软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1451−1462 [doi: 10.13328/j.cnki.jos.004999]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

具有私钥可恢复能力的云存储完整性检测方案*

沈文婷[1], 于 佳[1,2], 杨光洋[1], 程相国[1], 郝 蓉[1]

[1](青岛大学 信息工程学院,山东 青岛 266071)

[2](信息安全国家重点实验室(中国科学院 信息工程研究所),北京 100093)

通信作者: 于佳, E-mail: qduyujia@gmail.com

摘 要: 共享数据云存储完整性检测用来验证一个群体共享在云端数据的完整性,是最常见的云存储完整性检测方式之一.在云存储完整性检测中,用户用于生成数据签名的私钥可能会因为存储介质的损坏、故障等原因而无法使用.然而,目前已有的共享数据云存储完整性检测方案均未考虑到这个现实问题.探索了如何解决共享数据云存储完整性检测中私钥不可用的问题,提出了第 1 个具有私钥可恢复能力的共享数据云存储完整性检测方案.在方案中,当一个群用户的私钥不可用时,可以通过群里的 t 个或者 t 个以上的用户帮助其恢复私钥.同时,设计了随机遮掩技术,用于确保参与成员私钥的安全性.用户也可验证被恢复私钥的正确性.最后,给出安全性和实验结果的分析,结果显示所提方案是安全高效的.

关键词: 云存储;数据完整性检测;私钥恢复;秘密共享;安全性

中图法分类号: TP309

中文引用格式: 沈文婷,于佳,杨光洋,程相国,郝蓉.具有私钥可恢复能力的云存储完整性检测方案.软件学报,2016,27(6):1451−1462. http://www.jos.org.cn/1000-9825/4999.htm

英文引用格式: Shen WT, Yu J, Yang GY, Cheng XG, Hao R. Cloud storage integrity checking scheme with private key recovery capability. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1451−1462 (in Chinese). http://www.jos.org.cn/1000-9825/4999.htm

Cloud Storage Integrity Checking Scheme with Private Key Recovery Capability

SHEN Wen-Ting[1], YU Jia[1,2], YANG Guang-Yang[1], CHENG Xiang-Guo[1], HAO Rong[1]

[1](College of Information Engineering, Qingdao University, Qingdao 266071, China)

[2](State Key Laboratory of Information Security (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

Abstract: Verifying the integrity of cloud data shared by a group is one of the most common usage of cloud storage integrity checking. In cloud storage integrity checking, the private key which is used to generate data signatures by user may be unavailable because of the damage or the fault of storage medium. However, currently existing cloud storage integrity checking schemes for shared data do not consider this realistic problem. This paper first explores how to deal with the problem of the private key unavailability in cloud storage integrity checking for shared data. A new scheme that enables cloud storage integrity checking for shared data with private key-recovery ability is proposed. In this scheme, when a group user's private key is unavailable, this user's private key can be recovered with the help

* 基金项目: 国家自然科学基金(61572267, 61272425, 60703089, 61402245); 山东省自然科学基金(ZR2014FQ010, ZR2010FQ019); 信息安全国家重点实验室开放课题基金; 华为科技基金(YB2013120027); 青岛市建设事业科技发展项目(JK2015-26)

Foundation item: National Natural Science Foundation of China (61572267, 61272425, 60703089, 61402245); Shandong Provincial Natural Science Foundation of China (ZR2014FQ010, ZR2010FQ019); Open Research Fund from the State Key Laboratory of Information Security; Huawei Technique Fund (YB2013120027); Qingdao Construction Project of Science and Technology Development (JK2015-26)

收稿时间: 2015-08-14; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21

CNKI 网络优先出版: 2016-01-22 10:14:50, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1014.006.html

of *t* or more users in the group. At the same time, a random masking technology is designed to guarantee the security of participating members' private keys. The user can also verify the correctness of the recovered private key. Finally, the analysis of security and experimental results are provided to show that the proposed scheme is secure and efficient.

Key words: cloud storage; data integrity checking; private key recovery; secret sharing; security

云存储作为云计算的一种重要应用形式,允许数据拥有者将其数据外包到云服务器,由云服务器来存储和管理这些数据.这种存储方法从根本上改变了资源部署和服务提供的方式,避免了用户对本地软硬件和数据管理维护的大量投入,与此同时,也使用户可以享受到云存储技术带来的高质量的服务.云存储在给用户带来诸多好处的同时,也带来一些新的安全问题[1].其主要原因是,用户把数据存储到云服务器后,就失去了对数据的直接控制,存储在云服务器上的数据可能会因为软硬件的故障或者人为的失误造成丢失或者损坏[2−6].更严重的是,云服务提供商(CSP)可能为了维护自己的声誉而隐瞒数据丢失的事实,甚至可能为了节约存储空间、提高经济利益,而故意删除用户不经常访问的数据[7,8].因此,云用户完全有理由怀疑自己存储在云服务器上的数据是否仍然完整可用,这使得云用户定期检测存储在云服务器上的数据是否完整是完全必要的.如果周期性的数据完整性检测工作由用户自己完成,则需消耗用户大量的资源,给用户带来很大的负担,这显然不是一个可行的方法.更好的方法是引入第三方审计者(TPA)去帮助用户定期检查存储在云服务器上的数据是否完整.基于这个思想,大量的云存储数据完整性检测方案被提出[9−14].

在云存储的实际应用中,群体共享数据存储是一种非常重要的应用形式,比如 iCloud,Google Drive 和 Dropbox.在这种共享数据的云存储形式中,属于某个群体的任何用户均可以对存储在云服务器上的数据进行访问、下载等操作.针对这种云存储方式,很多数据完整性检测方案被提出.例如:文献[15]基于代理重签名技术设计了一个数据完整性检测方案,不仅支持群动态(用户的加入和撤销),还支持用户身份的隐私保护.文献[16]通过使用扩展的索引哈希表,提出了支持共享云数据动态操作的数据完整性检测方案.Yuan 等人[17]通过构造代理认证器和代理标签技术来设计一个满足群用户撤销的数据完整性检测方案.

观察目前已有的云存储数据完整性检测方案,均没有考虑到一个现实的安全问题——用户私钥不可用问题.在实际的场景中,私钥不可用的情况是完全可能发生的.一般来说,云用户存储私钥的常用方法包含以下两种.一种是将私钥的密文存储到磁盘中.然而,这可能会因为磁盘的磁头碰撞、电路故障、机械设备损坏等原因[18,19]导致存储在磁盘里的私钥无法使用.另一种方法是将私钥存储在诸如智能卡[20]的便携硬件中.但是智能卡可能会丢失,也可能因为外力破坏、芯片故障等原因造成损坏,从而导致存储在智能卡中的私钥无法使用.

