对称可搜索加密研究进展调研报告

引言

近年来，随着网络技术的高速发展，人们日常生活中产生的数据文件越来越多，本机的内存已经不能满足人们的基本存储需求，百度云等云存储的出现给人们带来了很多便利，人们已经逐渐习惯将数据文件存储在外部服务器上。然而，当用户享受云存储的便利时，用户也承受着隐私泄露的风险。当用户将文件存储在外部服务器时，用户就失去了对文件的控制，为了保障用户的隐私，需要对这些文件进行加密，传统的加密算法使得只有密钥拥有者才能对文件进行解密，这使得如果用户需要检索某数据时，服务器无法检索加密的文件，这就需要用户将所有文件下载下来，解密后再进行查找，这样的效率十分低下。于是，可搜索加密应运而生，可搜索加密是一种基于密文进行搜索查询的方案，用户可以直接在服务器上进行检索，而服务器无法知道检索的具体内容。本文将简要介绍可搜索加密及对称可搜索加密的研究进展。

一 可搜索加密

1.1 可搜索加密基本介绍

一般来说，可搜索加密可以分为四个过程。

（1）加密文件：用户在本地使用密钥对文件进行加密，然后将加密后的文件上传至服务器。

（2）陷门生成：合法用户利用陷门生成函数生成关键词相关的陷门，并将陷门提交给服务器。

（3）检索关键词：服务器利用陷门在服务器上存储的加密文件中进行检索，得到检索结果，将检索结果返还给用户。

（4）解密文件：用户利用密钥对服务器返还的加密文件进行解密。

1.2 可搜索加密的分类

从应用角度来看，可搜索加密可以分为四类：

（1）一对一模型：用户个人将加密文件存储于不可信赖的外部服务器上，只有用户个人进行加密并进行检索，该场景也是可搜索加密问题诞生的来源。

（2）多对一模型：多个不同的文件拥有者将其文件加密后上传至不可信赖的外部服务器，供特定的某个用户进行检索，实现文件上传者与文件接收者之间的传输。该模型要求文件的加密者和解密者不是同者，且只有文件接收者具有检索功能。

（3）一对多模型：单个文件拥有者将文件加密后上传至不可信赖的外部服务器，供特定的多个用户进行检索，以此实现与多个用户之间的数据共享。

（4）多对多模型：多个不同的文件拥有者将文件加密后上传至不可信赖的服务器，供特定的满足某条件的多个用户进行检索。

从构造可搜索加密方案的底层密码来看，可搜索加密可以分为两类：

（1）对称可搜索加密

对称可搜索加密所使用的加密方案是对称加密算法，加密和解密使用相同的密钥，陷门生成的过程中也需要该密钥。对称可搜索加密的特性使得该方案很适用于一对一模型：单个用户使用密钥加密个人文件，用相同的密钥生成陷门，将陷门提交给服务器进行检索。另外，在效率方面，对称可搜索加密方案通常具有算法简单开销小的优点。

（2）非对称可搜索加密

非对称可搜索加密在加解密时使用不同的密钥，在加密和检索目标密文时使用公钥，在解密和生成陷门时使用私钥。非对称可搜索加密公私钥分离的特性使其适用于多对一模型，发送者使用接收者的公钥加密文件，接收者使用自己的私钥生成想要检索的关键词相关的陷门，服务器根据陷门进行检索返回检索结果，接收者再用私钥解密检索结果文件，实现与发送者间的数据共享。然而，非对称可搜索加密方案所使用的加密算法往往非常复杂，开销较大。

本文接下来将主要针对对称可搜索方案阐述其研究进展。

二 对称可搜索加密的研究进展

2.1 对称可搜索加密的研究历史

对称可搜索加密起源于 2000 年 Song 等人撰写的论文[1],在这篇论文中 Song 首次提出了可搜索加密的思想，并且提出了第一个可搜索加密的方案，该方案的基本思想是将文档中的关键词划分开来，用户单独对每个关键词进行加密，用户将加密后的关键词分为两段，根据前半段 Li 生成密钥 Ki,用户再用流密码生成一系列 Si，对 Si 用 ki 产生伪随机数，将这两部分连接在一起与加密后的关键词进行异或得到的密文，将密文上传至服务器。检索时用户告知服务器陷门(E(Wi),Ki),服务器根据陷门对文档进行检索，将 E(Wi)与每一个Ci 进行异或，并检查异或结果是否为(Si,Fki(Si))的形式，如果是，则说明在该加密文档中找到了关键词，否则继续与下一个Ci进行异或。Song提出的该方案是基于序列扫描的，检索一个关键词要扫描整个文档，效率较低。

