一种适于云存储的数据确定性删除方法

王丽娜[1,2,3],任正伟[3],余荣威[2,3],韩 凤[3],董永峰[3]
(1.武汉大学软件工程国家重点实验室,湖北武汉 430072;
2.武汉大学空天信息安全与可信计算教育部重点实验室,湖北武汉 430072; 3.武汉大学计算机学院,湖北武汉 430072)

摘 要: 为保护云存储模式下数据的机密性,本文提出了一种适于云存储系统的数据确定性删除方法.该方法通过密钥派生树组织管理密钥,将密钥经秘密共享方案处理后分发到 DHT 网络中,利用 DHT 网络的动态特性实现密钥的定期删除,使得在非授权时间内密文数据不能被解密和访问,从而实现云存储系统中数据的确定性删除.实验结果表明,该方法能够有效地删除密钥,且性能开销低,满足云存储系统中过期数据或备份文件的确定性删除要求.

关键词: 云存储; 数据机密性; 数据删除; 密钥管理
中图分类号: TP302.7 **文献标识码:** A **文章编号:** 0372-2112 (2012) 02-0266-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.02.010

A Data Assured Deletion Approach Adapted for Cloud Storage

WANG Li-na[1,2,3], REN Zheng-wei[3], YU Rong-wei[2,3], HAN Feng[3], DONG Yong-feng[3]
(1. *State Key Laboratory of Software Engineering, Wuhan University. Wuhan, Hubei* 430072, *China*;
2. *Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University. Wuhan, Hubei* 430072, *China*; 3. *School of Computer, Wuhan University. Wuhan, Hubei* 430072, *China*)

Abstract: A data assured deletion approach adapted for cloud storage is proposed to protect data confidentiality of cloud storage system. We utilize a key derivation tree to organize and manage keys which are pushed to DHT network after partitioned by secret sharing scheme. The dynamic property of DHT network makes keys be deleted periodically causing ciphertext can not be decrypted or accessed when authorized time expires. And data assured deletion is implemented. The experiment results show that this method can delete keys effectively and the performance is low. It suffices for the requirements of assured deletion of expired data or expired data in the cloud storage system.

Key words: cloud storage; data confidentiality; data deletion; key management

1 引言

近年来,基于互联网的云存储服务模式由于成本低、部署灵活、可扩展性强而成为国内外工业界和学术界共同关注的研究领域[1~4].云存储系统由位于互联网上的大量存储资源以及将这些存储资源组织为可供用户透明访问的资源池的一整套资源管理与访问控制技术所组成[1],在为用户提供无限存储空间的同时,还提供了易于使用的数据接口和极强的可扩展性,使得存储于其上的数据能以服务的形式按需为用户所使用.云存储能在提高 IT(Information Technology)效率、成本节省以及绿色计算等方面带来了机遇,同时也面临着一些安全挑战.

在云存储模式下,数据托管存储于第三方云存储平台中,脱离了数据属主(Data Owner)的控制范围,其安全性高度依赖于云服务提供商(Cloud Service Provider, CSP).考虑到 CSP 不可信,Owner 为了保护数据的机密性,会先对数据加密,再将密文数据托管给 CSP,加密密钥则由自己保管.但是,即使数据以密文形式保存于 CSP 处也可能会存在一些安全隐患.例如,CSP 为了提高服务的可靠性,可能会对数据做多个备份[2],并且将这些备份数据分布在不同的服务器上.在此条件下,当数据过期了,Owner 要求 CSP 删除其数据时,CSP 可能并没有完全删除所有的数据及备份.一旦攻击者获取到了密钥,并从不可信的 CSP 处获取到了未被删除的密文数据或备份,数据的机密性将被破坏.因此,过期或备份数据的确定性删除,即数据是被彻底删除了或者是永远不可解密和访问的,是云存储安全研究的重要内容[3].

收稿日期:2011-04-22;修回日期:2011-09-01
基金项目:国家自然科学基金(No.60970114,No.61103219);湖北省自然科学基金(No.ZRZ0041);武汉市科技攻关计划项目(No.201110921292);信息网络安全公安部重点实验室开放基金(No.C10607);中央高校基本科研业务费专项(No.211275570)