在共享数据云存储完整性检测中,一旦群用户用于生成数据签名(又称认证器)的私钥不可用,就会导致群用户无法再为数据生成签名,从而使数据完整性检测工作无法进行.如果使用传统的方法解决这个问题,首先,需要给私钥不可用的群用户重新分发一个新的私钥;然后,群用户从云服务器上下载他签名的所有数据块,并用新的私钥对这些数据块重新签名;最后,将这些新的签名上传到云服务器.但是这种方法在云存储环境下是不可行的.这是因为用户需要下载的数据量非常大,使用这种方法会导致巨大的通信和计算开销.解决这个问题的另一种可能的方法是密钥托管的方法,云用户将私钥托管于或者存储在第三方托管中心[21].然而,密钥托管的方法需要引入一个或多个托管中心,这些托管中心的引入,一方面会增加系统复杂性,另一方面可能会导致新的安全问题,并存在安全争议,也不是理想的方法.因此,如何应对用户在共享数据云存储完整性检测中用户私钥不可用的问题有待进一步解决.

本文首次探索了在共享数据云存储完整性检测中如何解决云用户私钥不可用的问题.具体来说,本文的贡献归纳如下:

(1) 首次提出了具有私钥可恢复能力的共享数据云存储完整性检测的概念.在共享数据云存储的场景中,若某一个群用户的私钥不可用,则可通过其他的群用户帮助他恢复.我们给出了第 1 个具有私钥可恢复能力的实用方案.在方案中,私钥分发中心(KDC)使用秘密共享的方法[22,23]为群用户产生 n(假设群里有 n 个用户)个份额作为他们的私钥,当群中的用户私钥不可用时,可通过秘密共享的方法,让 t(t 是门限值)个或者多于 t 个群用户去帮助他恢复私钥.该方法既避免了私钥的重新颁发,也避免了密钥托管.

(2) 然而,如果直接使用秘密共享的方法,会存在新的安全问题.即,私钥恢复阶段可能会暴露参与私钥恢复用户的私钥.为了在私钥恢复过程中保护用户私钥的安全性,我们提出了一个随机遮掩的方法来盲化参与私钥恢复用户的私钥.此外,在私钥恢复阶段,我们使用可验证技术来保证参与成员提供盲化后私钥的正确性.

(3) 给出了方案安全性和实验结果的分析,分析结果显示提出的方案是安全、高效的.最后,也讨论了如何减少私钥恢复的运算量、如何选择 t 的值的问题.

1 系统模型和设计目标

1.1 系统模型

如图 1 所示,本文的系统模型包含以下 4 种不同的实体:云服务器、群用户、私钥分发中心(KDC)和第三方审计者(TPA).云服务器为群用户提供数据存储和数据共享服务.在群里,每一个群用户都可以上传数据到云服务器,并且可以把这些上传了的数据共享给群里的其他用户,即群里的每个用户均可以访问群里的任何一个用户上传到云服务器上的共享数据.KDC 给每一个群用户产生一个私钥和一个公钥,然后把这些私钥分别发送给群中相应的用户,并公开他们的公钥.TPA 可以代表群用户验证存储在云服务器上的共享数据的完整性.当 TPA 想要检查存储在云服务器上共享数据的完整性时,他会发送一个审计质询给云服务器.云服务器收到 TPA 发来的审计质询后,便会产生一个审计证明来回复 TPA,用来证明它真实、完整地拥有着这些数据.收到云服务器发来的审计证明后,TPA 通过验证收到的审计证明的正确性来判断存储在云服务器上的共享数据是否完整.

在私钥分发阶段,KDC 使用秘密共享的方法为群用户 $(U_1,U_2,...,U_n)$ 产生 n(假设群里有 n 个用户)个份额 $(s_1,s_2,...,s_n)$ 分别作为他们的私钥.在私钥恢复阶段,当群里有一个用户 U_m 的私钥 s_m 不可用时,可通过秘密共享的方法利用 t(t 是门限值)个群用户 U_{l_i} ($i\in[1,t]$且 $l_i\neq m$)盲化后的私钥 $(s'_{l_1},s'_{l_2},...,s'_{l_t})$ 去恢复私钥 s_m.私钥分发与私钥恢复过程如图 2 所示.

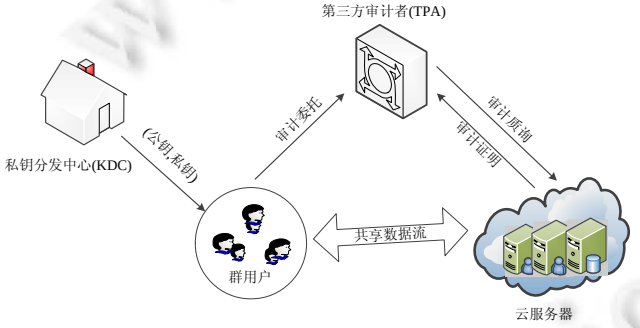

Fig.1 System model

图 1 系统模型

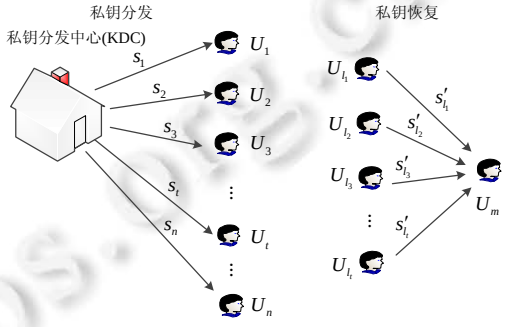

Fig.2 The model of private key distribution and private key recovery

图 2 私钥分发和私钥恢复模型

1.2 设计目标

为了安全有效地检查存储在云服务器上的共享数据的完整性,设计的方案应该具有以下几个性质:

(1) 私钥可恢复性:t 个或者多于 t 个群用户可以帮助私钥不可用用户恢复私钥.

(2) 被恢复私钥的可验证性:确保私钥不可用的群用户可以验证所恢复私钥的正确性.

(3) 参与私钥恢复用户私钥的安全性:确保私钥恢复过程中,参与者的私钥是安全、不可泄露的.

(4) 存储完整性:确保云服务器只有真实存储群用户的完整数据才能够通过 TPA 的验证.

(5) 公开审计性:允许 TPA 验证存储在云服务器上的共享数据的完整性,但是不需要从云服务器上下载全部的数据.

2 定义和预备知识

2.1 具有私钥可恢复能力的共享数据云存储完整性检测方案

一个具有私钥可恢复能力的共享数据云存储完整性检测方案包含以下 5 种算法:*KeyGen*,*SigGen*, *ProofGen*,*ProofVerify* 和 *KeyRecover*.

(1) $KeyGen(1^k)\to(pk,sk)$:公私钥产生算法,由 KDC 执行.输入安全参数 k,输出公私钥对(pk,sk).

(2) $SigGen(sk,m)\to\sigma$:签名产生算法,由群用户执行.输入私钥 sk 和消息块 m,输出对消息块 m 的签名σ.

