Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives 阅读笔记

李忆诺

2024 年 3 月 26 日

1 基本信息

1.1 论文来源

Bost R , Minaud B , Ohrimenko O .Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives[C]//Acm Sigsac Conference.ACM, 2017:1465-1482.DOI:10.1145/3133956.3133980.

1.2 概述

本文首次研究了可搜索加密的向后隐私概念。在给出不同类型的后向隐私的正式定义之后，本文提出了一个由距离约束伪随机函数构造前向私有方案的框架，并进行了各种效率权衡。

2 论文要点

2.1 背景

对于动态可搜索对称加密，其方案核心在于**效率**和**隐私**之间的权衡。自可搜索加密开始以来，在产生高吞吐量、低延迟和更具表现力的查询的高校解决方案上取得了巨大进展。

然而，随着人们对隐私问题的意识日益增强，产生了**前向隐私**和**后向隐私**这两个概念。前人一直致力于研究其前向隐私，希望更新不会透露任何关于它们执行的修改信息，并且服务器不可访问删除后的结果。而后向隐私却一直被忽视。

2.2 价值

在本文中，作者构建了一种技能实现前向隐私，又能实现弱形式的后向隐私的方案。其具体贡献如下：

- 提出了几种形式的后向隐私的正式定义，并描述了一种简单而通用的方法，在前向私有 SSE 方案上实现后向隐私；

- 定义了 $FS-RCPRF$ 框架，通过利用任意约束伪随机函数构建单关键字前向私有 SSE 方案，最终获得了前向安全 SSE 方案 $Diana$，其执行速度比最新方案快 10 倍。
- 描述了 $Janus$，该方案既是一种构建前向安全 SSE 方案的框架，又实现了一种弱形式的后向安全。

2.3 问题陈述

2.3.1 后向隐私定义

带插入模式的后向隐私：

泄露当前匹配 w 的文档、它们被插入的时间，以及 w 上总更新次数。

带更新模式的后向隐私：

泄露当前匹配 w 的文档、它们被插入的时间，以及 w 上所有更新发生的时间（但不包括它们的内容）。

弱后向隐私：

泄露当前匹配 w 的文档、它们被插入的时间、w 上所有更新发生的时间，以及哪个删除更新取消了哪个插入更新。

2.3.2 可穿刺加密

可穿刺加密可以支持对单个消息的撤销。核心思想在于使用标记 (tag)。消息用一组 tag 加密，穿刺算法使用某个 tag 更新密钥，使得新的密钥能够解密所有不包含该 tag 的密文。

2.3.3 实例化与目标

本文一开始提出了一种后向私有方案 B'，其核心思路为：服务器不存储索引 ind，而是上传一个密文 $E_{K_w}(ind, op)$，在搜索时，返回匹配的加密文档索引。交由客户机解密后，删除所有 $op = del$ 的索引，获得最终集合。将最终集合重新加密发给服务器，服务器以此进行相关删除操作。

使用 $\Sigma o\phi o\zeta$ 前向私有 SSE 方案作为基本模型，对文中提出的后向私有方案 B' 进行实例化，定义 $Fides$：一个前向和后向私有 SSE 方案基线。

$Fides$ 可被视作前向和后向私有设计基线：它易于构建，提供适度的计算开销，并表现出良好的安全性。

但区别于 $\Sigma o\phi o\zeta$，$Fides$ 在搜索过程中需要两轮，即**一轮搜索 + 一轮清理**过程。因此，本文的目标是提出一种避免额外往返和高通信开销的方案，其代价为只有**弱后向隐私**。

下文中假设所引入的可搜索加密方案均满足前向隐私。

2.4 方法

$Diana$：从基于树的 $GGMPRF$ 中构造一个简单有效的距离约束 PRF，用该范围实例化 $FS-RCPRF$，并将此实例化称之为 $Diana$。该方案的更新需要 $O(\log n_{max})$ 的计算复杂度，搜索过程中则有 $O(n_w)$ 个节点需要计算，这一方案更加有效。

Janus：客户机和服务器维护两个前向私有 SE 实例：Σ*add* 存储用可穿刺加密方案加密的新插入的索引（插入实例），Σ*del* 存储被穿刺的关键元素（删除实例）。每个关键字都有一个不同的加密密钥，客户端在本地存储每个密钥的 sk_0 部分。

在搜索 w 的过程中，客户端发送关联的密钥部分，分别从插入实例和删除实例中获得与 w 匹配的索引，最终能够获得并解密所有未删除的索引。

搜索后，客户端需要为 w 生成一个新密钥，并使用该新密钥加密 w 的新条目。

2.5 结果

隐私：

Janus 是一个满足前向私有和后向私有的 SSE 方案。前者直接来源于 Σ*add* 和 Σ*del* 方案，对于后者，服务器只有在对 w 进行搜索查询期间才能访问 w 条目的解密密钥。并且，该密钥允许她只解密自上次搜索 w 以来未删除的条目，因此，删除的索引仍然被隐藏，满足弱后向隐私。

效率：

Janus 的计算和通信复杂性可直接由 Σ*add* 和 Σ*del* 推导出来。根据文中计算所得，其搜索复杂度为 $T_{add}(a_w) + T_{del}(d_w) + O(n_w \cdot d_w)$；通信复杂度为 $C_{add}(a_w) + C_{del}(d_w)$。可以发现，当使用 $\Sigma o\phi o\zeta$ 和 *Diana* 进行实例化时，*Janus* 的搜索与通信复杂性是 $Optimal(constant)$。

3 评论

3.1 局限性

Janus 的存储开销相当高。客户端需要为每个关键字 w 存储 3 个组元素（每个至少 256 位），而服务器端每个密文由掩码索引、2 个组元素和标记组成，每个密钥共由 3 个组元素和 1 个标记组成。

本文中描述了一些减少开销的思路：从主密钥和秘密密钥上完成的穿孔次数中伪随机地生成加密方案参数和密钥元素。如此，客户端不需要存储公钥，而是直接从方案的参数加密明文索引。因此，客户端只需为每个 w 存储已删除的条目数量，减小了客户端开销。也可用类似的方案减少服务器端的开销。

3.2 扩展阅读

- Forward and Backward Private Conjunctive Searchable Symmetric Encryption
- ROSE: Robust Searchable Encryption with Forward and Backward Security

3.3 启示

通过阅读本文，我对**前向安全**和**后向安全**有了更加深刻的理解。前者是在插入前后泄露信息，后者是在删除前后泄露信息。通过将“删除”这一操作过程从服务器转移到客户端，由客户端分别获得插入实例和删除实例后，反向输入给服务器进行更新，从而达到后向安全的目的。

但阅读前向安全与后向安全相关工作，我发现其重心多在对于隐私泄露方面的追求。这种改进是否会影响到其搜索准确性？对关键词设置索引、并对索引层层加密，在搜索过程中其匹配准确度还能维持较高水平吗？

搜索性能与隐私安全处于一种相互制约的关系，如何找到其中更优的平衡点，也许是该领域工作的重心所在。