可验证对称加密搜索问题分析与研究

（申请清华大学工程硕士专业学位论文）

培 养 单 位: 计 算 机 科 学 与 技 术 系
学　　科: 计 算 机 技 术
申　请　人: 朱　洁
指 导 教 师: 李　琦　副　研　究　员

二〇一八年四月

Analysis and Research of Verifiable Searchable Symmetric Encryption

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the professional degree of

Master of Engineering

by

Zhu Jie

(Computer Technology)

Thesis Supervisor : Professor Li Qi

April, 2018

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

云存储的发展使得用户可以方便地存储、获取与分享数据。但与此同时，云存储也带来了很多安全问题，例如，数据隐私泄露等等。对称加密搜索的提出解决了数据隐私泄露问题，同时也保证了数据的可搜索性。通过使用对称加密搜索方案，用户可以在上传数据到云服务器之前，对数据进行加密，同时云服务器可以在用户的加密数据上进行搜索，从而确保数据隐私。

然而，对称加密搜索的假定是云服务器是诚实且好奇的，即云服务器会遵守协议，但现实情况中云服务器往往是不可靠的。为了解决该问题，可验证对称加密搜索技术相应提出，它通过结果验证技术可以到检测云服务器的恶意行为。但是，现有的可验证对称加密搜索方案都不完善，例如，不支持用户数据动态更新，依赖于特定的对称加密搜索方案，只支持单用户读写等等。

针对以上的问题，本文提出了一种通用的可验证对称加密搜索框架，该框架普适于任何加密搜索方案，支持用户数据更新，并且能够同时在单用户和多用户的场景下工作。本文的主要工作和创新点包括：

- 提出了一种单用户场景下的可验证对称加密搜索框架，并在此基础上设计了结果验证算法，该算法能同时保证数据新鲜性和数据完整性。该框架支持用户数据动态更新，并且将验证索引从对称加密搜索方案中解耦，使其可以为任何加密搜索方案提供结果验证功能。
- 提出了一种多用户场景下的可验证对称加密搜索框架。该框架支持单用户写，多用户读，确保了多用户场景下的数据新鲜性，并实现了数据共享场景下的结果验证。
- 本文采用了一个开源数据集作为测试数据，在本地环境对该框架进行了实验测试。安全性分析和实验表明，本文提出的可验证对称加密搜索框架不泄露数据隐私，并且给对称加密搜索方案引入的额外计算开销和通信开销很小，几乎可以忽略不计。

关键词：对称加密搜索，结果验证，云存储

Abstract

Cloud storage allows users to retrieve and share their data conveniently. Meanwhile, cloud storage also brings serious data privacy issues, i.e., the disclosure of private information. In order to ensure data privacy without losing data usability, Searchable Symmetric Encryption (SSE) has been proposed. By using SSE, users can encrypt their data before uploading to cloud services, and cloud services can directly operate and search over encrypted data, which ensures data privacy.

However, most SSE schemes only work with honest-but-curious cloud services that do not deviate from the prescribed protocols. This assumption does not always hold in practice due to the untrusted nature in storage outsourcing. To alleviate the issue, there have been studies on Verifiable Searchable Symmetric Encryption (VSSE), which functions against malicious cloud services by enabling results verification. But to our best knowledge, existing VSSE schemes exhibit very limited applicability, such as only supporting static database, demanding specific SSE constructions, or only working in the single-user model.

In this paper, we proposed a generic verifiable SSE framework in both single-user model and single-writer multiple-reader model, which provides verifiability for any SSE schemes and further supports data updates. In summary, our contributions are three-fold:

- We proposed a verifiable SSE framework in single-user model and designed the result verification algorithms. The framework seperate the verification index from the SSE construction and can provides generic verification for any SSE schemes. The algorithms guaranteed both data freshness and data integrity with support of data updates.
- We proposed the first verifiable SSE framework in the single-writer and multiple-reader model, which ensures data freshness across multiple users and provides result verification under data sharing scenario.
- We implemented our framework in a local enviroment and fed it with an open-source data set. Rigorous analysis and experimental evaluations show that our shceme is secure and introduces small overhead for result verification.

Key words: Searchable Symmetric Encryption; Result Verification; Cloud Storage

目 录

主要符号对照表

SSE	加密搜索 (Searchable Symmetric Encryption)
VSSE	可验证加密搜索 (Verifiable Searchable Symmetric Encryption)
MPT	默克尔帕特里夏树 (Merkle Patricia Tree)
IH	增量哈希 (Incremental Hash)
$\mathcal{W}$	关键字集合
$\|W\|$	关键字集合大小
w_i	关键字，其中 $i \in \{1, \cdots, \|W\|\}$
$\mathcal{D}$	明文文件集合
D_{w_i}	包含关键字 w_i 的明文文件集合
C	密文文件集合
C_{w_i}	包含关键字 w_i 的密文文件集合
d	明文文件
c	密文文件
W_d	文件 d 包含的关键字集合
τ	搜索令牌
λ	验证索引
π	鉴别符

