物联网环境下云数据存储安全及隐私保护策略研究

何 明[1] 陈国华[1,2] 梁文辉[1,3] 赖海光[1] 凌 晨[4]

(解放军理工大学指挥自动化学院 南京 210007)[1] (中国人民解放军 65655 部队 赤峰 024000)[2]
(中国人民解放军 61345 部队 西安 710004)[3] (总后后勤科学研究所后勤信息化研究中心 北京 100071)[4]

摘 要 物联网依托云计算强大的数据处理能力实现信息智能,而目前云计算对数据和服务的管理并不值得用户完全信赖。针对物联网环境下云数据安全性问题,在云计算中为了保证用户数据的准确性和隐私性,提出了一种物联网环境下云数据存储安全及隐私保护策略。实验结果表明该方案有效、灵活,且能抵御 Byzantine 失效、恶意修改数据甚至是服务器共谋攻击。

关键词 物联网,云计算,分布式数据库,隐私保护

中图法分类号 TP311 **文献标识码** A

Cloud Data Storage Security and Privacy Protection Policies under IoT Environment

HE Ming[1] CHEN Guo-hua[1,2] LIANG Wen-hui[1,3] LAI Hai-guang[1] LING Chen[4]

(Institute of Command Automation, PLA Science and Technology University, Nanjing 210007, China)[1]
(65655 Armies, PLA, Chifeng 024000, China)[2] (61345 Armies, PLA, Xi'an 710004, China)[3]
(Logistics Science Research Institute, Beijing 100071, China)[4]

Abstract The Internet of Things implements information intelligence relying on the powerful data processing capability of cloud computing. However, it is not worthy of user trust that cloud computing manages data and services. According to cloud data security issues in the environment of the IoT, the cloud data storage security and privacy policies were raised, in order to ensure the accuracy and privacy of user data in the cloud. The experimental results show that the program is effective and flexible, and can resist the Byzantine failure, malicious modification of data, and even server collusion attack.

Keywords IoT, Cloud computing, Distributed database, Privacy protection

1 引言

随着物联网的发展,物联网、互联网和 3G 手机的不断结合,未来物联网智能化程度将越来越高。同时,物联网在安全性方面的影响也不容忽视。物联网信号的窃取将直接影响整个物联网的信息安全。从信息安全和隐私保护的角度来说,物联网终端的广泛引入在提供更丰富信息的同时,也增加了暴露这些信息的危险。

在物联网环境下,云计算可作为数据处理层中完成信息智能处理的计算模型,使用户可以在任何地方通过与其连接的设备访问应用程序;数据库中的数据可以存储在云上(一个服务器集合中),而不是放在一个固定的物理位置。应用程序位于可大规模伸缩的数据中心,计算资源可在其中动态部署并进行共享,以便实现显著的规模经济。有了云计算,用户不再需要部署计算能力很强的客户端,而是直接从"云"里(服务器端)获得计算能力。由于不需要直接面对复杂的硬件管理,因此对用户来说,把数据转移到云中更为便利。虽然这些在线服务需要巨大的存储空间和计算资源,但计算平台的转移仍然减轻了本地计算机数据维护的责任,云服务提供商数据的实效性和准确性也面临挑战。

云计算将不可避免地面临许多安全威胁,而数据安全性直接制约着服务质量。首先,由于云计算中用户不直接控制数据,因此传统的通过加密来保护数据安全性的措施无法直接应用。云中存储数据的安全性核查需要在无法掌握全局数据的情况下进行。其次,云计算不仅仅是一个第三方数据仓库,其中存储的数据更新频率很快,包括插入、删除、修改、添加以及重排序等。在动态更新的情况下保证数据的准确性至关重要,然而传统技术无法应对动态性,必须寻找新的方法。最后,云计算的调度由同步、协作、分布式的用户掌握。单个用户数据冗余存储在多个物理空间,以此减低数据完整性的威胁,因此数据准确性核查的分布式协议是实际中云数据存储安全和健全至关重要的因素。

国内外学者就以上云计算的安全问题进行了相关研究[1,2],在一定程度上保证了数据的准确性,但也存在某些局限性,其主要体现在仅着眼于单个服务,没有考虑数据的动态操作,没有从根本上解决云数据存储面临的安全威胁。例如:

到稿日期:2011-07-29 返修日期:2011-10-23 本文受国家自然科学基金(60974086),江苏省自然科学基金(BK2010132)资助。

何 明(1978—),博士后,副教授,硕士生导师,主要研究方向为物联网、云计算、信息安全,E-mail:blue_horse@126.com;陈国华(1986—),硕士生,主要研究方向为物联网、信息安全。

Juels 等[3]提出一种正式的"可恢复证明"(POR)模型来保证远程数据的完整性,该方案采用点验证和纠错码来保证服务系统中文件的权属和可恢复性。Shacham 等[4]实现了上述模型并构建了基于随机线性函数的同型认证,可以不限制提问的次数且仅需要很少交互信息。Bowers 等[5]在前两者的基础上,提出改进的 POR 框架。然而所有这些研究都只涉及静态数据,其效率主要依赖于用户远程存储前的预处理过程。Ateniese 等[6]提出"可证明数据权属"(PDP)模型来保障不可信存储中文件的所有权,该模型利用基于公钥的同型标签审计数据文件,然而预处理过程耗费太大。在其后的研究中[7]提出了仅使用对称密钥加密法的 PDP,它减少了预处理的耗费并支持存储文件更新、删除和添加。但该改进方法仅研究了单个服务器的情况,并没有解决小规模数据损坏问题,并且分布式情况和错误数据恢复也没有研究。Schwarz 等[8]提出跨多分布式服务器保持文件完整的方案,其利用了纠错码和块级文件完整性检验,然而方案仅考虑了静态数据文件,也没有解决数据错误定位问题。Lillibrige 等[9]引入了 P2P 备份方案,数据文件块利用$(m+k,k)$纠错码离散到 $m+k$ 个终端,终端可以从其备份终端随机抽取数据块并依据块上的哈希密钥验证完整性,该方案可以发现终端的数据丢失,但不能保证所有数据未经改动。

这些方案没有考虑数据的动态操作而存在很大的局限性,为保证用户数据的准确性,本文提出了高效、灵活的分布式方案,以支持动态数据操作,包括块更新、删除和添加。其利用文件分发预过程的纠错码充当冗余奇偶向量,以此保证数据可信赖性,利用同型令牌和纠错码分布式验证,实现了存储准确性和错误定位的结合,即在跨分布式服务器存储准确性验证中发现数据损坏以后,基本可识别失误的服务器。通过详细的安全性和效率分析,该方案不仅高效,而且还可以抵御 Byzantine 失效和恶意数据篡改攻击以及服务器共谋攻击。

2 理论基础

针对云数据存储面临的主要安全威胁,构建云数据存储典型的网络架构(如图 1 所示)。此架构由 3 个网络实体构成:1)用户。需要将数据存储在云中,并依赖云进行数据计算的消费者或组织;2)云服务提供商(CSP)。拥有巨量的资源,对云存储服务器有专业的知识和管理能力,掌控云计算系统;3)第三方审计(TPA)。第三方审计是可选择的,拥有用户所没有的专业知识和能力,可以根据用户请求评估云存储服务的风险。

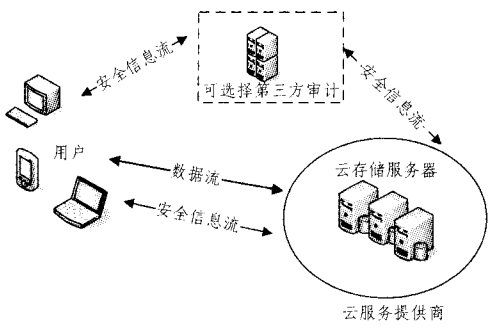

图 1 云数据存储结构图

在云存储中,用户通过云服务提供商将数据存储在一系列以同步、协调、分布式运行的云服务器中。随着用户数据增长,前向纠错或服务器失效校验码会产生数据冗余。用户通过云服务提供商协调云服务器存储或寻回其数据。某些情况下,用户需要对其数据执行块级别的操作,通常认为块操作包含块更新、删除、插入、添加。

由于用户不再将数据存储在本地,因此数据存储和管理的准确性就十分必要,用户需要用安全措施来持续保证数据的准确性,尤其是在没有本地备份的情况下。在没有时间、能力或者资源检测其数据时,用户可以委托其信任的第三方审计 TPA 代其完成。在模型中,假定每个云服务器与用户间点对点的通信渠道是经过可靠认证的(在实际中可能无法满足)。