(3) $ProofGen(F,\Phi,chal)\to P$:证明产生算法,由云服务器执行.输入共享数据文件 F、签名集合Φ和审计质询 $chal$,输出能够证明云服务器真实拥有完整共享数据的审计证明 P.

(4) $ProofVerify(pk,chal,P)\to\{\text{True,False}\}$:证明验证算法,由 TPA 执行.输入公钥 pk、审计质询 $chal$ 和审计证明 P.如果 TPA 验证这个证明 P 是有效的,输出 True;否则,输出 False.

(5) $KeyRecover(s_{l_1},s_{l_2},...,s_{l_t})\to s_m$:私钥恢复算法,由群用户执行.输入群用户的私钥 $s_{l_i}(i=1,2,...,t)$,输出私钥不可用用户 $U_m(m\notin[l_1,l_t])$ 的私钥 s_m.

2.2 双线性映射

设 G_1 和 G_2 是两个阶为大素数 p 的乘法循环群.定义双线性映射为 $\hat{e}:G_1\times G_1\to G_2$, 并满足以下性质:

(1) 可计算性:对于任意的 $P,Q\in G_1$, 存在有效算法可以容易计算 $\hat{e}(P,Q)$.

(2) 双线性:对于任意的 $P,Q\in G_1,a,b\in Z_p^*$, 满足 $\hat{e}(P^a,Q^b)=\hat{e}(P,Q)^{ab}$.

(3) 非退化性:存在 $P,Q\in G_1$, 使得 $\hat{e}(P,Q)\neq 1$.

2.3 Shamir的(*t*,*n*)秘密共享方案

假设 p 为大素数,私钥分发中心首先选择一个随机多项式 $f(x)=k+\sum_{j=1}^{t-1}a_jx^j(\text{mod }p)\in Z_p[x]$, 其中,$k$ 为共享的秘密.然后为每个成员 $P_i(i=1,2,...,n)$ 计算份额 $s_i=f(i)=k+\sum_{j=1}^{t-1}a_ji^j(\text{mod }p)$, 并将 s_i 发送给 $P_i(i=1,2,\ldots,n)$.

秘密重构:成员集合 $B(|B|=t)$拿出他们的份额,计算任何成员 P_i 的份额 $s_i=\sum_{P_i\in B}C_{Bi}(l)\ s_i(\text{mod }p)$, 其中,

$$C_{Bi}(l)=\prod_{P_j\in B\setminus\{P_i\}}\frac{l-j}{i-j}.$$

2.4 计算Diffie-Hellman问题(CDH问题)

设 G_1 是阶为大素数 p 的乘法循环群,将群 G_1 上的 CDH 问题定义为:给定 $g,g^a,h\in G_1$,其中, $a\in Z_p^*$, 计算 $h^a\in G_1$.

2.5 离散对数问题(DL问题)

设 G_1 是阶为大素数 p 的乘法循环群,将群 G_1 上的 DL 问题定义为:给定 $g,g^x\in G_1$,计算 $x\in Z_p^*$.

3 提出的方案

3.1 符号说明

令 G_1 和 G_2 为两个阶为大素数 p 的乘法循环群,g,u 是群 G_1 上的两个独立的生成元, $\hat{e}:G_1\times G_1\to G_2$ 是一个双线性映射.令 $H:Z_p^*\to G_1$ 是一个密码哈希函数.全体公共参数为 $(G_1,G_2,p,\hat{e},g,u,H)$.假设要上传的共享数据文件 F 被划分为 d 个数据块,记 $F=(m_1,m_2,...,m_d)$.群成员对共享数据文件 F 的签名集合记为 $\Phi=(\sigma_1,\sigma_2,...,\sigma_d)$.一个群由 n 个用户组成,记作 $\{U_1,U_2,...,U_n\}$.t 表示秘密共享方案的门限值.

3.2 方案描述

1) 算法 $KeyGen(1^k)\to(pk,sk)$

(1) KDC 选择一个随机多项式

$$f(x)=a_0+\sum_{i=1}^{t-1}a_i x^i\in Z_p[x],\ a_i\in Z_p^* \tag{1}$$

然后计算 $s_i=f(i)(i=1,2,...,n)$ 和 $g^{s_i}(i=1,2,...,n)$ 分别作为公私钥传输给相应群用户 $U_i(i=1,2,...,n)$.

(2) KDC 广播承诺值 $g^{a_i}(i=0,1,...,t-1)$.

(3) 每个群用户 $U_i(i=1,2,...,n)$ 收到 KDC 发来的承诺值 $g^{a_i}(i=0,1,...,t-1)$ 后,存储承诺值,并通过验证以下等式是否成立来判断 KDC 发来的私钥 $s_i(i=1,2,...,n)$ 的正确性:

$$g^{s_i}=\prod_{j=0}^{t-1}(g^{a_j})^{i^j} \tag{2}$$

若等式(2)成立,则说明群用户 $U_i(i=1,2,...,n)$ 相信 KDC 发送的私钥 s_i 是正确的;否则,认为是不正确的.

2) 算法 $SigGen(sk,m)\to\sigma$

(1) 要签名的数据块 $m=m_i\in Z_p^*(i\in[1,d])$, 群用户 $U_j(j=[1,n])$ 用私钥 $sk=s_j$ 给数据块 m_i 计算相应的签名 $\sigma_i=(H(i)\cdot u^{m_i})^{s_j}$.

(2) 群用户 U_j 发送$\{m_i,\sigma_i\}$到云服务器,随后从本地删除数据块 m_i 和其对应的签名σ_i.

3) 算法 $ProofGen(F,\Phi,chal)\to P$

(1) TPA 通过发送一个审计质询 *chal* 给云服务器来验证存储在云服务器上的共享数据完整性.TPA 产生审计质询 *chal* 的具体过程是:① 随机选择一个有 *c* 个元素的集合 *I*,其中,$I\subseteq[1,d]$.② 对于每个 $i\in I$,均产生一个随机值 $v_i\in Z_p^*$.③ 输出一个审计质询 $\{i,v_i\}_{i\in I}$ 发送给云服务器.

(2) 云服务器接收到审计质询 *chal* 后,产生一个审计证明 *P* 回复 TPA,证明它真实拥有完整的共享云数据.云服务器产生证明 *P* 的具体过程如下:

① 把集合 *I* 划分为 *n* 个子集,即 $I=\{I_1,I_2,...,I_n\}$, 其中,I_i 是被选择的数据块中被群用户 U_i 签名的数据块集合.集合 I_i 中有 c_i 个元素,所以有, $c=\sum_{i=1}^{n}c_i, I=I_1\cup I_2\cup...\cup I_n$ 以及 $I_i\cap I_j=\varnothing(i\neq j)$.