第 1 章　绪论

1.1　研究背景及选题意义

云存储使得用户可以随时随地地存取数据，并且极大地方便了用户之间的数据共享，降低了维护数据的成本[1–7]。但与此同时，云存储也带来了许多安全性问题，例如，数据丢失，数据隐私泄露等等。总体来说，云存储带来的安全性问题可以分为以下两类：

- 可用性问题。要求云服务器保证数据不丢失，用户可以将云端作为数据中枢进行数据备份和同步。目前，一般的云服务提供商都采用了多副本的方式保障数据的可用性，即将数据的多个副本分别写入其他的存储节点，当一个节点发生故障时，其他节点上的数据继续提供服务，同时通过其他节点中的数据副本，快速恢复故障节点上丢失的数据。目前，针对数据可用性的相关学术研究包括数据拥有证明 (Proof of Data Possession, PDP)[2,8–10] 以及数据可恢复性证明 (Proof of Retrievability, PoR)[1,5,11]。
- 隐私性问题。要求云服务器保证数据的隐私并且不泄露数据。目前，云服务提供商一般采用数据加密方式对隐私数据进行保护，但数据加密往往会导致数据可用性的降低，例如数据失去可搜索性。因此加密搜索 (Searchable Encryption, SE) 应运而生。加密搜索技术主要分为两类，一是对称加密搜索 (Searchable Symmetric Encryption, SSE)[12–16]，二是公钥加密搜索 (Public Key Encryption with Keyword Search, PEKS)[17]。

加密搜索的提出，使得用户可以在上传数据给云服务器之前，对其进行加密，并且使得云服务器可以在加密数据上进行搜索。从而既保证了数据隐私性，又保证了数据的可搜索性。目前，由于效率问题，应用较为广泛的为对称加密搜索技术。然而，大部分的对称加密搜索方案都基于服务器是诚实且好奇的假设[13–15]，即服务器会遵循协议但是可以从用户的查询中推断相关信息。这种假设在实际应用场景中往往是不成立的。因为云服务器可能会因为外部攻击，内部配置错误，软件错误等等问题而导致其违反原有协议[7,18]。这种协议违反所导致的最常见问题就是服务器返回的搜索结果不完整。例如，云服务器有可能为了节省计算开销和通信开销而返回少量搜索结果给用户，甚至有可能不返回搜索结果给用户。

为了解决该问题，可验证对称加密搜索技术也相应提出[3,18–24]。可验证对称加密搜索技术允许用户对搜索结果进行验证，从而来检测服务器的不诚信行为，保障加密搜索的正确性。然而，据我们所知，现有的可验证对称加密搜索方案都是

不完善的。例如，有的方案[19,20,23,24] 不支持数据更新，只能作用在静态数据库中，数据库若有变化则需要重建整个索引。有的方案[3,21,22] 无法防止服务器故意返回空结果来规避结果验证。特别需要说明的是，以上这些方案[3,21,22] 在用户提交的关键字不存在于数据库中时，是不返回任何搜索结果的，这就导致了服务器可以对任意关键字返回空结果来规避结果验证，除非用户在本地保留数据库的所有关键字集合。另外，大部分的可验证对称加密搜索方案[3,18–24] 仅仅支持在单用户场景下工作，即用户自己写自己读的场景，而现实情况中，数据往往有共享需求，即一方写多方读的多用户场景①。表格 1.1比较了现有的可验证对称加密搜索方案。

1.2　本文的主要内容

本文基于默克尔帕特里夏树 (Merkle Patricia Tree, MPT) 和增量哈希 (Incremental Hash, IH) 技术，提出了一种单用户场景下的通用可验证对称加密搜索框架。该框架将验证索引从对称加密搜索方案中解耦，使其可以与任何对称加密搜索方案结合，包括但不限于论文[14,15,22] 中的方案。该验证索引基于支持动态更新的数据结构 MPT 构建，因此支持用户动态更新其数据集，而不需要重新构建验证索引。该验证索引将加密后的关键字和其对应的文件存储于叶子节点中，从而使得 MPT 的根节点成为用户数据完整性的见证，用于后续支持结果验证。同时，本文还提出了基于该验证索引的一系列验证机制，来确保数据完整性和数据新鲜性的验证。与以往的方案不同[3,21,22]，我们的方案要求服务器不管搜索关键字存在与否，都需要给用户返回一个“证明”，用于让用户验证服务器是故意返回了空结果还是搜索关键字的确不存在与现有数据集中。需要特别说明的是，我们的方案不需要用户在本地维护文件集对应的关键字集合。

此外，基于以上方案，本文利用时间戳链和公钥加密机制，首次提出了一种多用户场景下的通用可验证加密搜索框架。该框架通过时间戳链和公钥加密机制构建了出了与 MPT 根哈希相关的鉴别符，解决了多用户共享数据情况下的数据新鲜性验证问题，实现了多用户下的结果验证。

本文通过严格的安全性证明，确保了方案不泄露用户的数据隐私信息。另外，本文通过实验表明，单用户场景和多用户场景下的可验证加密搜索框架效率很高，与加密搜索法方案结合时，给加密搜索引入的额外开销很小，几乎可以忽略不计。