一些研究者将超出了属主控制范围的数据删除问题等价为密钥管理问题,并开展了一些研究工作.如文献[5]提出了一种基于策略的文件确定性删除方法,以实现云存储系统中数据及其备份的确定性删除.其基本思想是由数据密钥对文件加密,再由与策略相关联的控制密钥加密数据密钥,删除控制密钥即可实现数据的确定性删除.然而,在该方案中控制密钥是由第三方的密钥管理者管理,是一种集中式的管理方式,存在密钥管理者不可信而未删除或泄漏控制密钥的安全隐患.相对于集中式的管理方法,分布式密钥管理方法的安全性更高,如 Vanish 系统[6]将密钥经门限密码处理后随机分发到采用了分布式哈希表(Distributed Hash Tables,DHT)[2,7,8]技术的 P2P 网络(DHT 网络)中,使得当授权时间到达后,密钥将被网络删除,导致加密数据不能被解密,实现了邮件服务器和网络中邮件副本的自销毁.由于 Vanish 系统直接将加密密钥分发到 DHT 网络中,攻击者就可以通过嗅探攻击[6]或跳跃攻击[9]获取到足够的密钥分片重构出密钥[9].文献[10]认为只销毁密钥不销毁数据存在着攻击者暴力破解密码算法的安全隐患,在 Vanish 系统的基础上,将密钥和部分密文数据一起分发到 DHT 网络中,使得攻击者暴力破解不完整密文数据所需的密钥空间增大,从而增加攻击的难度和代价.该方案将部分密文也分发到网络中,增加了网络通信开销.文献[11]通过对 Shamir 秘密共享算法[12]进行改进,扩展密钥份数的长度以抵抗 Vanish 系统中存在的跳跃攻击;对于嗅探攻击的抵抗则是通过公私钥加解密的方式实现.文献[6,10,11]都只考虑了单个密钥和少量数据的确定性删除,不能直接应用于云存储系统中.因为云存储系统中数据是海量的,用单个密钥加密全部数据不能对数据进行细粒度的管理和操作[2],不能按需提供服务.

本文基于密钥派生树和 DHT 网络,提出了一种适于云存储系统的数据确定性删除方法.该方法首先根据云存储系统中数据是海量的特点,借鉴结构化层次密钥管理[13]的思想,采用基于哈希函数的密钥派生树生成和管理密钥,从而有效减少了 Owner 所需维护的密钥数量及暴露给外部的密钥数量,并且通过数据块级的加密方式对数据提供细粒度的管理和操作.通过密钥派生树生成密钥后,将密钥分发到 DHT 网络中,利用 DHT 网络的动态变化特性确保密钥在授权时间到达后自动从网络中消失,使得密文数据不能被解密和访问,从而实现云存储系统中数据的确定性删除.为了减小网络的动态变化对系统可用性的影响,本文采用秘密共享方案对密钥处理后,将多个密钥分片分发到网络中以提高系统的可用性.

2　系统模型与安全假设

2.1　系统模型

本文在密钥派生树、DHT 网络和秘密共享方案的基础上,建立了如图 1 所示的系统模型.Owner 将数据加密后外包给 CSP,将授权和密钥等信息发送给 User.User 通过授权信息从 CSP 处获取密文数据,使用密钥解密和访问数据.

2.1.1　密钥派生树

为保护数据的机密性,并且支持数据的细粒度管理和操作,Owner 先在本地对数据进行分块后加密,再将密文数据块外包给 CSP.假设数据 M 被分成 n 个数据块$\{M_1, M_2, \cdots, M_n\}$,每个 M_i 用一个随机密钥加密,则 Owner 所需存储和维护的密钥量将随着 n 线性增长.若 User 需要访问 l 个数据块,Owner 和 User 之间传递的密钥量也与 l 成线性关系,增加了系统维护密钥的开销.针对该问题,本文借鉴文献[13]的方法,采用层次密钥及密钥派生树的方法来生成和管理密钥,如图 2 所示.

该方法通过二叉树生成并组织密钥,将密钥表示为 $k_{i,j}$,其中 i 表示 $k_{i,j}$在树中的层次,j 表示 $k_{i,j}$在第 i 层中的序号.除了根密钥 $k_{0,1}$,其他密钥都是由其父节点通过左(右)孩子派生规则 $f_L(f_R)$生成.

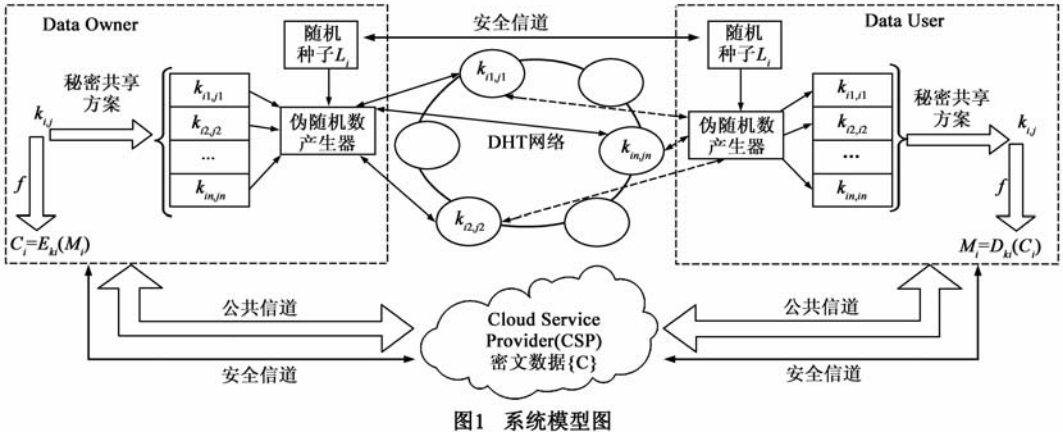

图1　系统模型图

考虑到直接将叶子节点作为加密密钥分发到网络中易受到嗅探攻击和跳跃攻击[6,9],降低系统的安全性.本文将叶子节点值进行某种变换后得到加密密钥 k_i,变换过程为 $k_i = f(k_{p,i})$ $(1 \leqslant i \leqslant n)$,其中,$p$ $(p \geqslant 0)$为树的高

