山东大学硕士研究生学位论文开题报告

<table>
<tr><td>姓名</td><td>张响鸽</td><td>学号</td><td>202017038</td><td>专业</td><td>网络空间安全</td></tr>
<tr><td>导师</td><td>陈宇</td><td>培养单位</td><td colspan="3">山东大学网络空间安全学院（研究院）</td></tr>
<tr><td>论文题目</td><td colspan="5">隐私集合计算中的关键技术研究</td></tr>
<tr><td rowspan="9">论文类型</td><td rowspan="3">基础理论</td><td rowspan="3"></td><td rowspan="6">选题来源</td><td>国家项目</td><td></td></tr>
<tr><td>部省(市)项目</td><td></td></tr>
<tr><td>学校项目</td><td></td></tr>
<tr><td rowspan="3">应用研究</td><td rowspan="3"></td><td>国际合作</td><td></td></tr>
<tr><td>专硕实践</td><td></td></tr>
<tr><td>其他</td><td></td></tr>
<tr><td rowspan="3">开发研究</td><td rowspan="3"></td><td rowspan="3">选题方式</td><td>校内导师推荐</td><td></td></tr>
<tr><td>校外导师推荐</td><td></td></tr>
<tr><td>研究生自选</td><td></td></tr>
<tr><td colspan="6">注：在相应栏内画“√”</td></tr>
</table>

与选题有关的国内外研究综述，选题的理论意义和实际意义（可加页）

1 选题的目的与意义

大数据时代，海量数据的交叉计算可以为科研、医疗、金融等领域提供更好的支持。然而现实中数据作为各机构或个人的核心资产，出于隐私保护及利益的考虑内部数据通常不对外开放，数据孤岛现象普遍存在，导致数据的价值无法体现。既要应用数据，又要保护数据安全。如何在分布式环境下保障数据和隐私安全的同时发挥数据价值，是当前亟待解决的问题。自 1982 年姚期智[]借“百万富翁”问题引入安全多方计算（Secure Mutiparty Computation，MPC）概念以来，其已成为解决多个参与者在协同计算过程中隐私保护问题的关键技术之一。

隐私集合计算（Private Set Operation，PSO）属于安全多方计算领域的热点问题，允许两个参与方在没有额外信息泄露的情况下对各自私有集合进行安全计算，包括隐私集合求交（Private Set Intersection，PSI）和隐私集合求并（Private Set Union，PSU）。以 PSI 协议为例，发送方输入集合 X，接收方输入集合 Y，两方运行 PSI 协议之后，接收方除获得交集 $X \cap Y$ 的信息外不能获得 $X \setminus Y$ 中的任何信息，而发送方不获得任何信息。在隐私保护的场景中，PSO 协议具有重要意义，如 PSI 的应用场景包括联系人匹配[]、计算广告转化率[]、DNA 检测与模式匹配[]和接触者追踪[]等；PSU 的应用场景包括利用 IP 黑名单联合查询进行网络风险评估[]和隐私保护数据聚合[]等。

在PSO协议的设计中，往往运用了一些高级的数据结构，对于降低协议的渐进复杂度起到了重要的作用。各类数据结构在具体协议中发挥的作用主要分为以下三类：

（1）**数据对齐**：在PSI和PSU协议中都利用了哈希分桶技术。发送方和接收方将各自集合中的元素分别映射到两个哈希表中，每个哈希表有B个桶，并向每个桶中填充哑元以防止额外信息泄露。两方相同的元素会被映射到相同索引的桶中。哈希分桶相当于将双方的输入集合分别划分成了B个不相交的子集，出现在子集的交集中的元素一定属于原始集合的交集。对于每个桶中的元素，直接逐桶执行PSI或PSU子协议。不同的协议中选用了不同的哈希函数构造哈希表以达到最佳的效率，包括简单哈希（Simple Hashing）[12]、置换哈希（Permutation-based Hashing）[11]和布谷鸟哈希（Cuckoo Hashing）[10]。

