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
$\mathcal{C}$	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[7,28]。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[3,22–28]。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[22,24–26] 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[3,23,27] 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[3,23,27] 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[3,22–28] 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[14,15,27] 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[3,23,27]，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[22]	×	-	×	$O(n)$	×
CG12[25]	×	-	✓	$O(log(\|W\|))$	×
KO13[23]	✓	✓	×	$O(n)$	×
SPS14[27]	✓	✓	×	$min\{\alpha + log(N), rlog^3(N)\}$	×
CYGZR15[26]	×	-	×	$O(\|W\|) + O(r)$	×
BFP16[28]	✓	✓	✓	$O(r)$	✓
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[22,24–26] 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[22,23,25–27] 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[27], 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　研究现状

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，并支持数据块产生破坏时对其进行恢复。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。

可验证数据结构： 可验证数据结构 (Authenticated data structure) 被广泛应用于不可信的云存储环境中，用来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)[18], 可验证哈希表 (Authenticated Hash Table, AHT)[19] 以及可验证跳表 (Authenticated Skip List)[20,21]。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] 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 [①] 累加器 (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。

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动态搜索鉴别符来支持用户对搜索结果的验证。具体的做法是，将加密关键字作为“键”，将该关键字对应的文件集合作为“值”，将其存储在默克尔树的叶子结点上。用户需要保留默克尔树的根哈希作为一个指纹信息，在进行结果验证时需要通过搜索关键字及其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。一个关键的原因是，当用户搜索的关键字不存在时，默克尔树上也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构建默克尔树时，将整个字典空间中所有可能的关

[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

键字集合都存储，但这将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[22–24]。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[22] 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[23] 支持了用户数据更新，但是搜索复杂度超过了线性时间。近期，Ogata 等人也提出了一个通用的可验证对称加密搜索框架[24], 它也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[25] [26] 也同样只是静态方案。由 Stefanov 等人提出的方案[27] 采用了消息验证码 (Message Authenticated Code, MAC) 机制来实现了结果验证，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[28] 是目前为止最完善的可验证对称加密搜索方案，但他们的方案在搜索时需要和服务器有两轮通信才能进行验证，无法并行进行验证，并且他们的方案同样也不支持多用户情况下的验证。

Kurosawa et al. [8] leveraged Message Authenticated Code (MAC) to ensure data integrity and utilizes RSA accumulator to ensure data freshness during data updates. However, the existing study [12] showed that users in [8] required maintaining all keywords in the datasets to detect cheating of servers. It introduces significant overhead for users to maintain a large set of keywords. Therefore, paper [12] proposed a no-dictionary VSSE scheme, which means users do not need to keep the keywords set as a dictionary. However, their scheme only works in the static setting and does not consider data update scenarios. Stefanov et al. [9] also used MAC and timestamp to ensure data integrity and freshness. However, they were still unable verify an empty search result unless the data owner maintained all keywords locally. In summary, verification on the three-party model (across multiple users) should allow users to verify an empty result in addition to verification of integrity and freshness of search results. Meanwhile, it is desirable that users do not need to maintain a large set of keywords locally for the verification. We clarified this issue in Section 1.

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 [18] 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 [19]，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。The first verifiable attribute-based keyword search (VABKS) was proposed by Zheng et al.[29]. Similar to the existing

SSE schemes above, VABKS only focused on search based on static encrypted data. Liu et.al[30] proposed a more efficient construction based on VABKS, and Sun et.al[7] also provided a verifiable scheme VCKS that support conjunctive keyword search. However, due to the limitations of asymmetric encryption schemes, both of the above schemes require an additional trusted authority.

多用户加密搜索方案 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 [2]，该方案允许数据拥有者将数据分享给其他用户，并且数据拥有者具有对用户的访问控制权限，可以随时撤销或者新增用户。Jerecki 等人随后又提出了一个基于 Oblivious Cross Tag 的加密搜索方案 [21]，然而该方案需要数据拥有者和数据用户频繁的交互。A few of non-verifiable multi-user schemes have been proposed[13,31–33]. Curtmola et al.[13] first proposed a multi-user SSE scheme based on broadcast encryption. Yang et al.[31] proposed a multi-user searchable encryption scheme by leveraging a bilinear map. However, the search delay of the scheme is proportional to the size of the database, which is not suitable for large-scale databases. Jarecki et al.[32] designed a multi-user scheme by using Oblivious Cross-Tags (OXT) protocol. However, their scheme required frequent communication between data owners and the users, which incured unnecessary communication overheads. Recently, Sun et al.[33] proposed a non-interactive multi-user searchable encryption schemes that reduced the interactions between data owner and users. However, the scheme did not support search under data update.

