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">隐私保护集合计算中基于哈希的数据结构分析</td></tr>
<tr><td rowspan="8">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）

一、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、金融等提供更好支持。但是许多企业和机构出于信息安全或利益的考虑，内部数据通常不对外开放。这种情况导致形成一个个数据孤岛，数据的价值无法体现。安全多方计算（Secure Mutiparty Computation，MPC）可以很好解决这一难题。长期以来，安全多方计算致力于在对多个来源的数据进行联合分析的同时，保证每个输入来源的隐私。

隐私保护集合计算属于安全多方计算领域一类特定应用问题，包括隐私保护集合求交（Private Set Intersection，PSI）和隐私保护集合求并（Private Set Union，PSU）。以两方 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，接收方只获得交集 $X \cap Y$ 的信息不能获得 $X \setminus Y$ 中的任何信息，而发送方不获得任何信息。隐私保护集合求交是安全多方计算领域应用最广泛的一类协议。其在实际生活中的应用场景包括联系人匹配[22]、计算广告转化率[20]、DNA 检测与模式匹配[40]、僵尸网络检测[28]和接触者追踪[12]等。

近些年，PSI 协议在计算开销和通信开销上得到了极大的优化，半诚实模型下两方的 PSI 方案效率几乎达到了与朴素 PSI 方案同一数量级。目前计算开销最小的 PSI 方案为 Kolesnikov 等人在[24]中提出的基于不经意传输/不经意伪随机函数（Oblivious Transfor，OT/Oblivious Pseudo-random Function，OPRF）的方案。这一方案在集合大小

为$n=2^{20}$时仅需2.4s的时间即可完成两方安全求交。其显著的计算开销优势得益于高效的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。为了优化通信开销，该方案沿用了Pinkas等人在[31][33]中提出的哈希分桶技术。目前通信开销最小的PSI方案为[39]中提出的基于不经意向量线性求值（Vector Oblivious Linear Evaluation，Vector-OLE）构造OPRF的PSI方案。Rindal和Schoppmann观察到用更高效的不经意键值对存储（Oblivious Key-Value Store，OKVS）[35]替换多项式与OPRF相结合构造可编程的OPRF（OPPRF）可以得到比[32]中通信开销更小的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。

在PSI协议的优化技术中，基于哈希的数据结构的应用起到了很大的作用。因此分析各类基于哈希的数据结构的特点、在PSI协议中的应用及其发挥的主要作用对PSI的研究进展有很大的帮助。

二、国内外研究概况

1. 哈希表在PSI中的应用

2005年Freedman等人在[17]中指出了OPRF与PSI之间的联系：一种简洁的PSI方案可以直接由OPRF协议得到。利用OPRF构造PSI方案的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，其中发送方得到密钥k可以任意地计算其输入元素的伪随机函数值$X'=\{OPRF_k(x):x\in X\}$，接收方无法自行计算只能得到OPRF协议中输出的伪随机函数值$Y'=\{OPRF_k(y):y\in Y\}$。发送方将X'发送给接收方进行对比，接收方输出$X\cap Y$。

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二选一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。基于高效的OT扩展协议，该方案有着计算开销小的优势。然而由于该方案中构造的是单点OPRF，其通信开销为$O(n^2)$。为了提高该协议的通信效率，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。构建哈希表时发送方选用简单哈希，接收方选用布谷鸟哈希[29]并将每个桶中的元素个数限制为1。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，即通过一次OT协议就可以判断两个元素是否相等。他们提出用N选一OT替换二选一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N选一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N选一OT。基于单

点 OPRF 的 PSI 协议尽管计算开销很小，且利用哈希技术后该类协议的通信开销得到了优化但仍然无法达到线性的复杂度。

除了在基于单点 OPRF 的 PSI 协议中的应用，哈希表同样应用到了基于不经意多项式求值的 PSI 协议中。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。接收方解密所有密文后根据结果是否与集合 Y 中的某一个元素 y 相等判断 y 是否在交集中。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。另外，Chen 等人在 2017 年[7]和 2018 年[8]提出的适用于在两方集合大小相差较大场景下的非平衡 PSI 方案中也用到了哈希分桶的技术降低多项式的次数。在 Chen 等人提出的非平衡 PSI 方案中，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。该方案的优势在于发送方和接收方都只需要发送 $|Y|$ 个密文，其通信开销只与小集合的大小相关。