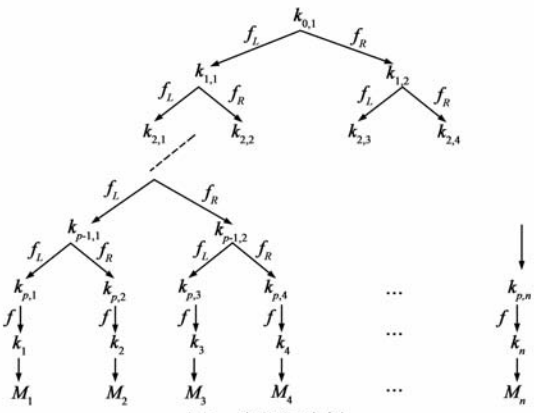

图2 密钥派生树

度,f 为变换函数.为简便起见,下文中将图 2 所示的密钥派生树中的所有密钥 $k_{i,j}$ 统称为树密钥,以与加密密钥 k_i 相区别.

通过该方法,Owner 只需维护根密钥 $k_{0,1}$,派生规则 f_L 和 f_R,变换函数 f,常量参数 n 和 p,就可以生成所有密钥,大大减少了 Owner 所需维护和存储的密钥数量.并且由于每个数据块都采用不同的密钥加密,因此只需分发被请求访问的数据块的密钥,从而减少了密钥泄露的可能性.如当 $n=8$ 时,对于数据块 M_5、M_6、M_7,分发的密钥集合为 $\{k_{2,3}, k_{3,7}\}$,而不是 $\{k_{1,2}\}$,从而避免 $k_{3,8}$ 暴露给外部.

2.1.2 DHT 网络

DHT 网络是指利用 DHT 表[7,8]存储数据和实现节点路由的 P2P 网络.DHT 网络的以下三个重要特性使得其可以应用在数据确定性删除方法中.

(1)可用性.DHT 网络可以提供可靠的分布式存储功能[7,8],很多 DHT 网络如 Vuze[14]等都已应用到实际系统中.该特性使得被保护的数据在授权时间(称之为 timeout)内是可用的,是 DHT 网络能够应用到本文研究中的基础.

(2)大规模且地理分布广.研究表明[14]在 Vuze 网络中同时并发存在的活动节点超过了一百万个,并且地理分布超过了 190 个国家.这种完全非集中的分布方式能够提供很健壮的抗攻击能力.

(3)动态性.由于不断有节点加入和离开,DHT 网络会随着时间动态地变化,存储在 DHT 网络中的信息也会定期被清除,使得被保护的数据在 timeout 达到后不可用.如在 Vuze 网络中,一旦某个节点的 IP 地址或者端口号发生改变,它在 DHT 网络中的位置就会改变,如研究[15]表明超过 80% 的 IP 地址在 7 天内会发生改变.DHT 网络的这种动态和定期清除数据的特性,为数据的确定性删除提供了实现思路.

2.1.3 秘密共享方案

秘密共享方案(Secret Sharing Scheme, SSS)[12]的基本思想是,将秘密 k 以某种方式分成 n 份(shares)$k_1, k_2, \cdots, k_n$,并且满足:

(1)通过任意 t 或 t 个以上 k_i 能够计算出 k;

(2)通过任意 $t-1$ 个或更少的 k_i 不能计算出 k.

这种方法也称为 (t, n)-门限方法,称 t 为门限阈值,n 为门限值,t/n 为门限率.

通过 SSS 将任意秘密 k 拆分成 n 份,且重构 k 需要至少知道任意 t 或 t 个以上 k_i,故泄露 $s<t$ 个 shares 也不会暴露秘密 k,而丢失 $s \leqslant n-t$ 个 shares 仍然可以重构出 k.SSS 是解决 DHT 网络因为动态性使得密钥丢失而不可用的一种有效方法.

2.1.4 模型描述

图 1 所示的模型由 7 个算法描述:(*DataKeyGen*, *Encryption*, *MixTrKeySet*, *TrKeyDis*, *TrKeyExtract*, *DataKeyRecover*, *Decryption*).

(1) $DataKeyGen(\kappa, n, p, k_{0,1}, f_L, f_R, f) \rightarrow \{k_i\}$,加密密钥生成算法.$\kappa$ 为系统安全参数,n 为数据块数,p 为树的高度,$k_{0,1}$ 为树根密钥,f_L 和 f_R 分别为左右孩子派生规则,f 为变换函数.算法根据上述参数,生成 n 个加密密钥 $\{k_i\}$.

(2) $Encryption(\{M_i\}, \{k_i\}) \rightarrow \{C_i\}$,加密算法.通过密钥 $\{k_i\}$ 加密明文数据 $\{M_i\}$,得到密文数据 $\{C_i\}$,即 $C_i = E_{k_i}(M_i)(1 \leqslant i \leqslant n)$.