【multi-user】Thanks for the comments. We emphasized that we did not aim to design a multi-user scheme for access control of search. Instead, our design aimed to provide results verification across multiple users. To the best of our knowledge, the existing VSSE schemes were all based on the two-party model and there did not exist a verifiable solution on the three-party model [11]. Specifically, verification on the three-party model faced more challenges than the two-party model. For example, when data was shared among users, a malicious server can easily mounted a data freshness attack because the users cannot detect a data update unless the data owner sent the update information to them. However, such an approach incurred a significant communication cost. In this paper, we proposed a VSSE scheme that is able to not only detect data integrity attack (especially when the server deliberately returns an empty result) but also capture data freshness attack by using a timestamp chain based mechanism. In our scheme, data owner only needed to send update information to the server, and data users can verify the search results by leveraging the information retrieved from the server without interacting with the data owner.

总结综上所述，现有的可验证加密搜索方案都不能满足多用户场景下的安全性保证，并且现有的方案无法完善地解决重放攻击和数据完整性攻击。这需要我们设计合理的机制来防御多用户场景下重放攻击，并且需要我们利用新型的数据结构来完善对数据完整性攻击的防御，尤其是防御服务器返回空结果来规避结果验证的情况。

2.2　先验知识

Incremental Hash. Incremental hash was proposed by Bellare et al.[?] and was used by existing SSE schemes, e.g., CS2[3]. An incremental hash function is a collision-resistant function $IH : \{0,1\}^* \rightarrow \{0,1\}^l$, with which the addition or the subtraction operation of two random strings on the IH does not produce a collision. For example, assuming F is a file collection that contains the keyword k. After a new file f is inserted to F, the file collection becomes F' (i.e., $F + f$), which means the new file f is a slight change according to F. Therefore, an incremental hash function can be used to quickly compute the corresponding collision-resist hash value after a file change. More detailed descriptions can be found in[3].

【IH】Thank the reviewer for pointing out this issue. We reviewed the incremental hash in Section 2. An incremental hash function is a collision-resistant function, which was first proposed by Bellare et al. [22]. We revised the description of the incremental hash in Section 2. In our scheme, if a file collection of a data owner is changed, the data owner uses the incremental hash function to quickly calculate a collision-resistance hash value of the changed data, which achieves high efficiency in computing hash values. Another advantage is that, when a file is added or deleted, a collision-resistance hash value ensures the security of our scheme during update.

Merkle Patricia Tree. The Merkle Patricia Tree (MPT) is first proposed in Ethereum [? ?], which combines the Trie Tree and the Merkle Tree for data update efficiency. There are three kinds of nodes in an MPT to achieve the goal. Leaf Nodes(LN) represents [key,value] pairs. Extension Nodes(EN) represent [key,value] pairs where keys are the public prefixes and their values are the hashes of the next nodes. The Branch Nodes (BN) are used to store possible branches when the prefixes of the keywords differ, which is presented with 17 elements. Among the 17 elements, the first 16 elements represent the 16 possible hex characters in a key and the last element stores a value if a key in a [key,value] pair matches the node. Fig. **??** shows insertion operations of a Merkle Patricia Tree (MPT) with the following four cases. First, to insert a [key,value] pair into a branch node, there are two possible cases. If the current key is empty, we can directly insert the value into the 17th bucket of the branch node. Otherwise, the unmatched key and value will be stored in a leaf node. Second, if we want to insert a [key,value] pair into a leaf node, there are also two possible cases. If the current key matches, we should modify the value of the leaf node directly. Otherwise, we should find the common prefix as the key of a newly created extension node. Meanwhile, we create a new branch node, and the original leaf node and the inserting [key,value] pair will be inserted as child node of the branch node. Note that, each node of the MPT is represented by its hash and is encoded using Recursive Length Prefix (RLP) code that is mainly used to encode arbitrarily binary data[?], which ensures the cryptographically security of the search operations. The root hash in MPT becomes a fingerprint of the entire tree and is computed based on all hashes of nodes below. Therefore, any modification in a node would incur recomputation of the root hash. Note that, the MPT is fully deterministic, meaning that an MPT with the same [key,value] pairs is exactly the same regardless of the order of insertion, which is different from the Merkle Tree.

