**前向加密搜索**

目录

[一、 Introduction介绍 3](#_Toc22840)

[二、 Related work相关工作 4](#_Toc20737)

[三、 Preliminaries 预备知识 5](#_Toc391)

[3.1 Trapdoor Permutations 陷门排列 5](#_Toc25729)

[3.2 Symmetric Searchable Encryption 对称可搜索加密 5](#_Toc26975)

[3.3 Security of SSE SSE的安全 5](#_Toc28433)

[四、 Problem formulations前向隐私 7](#_Toc2053)

[4.1 Definition 定义 7](#_Toc19008)

[4.2 The Need for Forward Privacy前向隐私的需要 7](#_Toc31423)

[4.3 Constraints Induced by Forward Privacy由前向隐私引起的约束 7](#_Toc26717)

[五、 Construction of VMKS scheme VMKS方案的构建 8](#_Toc13344)

[5.1 General ideas总体思路 8](#_Toc7473)

[5.2 Basic Construction基本建设 8](#_Toc7776)

[5.3 Security安全 8](#_Toc16731)

[5.4 Derived Constructions派生结构 8](#_Toc4551)

[5.5 Reducing Client-side Storage减少客户端存储 8](#_Toc11676)

[5.6 Security against Malicious Adversaries防范恶意对手的保安措施 8](#_Toc18238)

[5.7 Comparison with Other Constructions与其他结构的比较 8](#_Toc13734)

[六、 Implementation and Experimental Results实施与实验结果 9](#_Toc28789)

[6.1 Evaluation评价 9](#_Toc17951)

[6.2 Comparison with Existing Implementations与现有实现的比较 9](#_Toc10828)

[七、 Conclusions结论 10](#_Toc10230)

Abstract

可搜索对称加密的目的是使搜索一个加密的数据库存储在一个不受信任的服务器，同时保持查询和数据的隐私，允许一些小的可控泄漏到服务器。

最近的研究表明，动态方案——其中数据是有效可更新的——泄露一些关于更新关键字的信息，会受到破坏性的自适应攻击，破坏查询的隐私。阻止这种攻击的唯一方法是设计前向的私有方案，如果新插入的元素与以前的搜索查询匹配，则该方案的更新过程不会泄漏。

这项工作提出了Σoφo&向前私人SSE方案与性能类似于现有的不安全的方案,这是概念上比以前更简单(也更有效率)私人建设向前发展。特别地，它只依赖于暗门排列，而不使用类oram结构。我们还解释了为什么Σoφo&是一个最佳点的安全/ SSE的性能权衡。

最后，给出了具体的实施和评价结果。

# Introduction介绍

能够高效地、私下地查询存储在不受信任的服务器上的加密数据库对于安全的云应用程序至关重要。理想情况下，我们希望使用不向服务器泄漏任何信息的协议。这样的解决方案可以使用强大的技术来构建，例如多方计算、完全同态加密或遗忘RAM，但是，由于这些技术的通用性，它们是不切实际的:它们比一种简单的方法要慢，这种方法包括下载整个数据库并在本地进行搜索[Nav15]。

可搜索对称加密(SSE) [SWP00]是一种结构化加密[CK10]，用于搜索索引或搜索树等搜索结构，它通过交换泄漏以提高效率来解决这个问题。这种泄漏来自于使用(通常是对称的)确定性加密，这种加密将使服务器能够轻松地在加密的令牌之间找到匹配，而不需要运行昂贵的协议或计算。

在以前的工作中，泄漏的范围从全面揭示关键字出现在每个文件的模式(包括关键字顺序)，例如密云[Cip]或skyhigh[sky]等与遗产兼容的结构，到查询显示的出现模式，用于从[CGKO06]派生的基于反向索引的方案。

slam等[IKK12]和Cash等[CGPR15]研究了SSE方案泄漏的现实后果，并表明即使是很小的泄漏也可以被被动攻击者利用来揭示客户端的查询，从而导致泄漏滥用攻击。对于较大的泄漏，[CGPR15]的作者表明加密数据库的完全纯文本恢复是可能的。

Zhang等人对他们的工作进行了改进，在动态数据库中，攻击者可以在[ZKP16]中向数据库注入新文档。在本文中，作者展示了一种破坏性的适应性，通过插入最少10个新文档来揭示过去查询内容的攻击。这种攻击可以在几乎所有现有的SSE模式上运行，因为服务器可以知道新添加的文档与以前的搜索查询相匹配。

这项工作强调了对不泄露这一信息的SSE建设的需要，也被称为前向私人建设[SPS14]。据我们所知，除了TWORAM [GMP15]等基于oram的方案外，目前仅存的forward private scheme由Chang和Mitzenmacher [CM05]和by提出Stefanov等[SPS14]，但两者都存在效率低下的问题。前者具有从客户端到服务器的大带宽需求(用于搜索查询)和非常大的服务器存储空间，而后者使用了重要的oram相关技术，尽管支持高效的删除，但这会导致更新时产生大量带宽开销。