(3) $MixTrKeySet(\{index\}, c) \rightarrow \{K\}_{mix}$,最小树密钥集生成算法.根据数据块索引集合 $\{index\}$ 和数据块数 c 计算访问这些数据块所需的最小树密钥集合 $\{K\}_{mix}$.

(4) $TrKeyDis(R, \{L\}, \varphi, t, m, \{K\}_{mix}) \rightarrow \{Node\}$,树密钥分发算法.$R$ 为伪随机数产生器,$\{L\}$ 为随机种子集合,φ 为映射函数,t 为门限阈值,m 为门限值,$\{Node\}$ 为 DHT 网络的节点集合.算法根据上述参数,将最小树密钥集中的树密钥经秘密共享算法处理后分发到 DHT 网络的节点上.

(5) $TrKeyExtract(R, \{L\}, t, m) \rightarrow \{K\}_{mix}$,树密钥提取算法.根据输入参数从 DHT 网络中提取并重构出最小树密钥集 $\{K\}_{mix}$.

(6) $DataKeyRecover(f_L, f_R, f, \{K\}_{mix}) \rightarrow \{k_{i_s}\}$,加密密钥恢复算法.根据输入参数将最小树密钥集 $\{K\}_{mix}$ 中的树密钥转换为加密密钥 $\{k_{i_s}\}$.

(7) $Decryption(\{C_{i_s}\}, \{k_{i_s}\}) \rightarrow \{M_{i_s}\}$,解密算法.根据密钥 $\{k_{i_s}\}$ 解密密文数据 $\{C_{i_s}\}$,得到明文数据 $\{M_{i_s}\}$,即 $M_{i_s} = D_{k_{i_s}}(C_{i_s})$.

2.2 安全假设

对于上述模型,本文有如下安全假设:

(1)CSP 是不可信的,在向 Owner 和 User 提供服务的同时,可能利用其所知道的信息查看存储于其中的

数据的内容,并且泄露给其他非授权实体.

(2)User 是可信的[6],即 User 为授权用户,会遵守和 Owner 的协定,不会主动泄露明文数据(使用完明文数据后就删除),也不会主动泄露其获得的密钥和密文数据块.但是 User 可能因受法律干预或误操作等而导致密钥泄露.事实上,对于 User 的诸如截屏、拍照等恶意泄露行为从技术上是很难防范的,因此本文认为该假设是合理的.

(3)攻击者的攻击行为不是实时的,而是一种事后行为[6],因为攻击者不能实时知道哪些数据对其是有攻击价值的,只有在数据被使用后攻击者才会确定是否要发起攻击获取该数据.

(4)Owner 与 CSP,Owner 与 User,User 与 CSP 之间都有安全信道,其安全信道可以通过相互之间的密钥协商获得的共享密钥建立,也可以基于相互之间的公私钥加解密的方式建立.

3　基于 DHT 网络的数据确定性删除方法

本节首先给出系统模型中算法的详细描述,然后介绍通过这些算法和 DHT 网络实现数据访问的流程,最后阐述通过 DHT 网络实现数据确定性删除的方法.

3.1　算法描述

(1) $DataKeyGen(\kappa, n, p, k_{0,1}, f_L, f_R, f) \rightarrow \{k_i\}$

首先根据 κ 随机选取一个 $k_{0,1}$作为树根密钥. n 个数据块对应着 n 个叶子节点,根据二叉树的性质,树的最小高度 p 应满足 $2^{p-1} < n \leqslant 2^p$. 当 $k_{i,j}$不是叶子节点时,其左、右孩子分别为 $k_{(i+1),(2*j-1)}$和 $k_{(i+1),(2*j)}$,并且 $k_{(i+1),(2*j-1)} = f_L(k_{i,j})$, $k_{(i+1),(2*j)} = f_R(k_{i,j})$. 其中, $f_L(k_{i,j}) = h(k_{i,j} \| (2*j-1))$, $f_R(k_{i,j}) = h(k_{i,j} \| (2*j))$, $h()$为哈希函数 SHA-1, $\|$ 表示串联. 不断重复该过程,就可由 $k_{0,1}$计算出所有叶子节点 $k_{p,i}$. 最后,根据变换 $k_i = f(k_{p,i}) = [h([k_{p,i}]_{pre80}) \| h([k_{p,i}]_{pos80})]_{pre256}$将所有的 $k_{p,i}$变换为加密密钥 k_i,其中, $[X]_{preM}$表示截取 X 的前 M 位, $[X]_{posN}$表示截取 X 的后 N 位,即 k_i 是通过将 $k_{p,i}$的前 80 位 SHA-1 值和后 80 位的 SHA-1 值串联后截取前 256 位得到.

(2) $Encryption(\{M_i\}, \{k_i\}) \rightarrow \{C_i\}$

通过对称加密算法 AES,对每一个明文数据块 M_i 用 256 位的密钥 k_i 加密,得到密文数据块 C_i.