2003 年的时候 Goh 提出了基于索引的 ZIDX 方案[2]，该方案使用布隆过滤器来构建索引。布隆过滤器中每个元素s对应一个m位阵列，阵列初始化为0，如果该元素在集合中，则通过 r 个哈希函数将该元素对应的 h(s)映射成 1。检索时，判断 m 位阵列，如果位阵列内 h(s)有一个为 0，则该元素不在该集合中，如果都是 1，那么该元素可能在该集合中。在该方案中，用户使用伪随机函数处理文档唯一标识符和单词，使不同文档的同一单词所对应的码字不同，然后将码字插入布隆过滤器中，再在布隆过滤器添加些无关因素进行混淆，将构建好的索引上传至服务器，检索时用户利用伪随机函数生成陷门，将陷门和文档标识符上传至服务器，服务器利用布隆过滤器的性质来进行检索。该方案是索引结构，效率比Song的方案更高，然而该方案也存在着一些缺点：由于要存储索引，因此需要额外的空间；布隆过滤器存在一定的错误率。

以上两种方案是对称可搜索加密的两种经典方案，在此之后无数研究人员又在这些基础上提出了许多更新更高效的方案，下一部分我将阐述近些年对称可搜索加密的研究进展。

2.2 对称可搜索加密的研究进展

2.2.1 安全方面

很多可搜索加密算法会泄露 query pattern（即什么时候会重复一个询问）和 access pattern(即对于每个询问会返回哪个文件)。且大部分算法不满足前向安全（即可以在新的文件中搜索旧的令牌）。因此根据这些可加密算法的泄露信息，出现了一些对应的攻击。

2012 年 Islam 在第 19 届 NDSS 会议上发表相关论文[3]，该论文根据用户请求，请求返回的文档，部分曾出现在请求中的关键词，以及某关键词在某文档中的概率等，将某关键词在某文档中的概率的联合概率矩阵与检索的概率矩阵对比，找到最接近的，可以计算得出某请求所对应的具体关键词是什么，实现了对可搜索加密方案的攻击，这种攻击方案称

为查询恢复攻击(query recovery attack)。Islam的方法需要知道用户所有文档消息和部分请求消息。

2015 年 Cash 在论文[4]中提出了文件注入攻击（“File-Injection attack”）,即如果服务器向用户发送某邮件，用户会将该邮件正常加密上传至服务器，在用户进行下一次询问的时候，如果该邮件被返还，那么服务器就知道该邮件包括了用户请求的关键字，如果该邮件只含一个关键字，那么服务器就能够知道这次请求的对应关键字。但 Cash 并未将其用于查询恢复，而对于查询恢复，Cash 在 Islam 方案的基础上进行改进，对包含某关键词的文档进行计数，找到与某次询问的陷门数目相等的关键词。

2016 年 Yupeng Zhang 在 Usenix 会议上发表相关论文[5]，该论文将 Cash 提出的文件注入攻击用于询问恢复，先提出了二分检索方法，即对于 K=2^m 个关键词，服务器给用户发送 k 个文件，每个文件包括 K/2=2^(m-1)个关键词，根据返回文件结果可以得知这次询问对应关键词。（如 8 个关键词 k0k1k2k3…k7，第一个文件包含 k4k5k6k7，第二个文件包含 k2k3k6k7，第三个文件包含k1k3k5k7，最后只有第二个文件被返回，那么可以知道本次询问对应的是 k2 关键词。但二分检索方法有相应的应对措施，即设立阈值 T，将文档关键词限定在T以下，而 T远比K/2少，这样就使得服务器没法发送包含K/2个关键词的文件，二分检索失效。因此该论文又对二分检索攻击又进行了改进提出分层检索攻击，即将所有关键词分为 K/T 部分，每个部分包含T个关键词，对于某次询问服务器给用户发送包含各部分关键词的文件以确定关键词在哪个部分，然后再对部分关键词进行二分检索。Islam和Cash 的方案都需要知道用户所有文件的明文信息，而该方法不需要。因此在泄露消息少的情况下，该攻击方法要优于前两者的方法。

