

支持容错检索的数据共享方案

易磊^{1,2}, 仲红², 袁先平^{1,2}, 赵玉^{1,2}

(1. 安徽大学 计算与信号处理教育部重点实验室, 合肥 230039; 2. 安徽大学 计算机科学与技术学院, 合肥 230039)

(zhongh@mail.ustc.edu.cn)

摘要:针对数据共享方案中的容错检索和细粒度访问控制问题,设计一种新的数据共享方案,采用了位置敏感的哈希和谓词加密方法,使得用户可进行关键字的容错检索,对密文做简单修改即可更新用户的访问权限,并且更新的计算复杂度优于现有的方案;通过理论分析,表明该解决方案是正确、安全和有效的。

关键词:数据共享;位置敏感的哈希;谓词加密;容错检索;访问控制

中图分类号: TP309.7 **文献标志码:** A

Error-tolerant searchable data sharing scheme

YI Lei^{1,2}, ZHONG Hong², YUAN Xian-ping^{1,2}, ZHAO Yu^{1,2}

(1. Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, Hefei Anhui 230039, China;

2. School of Computer Science and Technology, Anhui University, Hefei Anhui 230039, China)

Abstract: A new data sharing scheme was proposed to solve the problem of error-tolerant search and fine-grained access control. This new scheme adopted the technology of locality-sensitive hashing and the predicate encryption, which allowed users to search for keywords in an error-tolerant manner, and modified the user's access rights easily by updating the encrypted data. The computational complexity of updating is more optimized than the existing scheme. The theoretical analysis shows that the proposed solution is correct, safe and effective.

Key words: data sharing; locality-sensitive hashing; predicate encryption; error-tolerant search; access control

0 引言

目前,有许多在线存储服务,如 Adrive, Box 和 Flickr 等,为用户提供在互联网上共享数据提供存储空间,此类服务通常将用户数据加密后存储,以防止恶意攻击者获取存储的数据信息,但这种方法并不能保证数据的安全性和用户的查询隐私。

2008年,文献[1]作者通过构造关键字和访问码的集合多项式,提出一种支持关键字检索和访问控制的数据共享方案。该方案只支持精确的关键字检索,并且未考虑更新用户访问的问题。文献[2]将位置敏感的哈希^[3](Locality-Sensitive Hashing, LSH)算法与传统的加密方法相结合,提出一个支持关键字容错检索的加密方案,但该方案不能够实现细粒度访问控制。文献[4]对数据共享方案中用户访问权限更新问题进行了研究,引入在线代理服务器用于更新用户的访问权限,但是需要重新加密密文数据和全部授权用户的密钥,计算复杂度较高。

针对现有数据共享方案中的容错检索和细粒度访问控制策略问题,提出一种新的数据共享方案(如图1),使得更新用户访问权限计算复杂度为 $O(n)$ (其中 n 为常数)。

1 预备知识

本文使用如下符号: $[k] = \{1, 2, \dots, k\}$; $GL(N, F_q)$ 为在 q 阶有限域 F_q 上的一般线性群,即 $N \times N$ 的可逆矩阵的群; $\alpha, \beta \xleftarrow{U} F_q$ 表示 α 和 β 均匀随机地取之于 F_q ; $d(x, y)$ 为二进制向量 x 和 y 之间的汉明距离; (x, y) 为向量 x 和 y 的内积。

1.1 位置敏感的哈希

LSH 以很高的概率降低相似数据的差异性,而差异较大的数据仍保持差异性,可用于实现数据的容错检索。

LSH 函数族^[3]: 设 $\lambda_{\max} > \lambda_{\min} > 0, 1 > \xi_1 > \xi_2 > 0, J > j$, 对于 $x, y \in F_2^J$ 称 $H = \{h_i | h_i: F_2^J \rightarrow F_2^j, i \in [k]\}$ 是 $(\lambda_{\max}, \lambda_{\min}, \xi_1, \xi_2)$ 敏感的:

1) 如果 $d(x, y) < \lambda_{\min}$, 则 $\Pr_H[h(x) = h(y)] > \xi_1$;