① 本文所述的多用户场景均指一方写入，多方读取的场景，以下不做特别说明。

表 1.1　现有可验证对称加密搜索方案比较

	动态性 [1]	新鲜性 [2]	完整性 [3]	验证效率 [4]	通用性 [5]
KPR11[3]	✓	✓	×	$O(\|W\|)$	✓
KO12[19]	×	-	×	$O(n)$	×
CG12[20]	×	-	✓	$O(log(\|W\|))$	×
KO13[21]	✓	✓	×	$O(n)$	×
SPS14[22]	✓	✓	×	$min\{\alpha + log(N), rlog^3(N)\}$	×
CYGZR15[23]	×	-	×	$O(\|W\|) + O(r)$	×
BFP16[18]	✓	✓	✓	$O(r)$	✓
OK16[24]	×	-	✓	$O(r)$	✓
我们的方案	✓	✓	✓	$O(log(\|W\|))$	✓

[1] 注意，动态性是指方案是否支持用户数据动态更新，由此可将可验证对称加密搜索方案分为静态和动态两种类型，后者在功能性上更完善。

[2] 注意，'×' 表示有实现的需求但是该方案没有实现, 而'-' 表示没有实现的需求。具体而言，静态的可验证对称加密搜索方案不存在数据新鲜性问题，因此方案[19,20,23,24] 也没有进行数据新鲜性验证的需求。

[3] 我们考虑各种数据完整性攻击，尤其包括服务器故意返回空结果来规避结果验证的场景。

[4] 验证效率是指服务器进行结果验证支持所需要的计算开销。对于表格中的非通用型方案[19–23] 来说，由于他们的方案并没有将验证索引从加密搜索方案中解耦，因此他们的验证效率和服务器进行加密搜索所需的计算开销是等价的。这里，n 代表所有文件的数量, $|W|$ 表示所有关键字的数量, r 表示包含某一特定关键字的文件数量, α 表示某一关键字历史上被加入到集合中的次数[22], N 表示 (文件，关键字) 对的数量。

[5] 一个通用的可验证对称加密搜索方案是指该方案可以为任何加密搜索方案提供结果验证，而非通用的可验证对称加密搜索方案表示该方案仅支持在特定的加密搜索方案下工作。

1.3　本文的结构安排

本文的结构如下，第 1 章为绪论，介绍了研究背景、选题意义以及主要工作内容；第 2 章为相关研究综述，介绍了对称加密搜索、可验证对称加密搜索等相关工作的研究现状，并对本文用到的相关概念和先验知识进行了介绍；第 3 章为单用户下的可验证对称加密搜索方案研究，从适用场景、方案流程、算法分析、安全性证明和实验验证几个角度，完整的介绍了单用户场景下可验证对称加密搜索框架方

案；第 4 章为多用户下的可验证对称加密搜索方案研究，整体结构与第 3 章类似；第 5 章总结了全文，并对可验证加密搜索领域未来可能的发展方向进行了分析。

第 2 章　相关工作及问题定义

2.1　相关研究工作

2.1.1　安全云存储方案

可验证的云存储服务已经被广泛的研究过，例如，数据拥有性证明 (Proof of Data Possession, PDP)[2,8–10]，数据可取回证明 (Proof of Retrievability, POR)[1,5,11] 等等。这些方案主要侧重于云端存储数据的完整性验证，并支持丢失数据的恢复。注意，这些方案与加密搜索场景下的结果验证是不同的，因为加密搜索的结果验证不仅需要验证某个文件本身的完整性，还需要验证整个搜索结果集合是否完整。而这些方案只能单纯验证数据块的完整性，不支持对搜索结果完整性的验证。

2.1.2　安全加密搜索方案

加密搜索的概念首次由 Song 等人[12] 在 2000 年提出，他们的方案允许用户将加密后的数据集存储到云端，并同时保证用户在该加密数据集上进行搜索的能力。随后，加密搜索方案被广泛的研究，总体来说可以分为以下两个分支：对称加密搜索 (Searchable Symmetric Encryption, SSE) 和公钥加密搜索 (Public Key Encryption with Keyword Search, PEKS)。其中，最经典的对称加密搜索方案[13] 由 Curtmola 等人提出，他们的方案利用了明文搜索中的倒排索引的思想，并且他们对加密搜索的安全性进行了严格的定义，提出加密搜索方案至少要在面对一个被动敌手的情况下是安全可靠的。目前还有许多不同的对称加密搜索方案实现了不同的搜索功能。例如，动态对称加密搜索 (Dynamic SSE) 方案[14,15,22] 允许用户更新其数据集，支持关键字排序 (Ranked Keyword Search) 的对称加密搜索方案[38] 允许用户获取根据某一影响因子排序后的搜索结果。最经典的公钥加密搜索方案[17] 由 Boneh 等人提出，他们的方案利用了双线性映射技术。总体来说，公钥加密搜索方案的性能是远远低于对称加密搜索方案的。

2.1.3　可验证数据结构

可验证数据结构 (Authenticated data structure) 在不可信的云存储环境中，主要被用于验证数据块的完整性。典型的可验证数据结构包括：默克尔树 (Merkle Tree, MT)[25], 可验证哈希表 (Authenticated Hash Table, AHT)[26] 以及可验证跳表 (Authenticated Skip List, ASL)[27,28]。其中，默克尔树是最常见的用于验证数据完整