• F—存储的数据文件。假定 F 可以描述为 m 个大小相同的数据向量组成的矩阵,每个向量由 L 块组成。数据块可以表示为伽罗瓦域(Galois Field)中的元素 $GF(2^p)$,p=8 or 16。

• A—Reed-Solomon 编码使用的散列矩阵。

• G—编码文件矩阵,包含 $n=m+k$ 个向量,每个向量包含 L 个块。

• $f_{key}(\cdot)$—伪随机函数(PRF),定义为 $f:\{0,1\}^* \times key \rightarrow GF(2^p)$。

• $\phi_{key}(\cdot)$—伪随机序列(PRP),定义为 $\phi:\{0,1\}^{\log_2(l)} \times key \rightarrow \{0,1\}^{\log_2(l)}$。

• ver—随着数据块索引变化的版本号,记录块修改的时间,所有数据块的 ver 初始值为 0。

• s_{ij}^{ver}—PRF 的种子,依赖于文件名、数据块索引 i、服务器位置 j 以及可选的块版本号 ver。

3 云数据存储安全

在云数据存储系统中,用户将数据存储在云中,本地则不再保存备份,因此必须保证存储在分布式云服务器中的数据文件的准确性和可用性。其关键问题是如何有效、及时地发现由 Byzantine 失效或服务器故障引起的未经授权的数据的修改或损毁,另外在分布式环境中如何确定损毁数据位于哪个服务器也很棘手,而这却是快速恢复损毁数据的第一步。

3.1 文件分发准备

众所周知,擦除纠错码可以纠正分布式系统中的多倍错误,在云数据存储中,利用该方法将数据文件 F 冗余散列到 $n=m+k$个分布式服务器上。利用 $A(m+k,k)$Reed-Solomon 擦除纠错码,从 $m+k$ 个数据向量中任选 m 个,产生 k 个冗余向量。将 $m+k$ 个向量分配到不同的服务器上,那么 $m+k$ 个服务器中的任何 k 个失效,原始数据文件不会丢失的概率为 k/m。为了使原始数据能够高效有序地输入输出,文件布局需是规则的,即 m 个没有改变的数据文件向量和 k 个奇偶向量分配到 $m+k$ 个不同的服务器。

令 $F=(F_1,F_2,\cdots,F_m)$,$F_i=(f_{1i},f_{2i},\cdots,f_{li})^{\mathrm{T}}(i\in\{1,\cdots,m\})$,其中 $l\leqslant 2^p-1$。所有块都有 $GF(2^p)$的元素。奇偶向量布局规则由信息散列矩阵 A 通过一个 $m\times(m+k)$范得蒙矩阵实现。

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_m & \beta_{m+1} & \cdots & \beta_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \beta_1^{m-1} & \beta_2^{m-1} & \cdots & \beta_m^{m-1} & \beta_{m+1}^{m-1} & \cdots & \beta_n^{m-1} \end{pmatrix}$$

式中,$\beta_j(j\in\{1,\cdots,n\})$是从 $GF(2^p)$随机抽取的离散元。

经过一连串的初级行变换，期望矩阵 A 可以写成：

$$A=(I|P)=\begin{pmatrix}1&0&\cdots&0&p_{11}&p_{12}&\cdots&p_{1k}\\0&1&\cdots&0&p_{21}&p_{22}&\cdots&p_{2k}\\\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\ddots&\ddots\\0&0&\cdots&1&p_{m1}&p_{m2}&\cdots&p_{mk}\end{pmatrix}$$

式中，I 是一个 $m\times m$ 识别矩阵，P 是 $m\times k$ 的生成矩阵。注意，A 来自范得蒙矩阵，因此 $m+k$ 中的任意 m 列都是可逆矩阵。

由 F 乘 A，可得编码文件：

$$G=F\cdot A=(G^{(1)},G^{(2)},\cdots G^{(m)},G^{(m+1)},\cdots G^{(n)})$$
$$=(F_1,F_2,\cdots,F_m,G^{(m+1)},\cdots,G^{(n)})$$

式中，$G^{(j)}=(g_1^{(j)},g_2^{(j)},\cdots g_l^{(j)})^{\mathrm{T}}(j\in\{1,\cdots,n\})$。该乘法还原了原始数据文件向量，剩余部分$(G^{(M+1)},\cdots,G^{(n)})$是 k 个 F 衍生的奇偶向量。