(3) $MixTrKeySet(\{index\}, c) \rightarrow \{K\}_{mix}$

对于两个树密钥 $k_{a,b}$和 $k_{c,d}$,若其父节点相同,即其下标满足条件: $a = c$, $d = b + 1$, $b \equiv 1 (\mathrm{mod}2)$, $d \equiv 0 (\mathrm{mod}2)$,或者 $a = c$, $b = d + 1$, $d \equiv 1 (\mathrm{mod}2)$, $b \equiv 0 (\mathrm{mod}2)$,则用其父节点代替这两个树密钥,称该过程为树密钥的合并.将由 $\{index\}$和 c 指定的数据块所对应的树密钥逐一合并,生成最小树密钥集合 $\{K\}_{mix}$.

(4) $TrKeyDis(R, \{L\}, \varphi, t, m, \{K\}_{mix}) \rightarrow \{Node\}$

以 random()函数作为 R,取系统当前时间为随机种子 L_i,共取 $|\{K\}_{mix}|$次系统当前时间得到 $\{L\}$(设 $|\{X\}|$表示集合 $\{X\}$中元素的个数).通过映射 $\varphi: L_i \rightarrow k_{i,j}$按顺序将 L_i 与 $\{K\}_{mix}$中的 $k_{i,j}$一一对应.再通过 Shamir 秘密共享方案将 $\{K\}_{mix}$中的每个 $k_{i,j}$及其下标 i 和 j 分为 m 个分片,以 L_i 作为 R 的输入,产生 m 个随机值,通过这 m 个随机值将每个 $k_{i,j}$及 i 和 j 的 m 个分片分发到 DHT 网络的 m 个节点上.

(5) $TrKeyExtract(R, \{L\}, t, m) \rightarrow \{K\}_{mix}$

以 L_i 作为 R 的输入,产生 m 个随机值,根据这 m 个随机值从 DHT 网络中提取 $k_{i,j}$及 i 和 j 的分片,再根据 Shamir 秘密共享方案恢复出 $k_{i,j}$,得到 $\{K\}_{mix}$.

(6) $DataKeyRecover(f_L, f_R, f, \{K\}_{mix}) \rightarrow \{k_{i_s}\}$

首先通过 f_L 和 f_R 将 $\{K\}_{mix}$中的所有非叶子节点转换为叶子节点,再通过 f 将所有的叶子节点转换为加密密钥 $\{k_{i_s}\}(1 \leqslant s \leqslant c)$.

(7) $Decryption(\{C_{i_s}\}, \{k_{i_s}\}) \rightarrow \{M_{i_s}\}$

用对称加密算法 AES,对每一个密文数据块 C_{i_s},通过 256 位的密钥 k_{i_s}解密,得到明文数据块 M_{i_s}.

3.2　数据访问流程

在图 1 所示的模型中,数据的访问过程分为 Initial, KeyDistribution 和 DataAccess 3 个阶段.

在 Initial 阶段,Owner 通过 *DataKeyGen* 生成加密密钥,用 *Encryption* 加密数据块,并将密文数据 $\{C_i\}$及 User 的访问控制信息一起存储到 CSP 处.之后,Owner 保留 $(\kappa, k_{0,1}, f_L, f_R, f, n, p)$,删除明文数据 $\{M_i\}$和密钥派生树以节省存储空间.

在 KeyDistribution 阶段,Owner 根据 User 的访问请求,执行 *MixTrKeySet* 计算访问数据所需的最小树密钥集合,通过 *TrKeyDis* 将树密钥分发到 DHT 网络中,并将 R、$\{L\}$、φ、(t, m)及其他附加信息(如授权证书及授权时间 timeout 等)发送给 User,将 User 的最新访问控制信息发送给 CSP.

在 DataAccess 阶段,User 将从 Owner 处获得的证书和期望访问的数据块的索引信息发送给 CSP.经 CSP 验证后,从 CSP 处获得密文数据块.然后 User 执行 *TrKeyExtract* 从 DHT 网络中提取并重构出最小树密钥集合,执行 *DataKeyRecover* 恢复出加密密钥,最后通过 *Decryption* 解密密文数据块,得到明文数据,实现数据的访问.

值得注意的是,本文方案中 User 可以在本地保留 $\{L\}$,以实现对数据的多次访问而无需多次向 Owner 提出访问请求.但是算法 *TrKeyExtract* 和 *DataKeyRecover* 是通过后台处理程序执行的(如浏览器插件),对 User 是

透明的,因此树密钥 $k_{i,j}(k_{p,i})$和加密密钥 k_i 的计算对 User 也是透明的.

3.3 数据确定性删除方法

对于某一数据块 M_i 的删除分为如下两种情况:

(1)未被 User 访问过的数据块 M_i,其对应的树密钥 $k_{p,i}$和加密密钥 k_i 都未暴露.若 M_i 过期了,需要被删除,Owner 只需对 M_i 作标记,之后不计算也不分发 $k_{p,i}$.没有 $k_{p,i}$将不能计算 k_i,就不能解密和访问 M_i,即实现了 M_i 的确定性删除.