② 给每个集合 I_j 计算数据块的线性组合 $\mu_j=\sum_{i\in I_j}v_i m_i$ 和签名的聚合 $\sigma_j'=\prod_{i\in I_j}\sigma_i^{v_i}$.

③ 将审计证明 $P=(\mu,\sigma)$发送给 TPA,其中, $\mu=\{\mu_1,\mu_2,...,\mu_n\},\sigma=\{\sigma_1',\sigma_2',...,\sigma_n'\}$.

4) 算法 $ProofVerify(pk,chal,P)\to\{\text{True},\text{False}\}$

当 TPA 收到审计证明 *P* 后,通过验证以下等式是否成立来判断证明 *P* 的正确性:

$$\hat{e}\left(\prod_{j=1}^{n}\sigma_j',g\right)()=\prod_{j=1}^{n}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right) \tag{3}$$

若等式(3)成立,则说明存储在云服务器上的共享数据是完整的;否则,认为至少有一个数据块是不正确的.等式(3)的正确性由下列推导保证:

$$\begin{aligned}\hat{e}\left(\prod_{j=1}^{n}\sigma_j',g\right)&=\prod_{j=1}^{n}\hat{e}(\sigma_j',g)\\&=\prod_{j=1}^{n}\hat{e}\left(\prod_{i\in I_j}\sigma_i^{v_i},g\right)\\&=\prod_{j=1}^{n}\hat{e}\left(\prod_{i\in I_j}\left((H(i)\cdot u^{m_i})^{s_j}\right)^{v_i},g\right)\\&=\prod_{j=1}^{n}\hat{e}\left(\prod_{i\in I_j}(H(i)\cdot u^{m_i})^{v_i},g^{s_j}\right)\\&=\prod_{j=1}^{n}\hat{e}\left(u^{\sum_{i\in I_j}m_i v_i}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right)\\&=\prod_{j=1}^{n}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right).\end{aligned}$$

5) 算法 $KeyRecover(s_{l_1},s_{l_2},...,s_{l_t})\to s_m$

(1) 私钥不可用的群用户 U_m 随机选择 *t* 个群用户帮助他恢复私钥 s_m.假定被选择的 *t* 个群用户为 $U_{l_i}(i\in[1,t],$且$l_i\neq m)$.

(2) 每个群用户 $U_{l_i}(i=1,2,...,t)$ 选择一个随机多项式

$$f_{l_i}(x)=\sum_{k=0}^{t-1}a_{l_ik}x^k(\bmod p) \tag{4}$$

然后分别计算用于盲化私钥 $s_{l_i}(i=1,2,...,t)$ 的 $u_{l_il_j}=f_{l_i}(l_j)(j=1,2,...,t)$ 和 $u_{l_im}=f_{l_i}(m)$. 最后,群用户 U_{l_i} 把 u_{l_im} 发送给群用户 U_m,把 $u_{l_il_j}(j\in[1,t],j\neq i)$ 发给除自己以外的其他 t–1 个参与私钥恢复的群用户,并广播承诺值 $g^{u_{l_im}},g^{a_{l_ik}}(k=0,1,\ldots,t-1)$ 和 $g^{u_{l_il_j}}$.

(3) 群用户 U_m 接收到消息 u_{l_im} 后,通过以下等式判断消息 u_{l_im} 的正确性:

$$g^{u_{l_im}}=\prod_{k=0}^{t-1}(g^{a_{l_ik}})^{m^k} \tag{5}$$

若等式(5)成立,则群用户 U_m 相信群用户 $(U_{l_1},U_{l_2},...,U_{l_t})$ 发来的消息 u_{l_im} 是正确的,然后计算恢复私钥时用于解盲化的 $u_m=\sum_{i=1}^{t}u_{l_im}$.

(4) 群用户 $U_{l_j}(j=1,2,...,t)$ 收到其他 t–1 个参与私钥恢复的群用户的消息 $u_{l_il_j}$ 后,通过验证以下等式是否成立来判断他接收到的消息 $u_{l_il_j}$ 的正确性:

$$g^{u_{l_il_j}}=\prod_{k=0}^{t-1}(g^{a_{l_ik}})^{(l_j)^k} \tag{6}$$

若等式(6)成立,则群用户 $U_{l_j}(j=1,2,...,t)$ 相信他接收到的消息 $u_{l_il_j}$ 是正确的,然后计算盲化后的私钥 $s'_{l_j}=s_{l_j}+\sum_{i=1}^{t}u_{l_il_j}$, 并把它发给群用户 U_m.

(5) 群用户 U_m 接收到群用户 $U_{l_j}(j=1,2,...,t)$ 发来的盲化后的私钥 s'_{l_j} 后,通过验证以下等式是否成立来判断 s'_{l_j} 的正确性:

$$g^{s'_{l_j}}=\prod_{k=0}^{t-1}(g^{a_{l_ik}})^{(l_j)^k}\cdot\prod_{i=1}^{t}g^{u_{l_il_j}} \tag{7}$$

等式(7)的正确性由以下推导保证:

$$\begin{aligned}g^{s'_{l_j}}&=g^{s_{l_j}+\sum_{i=1}^{t}u_{l_il_j}}\\&=g^{s_{l_j}}\cdot\prod_{i=1}^{t}g^{u_{l_il_j}}\\&=\prod_{k=0}^{t-1}(g^{a_{l_ik}})^{(l_j)^k}\cdot\prod_{i=1}^{t}g^{u_{l_il_j}}.\end{aligned}$$

若式(7)确实成立,则群用户 U_m 相信群用户 $U_{l_j}(j=1,2,...,t)$ 发来的盲化后私钥 s'_{l_j} 是正确的,然后计算私钥:

$$s_m=\sum_{j=1}^{t}C_{Bl_j}(m)s'_{l_j}-u_m(\bmod p) \tag{8}$$

其中, $C_{Bl_j}(m)=\prod_{i=\{1,2,\ldots,t\}\setminus\{j\}}\frac{m-l_i}{l_j-l_i}$.

4 安全分析

定理 1(被恢复私钥的正确性). 在私钥恢复过程中,如果被选择用于私钥恢复的 t 个群用户是诚实的,那么私钥不可用的群用户 U_m 可以得到正确的份额.

证明:

$$\begin{aligned}s_m&=\sum_{j=1}^{t}C_{Bl_j}(m)s'_{l_j}-u_m\\&=\sum_{j=1}^{t}C_{Bl_j}(m)\cdot(s_{l_j}+\sum_{i=1}^{t}u_{l_il_j})-u_m\\&=\sum_{j=1}^{t}C_{Bl_j}(m)s_{l_j}+\sum_{j=1}^{t}\sum_{i=1}^{t}C_{Bl_j}(m)u_{l_il_j}-u_m\\&=\sum_{j=1}^{t}C_{Bl_j}(m)s_{l_j}+\sum_{i=1}^{t}\sum_{j=1}^{t}C_{Bl_j}(m)f_{l_i}(l_j)-u_m\\&=\sum_{j=1}^{t}C_{Bl_j}(m)s_{l_j}+\sum_{i=1}^{t}u_{l_im}-\sum_{i=1}^{t}u_{l_im}\\&=\sum_{j=1}^{t}C_{Bl_j}(m)s_{l_j}.\end{aligned}$$

□

定理 2(被恢复私钥的可验证性). 在私钥恢复过程中,如果被选择的 t 个成员中,有不诚实的成员提供错误

的盲化私钥,则可通过提出的方案检测出来.