3.2 挑战令牌预计算

方案依靠预计算检验令牌来保证存储的准确性和数据错误定位的及时性，其主要思想如下：在文件分配以前，用户为每个向量 $G^{(j)}(j\in\{1,\cdots,n\})$计算一个一定数量的检验令牌，每个令牌包含数据块的随机子集。用户需要检验云中数据准确性时，则以一组随机生成的块索引挑战云服务器；接到挑战后，每个云服务器为块签名，然后反馈给用户；用户将签名和预计算的对应令牌比对。与此同时所有服务器计算块索引的同一子集，完整性检验响应值编码的有效与否由矩阵 P 决定。

如果用户需要挑战云服务器 t 次来确认数据存储的准确性，那么必须为每个 $G^{(j)}(j\in\{1,\cdots,n\})$预计算 t 个检验令牌，这需要用到伪随机函数 $f(\cdot)$和伪随机序列 $\phi(\cdot)$，以及挑战钥 k_{chal}和主序列钥 K_{PRP}。如果需要生成服务器 j 的第 i 个令牌，用户需要：

1)生成随机挑战值 α_i 和序列钥 K_{PRP}，$\alpha_i=f_{k_{chal}}(i)$。

2)计算 r 个随机抽取的索引：$\{I_q\in[1,\cdots,l]1\leqslant q\leqslant r\}$，其中 $I_q=\phi_{k_{prp}^{(t)}}(q)$。

3)计算令牌：$v_i^{(j)}=\sum_{q=1}^{r}\alpha_i^q*G^{(j)}[I_q]$，其中 $G^{(j)}[I_q]=g_{I_q}^{(j)}$。

令牌生成以后，用户可以将其存储在本地，也可以将其以加密的形式存储到云服务器，为了叙述简便，本文假定将其存储在本地。

3.3 准确性校验和错误定位

在存储系统中，错误定位是排错的先决条件，然而目前的研究方案没有深入考虑这个问题，本文方案将准确性校验和错误定位集成到挑战-应答协议中：服务器的应答值不仅仅可以判定分布式存储的准确性，还包含潜在错误位置信息。

因为所有服务器操作数都是索引的相同子集，所以 r 行$(R_i^{(1)},\cdots,R_i^{(n)})$线性组合必定是编码文件矩阵的码字。如果以上等式成立，挑战即结束，否则说明在 r 行中存在数据错误。

存储中一旦发现不一致，就可以根据预计算的验证令牌进一步确定错误数据的位置。

3.4 数据恢复和错误排除

因为文件矩阵的布局是规则的，用户可以从前 m 个服务器下载数据向量来重构原始文件，在此假设前 m 个服务器应答正确。监测方案是基于随机污点的检测，因此存储正确性的概率为 1。适当调整系统参数(比如 r,l,t)做足够多次数的检验，可以大概率地恢复文件。另一方面，无论何时发现数据错误，通过算出的令牌与应答值的比较找出全部失误的服务器，其概率是非常大的。总之，用户可以要求服务器返回挑战中指定的 r 行数据，纠错后重构正确数据块，最多还可以识别 k 个失误的服务器。最新恢复的数据块再分配到失误服务器，又可以保证数据的正确性。

4 云数据隐私保护策略

云计算的核心问题是基础设施的共享性，即数据远程存储操作带来的风险，因为不同用户共享的虚拟设备不断增多，存储在云中的个人机密敏感信息变得异常重要。云计算的另一个特点为环境是动态的，交互服务更是增加了传统电子商务的动态性。服务的潜在客户是不断变化的，服务商随时可以改变服务供应，如此个人敏感信息就可能跨越边界流动，必须对数据进行适当的保护和限制，在云计算的速度与数据安全之间需要进行折中。这就引入了一个问题，数据安全是客户最为关注的，尤其是财务和健康信息，云计算环境的快速变化对服务商保持安全标准的能力以及服务连贯性提出挑战。云计算随时可能出现新的应用，例如“即时打印”不仅提供打印服务，还可能附加存储服务，该过程比以前需要多个公司的联合服务更为便捷，其中的服务对安全和隐私保护要求也不尽相同。另一方面，服务必须收集存储个人敏感信息，这些信息在服务商边界流动。此外随着云计算的发展，新的风险会不断增加。