2) 如果 $d(x, y) > \lambda_{\max}$, 则 $\Pr_H[h(x) = h(y)] < \xi_2$ 。

本文中省略了 LSH 的构造方法的详细介绍。设 G_{lsb} 为 LSH 的生成算法: 以 $(J, j, k, \lambda_{\max}, \lambda_{\min}, \xi_1, \xi_2)$ 为输入, 输出函数族 H 。

1.2 谓词加密

谓词加密^[5-6] 是一个新出现的公钥加密方案, 于 2008 年, 首先由 Katz 等人在文献[5] 中提出。基本思想是: 在谓词层次加密方案中, 密钥 SK_f 对应于某个谓词 f , 密文关联于一组属性 I ; 当且仅当 $f(I) = 1$ 时, 对应于谓词 f 的密钥 SK_f 才能够解密关联于属性 I 的密文。为了实现容错检索和访问控制, 需对文献[7] 的谓词加密方案进行改进, 将生成一对正交基生成算法 G_{ob} 修改为可生成两对正交基的算法 $G_{2\text{ob}}$, 算法 1 给出改进后的算法的基本原理。

算法 1 $G_{2\text{ob}}$ // 生成两对正交基

输入: $(1^\lambda, N)$

输出: $(\text{param}_V, B, B^*, D, D^*)$

begin

$(q, V, G_T, A, e) \xleftarrow{R} G_{\text{dprv}}(1^\lambda, N),$

$\text{param}_V := (q, V, G_T, A, e),$

收稿日期: 2010-11-26; 修回日期: 2011-01-23。 基金项目: 国家自然科学基金资助项目(60773114); 安徽高校省级重点自然科学基金项目(KJ2010A009); 安徽省自然科学基金资助项目(11040606M141); 安徽大学 211 项目。

作者简介: 易磊(1986-), 男, 河南信阳人, 硕士研究生, 主要研究方向: 信息安全; 仲红(1965-), 女, 安徽合肥人, 教授, 主要研究方向: 网络与信息安全、分布式计算; 袁先平(1985-), 女, 安徽合肥人, 硕士研究生, 主要研究方向: 信息安全; 赵玉(1984-), 女, 安徽合肥人, 硕士研究生, 主要研究方向: 信息安全。

$$\begin{aligned} X &:= (x_{i,j}), Y := (y_{i,j}) \xleftarrow{U} GL(N, F_q), \\ (\vartheta_{i,j}) &:= (X^T)^{-1}, (\omega_{i,j}) := (Y^T)^{-1}, \\ b_i &:= \sum_{j=1}^N x_{i,j} a_j, B := (b_1, \dots, b_N), \\ b_i^* &:= \sum_{j=1}^N \vartheta_{i,j} a_j, B^* := (b_1^*, \dots, b_N^*), \\ d_i &:= \sum_{j=1}^N y_{i,j} a_j, D := (d_1, \dots, d_N), \\ d_i^* &:= \sum_{j=1}^N \omega_{i,j} a_j, D^* := (d_1^*, \dots, d_N^*), \end{aligned}$$

end

其中 B 和 B^* 用于构造和验证访问码, D 和 D^* 用于消息的加密和解密。

2 新型的数据共享方案

2.1 数据共享模型

本文的数据共享方案模型如图 1 所示:数据拥有者 Alice 将密文数据上传给公共服务器 PS, PS 提供数据的存储和共享服务, 响应授权用户 U_i (i 无特殊含义, 目的是区分不同的用户) 的查询请求, 并返回查询结果。在本文方案中 PS 无法得知 Alice 存储了什么, 以及 U_i 查询了什么。



图 1 数据共享模型

2.2 新方案详细描述

本方案主要有系统初始化、授权、加密、查询、解密和更新等 6 个阶段, 主要由 Setup, KeyGen, Enc, Query, Update 等算法构成。方案中使用全局变量 S , 为已授权用户属性向量集合。

1) 阶段 1: 系统初始化。

Alice 选取系统参数 $1^\lambda, t, N = 2n + 3, J, k, \lambda_{\max}, \lambda_{\min}, \xi_1, \xi_2$, 其中 $n > t + 4$; 然后执行算法 2 获取公私钥对 (pk_A, sk_A) 。