（2）**成员测试**：一种PSI协议的设计范式可以概括为两个参与方将各自集合中的元素盲化后直接对比得到交集，如基于不经意伪随机函数的PSI协议。利用布隆过滤器（Bloom Filter）[]或布谷鸟过滤器（Cuckoo Filter）[]进行成员测试可以降低通信开销。具体做法为，发送方将盲化后的数据集合X'中的元素插入过滤器并将过滤器发送给接收方，接收方查询Y'中的元素是否在过滤器中获得交集。

（3）**数据编码**：一些PSI和PSU协议中利用了多项式或乱码布隆过滤器（Gabled Bloom Filter）对集合中的元素进行编码。Garimella等人根据上述技术抽象出了不经意键值对存储（Oblivious Key-Value Store，OKVS）[]的概念。通过构造效率更高的OKVS实例化3H-GCT可以直接得到更高效的PSI和PSU协议。接收方将其集合中的元素y作为键并选取随机值r，对键值对$\{y,r\}$进行编码构造一个OKVS，发送方同样以其集合中的元素x作为键进行解码得到对应值。若$x \in Y$则接收方解码可以得到发送方选择的随机值。

综上所述，隐私集合计算是安全多方计算的一个重要研究分支，是近几年来国内外的研究热点。各类数据结构的应用使得PSO协议的效率得到了很大的提高：a.利用哈希表进行数据对齐；b.利用过滤器进行成员测试；c.利用OKVS进行数据编码。通过对各类数据结构提供性能分析与基准测试，设计PSO协议时可以选择更高效更合适的优化技术，对隐私保护场景下的数据共享问题具有重要的理论意义和实际意义。

2 国内外研究现状

2.1 PSI的研究现状

2.1.1 基于不经意多项式求值的PSI协议

基于不经意多项式求值的PSI协议的设计思想是：将发送方或接收方将其集合元素表示成多项式的根，利用多项式的性质并结合密码学工具进行交集求解。

2004年，Freedman首先给出了基于不经意多项式求值的PSI协议[16]。在该协议中，

接收方将其输入集合中的所有元素设为某一多项式$P(z)$的根，并利用插值公式计算出多项式的系数。接收方用 Paillier[30]或 ElGamal[13]半同态加密算法将多项式的系数加密发送给另一方。根据同态加密的性质，发送方对其输入集合中的所有元素进行密态求值并利用随机数r盲化后将结果$Enc(r\cdot P(x_i)+x_i)$返回给接收方比对。接收方解密，若元素y在交集中，则对任意r都有$r\cdot P(y)+y=y$；否则$r\cdot P(y)+y$是一个随机值。该方案的通信开销为线性的复杂度但其计算开销非常高，主要的计算开销在于产生一个次数为$|Y|-1$的高次多项式以及对高次多项式进行$|X|$次密态计算。因此论文指出利用平衡分配哈希[3]，发送方和接收方将各自集合中的元素平均的分配到B个桶中，每个桶中元素的个数最多为M，从而达到降低多项式次数的目的。

Chen 等人在 2017 年[7]和 2018 年[8]基于不经意多项式提出了适用于在两方集合大小相差较大场景下的非平衡 PSI 方案。主要构造为，发送方而非接收方以其集合中的元素作为根产生一个多项式$Q(z)$，接收方利用 Fan-Vercauteren[15]全同态加密算法对其集合中每个元素加密后发送给发送方进行密态求值。发送方盲化后将结果$Enc(r_j\cdot Q(y_j))$返回给接收方，接收方解密后根据结果是否为 0 判断交集元素。该方案的优势在于拥有小集合的接收方只执行相对较轻的计算且方案通信开销与大集合亚线性相关。

2.1.2 基于不经意伪随机函数的 PSI 协议

2005 年 Freedman 等人在[17]中指出了 OPRF 与 PSI 之间的联系。关键思想在于将发送方和接收方输入集合中的元素替换为伪随机函数值即随机盲化，其中发送方得到密钥k可以任意地计算其输入元素的 PRF 值$X'=\{OPRF_k(x):x\in X\}$，接收方无法自行计算只能得到 OPRF 协议中输出的 PRF 值$Y'=\{OPRF_k(y):y\in Y\}$。接收方通过对比X'和Y'可以得到交集。