到目前为止，还没有一个SSE架构能够同时满足阻止这些漏洞滥用攻击的安全需求，并实现最优的更新效率。

Contributions：我们构造一个私人SSE方案与最优搜索和更新复杂性、计算和通信的复杂性,称为Σoφo&。

它既具有前向私有结构的安全性保证[SPS14]，又具有泄漏量较大的结构的渐近效率[CJJ+14]。我们的方案使用简单的密码工具(仅使用伪随机函数和活门排列，不依赖于ORAM思想)，非常容易理解。比较与先前的作品Σoφo&表1中给出。

我们给出了针对诚实但好奇的对手的安全性的充分证据，并展示了如何轻松地将其扩展到针对恶意对手的安全性，而无需在服务器端修改构造(Σoφo& -”)。

最后，我们提供了我们的scheme [Bos16]的一个开源实现，并表明，即使在持久存储上，它在搜索和更新方面也非常有效。

# Related work相关工作

SSE由Song等人[SWP00]引入，其解决方案的搜索时间与数据库的大小成线性关系。Chang和Mitzenmacher [CM05]后来提出了一种结构，其搜索时间也是线性的，但实现了前向隐私。

十年前，Curtmola等人[CGKO06]是第一个正式考虑泄漏的人，并在静态设置下设计了第一个基于索引的SSE结构，实现了次线性搜索的复杂性。第一个动态的次线性方案是由Kamara等人提出的[KPR12]，但是泄露了更新文档中包含的关键字散列。Kamara和Papamanthou [KP13]后来改进了这种结构，并以增加服务器空间复杂性为代价减少了泄漏。不过，这种建设并不是私人的

另一项工作提出了支持布尔查询的基于索引的方案[CJJ+13]，针对非常大的数据库和动态[CJJ+14]或三方设置[JJK+13]进行了优化，其中一些特性也在[PKV+14]中进行了研究。

Stefanov等人在[SPS14]中首次明确考虑了前向隐私。作者在他们的论文中建立了一个动态的、次线性的方案来实现前向隐私。

黑泽明和他的团队已经研究了可验证的SSE方案(即针对活跃对手的安全方案)Ohtaki [KO12]，以及Bost等人[BFP16]最近提出的高效和动态结构，包括一个来自[SPS14]的前向私有结构。

最近，Cash和Tessaro [CT14]证明了可搜索加密的存储位置的一个下界:不可能同时实现最优的服务器存储和最优的位置。Asharov等人[ANSS16]研究了它们的下界。

Chase和Kamara [CK10]将SSE扩展到任意结构的数据，如图、矩阵或标记数据。最近出现了结构化加密的新应用，例如加密图上的近似最短距离查询[MKNK15]。

SSE可以由不经意的RAM [GO96]构建，可以是黑箱方式，也可以使用特定的ORAM子组件，可以是灰箱方式，如[GMP15]。不幸的是，ORAM会导致大带宽、大客户端存储和/或多次往返。这使得ORAM对于SSE的使用变得不现实，正如Naveed [Nav15]的研究表明的那样。

Islam等人[IKK12]研究了可搜索加密中泄漏的后果。Cash等人[CGPR15]改进了他们的攻击，最近Zhang等人[ZKP16]也证明了SSE需要转发隐私。

# Preliminaries 预备知识

在论文中,λ是安全参数和negl(λ)代表一个微不足道的函数在安全参数。我们的构造使用伪随机函数(PRF)和(键控)散列函数，为此我们使用了标准的安全定义[Gol04]。

除非指定的影响,λ比特的对称密钥是字符串,和密钥生成算法均匀样本f0的关键;1 gλ。我们只考虑(可能概率)算法和协议运行在安全时间多项式参数λ。特别是，对手是概率多项式时间算法。

对于有限集X, X $ X表示X是从X中均匀采样的。

## 3.1 Trapdoor Permutations 陷阱门排列

一扇门排列(TDP)π是一个排列在一组D,使用公钥PK,π很容易评估,但逆π−1可以有效地计算只与SK的秘密。

## 3.2 Symmetric Searchable Encryption 对称可搜索加密

## 3.3 Security of SSE SSE的安全

SSE方案必须实现两个安全特性:正确性和机密性。

Correctness： SSE模式的正确性是一个基本属性:搜索协议必须为每个查询返回正确的结果，除非概率很小。

Confidentiality： SSE方案的机密性定义使用了现实世界与理想世界的形式化[CGKO06, KPR12, CJJ+14]。通过泄漏函数L = (LStp;LSrch;描述协议泄露给对手的内容，并形式化为有状态算法。这个定义确保了该方案除了从泄漏函数中推断出的信息外，不会泄露任何其他信息

更准确地说，我们定义了两个游戏SSEReal和SSEIdeal，在这两个游戏中，对手A选择一个数据库DB，并返回在真实情况下使用Setup(DB)生成的EDB，或在理想情况下使用S(LStp(DB))生成的EDB。然后，他重复执行搜索并使用输入q (resp)更新查询。接收运行搜索时生成的文本(q) (resp)。更新(op;在))协议在真实的游戏，或模拟器 S (LSrch (q))(分别地。S (LUpdt (op;在理想的游戏中。 最后，A输出一个位。我们说，该方案是l -自适应安全的，如果对每个对手A，存在一个有效的模拟器S这样的对手A不能以不可忽略的概率将真实博弈的执行与理想博弈的执行区分开来