算法 2 Setup // 系统初始化

输入: $(1^\lambda, n, t, J, k, \lambda_{\max}, \lambda_{\min}, \xi_1, \xi_2)$

输出: (pk_A, sk_A)

begin

$(param_V, B, B^*, D, D^*) \leftarrow G_{2ob}(1^\lambda, N)$,

$H_1 := \{h_1, \dots, h_k\} \leftarrow G_{ish}(param_{ish})$,

/* $param_{ish} = (J, j, k, \lambda_{\max}, \lambda_{\min}, \xi_1, \xi_2)$ */

/* $h_i: F_2^J \rightarrow F_2^k, \forall i \in [k]$ */

$H_2: G_T \rightarrow \{0, 1\}^\lambda$,

$T := (t_1, \dots, t_k) \xleftarrow{R} F_q^k$,

$\hat{D} := (d_1, \dots, d_n, d_{2n+1}, d_{2n+3})$,

$pk_A := (1^\lambda, k, param_V, \hat{D}, H_1, H_2)$

$sk_A := (t, T, B, B^*, D^*)$

end

2) 阶段 2: 授权。

Alice 选择 $S_i := (s_{i,1}, \dots, s_{i,t}) \xleftarrow{R} F_q^t / S \cup \{0\}$ 作为 U_i 的属性向量, 要求 $(S_i, S_j) = 0, \forall S_j \in S$ 。Alice 执行算法 3, 生成与 S_i 相关联的访问码 $k_{acc,i}$ 和解密密钥 $k_{dec,i}$ 。最后 Alice 更新 $S := S \cup \{S_i\}$, 并将授权的 k_i 发送给用户 U_i 。

算法 3 KeyGen // 生成 U_i 的授权密钥

输入: (sk_A, S_i)

输出: k_i

begin

$x', y' \xleftarrow{U} (n_1, n]$

$\varepsilon_0, \varepsilon_{n_1+1}, \dots, \varepsilon_{x'}, \varepsilon_{2n+2}, \varphi_0, \varphi_{n_1+1}, \dots, \varphi_{y'}, \varphi_{2n+2} \xleftarrow{U} F_q$

$k_{acc,i} := \varepsilon_0 \sum_{j=1}^t s_{i,j} b_j^* + \sum_{j=n_1+1}^{x'} \varepsilon_j b_j^* + b_{2n+1}^* + \varepsilon_{2n+2} b_{2n+2}^*$

$k_{dec,i} := \varphi_0 \sum_{j=1}^t s_{i,j} d_j^* + \sum_{j=n_1+1}^{y'} \varphi_j d_j^* + d_{2n+1}^* + \varphi_{2n+2} d_{2n+2}^*$

$k_i := (T, k_{acc,i}, k_{dec,i})$

end

3) 阶段 3: 加密。

Alice 执行算法 4, 用公钥 pk_A 加密消息 m , 并用私钥 sk_A 构造消息 m 的访问结构 c_1, c_2 和关键字 w 的查询陷门 T_w 。Alice 将 $(c_1, c_2, c_3, c_4, c_5)$ 发送给 PS。

算法 4 Enc

// 加密 m 和产生 w 的查询陷门

输入: (pk_A, sk_A, m, w)

输出: $(c_1, c_2, c_3, c_4, c_5)$

begin

/* $R = \cup S_j$ 为不拥有访问权限的属性向量集合 */

$Z := \sum_{S_j \in R} S_j = (z_1, \dots, z_t)$

$L_w := (h_1(w), \dots, h_k(w))$

$x, y \xleftarrow{U} (t, n_1], \beta_{t+1}, \dots, \beta_{x'}, \beta_{2n+3} \xleftarrow{U} F_q$,

$\eta_{2k+1}, \dots, \eta_y, \eta_{2n+1} \xleftarrow{U} F_q$,

$\delta_{t+1}, \dots, \delta_{x'}, \delta_{2n+3}, \zeta_1, \zeta_2 \xleftarrow{U} F_q$