（1）基于单点 OPRF 的 PSI 协议

Pinkas 等人在 2014 年提出了基于 OT/OPRF 的 PSI 协议[31]。该方案的主要思想为将 1-out-of-2 OT 看作接收方输入元素定义域为$r\in\{0,1\}$的单点 OPRF，用函数表示为$F((m_0,m_1),r)=m_r$，其中(m_0,m_1)作为 OPRF 的密钥。得益于高效的 OT 扩展协议[2][21][23]，仅需要少量的公钥加密操作和快速的对称加密操作即可获得大量 OT 实例，该方案有着显著的计算开销优势。然而由于该方案中构造的是单点 OPRF，其通信开销为$O(n^2)$。为了提高通信效率，论文指出可以借鉴哈希分桶的思想，将元素映射到哈希表中的每个桶中，然后逐桶执行元素比较得到交集。通过合理选取参数如哈希函数的个数和哈希表中桶的个数，该方案的通信开销可以减小到$O(n\log n)$的复杂度。

观察到上述方案需要的 OT 实例个数不仅与集合中元素的个数相关还与元素的长度相关，2015 年 Pinkas 等人提出使用置换哈希构造哈希表减小每个桶中放入元素的长度实

现了优化[33]。2016 年，在文献[24]中 Kolesnikov 等人对该方案进行了进一步优化使得 OT 实例的个数只与集合中元素的个数相关。他们提出用 1-out-of-N OT 替换 1-out-of-2 OT 对每个元素进行逐 N 比特比较，并在 Kolesnikov 和 Kumaresan 提出从编码的角度看 OT 扩展技巧[23]的基础上指出，构造 1-out-of-N OT 不需要纠错码纠错的性质，因此利用伪随机码可以得到 N 为线性大小的 1-out-of-N OT。该方案为目前计算开销最小的 PSI 方案，在集合大小为 $n=2^{20}$ 时仅需 2.4s 的时间即可完成两方安全求交。

（2）基于多点 OPRF 的 PSI 协议

2019 年 Pinkas 等人结合多项式插值技术和 OT 扩展设计了一个基于稀疏 OT 扩展的 PSI 协议[]，其允许接收方从 n 个随机秘密中不经意的选取 k 个以实现多点 OPRF，使得通信开销达到了线性的复杂度。然而该协议需要在一个大的域上计算高阶多项式导致其计算开销较高。通过选择一个新的哈希结构：2-选择哈希[]，该哈希函数几乎不需要引入哑元，并达到了降低计算开销的目的。2020 年，Chase 和 Miao 在文献[]中提出了一个更加高效的多点 OPRF，仅利用 OT、伪随机函数、哈希函数和位运算。

Pinkas 等人在 2020 年首次将布谷鸟哈希表应用到了恶意模型下的 PSI 协议[34]。通过改进布谷鸟哈希哈希算法构造乱码布谷鸟哈希表（Probe-and-XOR of Strings，PaXoS）并结合具有同态性质的 OT 扩展协议 OOS[]得到了一个高效的恶意模型下的 PSI 协议，该协议的计算效率几乎与已知最快的半诚实模型下的 PSI 协议[24]一样高。PaXoS 利用 2 个哈希函数将集合元素映射到对应桶并放入该元素的秘密分享值，以消除布谷鸟哈希构建恶意 PSI 时泄露发送方未在集合交集中的集合信息问题[]。

2021 年，在文献[39]中 Rindal 和 Schoppmann 基于不经意向量线性求值（Vector Oblivious Linear Evaluation，Vector-OLE）构造了一个新的 OPRF，并且他们观察到用更高效的 OKVS 实例化 3H-GCT 替换多项式与 OPRF 相结合构造可编程的 OPRF，可以得到比[32]中通信开销更小的 PSI 方案。[]中的方案为目前通信开销最小的 PSI 方案，在集合大小为 $n=2^{20}$ 时，该方案的通信开销为 53MB 比基于 Diffie-Hellman 的 PSI 协议[26]的通信开销更低。

