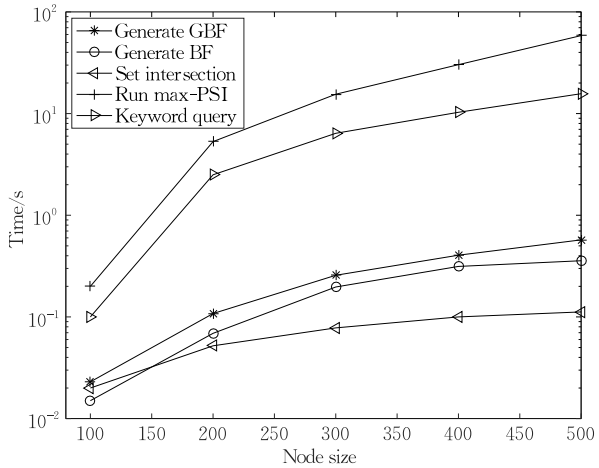


图 5　各个阶段的时间开销

① http://vlado.fmf.uni-lj.si/pub/networks/data/

从图 5 可知，方案的时间开销主要在关键词查询与 max-PSI 协议运行阶段，其次是 GBF 的构建. 对于关键词查询，主要取决于构建的 *Index* 索引，本文关键词查询的时间复杂度为 $O(|W|)$. 对于协议运行过程，每个数组填充后的容量为 max_L，k 为布隆过滤器所用哈希函数个数，则每个 BF 与 GBF 要进行 $max_L \times k$ 次哈希运算，则完整运行要进行 $|R(w)| \times max_L \times k$ 次哈希运算.

7.4 通信开销

当 λ 取 256 bits 时，我们测试了方案的通信开销. 我们同样取 BF 和 GBF 长度为 $l=2\times max_L$，哈希函数个数 k 为 50 个，构建 GBF 生成的随机字符串长度 λ_1 为 128 bits. 通信开销如图 6 所示.

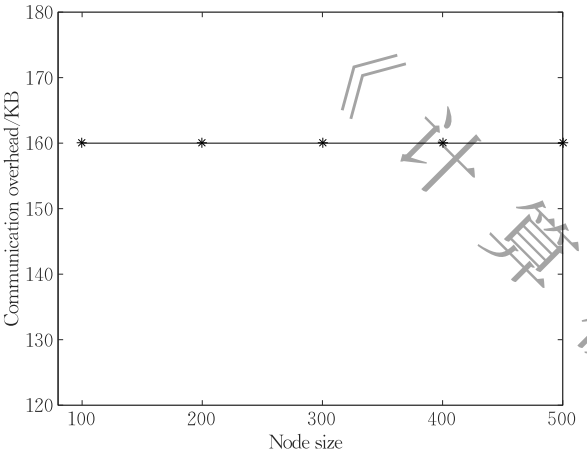


图 6 通信开销

从图 6 可知，随着节点个数的增加，通信量不变. 本方案的通信量主要包括：用户发送待搜索关键词以及感兴趣的数据项 $m_a(a\in n)$、数据拥有者返回检索陷门 $\tau:=(P_{K_1}(w),F_{K_2}(w))$ 及一个半保密数据项 $v_a=P_{K_1}(m_a)$、用户将陷门及半保密数据项发送给云服务器及云服务器返回最终加密数据项. 但由于本文方案只返回与用户感兴趣的数据项 $m_a(a\in n)$ 具有最多共同邻居节点的节点，因此，通信量趋于不变.

7.5 小 结

通过对比可知，我们提出 max-PSI 协议的通信复杂度与文献[33,37]方案一样，比文献[35]的方案优. max-PSI 协议的时间复杂度在 4 个方案中是最优的，同时，它的准确率也是最优的，始终始终可以保持在 98.5% 以上. 关于时间开销，本文方案主要的时间开销在于关键词查询和 max-PSI 协议运行阶段，其中关键词查询的时间开销与关键词个数 $O(|W|)$ 呈线性增长，max-PSI 协议运行阶段时间开销为 $O(|R(w)|\times max_L\times k)$，其中 $|R(w)|$、max_L 及 k 都要远远小于总节点个数 n.

8 结论与下一步工作

在结构化加密的基础上，提出了一种安全高效的支持隐私保护集合交集运算的结构化加密方案，可解决如婚姻介绍精确匹配的问题. 为了使我们的方案能够应用于复杂的数据结构，我们的方案引入了关联性属性. 方案通过引入混乱布隆过滤器设计了一种 max-PSI 协议，可以实现在密文下的交集运算，使得对于图中任意多个节点，通过比较其共有的邻居节点个数来返回最亲密的节点.