性的数据结构，但是默克尔树对数据更新的支持不够灵活。采用默克尔树实现的可验证对称加密搜索方案[3] 由于没法在中间节点存储关键字信息，因此也不支持与关键字相关的搜索。可验证哈希表采用了 RSA 累加器 ① (RSA Accumulator) 方法来实现数据验证，但是它的搜索和更新性能都较低。具体而言，可验证哈希表的搜索与更新速度在毫秒级别，而我们采用的默克尔帕特里夏树的搜索更新速度在微秒级别。可验证跳表采用了类似多级链表的方式来实现，一定程度上提升了搜索性能，但如果它将关键字信息存储于搜索路径上，存储空间将比默克尔帕特里夏树大很多。

2.1.4　可验证对称加密搜索方案

由 Kamara 等人提出的 CS2 方案[3] 通过使用默克尔树构建验证索引来支持用户对搜索结果的验证。具体的做法是，以加密的关键字作为“键”，以该关键字对应的加密文件集合作为“值”，将该“键值对”存储在默克尔树的叶子结点上。用户在本地需要保留默克尔树的根哈希作为一个指纹信息。在进行结果验证时，用户需要通过其搜索的关键字本身及服务器返回的该关键字对应在默克尔树上的路径来重构出该根哈希，并与保留的根哈希进行比对，从而来进行结果验证。但是他们的方案无法检测服务器恶意返回空结果的情况。关键的原因是，当用户搜索的关键字不存在时，默克尔树上不会存在该关键字对应的路径，因此服务器无法返回任何信息给用户。解决该问题的一个简单的方法是在构建默克尔树时，将整个字典空间中所有可能的关键字集合都存储在默克尔树中，但这样做会导致大量的空间浪费。

近期，Kurosawa 等人提出了一系列可验证对称加密搜索方案[19,21,24]。但是他们的方案要么效率很低，要么不支持用户数据动态更新。其中方案[19] 需要线性搜索时间并且不支持数据动态更新。他们的扩展方案[21] 支持了用户数据更新，该方案通过消息验证码 (Message Authenticated Code, MAC) 来确保了数据完整性，通过 RSA 累计器确保了数据新鲜性，但是方案的搜索复杂度超过了线性时间，并且该方案需要用户在本地维护一个关键字集合来探测服务器故意返回空结果的情况，这将引入较大的空间开销。Ogata 等人也提出了一个通用的可验证对称加密搜索框架[24]，该方案可以为任何对称加密搜索方案提供结果验证服务，并且不需要用户自己在本地维护一个关键字集合，但是他们的方案仍然是一个静态的方案，即不支持用户数据更新。方案[20] [23] 也同样只是静态方案。

由 Stefanov 等人提出的方案[22] 采用了时间戳和消息验证码机制来实现了结

① RSA 为提出该算法的三个密码学家名字的首字母，分别为 Ron Rivest, Adi Shamir, 和 Leonard Adleman

果验证，但是他们的方案没法防御服务器故意返回空结果来规避结果验证的情况。Bost 等人提出的方案[18] 是目前为止最完善的普适性可验证对称加密搜索方案，但他们的方案在搜索时需要与服务器进行两轮通信，加密搜索和结果验证过程在服务器端无法并行进行，即用户需要在拿到加密搜索的结果后再与服务器进行通信来进行验证，这将导致较大的验证时延和通信开销，并且他们的方案同样也不支持多用户情况下的验证。

总体来说，一个完善的普适性可验证对称加密搜索方案首先应该支持数据新鲜性和数据完整性验证，尤其要关注搜索结果为空时的验证，这一点被大部分的方案忽略。其次该方案应该在支持结果验证的同时，尽量降低用户本身的存储和计算开销，例如不需要用户本身去维护一个本地关键字集合。另外，该方案还应该支持用户数据的更新，并且能够支持多用户场景下的结果验证。综上所述，在单用户场景中，现有的可验证加密搜索方案无法在保证验证效率的同时，完善地验证数据新鲜性和数据完整性。并且现有的方案都不能满足多用户场景下的对称加密搜索结果验证。这需要我们利用合理的数据结构，并设计合理的机制来设计一个普适的可验证对称加密搜索框架。

2.1.5　可验证公钥加密搜索方案

第一个可验证的非对称加密搜索方案[29] 由 Zheng 等人提出，他们的方案采用了基于属性的关键字（Attribute-based keyword，ABK），但是他们的方案也只适用于数据库静态的情况。基于他们的工作，Liu 等人又提出了一个更高效的可验证非对称加密搜索方案[30]，Sun 等人也提出了一个支持多关键字搜索的可验证公钥加密搜索方案[7]。然而，由于非对称加密本身的限制，他们的方案必不可少地需要引入一个可信第三方，并且搜索的性能大大低于可验证对称加密搜索方案。