2.2.2 动态可搜索加密方案

亚线性的时间复杂度是可搜索加密方案所追求的，然而，在实际应用方面，也需要考虑到文件的查改增删，因此，有学者们先后提出了一些动态可搜索加密方案。

2005 年 Chang 和 Mitzenmacher 提出了第一个满足前向安全动态可搜索加密方案[6]，但直到 2014年才出现了第一个亚线性时间复杂度的可搜索加密方案，该方案由 Stefanov 提出[7]，在该方案的交互更新过程中，客户端从服务器获取不可忽略的数据量，并要求 O(N^α)（0<α<1）的工作存储来运行不经意的排序算法，此外，删除数据时需要进行的操作是将删除信息添加到加密数据库中，并在搜索过程中对它们进行重新划分，另外，当需要删除的数据超过总数据量的一半时，可以重新构建整个加密数据库。

2015 年 Yavuz 和 Guajardo 提出了更高效安全的动态可搜索加密方案[8]，使用了一个简单的基于矩阵的 O(m×n)数据结构，其中行和列分别表示关键字和文档标识符。在他们的方案中， 两个哈希表存储在客户端，其中一个必须与服务器共享，以便实时同步某些状态。这些哈希表分别表示倒排索引和前向索引，引入了将这两个哈希表连接起来的矩阵结构，提供了基于倒排索引的搜索操作和基于正排索引方法的更新操作，但该数据结构需要预先确定 m 和 n，这是该方案的缺陷，因此该方案更类似于静态方案。

2017 年 Kee Sung Kim 等人在 ACM CCS 会议上发表相关论文[9],提出了既满足前向安全又能实现高效更新的方案，该方案构建了一个同时利用了正排索引和倒排索引的新型数据结构 dual dictionary, 正排索引便于更新，倒排索引便于搜索，同时利用两对计数器和密钥分别加密现有的和新添加的数据，每次搜索时更新计数器和密钥对用以实现前向安全。虽然该方案需要额外的空间来存储两个索引，但相比之前所有的动态可搜索加密方案，该方案的复杂度是最低的，客户端的存储量与数据库中的关键词数量线性相关，服务器端的存储量与数据库中的文档-密钥对的数量线性相关，搜索的复杂度与包含关键词的搜索结果集的大小线性相

关，更新的时间复杂度与更新的文件中含有的不重复的关键词的数目线性相关。

2.2.3 连接关键词搜索

在实际的应用场景中，单个关键词的搜索显然是不够的，人们经常需要多个关键词来达到搜索目的，因此出现了连接关键词搜索问题。连接关键词搜索问最朴素的解决方案是分别检索单个关键词，然后在这些单个关键词的检索结果里进行交叉，得到想要的结果。但这效率十分低下，例如当搜索的关键词为“男性”和“李明”时，单独检索“男性”就会返回大量的文件，浪费资源，另外，这种方法也导致了不可忽视的数据泄露，因为它显示了匹配关键词的每个关键字的文档。

2004 年 Golle 等人[10]提出的方案安全性较好，他们展示了如何为每个连接关键词查询构建一组令牌，这些令牌可以根据数据库中的每个文档(更准确地说，针对编码的文档)进行测试以标识匹配的文档。这些解决方案只泄漏匹配的文档集(可能还有正在搜索的属性集)，但由于需要 O（d）的工作量（d 为数据库内文档数目）不适用于大型数据库，且无论结果集和匹配搜索关键词的文档数目有多少，都需要这么多的工作量。这种方法甚至需要 O（d）的用户与服务器间的交流和指数运算或 O（d）的配对操作。这种方法的另一个严重限制是它只适用于结构化属性值类型数据库，不支持自由文本搜索。