2.1.3 基于布隆过滤器的 PSI 协议

Dong 等人在 2013 年提出了利用布隆过滤器[5]和乱码布隆过滤器构造的半诚实模型下的 PSI 协议[11]。与布隆过滤器不同，乱码布隆过滤器中每个桶存放的元素不再是 1 比特而是 λ 比特关于插入元素的秘密分享值。发送方和接收方根据其各自集合中的元素分别生成一个乱码布隆过滤器和布隆过滤器，并逐桶执行消息长度为 λ 的 OT 协议。该 PSI 协议能处理的集合数量首次突破了亿级别。此后，对于布隆过滤器的改进也成为优化 PSI 协议的一个重要方向。2017 年 Rindal 和 Rosulek 在文献[59]中通过生成比所需的布隆过滤器比特数略多的 1-out-of-2 OT 构建 Cut-and-Choose 技术，以此实现抵抗恶意敌

手的 PSI 协议。

2.2 PSU 的研究现状

相比于 PSI 协议，关于 PSU 协议的研究工作较少。Kissner 和 Song[]在 2005 年提出了第一个 PSU 协议。该协议的构造基于多项式和加法同态。主要思想为，发送方和接收方将各自集合中的元素分别用多项式 f 和 g 表示，则多项式 $f \times g$ 的根即为并集 $X \cup Y$ 中元素。具体来说，若元素 e 在集合 X 或 Y 中，则有 $(f \times g)(e) = f(e) \times g(e) = 0$。利用加法同态算法，发送方将加密后的多项式 $Enc(f)$ 发送给接收方，接收方计算 $Enc(f \times g)$，解密后求解多项式的根即可得到并集。该方案的计算开销和通信开销都为 $O(n^2)$。2007 年 Frikken 在[]中提出了一个通信开销为 $O(n)$ 的 PSU 协议。接收方将其集合 Y 中的元素用多项式 f 表示并利用加法同态将加密后的 $Enc(f)$ 发送给发送方。发送方对于任意 $x \in X$ 计算 $(Enc(xf(x)), Enc(f(x)))$ 并将结果返回接收方。若 $x \in Y$ 接收方解密后只能获得 $(0,0)$ 不会泄露任何关于 x 的信息，若 $x \notin Y$ 接收方解密后获得 $(xf(x), f(x))$ 可以恢复出属于并集的元素 x。与[]中的思想相似，Davidson 和 Cid 在文献[]中将多项式替换为逆转布隆过滤器提出了一个新的 PSU 协议。

上述所有协议都基于公钥加密体系实现 PSU，即利用 AHE 加密接收方集合的(多项式或布隆过滤器)表示，并对密文执行大量计算。2019 年 Kolesnikov 等人在[]中首次提出了仅利用对称加密技术实现的 PSU 协议。该协议的核心是一个名为反向私有成员测试(Reverse Private Membership Test，RPMT)的子协议，它可以测试发送方的元素 x 是否属于接收方的集合 Y，并让接收方获得结果。之后，发送方将 $\{x, \perp\}$ 作为 OT 协议的输入，接收者获得 $\{x\} \cup Y$。为了降低多项式构造 RPMR 造成的计算和通信开销，该协议利用哈希分桶技术使技术开销达到了 $O(n \log n \log\log n)$ 通信开销达到了 $O(n \log n)$。2022 年，Zhang 等人对[]中的方案做出了改进，将 OKVS 应用到了 PSU 协议中[]。接收方选择一个随机值 w 并加密得到密文 $Enc(w)$。对于集合中的元素 $y \in Y$ 接收方利用 $\{y, Enc(y)\}$ 作为键值对构造一个 OKVS。发送方以 $x \in X$ 作为键解码，将得到的密文重随机化后发送给接收方。接收方解密后根据结果是否为 w 判断发送方的元素 x 是否属于并集。

所要解决的主要问题及研究途径与方法（可加页）

3 拟研究解决的主要问题

拟研究的 PSO 协议中应用的数据结构主要分为三类，分别是哈希表、过滤器和不经意键值对存储。其中哈希表包括简单哈希表、置换哈希表、布谷鸟哈希表；过滤器包括布隆过滤器和布谷鸟过滤器；不经意键值对存储包括乱码布隆过滤器和乱码布谷鸟哈希表（3H-GCT）。拟解决的主要问题如下：