2. 过滤器在 PSI 中的应用

Dong 等人在 2013 年提出了利用布隆过滤器[5]和混乱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，混乱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混乱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。通过改进布隆过滤器，2016 年 Rindal 和 Rosulek 在文献[38]中给出了第一个恶意模型下的 PSI 协议[44]。

2018 年，Resende 和 Aranha 在文献[37]中首次将布谷鸟过滤器应用到了 PSI 方案中。他们指出将布谷鸟过滤器与 Baldi 等人在 2011 年提出的基于 Diffie-Hellman 多点 OPRF 的 PSI 方案[4]结合，可以得到一个通信开销更小的优化方案。[4]中的 OPRF 协议用函数可表示为 $F_\alpha(y)=H'(H(y)^\alpha)$。具体来说，（1）发送方选择密钥 $\alpha\in Z_q^*$ 和输入集合 X，接收方选择输入集合 Y；（2）对于每个元素 $y_j\in Y$ 接收方随机选择不同的 $\beta_j\in Z_q^*$ 并将 $H(y_j)^{\beta_j}$ 发送给发送方，发送方返回 $(H(y_j)^{\beta_j})^\alpha$，接收方将指数 β_j 移除输出

$H'(H(y_j)^{\alpha}) = H'(((H(y_j)^{\beta_j})^{\alpha})^{1/\beta_j})$；（3）对于每个元素 $x_i \in X$ 发送方计算 $H'(H(x_i)^{\alpha})$ 并发送给接收方计算交集。在优化的协议中，发送方不直接发送盲化后的元素而是向接收方发送一个布谷鸟过滤器，其中插入盲化后的元素 $H'(H(x_i)^{\alpha})$，接收方依次查询元素 $H'(H(y_j)^{\alpha})$ 是否在过滤器中。利用布谷鸟过滤器将发送的集合进行了压缩实现了通信开销的优化，相比于 Baldi 等人的方案，该优化方案传输的数据减少到原来的十分之三，相应地运行速度快了 3.3 倍。2020 年，在文献[25]中 Liang 等人指出其他近似成员测试数据结构如布隆过滤器[5]和真空过滤器[42]也可以用于该优化技术，并给出了当假阳率固定为 2^{-40} 时三种数据结构插入不同数据量时的压缩率对比。其中，布谷鸟过滤器和真空过滤器的压缩率比布隆过滤器低但不够平滑稳定。当插入元素个数超过 2^{16} 时真空过滤器的压缩率趋于稳定，因此作者建议元素个数超过 2^{16} 时选择真空过滤器。过滤器还可用于其他基于 OPRF 的 PSI 协议中的通信开销优化，如基于 Blind-RSA 的 PSI 协议[10]、基于稀疏 OT 的 PSI 协议[36]和基于轻量多点 OPRF 的 PSI 协议[6]。

3. 不经意键值对存储在 PSI 中的应用

布谷鸟哈希表无法应用于恶意模型下的 PSI 协议。在[24][31][33]中，接收方利用布谷鸟哈希将元素 y 映射到 $h_1(y)$ 或 $h_2(y)$ 其中一个桶中，发送方利用简单哈希将元素 x 映射到 $h_1(x)$ 和 $h_2(x)$ 两个桶中。该类方案无法抵抗恶意敌手在于恶意的发送方可以只将元素 e（两方交集中的元素）放置到 $h_1(e)$ 或 $h_2(e)$ 一个桶中，则接收方输出的交集中是否包含元素 e 取决于接收方对元素 e 的放置位置。这种攻击导致恶意的发送方知道接收方将其输入集合中的元素映射到了哪个位置，而接收方对该位置的选择依赖于其集合中的其他元素，从而导致非交集元素位置的泄露。2020 年，Pinkas 等人在文献[34]中提出混乱布谷鸟哈希表（Probe-and-XOR of Strings，PaXoS）解决了上述问题并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。混乱布谷鸟哈希表的构造结合了混乱布隆过滤器和布谷鸟哈希的特点，即用 2 个哈希函数而不是 $k\,(k>2)$ 个哈希函数构造混乱布隆过滤器。与混乱布隆过滤器中元素以任意顺序插入不同，PaXoS 由于仅仅只使用 2 个哈希函数导致失败概率变高，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。文献[18]在[34]的基础上抽象出了不经意键值对存储的概念。基于 3 个哈希函数构造布谷鸟哈希表比 2 个哈希函数构造布谷鸟哈希表效率更高的事实，Garimella 等人使用 3 个哈希函数构造混乱布谷鸟哈希表（3-Hash Garbled Cuckoo Table，3H-GCT）得到了一个更加高效的 OKVS。2021 年，在文献[39]中 Rindal 和 Schoppmann 观察到用更高效的 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。