2.1.6　多用户加密搜索方案

目前有一些多用户场景下的加密搜索方案[13,31–33]，但这些方案都不支持结果验证。Curtmola 等人在 2006 年即提出了一个基于广播加密的多用户加密搜索方案[13]，该方案允许数据持有者将数据分享给其他用户，并且数据持有者可以设定其他用户的访问控制权限，可以随时撤销或者新增用户。Yang 等人也通过双线性映射 (Bilinear Mapping) 技术提出了一种支持多用户读多用户写的方案[31]，但是该方案的搜索效率与数据集合的大小成正比，无法应用于数据量很大的场景中。Jerecki 等人随后又提出了一个多用户加密搜索方案[32]，然而该方案需要数据持有者和搜索用户进行频繁的交互，给数据持有者带来了很大的通信开销。近期，Sun 等人提

出了一个非交互式的多用户加密搜索方案[33]，该方案降低了数据持有者的通信开销，但他们的方案不支持用户数据更新。注意，这里我们需要强调，下文我们提出的多用户场景下的可验证对称加密搜索方案旨在为实现了多用户加密搜索的方案提供一个跨用户的结果验证，而不是旨在设计一个多用户方案。据我们目前所知，现有的可验证对称加密搜索方案都只支持单用户场景下的结果验证，而不支持多用户场景下的结果验证，因为多用户场景下的结果验证会面临更多的困难。例如，当数据在不同用户之间共享时，由于数据搜索用户无法探知数据持有者是否对数据集合进行了更新，因此一个恶意的服务器可以返回旧数据集合的搜索结果。除非数据持有者在每次更新时都通知所有的搜索用户，但这将会带来很大的通信开销。我们将在第 4 章具体讲述我们的多用户方案。

2.2　先验知识

2.2.1　增量哈希

增量哈希 (Incremental Hash, IH) 由 Bellare 等人提出[34]，并被已有的加密搜索方案[3] 所使用。增量哈希函数是一个抗碰撞的函数：$IH : \{0,1\}^* \rightarrow \{0,1\}^l$，两个随机字符串通过增量哈希函数相加或相减后，生成的哈希值不会产生碰撞。举例来说，假设 D 是一个包含关键字 w 的数据集合，它的增量哈希值为 H。当一个新数据 d 加入到 D 中后，新的数据集合变为 D'，即 $D + d$。对于原有数据集 D 来说，数据 d 的加入只是微小的变动。这使得增量哈希函数可以基于数据 d 和现有哈希值 H，并通过“加法”操作快速的计算出新文件集 D' 的一个抗碰撞哈希值，而不需要基于新文件集 D' 重新计算哈希值，这使得哈希操作的性能得到了较大的提升。

2.2.2　默克尔帕特里夏树

默克尔帕特里夏树 (Merkle Patricia Tree, MPT) 最早在以太坊 [35,36] (Ethereum) 中提出，它将传统的字典树 (Trie Tree) 和默克尔树结合，使得该树同时具有查找和验证的功能。MPT 具有四种类型的节点，分别为空节点（Blank Node，BN），叶子节点（Leaf Node，LN），分支节点（Branch Node，BN）和扩展节点（Extension Node，EN）。其中空节点只是一个不存任何信息的节点，叶子节点存储了键值对（key-value pair），扩展节点也存储了键值对，但扩展节点的键值分别为其子节点的公共前缀和子节点的哈希值。分支节点有 17 个元素，其中前 16 个元素代表了该节点上有可能的分支，即 16 个十六进制数字，第 17 个元素为值。当某一个关键字在该分支节点匹配完成时，该关键字对应的值就存储该元素中。