证明:不失一般性,假设在私钥恢复阶段,成员 U_{l_j} 是不诚实的,提供一个不正确的盲化私钥 $s''_{l_j} \neq s'_{l_j}$,则

$$g^{s''_{l_j}} \neq g^{s_{l_j}+\sum_{i=1}^{t} u_{l_j l_j}} = g^{s_{l_j}} \cdot \prod_{i=1}^{t} g^{u_{l_j l_j}} = \prod_{k=0}^{t-1} (g^{a_{l_j k}})^{(l_j)^k} \cdot \prod_{i=1}^{t} g^{u_{l_j l_j}}.$$

可见,它不能通过等式(7)的验证.因此,不诚实成员提供的错误盲化私钥总能检测出来. □

定理 3(被恢复私钥的安全性). 即使攻击者可以收买 t–1 个参与私钥恢复的群用户,他仍然不知道私钥不可用用户 U_m 的私钥 s_m.

证明:假定在私钥恢复过程中,被攻击者收买的群用户是 $\{U_{l_1},U_{l_2},...,U_{l_{t-1}}\}$,这意味着攻击者能够知道这 t–1 个群用户的私钥 $s_{l_1},s_{l_2},...,s_{l_{t-1}}$.而私钥 $s_{l_1},s_{l_2},...,s_{l_{t-1}}$ 是多项式 $f(x)$上的 t–1 点.根据 Shamir 的(t,n)秘密共享方案的安全性,可以知道,如果只知道这个多项式 $f(x)$上的 t–1 个点,不能计算出份额 s_m.又因为 G_q 中离散对数问题是难解的,攻击者不能从任何的公开承诺值中得到关于多项式 $f(x)$上任何点的值.因此,即使攻击者可以收买参与私钥恢复的 t–1 个群用户,他仍然不知道私钥 s_m. □

定理 4(参与私钥恢复用户私钥的安全性). 用户 U_m 不能从参与恢复私钥用户 U_{l_j} $(j=1,2,...,t)$ 发来的盲化私钥 s'_{l_j} 中,得到真实的私钥 s_{l_j}.

证明:在私钥恢复阶段,用户 U_{l_j} 选择的多项式(4)是随机多项式,因此通过多项式(4)计算出的 $u_{l_j l_j}$ 是随机的.这意味着 $\sum_{i=1}^{t} u_{l_j l_j}$ 也是随机数,用户 U_m 无法知道 $\sum_{i=1}^{t} u_{l_j l_j}$.因此,通过 $s'_{l_j} = s_{l_j} + \sum_{i=1}^{t} u_{l_j l_j}$ 计算的 s'_{l_j} 对于用户 U_m 而言是完全随机的,用户 U_m 无法从 s'_{l_j} 中得到真正的 s_{l_j}. □

定理 5(存储的完整性). 在提出的共享数据云存储完整性检测方案中,只有云服务器真实存储群用户完整数据才可以通过 TPA 的验证.

证明:我们仍然使用文献[24]的知识证明方法构造一个知识提取器.如果云服务器没有存储完整的数据而通过 TPA 验证,则可以通过知识提取器与协议的反复交互提出完整的质询数据块.采用类似文献[24]的一系列游戏的方法进行证明.

游戏 1:参见文献[24]游戏 0.

游戏 2:游戏 2 和游戏 1 相似,仅有一处不同.即质询者保存所有它对敌手查询的应答,并观察它与敌手之间的质询、应答过程中的每次实例,如果发现某个聚合的签名 σ'_l 不等于 $\prod_{i\in I_l} \sigma_i^{v_i}$,则质询者失败并退出.

分析.假定诚实证明者提供的成员 U_j 的正确证明为 $\{\mu_j,\sigma'_j\}$,由方案的正确性,可知下列验证等式成立:

$$\hat{e}\left(\prod_{j=1}^{n} \sigma'_j, g\right) = \prod_{j=1}^{n} \hat{e}\left(u^{\mu_j} \cdot \prod_{i\in I_j} H(i)^{v_i}, g^{s_j}\right) \tag{9}$$

假设敌手给出的相对应于某个成员 U_i 的不同应答是 $\{\mu'_l,\sigma''_l\}$,而相对应于其他成员的应答是正确的证明 $\{\mu_j,\sigma'_j\}(j\in[1,n], j\neq l)$,由于伪造是成功的,因此下列等式成立:

$$\hat{e}\left(\sigma''_l \cdot \prod_{j\in[1,n], j\neq l} \sigma'_j, g\right) = \hat{e}\left(u^{\mu'_l} \cdot \prod_{i\in I_l} H(i)^{v_i}, g^{s_l}\right) \cdot \prod_{j\in[1,n], j\neq l} \hat{e}\left(u^{\mu_j} \cdot \prod_{i\in I_j} H(i)^{v_i}, g^{s_j}\right) \tag{10}$$

显然,$\mu'_l \neq \mu_l$.否则,$\sigma''_l = \sigma'_l$,这与质询者失败并退出的假设矛盾.定义 $\Delta\mu = \mu_l' - \mu_l$,构造模拟器利用敌手来攻破 CDH 实例,具体过程如下:

给定 $(g, g^{\alpha} = g^{s_l}, h) \in G_1$,模拟器的目标是计算 h^{α}.模拟器选择两个随机元素 $a,b \in Z_p^*$,并且设置 $u=g^a h^b$,并令成员 U_i 的公钥表示为 $v = g^{\alpha} = g^{s_l}$ (模拟器并不知道 s_l 的值).

为了回答敌手的随机预言查询,模拟器对质询中的每个 i 选择一个随机值 $r_i \in Z_p^*$,定义随机预言值:

$$H(i) = g^{r_i} / (g^{am_i} \cdot h^{bm_i}) \tag{11}$$

因此,可以得到

$$H(i) \cdot u^{m_i} = g^{r_i} / (g^{am_i} \cdot h^{bm_i}) \cdot u^{m_i} = g^{r_i} / (g^{am_i} \cdot h^{bm_i}) \cdot g^{am_i} \cdot h^{bm_i} = g^{r_i} \tag{12}$$

这样,模拟器可以计算 $\sigma_i = (H(i) \cdot u^{m_i})^{\alpha} = (g^{\alpha})^{r_i}$.