Secure Searchable Encryption. Searchable Encryption was first proposed by Song et al.[12], their solution allows a user to outsource its encrypted data to cloud services, and meanwhile retaining the ability to search over it. Normally, searchable encryption has been divided into two categories, i.e., Searchable Symmetric Encryption(SSE) and Public Key Encryption with keyword search(PKE). The most classical SSE scheme was proposed by Curtmola et al. in[13]. They defined privacy against passive adversaries (i.e., honest but curious servers) and developed their scheme by using an inverted index. There exist various SSE schemes with different secure searching functionalities. For example, dynamic SSE schemes[14,15,27] allow a user to update his dataset and ranked keyword search scheme[?] that allow a user to retrieve ranked search results from the server. The most famous PKE scheme was proposed by Boneh et al.[17] with the bilinear map. Normally, the efficiency of the PKE schemes are much lower than the SSE schemes.

【SSE】Thanks for the suggestions. We explained the secure searchable encryption schemes in Section 2. Searchable encryption allows the server to perform search operations without seeing plaintext data. It empowers the server an ability to search over ciphertext and ensures the security of data on the server. In this revised manuscript, we discussed the existing categories of searchable encryption in Section 2.

2.3 问题定义

2.3.1 Threat Model

We assume that the data owner is trusted and the data users authorized by the data owner are also trusted[①]. We consider cloud services performing searchable symmetric encryption (SSE) to be untrusted, which means 1) cloud services intends to derive some sensitive information from the encrypted data and the queries; 2) cloud services may deviate from the prescribed protocols and mount a data freshness attack or a data integrity attack to save its computation or communication cost. The definitions of the data freshness attack and the data integrity attack are presented as follow:

定义 2.1 (**Data Freshness Attacks**)：*A data freshness attack in SSE is that a malicious server (or an attacker) attempts to return the historical version of the search result, not the most recently updated version. Formally, let* $\Delta_{n-1} = \{\delta_1, \delta_2, \cdots, \delta_{n-1}\}$ *denote the historical version of the dataset and* δ_n *is the latest version. However, the search result returned by the server is retrieved from* δ_i *where* $1 \leq i \leq n-1$.

定义 2.2 (**Data Integrity Attacks**)：*A data integrity attack in SSE is that a malicious server (or an attacker) attempts to tamper with the search result to prevent authenticated users from accessing the complete and correct search result. Formally, let* τ *be the search token of the SSE scheme, and* δ_i *be the dataset, where* $1 \leq i \leq n$*, the corresponding search result should be* $\mathcal{F}(\delta_i, \tau)$*, but the result returned by the server is* $\mathcal{G}(\delta_i, \tau)$*, where* $\mathcal{G}(\delta_i, \tau) \neq \mathcal{F}(\delta_i, \tau)$.

2.3.2 Design Goal

In this paper, we aim to design a generic verifiable SSE scheme that enables verifiable searches on the three-party model. In particular, the scheme should satisfy the following privacy and efficiency requirements:

1. **Confidentiality:** The confidentiality of data and keywords is the most important privacy requirements in SSE. It ensures that users' plaintext data and keywords cannot be revealed by any unauthorized parties, and an adversary cannot learn any

[①] Please refer to Section **??** for details on how we can enforce such assumption in practice with multi-user access control techniques.

useful information about files and keywords through the proof index and update tokens used in

2. **Verifiability:** A verifiable SSE scheme should be able to verify the freshness and integrity of the search results for users.
3. **Efficiency:** A verifiable SSE scheme should achieve sublinear computational complexity, e.g. logarithmic $O(log(|W|))$, where $|W|$ is the number of keywords, even with file update. Note that, the computational complexity only refers to the cost of searching operations for verification, which does not include the complexity of the searching operations in the existing SSE schemes.

This paper aims to provide result verification for any SSE schemes, including but not limited to[14,15,27]. Therefore, we treat an existing SSE scheme as a black box such that our proposed scheme can be applied to these SSE schemes for result verification.

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

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

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

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

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

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

[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.

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)