（1）研究各类数据结构的区别与联系

简单哈希表：哈希表 T 由 m 个桶构成，利用 k 个哈希函数 $h_k:\{0,1\}^* \to [m]$ 将元素 x 映射到哈希表的 k 个桶 $T[h_k(x)]$ 中。

置换哈希表：将元素转化为更短的字符串并存储在哈希表中，以此减少存储空间。元素 x 表示为比特形式并拆分为两部分 $x_L \| x_R$，x_L 部分的长度 $|x_L| = \log m$。获取元素在哈希表中的索引 $x_L \oplus h(x_R)$，并将 x_R 存放在对应桶中。相比于简单哈希表，桶中存储的元素长度减少了 $\log m$ 比特。

布谷鸟哈希表：使用 k 个哈希函数 $h_k(\cdot)$ 将元素 x 映射到表 T 的某一个桶中，且每个桶中最多存放一个元素。计算 $h_1(x), h_2(x), \ldots, h_k(x)$，若 $T[h_1(x)], T[h_2(x)], \ldots, T[h_k(x)]$ 存在空桶则元素 x 随机插入空桶中；否则随机选取桶 $T[h_i(x)]$ 逐出其中的元素，对逐出的元素 x' 同样执行上述操作。若逐出操作达到了阈值，将无法插入的元素放入额外空间 stash 中。

布隆过滤器：利用 k 个哈希函数 $h_k(\cdot)$ 将元素 x 映射到一个二进制数组 bf 中。计算 $h_k(x)$ 作为 bf 的索引并将对应位置设置为 1。数组中未映射到的位置为 0。

布谷鸟过滤器：与布谷鸟哈希表相似，但仅使用 2 个哈希函数 $h_1(\cdot)$ 和 $h_2(\cdot)$ 计算索引值且桶中存放元素的指纹而非元素本身。插入元素 x 时，计算索引的规则为 $i_1 = h_1(x)$，$i_2 = i_1 \oplus h_2(f_x)$ 并将 x 放入 $T[i_1]$ 或 $T[i_2]$ 其中一个桶中。遇到两个桶都放满时，元素的逐出规则与布谷鸟哈希表相同。

乱码布隆过滤器：将 BF 数组中每个位置存放 1 比特信息改为存放 λ 比特信息。插入元素 x 时仍然使用 k 个哈希函数 $h_k(\cdot)$ 并计算 $h_k(x)$ 作为索引，共用 GBF 数组中已有的字符串，剩下的位置通过异或得到共享字符串并填充入对应位置。

乱码布谷鸟哈希表：与 GBF 相似，但仅使用 3 个哈希函数计算插入元素的索引值。使用 3 个哈希函数元素插入失败概率变高，需要处理遇到“环”的情况使得元素只能以特定的顺序插入保证成功率。目前 3H-GCT 还没有一个高效的实现，拟在论文中给出一个高效的实现。

（2）各类数据结构的代码实现并进行性能分析与基准测试

失败概率分析，如何选取合适的参数包括哈希函数的个数和桶的个数等使得各类数据结构构造时失败概率足够小。

构造效率分析，各类数据结构是否可以并行插入元素。

空间效率分析，各类数据结构插入相同元素个数时表的大小。

4 研究途径与方法

本论文在大量文献调研的基础上，进行资料分析，代码设计，对隐私保护集合计算中的数据结构进行分析研究。首先对于上述归纳出的 7 种数据结构，需要明确其基本构造方法包括如何插入元素，如何处理碰撞，在这个过程中分析其之间的联系与区别；其次关于性能对比部分，需要罗列出性能指标包括负载因子（Load Factor），插入性能，查询性能，假阳率和失败概率等；最后关于代码设计部分，拟采用 C++实现并需要考虑如下问题：