接下来的工作中，我们将进一步地研究在加密数据上的统计分析. 我们将在社交网络的结构化加密数据上进行各类统计分析. 在社会网络图中，一般认为节点的出入度越高，则该节点在社交网络中越重要，当我们计算出加密形式下社交网络图的度最大值、最小值及分析社交网络图中某个节点的度中心性等，就可以分析该节点在社交网络中的重要程度，进而实施更细化的隐私保护.

参 考 文 献

[1] Ren K, Wang C, Wang Q. Security challenges for the public cloud. IEEE Internet Computing, 2012, 16(1): 69-73

[2] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data//Proceedings of the 2000 IEEE Symposium on Security and Privacy. Berkeley, USA, 2000: 44-55

[3] Goh E J. Secure indexes. Cryptology ePrint Archive, 2003. https://eprint.iacr.org/2003/216

[4] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data//Proceedings of the International Conference on Applied Cryptography and Network Security. Berlin, Germany: Springer, 2005: 442-455

[5] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions//Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, USA, 2006: 79-88

[6] Chen L, Xue Y, Mu Y, et al. CASE-SSE: Context-aware semantically extensible searchable symmetric encryption for Encrypted Cloud Data. IEEE Transactions on Services Computing, 2022, DOI: 10.1109/TSC.2022.3162266

[7] Kamara S, Moataz T. Boolean searchable symmetric encryption with worst-case sub-linear complexity//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Paris, France, 2017: 94-124

[8] Chen L, Zhang N, Sun H M, et al. Secure search for encrypted personal health records from big data NoSQL databases in cloud. Computing, 2020, 102(6): 1521-1545

[9] Kim K S, Kim M, Lee D, et al. Forward secure dynamic searchable symmetric encryption with efficient updates// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 1449-1463

[10] Asharov G, Segev G, Shahaf I. Tight tradeoffs in searchable symmetric encryption. Journal of Cryptology, 2021, 34(2): 1-37

[11] Chen L, Lee W K, Chang C C, et al. Blockchain based searchable encryption for electronic health record sharing. Future Generation Computer Systems, 2019, 95: 420-429

[12] Du L, Li K, Liu Q, et al. Dynamic multi-client searchable symmetric encryption with support for Boolean queries. Information Sciences, 2020, 506: 234-257

[13] Patranabis S, Mukhopadhyay D. Forward and backward private conjunctive searchable symmetric encryption//Proceedings of the 28th Annual Network and Distributed System Security Symposium. San Diego, USA, 2021: 1-18

[14] He K, Chen J, Zhou Q, et al. Secure dynamic searchable symmetric encryption with constant client storage cost. IEEE Transactions on Information Forensics and Security, 2020, 16: 1538-1549

[15] Song Q, Liu Z, Cao J, et al. SAP-SSE: Protecting search patterns and access patterns in searchable symmetric encryption. IEEE Transactions on Information Forensics and Security, 2020, 16: 1795-1809

[16] Li J, Huang Y, Wei Y, et al. Searchable symmetric encryption with forward search privacy. IEEE Transactions on Dependable and Secure Computing, 2019, 18(1): 460-474

[17] Chen L, Qiu L, Li K C, et al. DMRS: An efficient dynamic multi-keyword ranked search over encrypted cloud data. Soft Computing, 2017, 21(16): 4829-4841

[18] Chase M, Kamara S. Structured encryption and controlled disclosure//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Singapore, 2010: 577-594

[19] Yao A C. Protocols for secure computations//Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. Washington, USA, 1982: 160-164

[20] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 1-19

[21] Dong C, Chen L, Wen Z. When private set intersection meets big data: An efficient and scalable protocol//Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. Berlin, Germany, 2013: 789-800

[22] Liu X, Yang G, Mu Y, et al. Multi-user verifiable searchable symmetric encryption for cloud storage. IEEE Transactions on Dependable and Secure Computing, 2018, 17(6): 1322-1332

[23] Huang Y, Lv S, Liu Z, et al. Cetus: An efficient symmetric searchable encryption against file-injection attack with SGX. Science China Information Sciences, 2021, 64(8): 1-18

[24] Cao N, Yang Z, Wang C, et al. Privacy-preserving query over encrypted graph-structured data in cloud computing// Proceedings of the 2011 31st International Conference on Distributed Computing Systems. Minnesota, USA, 2011: 393-402

[25] Liu C, Zhu L, Chen J. Graph encryption for top-k nearest keyword search queries on cloud. IEEE Transactions on Sustainable Computing, 2017, 2(4): 371-381

[26] Wang Q, Ren K, Du M, et al. SecGDB: Graph encryption for exact shortest distance queries with efficient updates// Proceedings of the International Conference on Financial Cryptography and Data Security. Sliema, Malta, 2017: 79-97