用等式(9)去除等式(10),可以得到 $\hat{e}(\sigma_l''/\sigma_l',g)=\hat{e}(u^{\Delta\mu},v)=\hat{e}((g^a h^b)^{\Delta\mu},v)$. 所以,

$$\hat{e}(\sigma_l''\cdot\sigma_l'^{-1}v^{-a\Delta\mu},g)=\hat{e}(h,v)^{b\Delta\mu}=\hat{e}(h^{\alpha},g)^{b\Delta\mu} \tag{13}$$

所以,可以知道 $h^{\alpha}=(\sigma_l''\cdot\sigma_l'^{-1}v^{-a\Delta\mu})^{1/(b\Delta\mu)}$. 注意到,游戏失败的概率与 $b\Delta\mu=0 \bmod p$ 概率相同.然而,$b\Delta\mu_j=0 \bmod p$ 的概率只有 $1/p$,所以,这个概率是可以忽略的.这就意味着,在游戏 1 和游戏 2 中,如果敌手获胜的概率存在不可忽略概率的不同,我们就可以构造一个模拟器利用敌手来解决 CDH 问题.

游戏 3:游戏 3 和游戏 2 相似,仅有一处不同.与先前一样,质询者仍然存储和观察质询、应答中的每个实例.如果对于某个实例,聚合的消息 μ_l 不等于期盼的 $\sum_{i\in I_l}v_i m_i$,则质询者失败并退出.

分析.假定诚实证明者提供的成员 U_j 的正确的证明为 $\{\mu_j,\sigma_j'\}$,由方案的正确性,可知下列验证等式 $\hat{e}\left(\prod_{j=1}^{n}\sigma_j',g\right)=\prod_{j=1}^{n}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right)$ 成立.假设敌手给出的相对应于某个成员 U_i 的不同应答是 $\{\mu_l',\sigma_l''\}$,而相对应于其他成员的应答是正确的证明 $\{\mu_j,\sigma_j'\}(j\in[1,n],j\neq l)$,由于伪造是成功的,因此等式

$$\hat{e}\left(\sigma_l''\cdot\prod_{j\in[1,n],j\neq l}\sigma_j',g\right)=\hat{e}\left(u^{\mu_l'}\cdot\prod_{i\in I_l}H(i)^{v_i},g^{s_l}\right)\cdot\prod_{j\in[1,n],j\neq l}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right)$$

成立.由游戏 2 中分析可知 $\sigma_l''=\sigma_l'$. 定义 $\Delta\mu=\mu_l'-\mu_l$. 构造模拟器利用敌手来攻破 DL 问题,具体过程如下:

给定 $(g,h)\in G_1$,模拟器的目标是计算出某个 x 满足 $h=g^x$.模拟器选择两个随机元素 $a,b\in Z_p^*$,并且设置 $u=g^a h^b$.

从上述分析成立的两个验证等式可以得到

$$\begin{aligned}\prod_{j=1}^{n}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right)&=\hat{e}\left(\prod_{j=1}^{n}\sigma_j',g\right)\\&=\hat{e}\left(\sigma_l''\cdot\prod_{j\in[1,n],j\neq l}\sigma_j',g\right)\\&=\hat{e}\left(u^{\mu_l'}\cdot\prod_{i\in I_l}H(i)^{v_i},g^{s_l}\right)\cdot\prod_{j\in[1,n],j\neq l}\hat{e}\left(u^{\mu_j}\cdot\prod_{i\in I_j}H(i)^{v_i},g^{s_j}\right).\end{aligned}$$

因此,可以获得

$$\hat{e}\left(u^{\mu_l}\cdot\prod_{i\in I_l}H(i)^{v_i},g^{s_l}\right)=\hat{e}\left(u^{\mu_l'}\cdot\prod_{i\in I_l}H(i)^{v_i},g^{s_l}\right)\Rightarrow 1=u^{\Delta\mu}=g^{a\Delta\mu}h^{b\Delta\mu}.$$

若 $\Delta\mu=0 \bmod p$,则 $\mu_l=\mu_l' \bmod p$,这与假设矛盾.因此解决 DL 问题:

$$h=g^{\frac{-a\Delta\mu}{b\Delta\mu}}=g^{\frac{-a}{b}},x=-\frac{a}{b}.$$

b 为 0 的概率是 $1/p$,由于 p 是一个大素数,是可以忽略的.然后,解决 DL 问题的概率为 $1-1/p$.所以,这个概率与在 G_1 中计算 DL 问题是困难的假设矛盾.

这就意味着,在游戏 2 和游戏 3 中,如果敌手获胜的概率存在不可忽略概率的不同,我们就可以构造一个模拟器利用敌手来解决 DL 问题.因此,定义的这些游戏之间仅存在可忽略的概率不同.

下面,通过构造知识提取器来提取所有质询的数据块 $m_i(i\in I,|I|=c)$,通过选择 c 次不同的系数 $v_i(i\in I,|I|=c)$质询在相同的数据块 $m_i(i\in I,|I|=c)$上,可以得到一个关于 $m_i(i\in I,|I|=c)$的独立的线性等式,知识提取器通过求解这个 c 元一次线性方程组,即可求得所有的解 $m_i(i\in I,|I|=c)$.因此,提取器可以提取整个质询数据块.这意味着如果云服务器可以通过 TPA 的审计验证,它必定真实地存储着群用户质询的完整数据. □

5 性能评估

在实验中,采用 C 语言,利用 GMP 和 PBC 函数库[25]进行实现.运行平台是具有 2.70GHz 英特尔奔腾处理器和 4GB 内存的 Linux 服务器,运行的操作系统是 Linux 系统.在实验中,我们设置基域的大小为 512 比特,在 Z_p^* 中一个元素的大小为 $|p|=160$ 比特,选择的文件大小为 20MB,每个文件被分成 1 000 000 个数据块.

5.1 审计性能

方案的计算开销主要来源于审计任务.审计任务包括 3 个阶段:审计质询的产生、审计证明的产生、审计证明的验证.见表 1,为了有效地评估方案的审计开销,选择 3 种不同数量的质询数据块进行实验.为了简单起见,假定在共享群里有 10 个用户,即 n=10.从表 1 可以知道,在审计任务的 3 个阶段中,用于产生审计质询的时间最少,验证审计证明需要花费最多的时间.此外,还可以推断出当被质询的数据块的数量增加时,审计任务中的这 3 个阶段的计算开销均增加.然而,被质询的数据块的数量越多,检查得到的结果就越精确.因此,需要在减少审计计算开销和确保数据完整性之间做一个权衡.如图 3 显示的实验结果是为了有效评估群用户数量 n 的大小对审计计算开销的影响.实验中,假设群用户数量的范围是 $n\in[2,50]$,被质询的数据块的数量 c=460.从图 3 可以看出,随着群用户数量 n 的增大,产生审计质询的时间与产生审计证明的时间基本没有变化,而验证审计证明的时间缓慢增加.因此,可以推断群用户数量 n 的大小对审计计算开销影响不大.