所要解决的主要问题及研究途径与方法（可加页）

三、拟研究解决的主要问题

基于哈希的数据结构在各类 PSI 协议中都起到了优化通信开销和计算开销的作用。本文的主要目的为对 PSI 协议中应用的数据结构进行总结分析对其适用的场景给出建议并在同一运行环境下进行性能对比分析。

论文中拟要研究的基于哈希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（Simple Hashing）、平衡分配哈希（Balanced Allocation Hashing）、布谷鸟哈希（Cuckoo Hashing）；过滤器包括布隆过滤器（Bloom Filter）、混乱布隆过滤器（Gabled Bloom Filter）和布谷鸟过滤器（Cuckoo Filter）；不经意键值对存储包括混乱布谷鸟哈希表（Garbled Cuckoo Table）。拟解决的主要问题如下：

1. 利用哈希表进行哈希分桶时，不同的 PSI 方案选用了不同的哈希函数构建哈希表，各类哈希表的特点是什么，适用于什么样的场景？如：
（1）三类哈希表构造时处理碰撞的方式不一样：简单哈希表不处理碰撞的情况，元素直接放置到每个哈希函数映射的位置；平衡分配哈希遇到碰撞时将元素放置到存储元素少的一个桶中；布谷鸟哈希采用“逐出”操作处理碰撞。
（2）基于单点 OPRF 的 PSI 协议中只能同时利用简单哈希和布谷鸟哈希优化通信效率，且布谷鸟哈希表只能用于构造半诚实模型下的 PSI 协议。

2. 哈希表、过滤器和不经意键值对存储之间的联系与区别？如：
（1）哈希表和过滤器都可用于成员测试；区别在于过滤器的空间效率比哈希表更高，且过滤器进行成员测试时存在假阳率使得原本不在集合中的元素的查询结果为“真”。
（2）哈希表和过滤器包含的基础算法为 *Insert* 和 *Lookup*；不经意键值对存储包含的基础算法为 *Encode* 和 *Decode* 。
（3）混乱布隆过滤器本身也可以用于实例化不经意键值对存储，混乱布谷鸟哈希表是使用 2 个哈希函数构造的混乱布隆过滤器。

3. 各类数据结构的代码实现并进行性能分析与测试。
（1）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。
（2）构造效率分析，各类数据结构是否可以并行插入元素。
（3）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。

四、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基于哈希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，我需要明

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，我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，我将采用 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，我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“环”的处理。混乱布谷鸟哈希表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。

五、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.

[40] Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. “Privacy preserving error resilient DNA searching through oblivious automata”. In: Proceedings of the 14th ACM conference on Computer and communications security. 2007, pp. 519–528.

[41] Jaideep Vaidya and Chris Clifton. “Secure set intersection cardinality with application to association rule mining”. In: Journal of Computer Security 13.4 (2005), pp. 593–622.
[42] Minmei Wang and Mingxun Zhou. “Vacuum filters: more space-efficient and faster replacement for bloom and cuckoo filters”. In: Proceedings of the VLDB Endowment (2019). 6
[43] 申立艳 et al. “隐私保护集合交集计算技术研究综述”. In: 计算机研究与发展 54.10 (2017), p. 2153.
[44] 黄翠婷 et al. “隐私集合求交技术的理论与金融实践综述”. In: 信息通信技术与政策 47.6 (2021), p. 50.

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
2022.06~2022.09	各类数据结构的代码实现与对比分析	代码实现
2022.09~2023.02	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	答辩通过

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