Common Leakage 常见的泄露

许多SSE方案会泄漏客户机发送到服务器的令牌的重复，因此也会泄漏查询关键字的重复。与以前的工作类似[CGKO06]，当这种泄漏仅限于搜索查询(例如在静态模式中)时，我们称之为搜索模式。当更新的关键字的重复也泄漏时，我们称之为查询模式。

更正式地说，泄漏函数L将保存查询列表Q:到目前为止发出的所有查询的列表，其条目为(i;对于关键字w的搜索查询，或(i;人事处;对于输入为in的op update查询。整数i是一个时间戳，最初设置为0，并在每个查询中递增。

在本文中，我们还介绍了HistDB(w)符号:它是历史上添加到的文档列表DB匹配关键字w，按插入顺序排列。特别是，它包括已添加和稍后删除的文档，甚至包括已添加两次的文档。

最后，我们使用符号Hist(w)，关键字w的历史。它列出了所有的修改 DB(w)除以时间。它包含一个元组，它的第一个元素是DB0(w)，这是一组在设置时匹配w的文档索引，它的第二个元素是匹配w的文档更新的列表维护器(w)，称为更新历史记录。 例如，考虑两个与w匹配的文档Dind1和Dind2，假设我们首先在第二次更新时添加了Dind1，然后在第5次更新时添加了Dind2，最后在第14次更新时删除了Dind1。那么UpHist(w) = [(2;添加;ind1);(5);添加;ind2);(14;德尔;ind1)]。还可以使用查询列表正式定义它们。

# Problem formulations前向隐私

## 4.1 Definition 定义

前向私密性是动态SSE方案中SSE泄漏函数的一个强特性。非正式地说，它意味着更新不会泄漏关于更新的关键字的任何信息。特别是，服务器无法知道更新的文档是否与我们之前查询的关键字匹配。更正式地说，我们给出了以下关于转发隐私的定义。

这里给出的定义稍微扩展了Stefanov等人对转发隐私的非正式定义[SPS14]，该定义只关注添加的文档，而不是更新的(即添加和删除)文档。他们还描述了反向隐私，这是一种防止对已删除文档执行搜索的属性。

目前仅有的同时支持前向和后向隐私的方案是基于不经意RAM的，但效率不是特别高，而高效的前向私有方案已经被设计出来[CM05，，但遭受一个大带宽或存储爆炸。

## 4.2 The Need for Forward Privacy前向隐私的需要

Islam等人[IKK12]和Cash等人[CGPR15]研究了SSE方案泄漏的现实后果，并表明即使是很小的泄漏也可以被攻击者利用来揭示客户端的查询。他们的工作不仅考虑了静态数据库，还考虑了动态数据库，在这种情况下，服务器可以提交新文档并将其添加到数据库中，从而帮助他打破查询隐私。Zhang等人[ZKP16]改进了这种文件注入攻击，并证明它可以是毁灭性的。

它们实际上描述了非适应性攻击和适应性攻击。自适应攻击非常有效，但只适用于非前向私有的方案。这种攻击，揭示了以前搜索关键字w提交日志2T新文件，如果对手有部分知识的数据库，或使用W=T +log T新文件，如果他还没有。T是阈值参数，一个用于避免非自适应攻击的公共参数，需要保持较小的值以作为有效的反措施(但不要太小，以避免较大的效率损失)。在[ZKP16]中，作者为他们的实验选择了T = 200，在这种情况下，查询的隐私可以被发布少于10个新文档的对手破坏。

即使从功能的角度来看，前向隐私也是重要的。确实，前向私有方案允许在线构建加密的数据库。在大多数其他SSE结构中，必须首先执行索引步骤:设置阶段需要一个反向索引，其构造需要时间和空间。前向的私人计划避免了这种低效率的来源

## 4.3 Constraints Induced by Forward Privacy由前向隐私引起的约束

不幸的是，forward privacy在效率方面有很多缺点。本节解释动态SSE方案必须在存储效率、位置和安全性之间做出妥协。

On Storage

On Locality

# Construction of VMKS scheme VMKS方案的构建

在本节中，我们首先构造一个前向安全SSE方案，它的搜索和更新协议在一次往返中执行，但是以客户端上O(W(log D + log jMj)存储为代价。我们还首先考虑只支持添加而不支持删除的方案。然后，我们将描述如何将此基本构造转换为支持添加和删除的SSE构造，同时减少客户端存储，以及如何使其免受恶意对手的攻击。

## 5.1 General ideas总体思路

## 5.2 Basic Construction基本建设

## 5.3 Security安全

## 5.4 Derived Constructions派生结构

## 5.5 Reducing Client-side Storage减少客户端存储

## 5.6 Security against Malicious Adversaries防范恶意对手的保安措施

## 5.7 Comparison with Other Constructions与其他结构的比较

# Implementation and Experimental Results实施与实验结果

## 6.1 Evaluation评价

## 6.2 Comparison with Existing Implementations与现有实现的比较

# Conclusions结论