Table 1 The computational overheads of the diffirent numbers of challenged data blocks (n=10)(s)

表 1 不同数量的质询数据块下的计算开销(n=10) (s)

	c=300	c=460	c=1000
质询产生	0.13	0.20s	0.43s
证明产生	1.18	1.79s	3.84s
证明验证	3.89	5.93s	12.61s

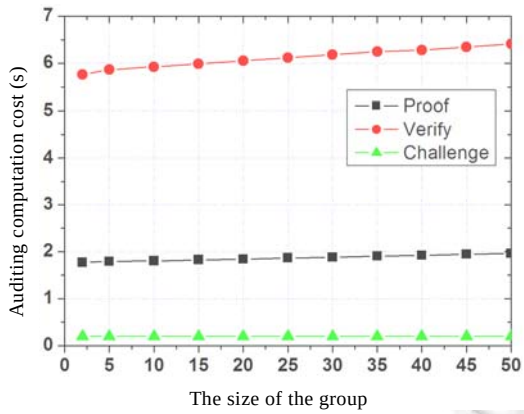

Fig.3 Impact of the number n of group users on auditing computational overheads

图 3 群用户数量 n 的大小对审计计算开销的影响

5.2 私钥分发和私钥恢复的性能

为了讨论门限值 t 值和群用户数量 n 的大小对私钥分发和私钥恢复时间的影响,进行如下 4 个实验.其中两个分别是在私钥分发和私钥恢复阶段,门限值 t 值不变(t=6),群用户数量 n 的大小改变,如图 4 和图 5 所示.另外两个实验是在上述两个阶段中,门限值 t 值改变,群用户数量 n 的大小不变(n=55),如图 6 和图 7 所示.

在实验中,为了显示私钥恢复时间是可接收的,我们选择一个更大的阶数 p,其长度为 1 024 比特,在这个条件下可以远远确保群中离散对数是难解的.通过比较图 4 和图 6,可以知道在私钥分发阶段,无论门限值 t 值增大还是群用户数量 n 增大,私钥分发所用的时间均会增大.通过比较图 5 和图 7,可以知道在私钥恢复阶段,当门限值 t 不变(t=6),群用户数量 n 增大时,私钥恢复的时间不改变.但是当群用户数量 n 不变(n=55),门限值 t 值增大时,私钥恢复时间增大.而且随着 t 增大,私钥恢复时间增长的幅度更明显.因此,可以得出私钥分发时间受门限值 t 值和群用户数量 n 大小的影响,而私钥恢复时间只与门限值 t 值有关,与群用户数量 n 大小无关的结论.从图 7 可以看出,当 t=20 时,私钥恢复的时间为 1.783s,这个时间并不长,而且在通常情况下是可以达到安全要求的.即使 t 达到 50 时,私钥恢复时间仍然可以接受.

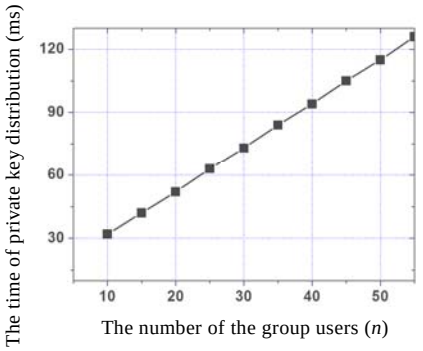

Fig.4 Impact of the number *n* of group users on private key generation

图 4 群用户数量 *n* 的大小对私钥分发时间的影响

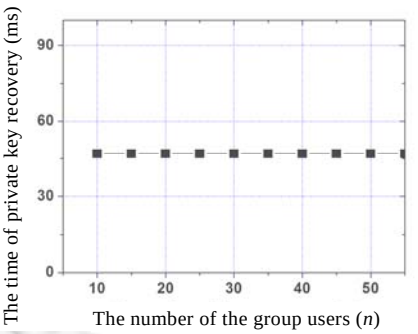

Fig.5 Impact of the number *n* of group users on private key recovery

图 5 群用户数量 *n* 的大小对私钥恢复时间的影响

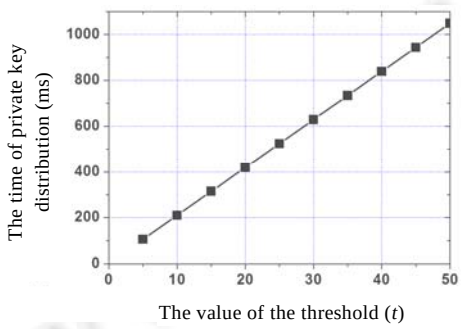

Fig.6 Impact of the value *t* of threshold on private key generation

图 6 门限值 *t* 对私钥分发时间的影响

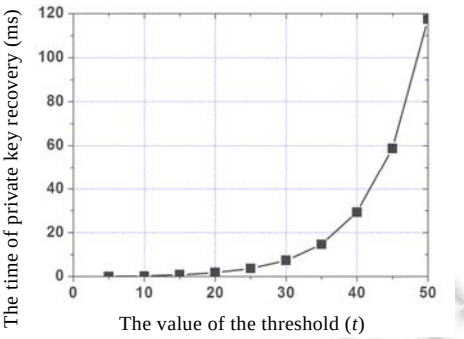

Fig.7 Impact of the value *t* of threshold on private key recovery

图 7 门限值 *t* 对私钥恢复时间的影响

6 进一步讨论

6.1 如何减少私钥恢复的运算量

可以通过以下修改来减少私钥恢复的运算量:在私钥恢复阶段,为了帮助私钥不可用的群用户 U_m 恢复私钥,群用户 $U_{l_i}(i=1,2,...,t)$ 随机选择 $a_{l_i1},a_{l_i2},...,a_{l_i(t-1)}\in Z_p^*$, 计算 $a_{l_i0}=-\sum_{k=1}^{t-1}a_{l_ik}m^k$, 令 $f_{l_i}(x)=\sum_{k=0}^{t-1}a_{l_ik}x^k$. 因此, $u_{l_im}=\sum_{k=0}^{t-1}a_{l_ik}m^k=f_{l_i}(m)=0$, 进而可以得到 $u_m=\sum_{i=1}^{t}u_{l_im}=0$. 在原始的方案中,群用户 U_m 可以通过计算等式(8) $s_m=\sum_{j=1}^{t}C_{Bl_j}(m)s'_{l_j}-u_m(\mathrm{mod}\ p)$ 来计算他的私钥 s_m.但是,还需要计算用于解盲化的 u_m 值.然而,修改方案后, $s_m=\sum_{j=1}^{t}C_{Bl_j}(m)s'_{l_j}(\mathrm{mod}\ p)$, 与等式(8)相比,不需要计算 u_m 的值.这将减少私钥恢复的运算量,减少计算开销,提高方案的效率.