(2)已经被 User 访问过的数据块 M_i,其对应的树密钥 $k_{i,j}(k_{p,i})$的分片在 timeout 到达前存在于 DHT 网络中,User 和攻击者都可以通过网络中的密钥分片重构出 $k_{i,j}(k_{p,i})$.经过授权的 User 虽是可信的,但算法 *TrKeyExtract* 和 *DataKeyRecover* 的执行对 User 是透明的,即树密钥 $k_{i,j}$、叶子节点 $k_{p,i}$及加密密钥 k_i 的计算都是由后台处理程序进行的.因此,User 不知道只存在于内存中的 $k_{i,j}(k_{p,i})$和 k_i,只知道有时效限制的随机种子 L_i.当授权时间 timeout 到达后,DHT 网络的动态特性会使得网络节点中存储的密钥分片被确定性地清除.这时通过 L_i 定位到的网络节点上将不再存有密钥分片信息,因而不能重构 $k_{i,j}(k_{p,i})$和计算 k_i,不能解密和访问 M_i.在 User 看来,M_i 是不可解密和访问的,是被确定性删除了的.若 M_i 过期了,需要被删除,Owner 只需标记 M_i 是被删除了的,不再计算和分发 $k_{p,i}$,就能实现 M_i 的确定性删除.

攻击者不是授权用户,没有实现 f_L、f_R 和 f 的算法和后台处理程序.因此,即使攻击者通过嗅探攻击或跳跃攻击获取到了 $k_{i,j}(k_{p,i})$,也无法计算出加密密钥 k_i,不能解密和访问 M_i,即 M_i 对于攻击者而言是被确定性删除了.

因此,对于某一数据块 M_i,无论其是否被访问过,通过本文的方法都可以实现其确定性删除.

当所有数据都过期了,需要被删除时,Owner 只需要删除其保存的根密钥 $k_{0,1}$,就能删除所有的树密钥,没有树密钥就不能计算加密密钥,密文数据对于任何一方都是不可恢复和访问的,即实现了数据的确定性删除.

4 实验及分析

本文通过实验对上述基于 DHT 网络的数据确定性删除方法进行了分析验证,主要是测试了本文方法的时间开销和密钥的确定性删除效果,其中时间开销的测试包括密钥派生树的生成时间和文件的加解密时间.

4.1 实验环境

本文的实验环境为 Intel Core2 3.0GHz,2GB 内存,操作系统为 Ubuntu 10.04,内核版本为 2.6.31.选取的 DHT 网络为 Vuze DHT[14],timeout 为 Vuze DHT 的默认配置 8 小时,选用开源项目 Vanish 的源代码[16]连接 Vuze DHT 网络.加密算法 AES 采用 OpenSSL1.0.0 库中的算法,密钥长度为 256 位.

4.2 实验结果及分析

(1)密钥派生树生成时间

图 3 显示了生成不同高度的密钥派生树所用的时间,从图中可见,对于 1GB 的数据,若每个数据块的大小为 4KB,则需生成一棵高度为 18 的密钥派生树,所用时间为 0.84 秒左右,性能开销是很小的.

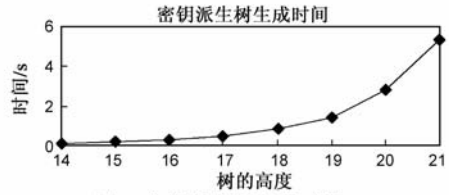

图3 密钥派生树生成时间

(2)文件加密和解密时间

本文对文件加解密的测试是将文件分块后再对文件块加密,解密时先对文件块解密再组合成文件,其中文件块的大小为 4KB.图 4 显示了不同大小的文件分块后加密所用的时间,图 5 为将文件块解密后再组合生成不同大小文件所用的时间关系图.从图 4 和图 5 中可见,文件的加解密时间与文件大小基本上成线性关系.当文件在 100MB 时,加解密时间为 10 秒左右.当文件为 1GB 时,加解密所用时间在 100 秒左右,都在可接受范围内.

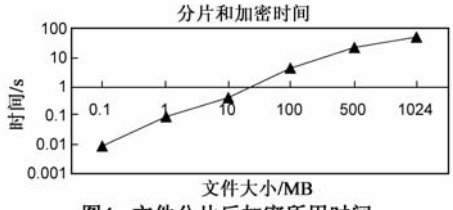

图4 文件分片后加密所用时间

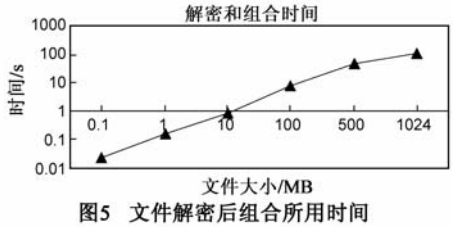

图5 文件解密后组合所用时间

(3)密钥的确定性删除

本文在不同的门限值 N 和不同的门限率下,对单个密钥在网络中的可用率进行了相应的实验.实验的基本思路为:将同一密钥随机向网络中分发多次,然后每隔一段时间,从网络中提取并重构密钥,根据重构出来的密钥来计算网络中密钥的可用率.其中,N 的取值

分别 1,10,20,50,100,150,门限率分别为 50%,60%,70%,80%,90%,100%.当 N 为 1,10,20 时,随机产生 20 个种子,根据这些种子将同一密钥向网络中分发 20 次;当 N 为 50 时,分发的次数为 15;N 为 100 和 150 时,分发的次数为 10.当 N 不同时,分发的次数不相同,是因为分发一次密钥所用的时间与 N 有关,如图 6 所示.为使得密钥能够在一个提取周期,即 1 个小时内全部分发到网络中,因此就针对不同的 N 选择了不同的分发次数.