$c_1 = \sum_{j=1}^t z_j b_j + \sum_{j=t+1}^{x'} \beta_j b_j + \beta_1 b_{2n+1} + \beta_{2n+3} b_{2n+3}$

$c_2 = T_w := T_w + T$ // w 的查询陷门

$c_3 := H_2(e(g, g)^{c_1})$

$c_4 := \sum_{j=1}^t z_j d_j + \sum_{j=t+1}^y \delta_j d_j + \zeta_2 d_{2n+1} + \delta_{2n+3} d_{2n+3}$,

$c_5 := e(g, g)^{c_2} m$,

end

算法 4 中 $n_1 := \lfloor \frac{n+t}{2} \rfloor$, 下文中的 n_1 若未做特殊说明, 皆如此定义。

4) 阶段 4: 查询。

① U_i 检索关键字 w' , 需先构造 w' 的查询陷门 $T_{w'}$, 然后将其和访问码 $k_{acc,i}$ 发送给 PS, 查询陷门构造方法下: $T_{w'} := L_{w'} + T = (h_1(w'), \dots, h_k(w')) + T$ 。

② 公共服务器接收到 U_i 的 $k_{acc,i}$ 和 $T_{w'}$ 后, 判断 $c_3 = H_2(e(c_1, k_{acc,i}))$ 和 $c_2 = T_{w'}$ 是否成立, 若两等式同时成立, 则将 c_4, c_5 发送给用户。

5) 阶段 5: 解密。

U_i 计算 $m' := c_5 / e(c_4, k_{dec,i})$, 若 $S_i \in R$ 则 $m' = m$, 否则 $m' = \perp$ 。

6) 阶段 6: 更新。

删除、恢复某些用户对消息 m 的访问权限。设 P 为待删除访问权限的用户属性向量集合, Q 为待恢复访问权限的用户属性向量集合, $P \cap Q = \emptyset$ 。

Alice 更新消息 m 访问结构, 执行算法 5, 生成与待更新访问权限用户相关的访问结构。Alice 将 c'_1, c'_4 发送给 PS, PS 接收到后做如下操作: $c_1 := c_1 + c'_1, c_4 := c_4 + c'_4$ 。

算法 5 Update

// 生成与待更新访问权限用户的访问结构

输入: (P, Q)

输出: (c'_1, c'_4)

begin

$Z := \sum_{S_i \in P} S_i - \sum_{S_j \in Q} S_j = (z_1, \dots, z_t)$,

$x, y \xleftarrow{U} (t, n_1]$

$$\begin{aligned} & \beta'_{t+1}, \dots, \beta'_x, \beta'_{2n+3}, \delta'_{t+1}, \dots, \delta'_y, \delta'_{2n+3} \xleftarrow{U} F_q \\ & c'_1 := \sum_{j=1}^t z_j b_j + \sum_{j=t+1}^x \beta'_j b_j + \beta'_{2n+3} b_{2n+3} \\ & c'_4 := \sum_{j=1}^t z_j d_j + \sum_{j=t+1}^y \delta'_j d_j + \delta'_{2n+3} d_{2n+3} \\ & \text{end} \end{aligned}$$

3 方案分析

文献[6]的谓词加密方案,其安全性基于已有的难解问题。本文的阶段1和阶段2参考了文献[6]的构造方法,不再做分析。下面主要对阶段3、阶段4和阶段5进行分析。

3.1 正确性

定理1 阶段4中对 U_i 的访问码和关键字的判断是正确的。

证明 不妨设 $R := \cup S_j$ 为禁止访问消息 m 的用户属性向量集合,PS接收到 U_i 的 $k_{acc,i}$ 和 $T_{w'}$ 时:阶段4①中若 U_i 的属性向量 $S_i \notin R$ 时 $c_3 = H_2(e(c_1, k_{acc,i}))$ 成立,反之不成立,因为:

$$\begin{aligned} c_1 &= \sum_{j=1}^t z_j b_j + \sum_{j=t+1}^x \beta_j b_j + \zeta_1 b_{2n+1} + \beta_{2n+3} b_{2n+3} = \\ & (Z, \beta_{t+1}, \dots, \beta_x, \overbrace{0, \dots, 0}^{2n-t-x}, \zeta_1, 0, \beta_{2n+3}) B \\ k_{acc,i} &= \varepsilon_0 \sum_{j=1}^t s_{i,j} b_j^* + \sum_{j=t+1}^{x'} \varepsilon_j b_j^* + b_{2n+1}^* + \varepsilon_{2n+2} b_{2n+2}^* = \\ & (\varepsilon_0 S_i, \overbrace{0, \dots, 0}^{x'-t}, \varepsilon_{n_1+1}, \dots, \varepsilon_{x'}, \overbrace{0, \dots, 0}^{2n-x'}, 1, \varepsilon_{2n+2}, 0) B^* \end{aligned}$$

其中 $t < x \leq n_1 \leq n_0$ 可知:若 $S_i \notin R$ 时 $(Z, S_i) = 0$,则有 $H_2(e(c_1, k_{acc,i})) = H_2(e(g, g)^{\varepsilon_0(Z, S_i) + \zeta_1}) = c_3$,否则为 \perp 。阶段4中②中若关键字 w 和 w' 的LSH向量相同,即 $L_w = L_{w'}$ 则有 $c_2 = T_{w'}$ 成立,反之不成立。

定理2 阶段5的解密算法是正确的。

证明 部分分析可参考定理1,证明简单描述如下:若

$$\begin{aligned} S_i \notin R \text{ 时 } (Z, S_i) = 0, \text{ 则: } m' &= \frac{c_5}{e(c_4, k_{dec,i})} = \frac{e(g, g)^{\varepsilon_2 m}}{e(g, g)^{\varepsilon_0(Z, S_i) + \zeta_2}} \\ &= m, \text{ 否则 } m' = \perp. \end{aligned}$$

3.2 安全性

定理3 阶段3中,PS无法得知Alice存储数据的任何信息。

证明 Alice存储在服务器上的数据形式为 $(c_1, c_2, c_3, c_4, c_5)$,其中:

$$c_4 = \sum_{j=1}^t Z_j d_j + \sum_{j=t+1}^y \delta_j d_j + \zeta_2 d_{2n+1} + \delta_{2n+3} d_{2n+3}$$

$$\text{设 } v_1 = \sum_{j=t+1}^y \delta_j d_j, v_2 = \sum_{j=t+1}^{n_1} \delta_j d_j, \text{ 在 } \hat{D} \text{ 公开的情况下, 由 DSP}$$

假设^[6]可知: v_1 与 v_2 是难以区分的,故PS无法得知 Z 和 ζ_2 。同理,在无法得知 B^* 情况下,PS更无法从 c_1 中无法得知 Z 和 ζ_1 。 $c_2 = L_w + T$,由于只有Alice和 U_i 拥有 T ,故PS从无法 c_2 中得知 w 的LSH向量 L_w ,继而也无法得知关键字 w 。由DDH假设可知,从 c_3, c_5 中无法得知 $\zeta_0, \zeta_1, \zeta_2$ 。

综上所述,PS无法得知Alice存储数据的任何信息。

定理4 除访问码 $k_{acc,i}$ 外,阶段4中不会泄露 U_i 的关键字 w' 的信息,并且访问码 $k_{acc,i}$ 的泄露不会威胁数据 m 的安全性。

证明 U_i 检索关键字 w' ,对应的查询陷门 $T_{w'} := L_{w'} + T$, U_i 将 $k_{acc,i}$ 和 $T_{w'}$ 发送给PS,从定理3中所述可知,PS无法得知关键字 w' 。由于只有Alice知晓 B 和 B^* ,故PS无法从 $k_{acc,i}$ 中得知 U_i 的属性向量 S_i ,又因为无法得知 D^* ,故PS无法构造对应的解密密钥 $k_{acc,i}$,所以用户访问码的公开,并不会威胁到数据的安全性。