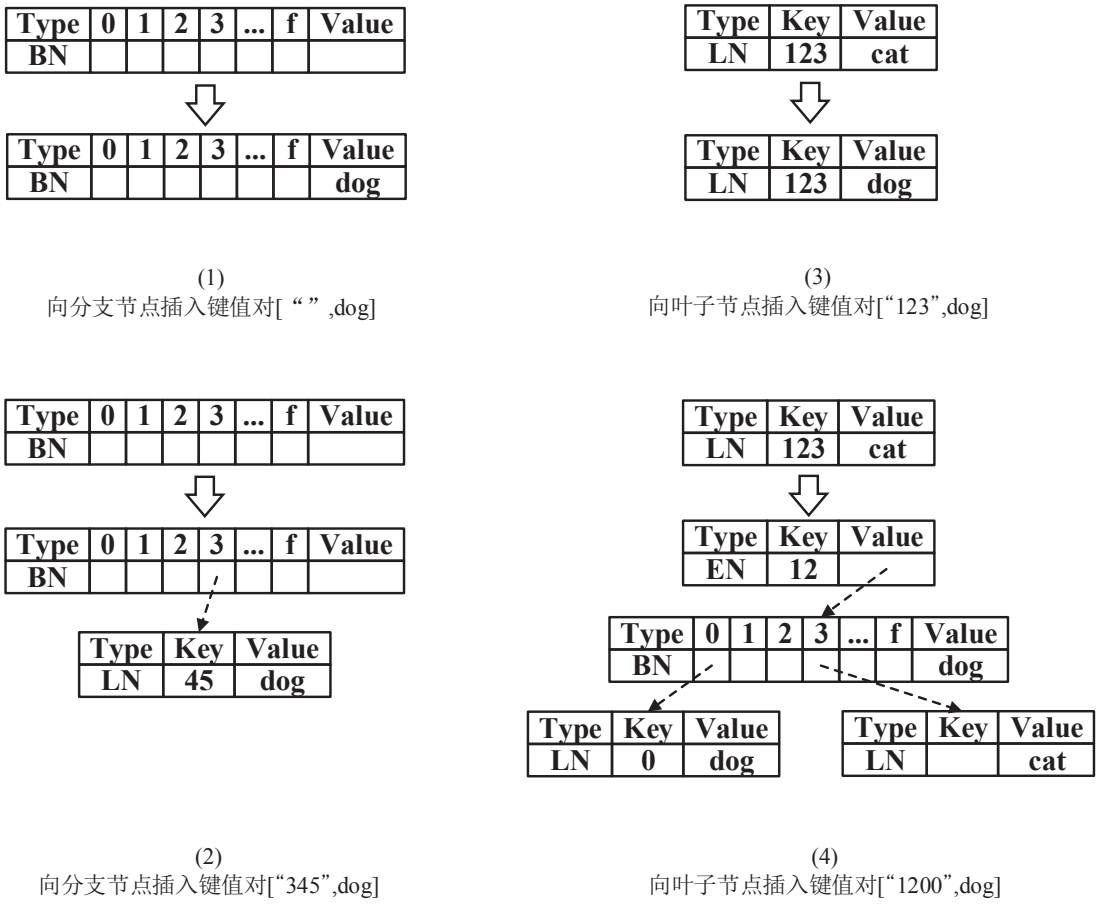

图 2.1　The Merkle Patricia Tree

图 2.1 通过四个简单的例子展示了 MPT 树的插入过程。首先是将一个“键值对”插入到分支节点，这分为两种情况。如果当前的键空间已经为空，我们可以直接将“值”插入到分支节点的第 17 个位置。否则，在经过了分支节点匹配后，键空间中“剩余键”和“值”将会存储在分支节点指向的一个新的叶子节点中。其次是将“键值对”插入到叶子节点，也分为两种情况。如果当前键空间中“剩余键”与叶子节点中的“键”正好匹配，直接将叶子节点中的“值”修改为新的“值”即可。否则，我们将找到当前键空间“剩余键”和叶子节点“键”的共同前缀，将其作为一个新建的扩展节点的“键”，并新建一个分支节点，将现有的叶子节点和新建的叶子节点作为子节点插入到分支节点对应的空间中。

注意，MPT 中的每一个节点都通过可递归长度前缀法[37](Recursive Length Prefix, RLP) 进行了编码并对编码值再进行了哈希。数据库中存储了每个节点的“键值对”键值对，其中“键”为该节点 RLP 编码的哈希，“值”为该节点的 RLP 编码。这样每个节点可以通过他的哈希值被引用，同时保证了 MPT 的可搜索性和可

验证性。通过这种方式，MPT 的根哈希成为了整棵树的指纹信息，根哈希的值由所有下层节点的哈希值所决定，任何节点的微小改变都会导致根哈希的值发生变化。此外，MPT 与默克尔树不同，MPT 是完全确定性的，即一组相同的“键值对”采用不同的顺序插入到 MPT 中，最终得到的根节点哈希值是相同的，而默克尔树不具有这个性质。

2.3　问题定义

在本节中，我们将正式定义方案的攻击模型，方案需要解决的问题以及方案需要实现的目标。

2.3.1　攻击模型

在单用户场景中，数据持有者和数据搜索用户是同一人，而在多用户场景中，这两者是分开的。我们假定数据持有者本身是可信的，而数据搜索用户不可信。此外，我们假定提供存储和搜索服务的云服务器是不可信的，即 1) 云服务器会试图从用户的加密数据和搜索请求中推断出一些隐私信息; 2) 云服务器有可能会因为外部攻击、配置错误、软件错误等原因背离原有协议，从而导致产生数据新鲜性攻击和数据完整性攻击，用以节省其自身的计算开销和通信开销。数据新鲜性攻击和数据完整性攻击的正式定义如下：

定义 2.1 (数据新鲜性攻击)： *在对称加密搜索中，数据新鲜性攻击是指一个恶意的云服务器试图从旧数据集中返回搜索结果，而不从最新的数据集中返回搜索结果。正式地，让 $\Delta_{n-1} = \{\delta_1, \delta_2, \cdots, \delta_{n-1}\}$ 代表用户数据集的历史版本，δ_n 代表用户的最新数据集，云服务器返回的搜索结果为 δ_i 的子集，其中 $1 \le i \le n-1$。*

定义 2.2 (数据完整性攻击)： *在对称加密搜索中，数据完整性攻击是指一个恶意的云服务器试图篡改搜索结果，阻止数据搜索用户获取到完整的搜索结果。正式地，让 τ 代表对称加密搜索方案中的搜索令牌，δ_i 代表数据集，其中 $1 \le i \le n$。对应的搜索结果应为 $\mathcal{F}(\delta_i, \tau)$，但云服务器返回的搜索结果为 $\mathcal{G}(\delta_i, \tau)$，其中 $\mathcal{G}(\delta_i, \tau) \neq \mathcal{F}(\delta_i, \tau)$。*