[27] Agarwal A, Herlihy M, Kamara S, et al. Encrypted databases for differential privacy. Cryptology ePrint Archive, 2018. https://eprint.iacr.org/2018/860

[28] Ghosh E, Kamara S, Tamassia R. Efficient graph encryption scheme for shortest path queries//Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. Hong Kong, China, 2021: 516-525

[29] Naor M, Pinkas B. Oblivious transfer and polynomial evaluation //Proceedings of the 31st Annual ACM Symposium on Theory of Computing. Atlanta, USA, 1999: 245-254

[30] Hazay C, Lindell Y. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries//Proceedings of the Theory of Cryptography Conference. New York, USA, 2008: 155-175

[31] Cristofaro E D, Tsudik G. Practical private set intersection protocols with linear complexity//Proceedings of the International Conference on Financial Cryptography and Data Security. Tenerife, Canary Islands, Spain, 2010: 143-159

[32] Freedman M J, Hazay C, Nissim K, et al. Efficient set intersection with simulation-based security. Journal of Cryptology, 2016, 29(1): 115-155

[33] Hallgren P, Orlandi C, Sabelfeld A. PrivatePool: Privacy-preserving ridesharing//Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium. Santa Barbara, USA, 2017: 276-291

[34] Badrinarayanan S, Miao P, Raghuraman S, et al. Multi-party threshold private set intersection with sublinear communication// Proceedings of the IACR International Conference on Public-Key Cryptography. Yokohama, Japan, 2021: 349-379

[35] Zhao Y, Chow S S M. Can you find the one for me?//Proceedings of the 2018 Workshop on Privacy in the Electronic Society. New York, USA, 2018: 54-65

[36] Sun Mao-Hua, Gong Zhe. A privacy-preserving outsourcing set union protocol. Journal of Cryptologic Research, 2016, 3(2): 114-125(in Chinese)
(孙茂华，宫哲. 一种保护隐私集合并集外包计算协议. 密码学报，2016, 3(2): 114-125)

[37] Ghosh S, Nilges T. An algebraic approach to maliciously secure private set intersection//Proceedings of the 2018 Annual International Conference on the Theory and Applications of Cryptographic Techniques. Darmstadt, Germany, 2019: 154-185

[38] Chen H, Huang Z, Laine K, et al. Labeled PSI from fully homomorphic encryption with malicious security//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2018: 1223-1237

[39] Baldi P, Baronio R, De Cristofaro E, et al. Countering gattaca: Efficient and secure testing of fully-sequenced human genomes//Proceedings of the 18th ACM Conference on Computer and Communications Security. New York, USA, 2011: 691-702

[40] Narayanan A, Thiagarajan N, Lakhani M, et al. Location privacy via private proximity testing//Proceedings of the Network and Distributed System Security. San Diego, USA, 2011: 1-17

YANG Jia-Hui, M. S. candidate. His research interests include structured encryption and PSI protocol.

CHEN Lan-Xiang, Ph. D., professor, Ph. D. supervisor. Her research interests include cryptography and its application, privacy computing.

MU Yi, Ph. D., professor, Ph. D. supervisor. His research interest is public cryptography.

ZENG Ling-Fang, Ph. D., professor, Ph. D. supervisor. His research interests include distributed computing and parallel processing.

XUE Yu-Jie, M. S. candidate. His research interests include structured encryption and knowledge engineering.

Background

Searchable encrypted database has been a trend to enhance data security and privacy on outsourced databases while it can be searched without decryption. With great computational efficiency and applicability, searchable symmetric encryption (SSE) has demonstrated its practicability of real-world applications. The traditional SSE scheme is mainly for text data types. In order to achieve searchable encryption of a wider range of data types, Kamara et al. proposed the concept of structured encryption (STE) in 2010. Structured encryption can realize searchable encryption of complex data structures. It can be used to encrypt social network graphs to achieve retrieval of encrypted data, but cannot be applied to the computation and statistical analysis of social network graphs. Therefore, we design the private set intersection protocol max-PSI, and apply it to the structured encrypted social network graph data, and propose a structured encrypted PSI scheme, STE_max-PSI, to realize any number of nodes in the encrypted graph and the calculation of the maximum intersection size of the set of neighbor nodes. In the encrypted social network graph, it can query the node with the highest affinity of any node (we think two nodes with the most common neighbors have the highest affinity). The proposed scheme utilized a garbled Bloom filter(GBF) to achieve richer query functions for more complex data structures, while protecting the privacy of data. The GBF has negligible false positives when querying elements. Compared with the existing PSI protocols, our scheme can greatly improve the query accuracy. Experiments on the real dataset show that the query accuracy of our scheme is better than the existing schemes.