3.3 计算复杂度

一次查询过程分为预处理阶段和在线查询阶段;预处理阶

段主要是 U_i 构造查询关键字 w' 的查询陷门 $T_{w'}$,该计算的计算复杂度主要为加法运算的复杂度,此计算的复杂度为 $O(k)$ (k 为LSH的哈希函数个数),文献[7]经过实验数据分析, $k = 15$ 时就可满足容错检索的要求;在线查询阶段,主要是判断 $c_3 = H_2(e(c_1, k_{acc,i}) \bmod q)$ 是否成立,可知此运算复杂度主要为模幂运算的复杂度,此计算的复杂度为 $O(n \log^3 q)$ 。

另外,对于更新用户的访问权限问题,分析阶段6可知,计算复杂度主要为 $8n + 12$ 次加法运算和不多于 $n + t$ 次的乘法运算,可知本文更新用户访问权限的计算复杂度为 $O(n)$,其中 $n = \frac{N-3}{2}$, N 为正交基的维数,且 $N \leq q$;而对于文献

[4],此种情况下,需要重加密密文数据和更新全部授权用户的密钥,其计算复杂度为 $O(\theta \log^3 q)$ (其中 θ 为用户的数量),相对于本文的计算复杂度 $O(n)$,可知该方案优于文献[4]。

3.4 通信复杂度

阶段4查询过程中, U_i 将 $T_{w'}$ 和 $k_{acc,i}$ 发送给服务器, $T_{w'} := L_{w'} + T$,其中 $L_{w'} = (h_1(w'), \dots, h_k(w'))$, $T \in F_q^k$, $T_{w'}$ 的存储信息为 $k \log q$ bit,由于 $b_i^* \in F_q^{2n+3}$, $k_{acc,i} \in \text{span}\{b_1^*, \dots, b_{n_1}^*, b_{2n+1}^*, b_{2n+2}^*\}$,故 $k_{acc,i}$ 的存储信息为 $(2n+3) \log q$ bit;查询成功后返回 c_4 和 c_5 ,因为 $c_4 \in \text{span}\{d_1, \dots, d_{n_1}, d_{2n+1}, d_{2n+3}\}$, $c_5 = e(g, g)^{\varepsilon_2 m} \in F_q$,可知 c_4 和 c_5 的存储信息为 $(2n+4) \log q$ bit。综上所述,一次查询的通信复杂度:

$$k \log q + (2n+3) \log q + (2n+4) \log q = O(n \log q)$$

4 结论

基于LSH和谓词加密的数据共享方案,很好地解决了关键字容错检索,以及在无需更改用户密钥的前提下,更改用户访问权限等问题;另外该方案具有比较好的安全性:数据拥有者的数据隐私和查询用户的查询隐私不会泄露。该方案也存在一些缺陷:例如密文结构复杂,空间复杂度高。下一步将研究如何降低空间复杂度,使方案具有更好的实用性。

参考文献:

- [1] YAU S S, YIN YIN. Controlled privacy preserving keyword search [C] // Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security. New York: ACM, 2008: 321–324.
- [2] BRINGER J, CHABANNE H, KINDARJI B. Error-tolerant searchable encryption [C] // Proceedings of IEEE International Conference on Communications. Dresden: IEEE, 2009: 1–6.
- [3] INDYK P, MOTWANI R. Approximate nearest neighbors: Towards removing the curse of dimensionality [C] // Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. New York: ACM, 1998: 604–613.
- [4] YU SHUCHENG, WANG CONG, REN KUI, et al. Attribute based data sharing with attribute revocation [C] // Proceedings of the Fifth ACM Symposium on Information, Computer and Communications Security. New York: ACM, 2010: 261–270.
- [5] KATZ J, SAHAI A, WATERS B. Predicate encryption supporting disjunctions, polynomial equations, and inner products [C] // Proceedings of 27th Annual International Conference on Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2008: 146–162.
- [6] LEWKO A, OKAMOTO T, SAHAI A, et al. Fully secure functional encryption attribute-based encryption and (hierarchical) inner product encryption [C] // Proceedings of 29th Annual International Conference on Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2010: 62–91.
- [7] 蔡衡, 李舟军, 孙健, 等. 基于LSH的中文文本快速检索[J]. 计算机科学, 2009, 36(8): 201–204.