2.3.2　设计目标

本论文旨在设计一种普适的可验证加密搜索框架，即该方案可以和任意加密搜索方案相结合，包括但不限于[14,15,22]，使其能够完成结果验证的功能。本方案将现有的加密搜索方案当做黑盒，总体来说，需要满足以下几个需求：

1. **机密性：**数据和关键字的机密性是加密搜索最近本的安全需求。它保证了用户的明文数据和关键字信息无法被其他不可信第三方所推断。并且保证了敌手无法从方案的加密数据集，验证索引以及搜索关键字中推断出任何有用的隐私信息。
2. **可验证性：**一个可验证的对称加密搜索方案应该能够验证搜索结果的正确性和完整性，即防止重放攻击和数据完整性攻击。
3. **高效性：**一个可验证对称加密搜索方案应该达到次线性的计算复杂度，即对数复杂度 $O(log(|W|))$，其中 $|W|$ 是关键字的总数，并且应该在支持用户数据更新的情况下仍然能达到该复杂度。注意，这里的计算复杂度仅仅指服务器提供结果验证服务时所需的额外计算复杂度，不包括加密搜索方案本身带来的计算复杂度。

第 3 章　单用户下的可验证对称加密搜索方案研究

第 4 章　多用户下的可验证对称加密搜索方案研究

第 5 章　总结与展望

参考文献

[1] Juels A, Kaliski Jr B S. Pors: Proofs of retrievability for large files[C]//Proc. of CCS. [S.l.: s.n.], 2007: 584–597.

[2] Ateniese G, Di Pietro R, Mancini L V, et al. Scalable and efficient provable data possession[C]// Proc. of Security and privacy in communication netowrks (SecureComm). [S.l.: s.n.], 2008.

[3] Kamara S, Papamanthou C, Roeder T. Cs2: A semantic cryptographic cloud storage system[R]. [S.l.]: Tech. Rep. MSR-TR-2011-58, Microsoft Technical Report (May 2011), http://research. microsoft. com/apps/pubs, 2011.

[4] Wang Q, Wang C, Ren K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE TPDS, 2011, 22(5): 847–859.

[5] Stefanov E, van Dijk M, Juels A, et al. Iris: A scalable cloud file system with efficient integrity checks[C]//Proc. of Annual Computer Security Applications Conference (ACSAC). [S.l.: s.n.], 2012.

[6] Kamara S, Papamanthou C. Parallel and dynamic searchable symmetric encryption[C]//Proc. of International Conference on Financial Cryptography and Data Security(FC). [S.l.: s.n.], 2013.

[7] Sun W, Liu X, Lou W, et al. Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data[C]//Proc. of INFOCOM. [S.l.: s.n.], 2015.

[8] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores[C]//Proc. of CCS. [S.l.: s.n.], 2007.

[9] Erway C C, Küpçü A, Papamanthou C, et al. Dynamic provable data possession[J]. ACM TISSEC, 2015, 17(4): 15.

[10] Zhu Y, Hu H, Ahn G J, et al. Cooperative provable data possession for integrity verification in multicloud storage[J]. IEEE TPDS, 2012, 23(12): 2231–2244.

[11] Bowers K D, Juels A, Oprea A. Proofs of retrievability: Theory and implementation[C]//Proc. of the workshop on Cloud computing security (SCC). [S.l.: s.n.], 2009.

[12] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]//Proc. of S&P. [S.l.: s.n.], 2000.

[13] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5): 895–934.

[14] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption[C]//Proc. of CCS. [S.l.: s.n.], 2012: 965–976.

[15] Cash D, Jaeger J, Jarecki S, et al. Dynamic searchable encryption in very-large databases: Data structures and implementation.[C]//NDSS: volume 14. [S.l.: s.n.], 2014: 23–26.

[16] Wang Q, He M, Du M, et al. Searchable encryption over feature-rich data[J]. IEEE Transactions on Dependable and Secure Computing, 2016.

[17] Boneh D, Di Crescenzo G, Ostrovsky R, et al. Public key encryption with keyword search[C]// Proc. of EUROCRYPT. [S.l.: s.n.], 2004.

[18] Bost R, Fouque P A, Pointcheval D. Verifiable dynamic symmetric searchable encryption: Optimality and forward security.[J]. IACR Cryptology ePrint Archive, 2016, 2016: 62.

[19] Kurosawa K, Ohtaki Y. Uc-secure searchable symmetric encryption[C]//Proc. of International Conference on Financial Cryptography and Data Security (FC). [S.l.: s.n.], 2012.

[20] Chai Q, Gong G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers[C]//Proc. of International Conference on Communications (ICC). [S.l.: s.n.], 2012.

[21] Kurosawa K, Ohtaki Y. How to update documents verifiably in searchable symmetric encryption [C]//Proc. of International Conference on Cryptology And Network Security (CANS). [S.l.: s.n.], 2013.

[22] Stefanov E, Papamanthou C, Shi E. Practical dynamic searchable encryption with small leakage [C]//Proc. of NDSS. [S.l.: s.n.], 2014.