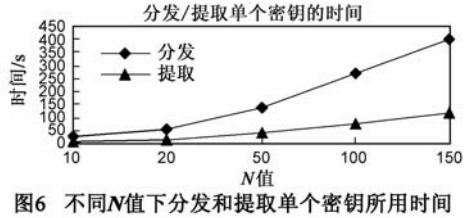

图6　不同N值下分发和提取单个密钥所用时间

本文从第一次开始分发密钥计时,每隔 1 个小时从网络中提取并重构密钥,整个提取过程持续 12 个小时.图 7 为当 N 不同,门限率都为 70%(除 $N=1$ 外)时,网络中密钥的可用率与时间的关系图.

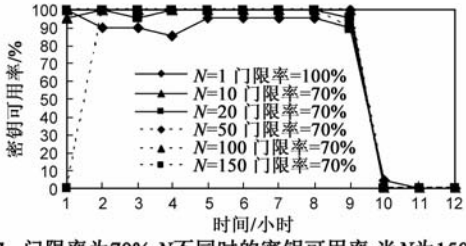

图7　门限率为70%,N不同时的密钥可用率.当N为150时,曲线在1小时处为0,是由于此时密钥分发还未结束,密钥提取还未开始

从图 7 中可以看出,经过 9 个小时后,将不能再从网络中提取并重构出密钥.而在 9 个小时前,分发到网络中的密钥基本上都是可用的.值得注意的是,图 7 中所有曲线在 9 小时后才降为 0,与 Vuze DHT 网络的默认 timeout 配置 8 小时是不同的.这是由于在默认配置的 8 个小时后,Vuze 不会立刻就清除数据,还会将数据额外保留一段缓冲时间(该时间从几分钟到一个小时不等),以便于数据分发者更新分发到网络中的数据,减少数据分发者和网络节点由于时钟不同步而导致的数据丢失率.若在该缓冲时间内,数据分发者不更新也不再向网络中分发数据,那么数据在该缓冲时间过后会被 DHT 网络确定性删除.

在图 6 中,单个密钥的分发和提取所用时间不仅与门限值 N 有关,还与实际的网络状况(如网速)相关.对于多个密钥,则通过并行化方法同时分发(提取)多个密钥,使其时间开销和单个密钥的时间开销维持在一个数量级上.综合图 6、图 7 和图 8 的实验结果,本文方法在 N 为 50,门限率为 70%或 80%时,性能开销较低,兼顾了系统的可用性和安全性.

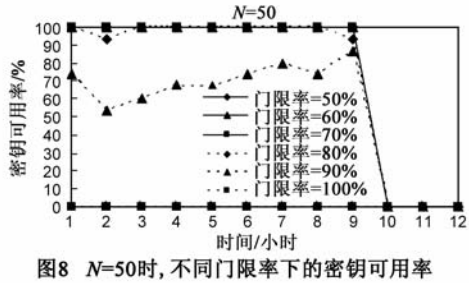

图8　N=50时,不同门限率下的密钥可用率

5　安全性分析

本文的密钥派生方法的安全性由 SHA-1 函数的单向特性保证,即由父节点可以很容易计算出其子节点值,但是由子节点却很难逆向计算出其父节点和兄弟节点的值.

分别称文献[5,6,10,11]中的方案为 FADE 方案、Vanish 方案、SSDD 方案和 SafeVanish 方案.表 1 为本文方案上述方案的对比分析表,表中 O 表示不存在该项.

在本文中,分发到 DHT 网络中的密钥为树密钥,不是最终的加密密钥.因此,即使攻击者通过嗅探攻击或跳跃攻击在 timeout 到达之前,从 DHT 网络中获取到了足够的密钥分片并重构出了树密钥 $k_{i,j}(k_{p,i})$,但由于攻击者不知道变换函数 f,不能将树密钥变换为解密密钥 k_i,也就不能解密并访问数据.因此,本文方案可以有效地抵抗嗅探攻击和跳跃攻击.