2013 年 Cash 提出了一种可搜索加密的方案[11]，可适用与大型数据库且适用于任意结构的文本搜索，支持布尔查询。该方法的核心思想是构建 Xset,对于每个关键字 w，都有 xtrap=F(Kx,w),而对于每个包含关键字 w 的索引 ind，都有 xtag=f(xtrap,ind),所有的 xtag 构成 Xset。然后进行多个关键字检索（w1,w2…wn）时，先检索认为返回数据会最少的关键字 w1，返回包含 w1 的 ind，再在 Xset 里检索是否有其余关键字与 ind 的配对，如果某个 Ind 满足所有关键字都存在相关配对，就返回这个 Ind。这样的话复杂度就减少很多。但该方法的缺点是仍然存在 equality pattern(哪些询问有同样的最不频繁的关键词 s-term),size pattern(每个询问中匹配第一个关键词的文档数目)等泄露,这些泄露仍可能被用于攻击。

2.2.4 其他

原有的可搜索加密方案都要求关键词精确，但实际应用中可能会出现关键词模糊的情况，2010 年 Jin Li 等人提出了模糊关键词检索的方案[12]，2012 年，Wang 等人进一步研究模糊关键词检索方案，并给出了形式化的安全性证明。

另外，大多数对称可搜索加密方案中假定的服务器都是诚实且好奇的，没有考虑过服务器不诚实且返回错误消息的情况。2012 年 QiCha 等人发表的论文指出[13]，在考虑半可信且好奇(semi-honest but curious)模型下，服务器可能为了节省流量而返回错误的搜索结果或者不完全的结果，因此，该论文提出了可验证的对称可搜索加密方案来解决这个问题，其中引入了基于哈希的检索树，要求服务器将检索路径的哈希序列作为证据一并返还给用户，用户可根据证据对服务器的检索结果进行完整性和正确性验证。2018 年 XueQiao Liu 等人[14]扩展了可验证的对称可搜索加密方案，将其推展至多用户的场景中，能更好地适应云存储。

结语

大数据的时代已经到来，云服务在人们的日常生活中应用得越来越多，学者们对可搜索加密的研究也一直在发展，但仍然有些问题没有解决，对称可搜索加密方案的效率、安全性以及可应用性还可以进一步提高。

参考文献

[1] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in 2000.

[2] E.-J. Goh, "Secure Indexes," Cryptology ePrint Archive, 2003.

[3] Islam, M. S., Kuzu, M., and Kantarcioglu, M. "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. "in 2012.

[4] Cash, D., Grubbs, P., Perry, J., And RistenpartI, T."Leakage-abuse attacks against searchable encryption" in 2015.

[5] Zhang, Yupeng, Katz, Jonathan, Papamanthou, Charalampos,"All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption." in 2016.

[6] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in 2005.

[7] Stefanov, E., Papamanthou, C. & Shi, E. (2013). Practical Dynamic Searchable Encryption with Small Leakage.. IACR Cryptology ePrint Archive, 2013, 832.

[8] Yavuz A A , Guajardo J . Dynamic Searchable Symmetric Encryption with Minimal Leakage and Efficient Updates on Commodity Hardware[C]// International Conference on Selected Areas in Cryptography. Springer, Cham, 2015.

[9] Kim, Kee & Kim, Minkyu & Lee, Dongsoo & Park, Je & Kim, Woo-Hwan. "Forward Secure Dynamic Searchable Symmetric Encryption with Efficient Updates". in 2017.

[10] P. Golle, J. Staddon and B. Waters, "Secure conjunctive keyword search over encrypted data," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3089, pp. 31-45, 2004.

[11] Cash D , Jarecki S , Jutla C , et al. "Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries".in 2013.

[12] J. Li et al, "Fuzzy keyword search over encrypted data in cloud computing," in 2010, . DOI: 10.1109/INFCOM.2010.5462196.

[13] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in 2012, . DOI: 10.1109/ICC.2012.6364125.

[14] X. Liu et al, "Multi-user Verifiable Searchable Symmetric Encryption for Cloud Storage," IEEE Transactions on Dependable and Secure Computing, pp. 1-1, 2018;2019;.