[23] Cheng R, Yan J, Guan C, et al. Verifiable searchable symmetric encryption from indistinguishability obfuscation[C]//Proc. of AsiaCCS. [S.l.: s.n.], 2015.

[24] Ogata W, Kurosawa K. Efficient no-dictionary verifiable sse[J]. IACR Cryptology ePrint Archive, 2016, 2016: 981.

[25] Merkle R C. A digital signature based on a conventional encryption function[C]//Proc. of EUROCRYPT. [S.l.: s.n.], 1987.

[26] Papamanthou C, Tamassia R, Triandopoulos N. Authenticated hash tables[C]//Proc. of CCS. [S.l.: s.n.], 2008.

[27] Pugh W. Skip lists: a probabilistic alternative to balanced trees[J]. Communications of the ACM, 1990, 33(6): 668–676.

[28] Goodrich M T, Tamassia R, Schwerin A. Implementation of an authenticated dictionary with skip lists and commutative hashing[C]//Proc. of DARPA Information Survivability Conference & Exposition (DISCEX). [S.l.: s.n.], 2001.

[29] Zheng Q, Xu S, Ateniese G. Vabks: verifiable attribute-based keyword search over outsourced encrypted data[C]//Proc. of INFOCOM. [S.l.: s.n.], 2014.

[30] Liu P, Wang J, Ma H, et al. Efficient verifiable public key encryption with keyword search based on kp-abe[C]//Proc. of Broadband and Wireless Computing, Communication and Applications (BWCCA). [S.l.: s.n.], 2014.

[31] Yang Y, Bao F, Ding X, et al. Multiuser private queries over encrypted databases[J]. International Journal of Applied Cryptography, 2009, 1(4): 309–319.

[32] Jarecki S, Jutla C, Krawczyk H, et al. Outsourced symmetric private information retrieval[C]// Proc. of CCS. [S.l.: s.n.], 2013.

[33] Sun S F, Liu J K, Sakzad A, et al. An efficient non-interactive multi-client searchable encryption with support for boolean queries[C]//Proc. of ESORICS. [S.l.: s.n.], 2016.

[34] Bellare M, Goldreich O, Goldwasser S. Incremental cryptography: The case of hashing and signing[C]//Proc. of CRYPTO. [S.l.: s.n.], 1994.

[35] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum Project Yellow Paper, 2014, 151: 1–32.

[36] Merkle patricia tree[M]. [S.l.: s.n.].

[37] Rlp code[M]. [S.l.: s.n.].

[38] Wang C, Cao N, Li J, et al. Secure ranked keyword search over encrypted cloud data[C]//Proc. of ICDCS. [S.l.: s.n.], 2010.

致　谢

衷心感谢导师 xxx 教授和物理系 xxx 副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 xxx 教授热心指导与帮助，不胜感激。感谢 xx 实验室主任 xx 教授，以及实验室全体老师和同学们的热情帮助和支持！本课题承蒙国家自然科学基金资助，特此致谢。

感谢 LaTeX 和 ThuThesis[?]，帮我节省了不少时间。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

xxxx 年 xx 月 xx 日出生于 xx 省 xx 县。

xxxx 年 9 月考入 xx 大学 xx 系 xx 专业，xxxx 年 7 月本科毕业并获得 xx 学士学位。

xxxx 年 9 月免试进入 xx 大学 xx 系攻读 xx 学位至今。

发表的学术论文

[1] Yang Y, Ren T L, Zhang L T, et al. Miniature microphone with silicon- based ferroelectric thin films. Integrated Ferroelectrics, 2003, 52:229-235. (SCI 收录, 检索号:758FZ.)

[2] 杨轶, 张宁欣, 任天令, 等. 硅基铁电微声学器件中薄膜残余应力的研究. 中国机械工程, 2005, 16(14):1289-1291. (EI 收录, 检索号:0534931 2907.)

[3] 杨轶, 张宁欣, 任天令, 等. 集成铁电器件中的关键工艺研究. 仪器仪表学报, 2003, 24(S4):192-193. (EI 源刊.)

[4] Yang Y, Ren T L, Zhu Y P, et al. PMUTs for handwriting recognition. In press. (已被 Integrated Ferroelectrics 录用. SCI 源刊.)

[5] Wu X M, Yang Y, Cai J, et al. Measurements of ferroelectric MEMS microphones. Integrated Ferroelectrics, 2005, 69:417-429. (SCI 收录, 检索号:896KM)

[6] 贾泽, 杨轶, 陈兢, 等. 用于压电和电容微麦克风的体硅腐蚀相关研究. 压电与声光, 2006, 28(1):117-119. (EI 收录, 检索号:06129773469)

[7] 伍晓明, 杨轶, 张宁欣, 等. 基于 MEMS 技术的集成铁电硅微麦克风. 中国集成电路, 2003, 53:59-61.

研究成果

[1] 任天令, 杨轶, 朱一平, 等. 硅基铁电微声学传感器畴极化区域控制和电极连接的方法: 中国, CN1602118A. (中国专利公开号)

[2] Ren T L, Yang Y, Zhu Y P, et al. Piezoelectric micro acoustic sensor based on ferroelectric materials: USA, No.11/215, 102. (美国发明专利申请号)