（1）关于不同哈希函数的调研。各类基于哈希的数据结构的性能与哈希函数的散列值是否均匀密切相关，需要选择一个高效的用于检索的哈希函数，可供选择的哈希函数包括 Murmur 哈希和 Bob 哈希等。

（2）关于 3H-GCT 的代码实现。3H-GCT 与其他数据结构不同，构造时元素需以特定的顺序插入，即需要处理遇到“环”的情况保证成功率。

（3）函数接口的设计以及各种数据结构产生的类之间是否存在继承与派生关系。

5 参考文献

[1] Yuriy Arbitman, Moni Naor, and Gil Segev. “Backyard cuckoo hashing: Constant worst-case operations with a succinct representation”. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. IEEE. 2010, pp. 787–796. 4

[2] Gilad Asharov et al. “More efficient oblivious transfer and extensions for faster secure computation”. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013, pp. 535–548.

[3] Yossi Azar et al. “Balanced allocations”. In: Proceedings of the twenty-sixth annual ACM symposium on theory of computing. 1994, pp. 593–602.

[4] Pierre Baldi et al. “Countering gattaca: efficient and secure testing of fully-sequenced human genomes”. In: Proceedings of the 18th ACM conference on Computer and communications security. 2011, pp. 691–702.

[5] Burton H Bloom. “Space/time trade-offs in hash coding with allowable errors”. In: Communications of the ACM 13.7 (1970), pp. 422–426.

[6] Melissa Chase and Peihan Miao. “Private set intersection in the internet setting from lightweight oblivious PRF”. In: Annual International Cryptology Conference. Springer. 2020, pp. 34–63.

[7] Hao Chen, Kim Laine, and Peter Rindal. “Fast private set intersection from homomorphic encryption”. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, pp. 1243–1255.

[8] Hao Chen et al. “Labeled PSI from fully homomorphic encryption with malicious security”. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, pp. 1223–1237.
[9] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. “Fast and private computation of cardinality of set intersection and union”. In: International Conference on Cryptology and Network Security. Springer. 2012, pp. 218–231.
[10] Emiliano De Cristofaro and Gene Tsudik. “Practical private set intersection protocols with linear computational and bandwidth complexity”. In: Cryptology ePrint Archive (2009).
[11] Changyu Dong, Liqun Chen, and Zikai Wen. “When private set intersection meets big data: an efficient and scalable pro tocol”. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013, pp. 789–800.
[12] Thai Duong, Duong Hieu Phan, and Ni Trieu. “Catalic: Delegated PSI cardinality with applications to contact tracing”. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer. 2020, pp. 870–899.
[13] Taher ElGamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: IEEE transactions on information theory 31.4 (1985), pp. 469–472.
[14] Bin Fan et al. “Cuckoo filter: Practically better than bloom”. In: Proceedings of the 10th ACM International on Confer ence on emerging Networking Experiments and Technologies. 2014, pp. 75–88.
[15] Junfeng Fan and Frederik Vercauteren. “Somewhat practical fully homomorphic encryption”. In: Cryptology ePrint Archive (2012).
[16] Michael J Freedman, Kobbi Nissim, and Benny Pinkas. “Efficient private matching and set intersection”. In: Interna tional conference on the theory and applications of cryptographic techniques. Springer. 2004, pp. 1–19.
[17] Michael J Freedman et al. “Keyword search and oblivious pseudorandom functions”. In: Theory of Cryptography Con ference. Springer. 2005, pp. 303–324.
[18] Gayathri Garimella et al. “Oblivious key-value stores and amplification for private set intersection”. In: Annual Interna tional Cryptology Conference. Springer. 2021, pp. 395–425.
[19] Gayathri Garimella et al. “Oblivious key-value stores and amplification for private set intersection”. In: Annual Interna tional Cryptology Conference. Springer. 2021, pp. 395–425.
[20] Mihaela Ion et al. “Private intersection-sum protocol with applications to attributing aggregate ad conversions”. In: Cryptology ePrint Archive (2017).
[21] Yuval Ishai et al. “Extending oblivious transfers efficiently”. In: Annual International Cryptology Conference. Springer. 2003, pp. 145–161. 5
[22] Ágnes Kiss et al. “Private Set Intersection for Unequal Set Sizes with Mobile Applications.” In: Proc. Priv. Enhancing Technol. 2017.4 (2017), pp. 177–197.
[23] Vladimir Kolesnikov and Ranjit Kumaresan. “Improved OT extension for transferring short secrets”. In: Annual Cryp tology Conference. Springer. 2013, pp. 54–70.
[24] Vladimir Kolesnikov et al. “Efficient batched oblivious PRF with applications to private set intersection”. In: Proceed ings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016, pp. 818–829.
[25] Ziyuan Liang et al. “A Framework of Private Set Intersection Protocols.” In: Cryptology ePrint Archive (2020).
[26] Catherine Meadows. “A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party”. In: 1986 IEEE Symposium on Security and Privacy. IEEE. 1986, pp. 134–134.