文献[5]中的密钥删除是由一个第三方的密钥管理者来实现的,正如其文中所提到的,密钥管理者可能不可信(被攻陷了或者是与 CSP 合谋).那么密钥管理

表 1　方案对比表

	密钥	密钥管理方式	密钥删除方式	抗嗅探攻击	抗跳跃攻击
FADE 方案	数据密钥为单密钥,控制密钥为多密钥	策略的逻辑组合	密钥管理者删除	O	O
Vanish 方案	单密钥	O	DHT 网络删除	否	否
SSDD 方案	单密钥	O	DHT 网络删除	否	否
SafeVanish 方案	单密钥	O	DHT 网络删除	是	是
本文方案	多密钥	密钥派生树	DHT 网络删除	是	是

者可能就没有删除密钥,数据的机密性就可能被破坏.这是由第三方集中管理密钥所导致的安全隐患.在本文的方法中,分发到网络中的树密钥是由动态分散的DHT网络自动删除的,是不受任何实体控制的,避免了第三方集中管理密钥可能导致的安全隐患.Owner对于密钥的删除,只是对密钥作标示或者删除树根密钥,不会由于复杂的密钥删除而导致其他安全隐患.

6 结束语

本文为了保护云存储系统数据的机密性,在密钥派生树和DHT网络的基础上,提出了一种适于云存储系统的数据确定性删除方法.该方法首先通过基于SHA-1的密钥派生树组织和管理密钥,能够支持云存储系统中数据块级的加密和操作.然后,本文将密钥经Shamir秘密共享方法处理后随机分发到DHT网络中,利用DHT网络的动态特性使得密钥在特定的授权时间到达后自动从网络中消失,从而确保在非授权时间内不能解密和访问数据,即实现了云存储模式下的数据确定性删除.实验结果表明,该方法生成密钥树和文件加解密等所用的时间开销较小,在授权时间到达后能够确定性删除分发到DHT网络中的密钥,满足云存储系统中过期数据或备份文件的确定性删除要求.并且该方法能有效地抵抗嗅探攻击和跳跃攻击,防止密钥和数据的泄漏,同时也避免了密钥的第三方集中管理方式中可能存在的安全隐患.

参考文献

[1] 武永卫,黄小猛.云存储[J].中国计算机学会通讯,2009,5(6):44-52.

[2] 王铁军,刘恒,孙明,等.资源定位服务的分布式生成树模型及算法研究[J].电子学报,2011,39(1):364-369.
Wang Tiejun, Liu Heng, Sun Ming, et al. Research on the model and algorithms based on distributed spanning tree for resource location service[J]. Acta Electronica Sinica, 2011, 39(1): 364-369. (in Chinese)

[3] 冯登国,张敏,张妍,等.云计算安全研究[J].软件学报,2011,22(1),71-83.

[4] Amazon. [EB/OL]. http://aws.amazon.com/s3/, 2011-07-29/2011-07-29.

[5] Yang Tang, Patrick P.C. Lee, John C.S. Lui, et al. FADE: Secure overlay cloud storage with file assured deletion[A]. Proc of the SecureComm'10[C]. New York: ACM Press, 2010. 380-397.

[6] R Geambasu, T Kohno, Amit A Levy, et al. Vanish: Increasing data privacy with self-destructing data[A]. Proc of USENIX Security'09[C]. Berkeley, CA: USENIX Association, 2009. 299-316.

[7] I Stoica, R Morris, D Karger, et al. Chord: A scalable peer-to-peer lookup service for internet applications[A]. Proc of the SIGCOMM'01[C]. New York: ACM Press, 2001. 149-160.

[8] Frank Dabek. A Distributed Hash Table[D]. Massachusetts, Massachusetts Institute of Technology, 2005.

[9] Scott Wolchok, Owen S. Hofmann, Nadia Heninger, et al. Defeating vanish with low-cost sybil attacks against large DHTs[A]. Proc of NDSS 2010[C]. Washington, DC: Internet Society, 2010. 37-51.

[10] Fengshun Yue, Guojun Wang, Qin Liu. A secure self-destructing scheme for electronic data[A]. Proc of EUC 2010[C]. New York: IEEE Press, 2010. 651-658.

[11] Lingfang Zeng, Zhan Shi, Shengjie Xu, et al. Safevanish: An improved data self-destruction for protecting data privacy[A]. Proc of CloudCom 2010[C]. New York: IEEE Press, 2010. 521-528.

[12] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[13] Weichao Wang, Zhiwei Li, Rodney Owens, et al. Secure and efficient access to outsourced data[A]. Proc of CCSW'09[C]. New York: ACM Press, 2009. 55-65.

[14] J. Falkner, M. Piatek, J John, et al. Profiling a million user DHT[A]. Proc of the 7th ACM SIGCOMM conference on Internet measurement[C]. New York: ACM Press, 2007. 129-134.

[15] Y Xie, F Yu, K Achan, et al. How dynamic are IP addresses? [A]. Proc of SIGCOMM'07[C]. New York: ACM Press, 2007. 301-312.

[16] Vanish. [EB/OL]. http://vanish.cs.washington.edu/, 2011-07-29/2011-07-29.

作者简介

王丽娜 女,1964年生,博士,教授,武汉大学计算机学院副院长,空天信息安全与可信计算教育部重点实验室(B类)主任.研究方向为信息安全、云计算安全等.

任正伟 男,1986年生,武汉大学计算机学院博士生.研究方向为应用密码学、云计算安全.E-mail: zhengwei_ren@163.com