6.2 如何选择*t*的值

从提高性能的角度考虑,门限值 *t* 减小,方案在私钥分发和私钥恢复时的性能越高,但与此同时,其安全性会降低.为了提高安全性,可以增大 *t* 的值,*t* 值越大,安全性越高.当 *t* 大到一定程度,即 *t*=*n*–1 时,安全性最高.这时,除非参与私钥恢复的 *n*–1 个用户都是不诚实的,用户私钥才可能被泄漏.然而从图 7 可知,当门限值 *t* 增大时,效率会随之降低.因此,需要在效率和安全性之间做一个权衡,根据实际场景的需求来选择 *t* 的值.

7 结 论

本文解决了在云存储数据完整性检测中群用户私钥不可用的问题,提出了具有私钥可恢复能力的共享数据云存储完整性检测的方案.在提出的方案中,当一个群用户私钥不可用时,可以通过群里的t个或者多于t个用户帮助他恢复私钥.为了保护私钥隐私和提高私钥恢复安全性,在私钥恢复阶段,设计了随机遮掩技术去盲化参与私钥恢复用户的私钥,使得参与私钥恢复用户的私钥不被泄漏.此外,使用可验证技术去检查参与私钥恢复的群用户是否是诚实的.安全性分析和实验结果说明了我们方案是安全、高效的.如何设计其他有效的机制来实现具有私钥可恢复能力的共享数据云存储完整性检测是值得进一步解决的问题.

References:

[1] Ren K, Wang C, Wang Q, Yu J. Research progress of data security in cloud computing. In: China Cryptography Development Report 2013. Beijing: China Quality Inspection Press, 2014. 71−94 (in Chinese).

[2] Armbrust M, Fox A, Griffith R, Joseph A. D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM, 2010,53(4):50−58. [doi: 10.1145/1721654.1721672]

[3] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, Song D. Provable data possession at untrusted stores. In: Proc. of the ACM CCS. 2007. 598−609. [doi: 10.1145/1315245.1315318]

[4] Wang BY, Li BC, Li H. Oruta: Privacy-Preserving public auditing for shared data in the cloud. In: Proc. of the IEEE Cloud. 2012. 295−302. [doi: 10.1109/CLOUD.2012.46]

[5] Ren K, Wang C, Wang Q. Security challenges for the public cloud. IEEE Internet Computing, 2012,16(1):69−73. [doi: 10.1109/MIC.2012.14]

[6] Song D, Shi E, Fischer I, Shankar U. Cloud data protection for the masses. IEEE Computer, 2012,45(1):39−45. [doi: 10.1109/MC.2012.1]

[7] Wang C, Chow SMS, Wang Q, Ren K, Lou WJ. Privacy-Preserving public auditing for secure cloud storage. IEEE Trans. on Computers, 2013,62(2):362−375. [doi: 10.1109/TC.2011.245]

[8] Wang Q, Wang C, Ren K, Lou WJ, Li J. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. on Parallel and Distributed Systems, 2011,22(5):847−859. [doi: 10.1109/TPDS.2010.183]

[9] Singh R, Kumar S, Agrahari SK. Ensuring data storage security in cloud computing. Int'l Journal of Engineering and Computer Science, 2012,2319−7242.

[10] Wang Q, Wang C, Li J, Ren K, Lou WJ. Enabling public verifiability and data dynamic for storage security in cloud computing. In: Proc. of the ESORICS. 2009. 355−370. [doi: 10.1007/978-3-642-04444-1_22]

[11] Wang C, Wang Q, Ren K, Lou WJ. Privacy-Preserving public auditing for data storage security in cloud computing. In: Proc. of the IEEE INFOCOM. 2010. 1−9. [doi: 10.1109/INFCOM.2010.5462173]

[12] Chen B, Curtmola R, Ateniese G, Burns R. Remote data checking for network coding-based distributed storage systems. In: Proc. of the ACM CCSW. 2010. 31−42. [doi: 10.1145/1866835.1866842]

[13] Yu J, Ren K, Wang C, Varadharajan V. Enabling cloud storage auditing with key-exposure resistance. IEEE Trans. on Information Forensics and Security, 2015,10(6):1167−1179. [doi: 10.1109/TIFS.2015.2400425]

[14] Yang GY, Yu J, Shen WT, Su QQ. Fu ZJ, Hao R. Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. Journal of Systems and Software, 2016,113:130−139. [doi: 10.1016/j.jss.2015.11.044]

[15] Wang BY, Li H, Li M. Privacy-Preserving public auditing for shared cloud data supporting group dynamics. In: Proc. of the 2013 IEEE Int'l Conf. on Communications (ICC). 2013. 1946−1950. [doi: 10.1109/ICC.2013.6654808]

[16] Wang BY, Li BC, Li H. Panda: Public auditing for shared data with efficient user revocation in the cloud. IEEE Trans. on Services Computing, 2015,8(1):92−106. [doi: 10.1109/TSC.2013.2295611]

[17] Yuan JW, Yu SC. Efficient public integrity checking for cloud data sharing with multi-user modification. In: Proc. of the IEEE INFOCOM. 2014. 2121−2129.

[18] Hard Drive Data Corruption-iAfrica. http://users.iafrica.com/c/cq/cquirke/baddata.htm [doi: 10.1109/INFOCOM.2014.6848154]

[19] Hard Disk Drive Failure. https://en.wikipedia.org/wiki/Hard_disk_drive_failure

[20] RSA Laboratories-How should I store my private key. http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/store-private-key.htm

[21] What is Key Escrow?-Definition from Techopedia. http://www.techopedia.com/definition/3997/key-escrow

[22] Shamir A. How to share a secret. Communications of the ACM, 1979,22(11):612−613. [doi: 10.1145/359168.359176]

[23] Blakley GR. Safeguarding cryptographic keys. IEEE Computer Society, 1979. 313−317.

[24] Shacham H, Waters B. Compact proofs of retrievability. Journal of Cryptology, 2013,26(3):442−483. [doi: 10.1007/s00145-012-9129-2]

[25] Lynn B. The pairing-based cryptographic library. 2015. http://crypto.Stanford.edu/pbc/

附中文参考文献:

[1] 任奎,王聪,王骞,于佳.云计算中数据安全的研究进展.见:中国密码学发展报告 2013.北京:中国质检出版社,2014.71−94.

沈文婷(1991－),女,广西南宁人,硕士生,主要研究领域为大数据,云存储安全.

于佳(1976－),男,博士,教授,主要研究领域为密码学,云计算安全,大数据安全,网络安全.

杨光洋(1989－),男,硕士生,主要研究领域为大数据,云存储安全.

程相国(1969－),男,博士,教授,主要研究领域为密码学.

郝蓉(1976－),女,讲师,主要研究领域为密码学,网络安全.