[27] Michael Mitzenmacher. “The power of two choices in randomized load balancing”. In: IEEE Transactions on Parallel and Distributed Systems 12.10 (2001), pp. 1094–1104.
[28] Shishir Nagaraja et al. “{BotGrep}: Finding {P2P} Bots with Structured Graph Analysis”. In: 19th USENIX Security Symposium (USENIX Security 10). 2010.
[29] Rasmus Pagh and Flemming Friche Rodler. “Cuckoo hashing”. In: Journal of Algorithms 51.2 (2004), pp. 122–144.
[30] Pascal Paillier. “Public-key cryptosystems based on composite degree residuosity classes”. In: International conference on the theory and applications of cryptographic techniques. Springer. 1999, pp. 223–238.
[31] Benny Pinkas, Thomas Schneider, and Michael Zohner. “Faster private set intersection based on OT extension”. In: 23rd USENIX Security Symposium (USENIX Security 14). 2014, pp. 797–812.
[32] Benny Pinkas et al. “Efficient circuit-based PSI with linear communication”. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. 2019, pp. 122–153.
[33] Benny Pinkas et al. “Phasing: Private set intersection using permutation-based hashing”. In: 24th USENIX Security Symposium (USENIX Security 15). 2015, pp. 515–530.
[34] Benny Pinkas et al. “PSI from PaXoS: fast, malicious private set intersection”. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. 2020, pp. 739–767.
[35] Benny Pinkas et al. “PSI from PaXoS: fast, malicious private set intersection”. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. 2020, pp. 739–767.
[36] Benny Pinkas et al. “SpOT-light: lightweight private set intersection from sparse OT extension”. In: Annual International Cryptology Conference. Springer. 2019, pp. 401–431.
[37] Amanda C Davi Resende and Diego F Aranha. “Faster unbalanced private set intersection”. In: International Conference on Financial Cryptography and Data Security. Springer. 2018, pp. 203–221.
[38] Peter Rindal and Mike Rosulek. “Faster Malicious 2-Party Secure Computation with Online/Offline Dual Execution”. In: 25th USENIX Security Symposium (USENIX Security 16). 2016, pp. 297–314.
[39] Peter Rindal and Phillipp Schoppmann. “VOLE-PSI: fast OPRF and circuit-psi from vector-ole”. In: Annual Interna tional Conference on the Theory and Applications of Cryptographic Techniques. Springer. 2021, pp. 901–930.

研究进度及具体时间安排

起讫日期	主要研究内容	预期结果
2022.04~2022.06	查阅国内外文献，完成文献综述与开题报告	完成开题报告
2022.06~2022.10	各类数据结构的代码实现与对比分析	代码实现
2022.10~2023.02	撰写毕业论文，完成初稿	毕业论文初稿
2023.02~2023.04	结合导师意见不断完善论文	毕业论文定稿
2023.04~2023.06	进行论文答辩前的相关准备	通过答辩

专家对开题报告的评议

1．对选题依据、预期思路或技术路线的科学性、可行性、先进性及创新性的评价

2．存在的主要问题和改进措施

3．□优秀　□通过　□建议修改或补充　□不通过

专家组长签名：

年　月　日

参加学位论文开题报告的专家名单（第一位为组长）

姓名	专业技术职务	学科、专业	工作单位