4.1 数据和软件的传输协议

软件配置分为 3 大部分：隐私标签、受保护部分和非保护部分。

隐私标签由以下元素组成：

• CID：用户注册云隐私服务时，由云端提供的唯一标识用户的 ID 号；

• SID：由用户提供的唯一标识其软件应用的 ID 号；

• $E(PU_{CID},K_{SID})$：K_{SID} 是由云用户随机生成的一个对称密钥，用来加密保护软件的 S_{SID} 部分，$E(PU_{CID},K_{SID})$表示用用户的公钥 PU_{CID}对 K_{SID}进行加密。这个加密过程确保除了加密协处理器中的 RP 进程外，没有别的实体可以提取出 K_{SID}；

• DS：软件包的数字签名。可以象征性地表示为：$DS=Sign(PR_{CID},CID\|SID\|E(PU_{CID},K_{SID})\|Timestamp\|Identification\|S_{SID}\|S'_{SID})$，其中 ‖ 是串联符号；

• $Timestamp$：时间戳，代表数字签名创建的时间，用来防止重放攻击；

• $Identification$：软件应用的一般信息，比如软件名字、版本等；

• $E(K_{SID},S_{SID})$：使用密钥 K_{SID}对软件的受保护部分 S_{SID}进行加密。

基于以上的软件配置，用户构建软件包并上传到云端；接收到软件报文后，云端使用数字签名 DS 来验证机密性和完整性，然后存储到云中。

向云端传输数据，用户需要执行以下的协议步骤：

(1)按照 3 种隐私类别对数据进行分类；

(2)如果隐私类别是 NP 或 PTP，客户端直接发送数据到云端。对于 PTP 类数据，应该通过 SSL 加密发送。接收到数

据后，云端将其存储到相应的存储池中；

(3)对于 PNTP 类数据，用户到其注册的加密协处理器上执行 Diffie-Hellman 密钥管理协议，得到一个共享的密钥 K_{CID}。用户使用 K_{CID} 对数据进行加密并发送给云端，再存储到 PNTP 存储池中。

4.2 软件执行和数据处理协议

该协议描述了加密协处理器和主服务器安全执行用户软件的步骤，介绍了在云中处理用户敏感数据时，隐私增强机制是如何保证隐私的。协议步骤如下：

(1)主服务器加载非保护软件部分，将软件包(保护部分)发送给加密协处理器；

(2)加密协处理器上的 RP 进程读取软件的隐私标签，对 DS 进行验证，以确保软件包的真实性和完整性。另外，RP 还检查时间戳的有效性，使用用户私钥对 $E(PU_{CID}, K_{SID})$ 进行解密提取出 K_{SID} 密钥；

(3)RP 使用 K_{SID} 密钥对加过密的软件部分 $E(K_{SID}, S_{SID})$ 进行解密，然后在协处理器的地址空间执行；

(4)RP 进程通过(CID,SID)标识来存储 K_{SID}，这样避免了下次加载同样的软件时使用公钥解密 K_{SID} 的昂贵开销；

(5) 非保护部分的程序只能处理 NP 或 PTP 隐私类数据(事实上这部分程序也无法访问得到 PNTP 数据)。如果请求的数据属于 NP 存储池，则直接从云存储中心加载数据；如果请求的数据属于 PTP 存储池，则云端需要对数据进行解密然后加载。保护部分的程序其主要职责是处理 PNTP 数据(技术上也能处理 NP 或 PTP 隐私类数据)。PNTP 数据由用户的 K_{CID} 密钥进行加密，而 K_{CID} 密钥被安全存储在加密协处理器中。为访问到明文数据，受保护部分程序向 RP 进程提出申请，拥有 K_{CID} 访问权的 RP 进程加载加密数据，然后进行解密和检查完整性，将数据提供给受保护部分程序。

软件的输出结果也分为 NP、PTP 和 PNTP 3 个隐私类。NP 类数据生成的结果存储到 NP 存储池中(对于受保护部分程序产生的结果，应先递交给 RP，再由 RP 发送到 NP 存储池)；PTP 类数据与受保护部分程序生成的结果输出，由 RP 转交给主服务器的加密进程，加密进程再使用云端密钥对结果进行加密后，将其存储到 PTP 存储池中；而 PNTP 类数据生成的结果输出，由 RP 使用 K_{CID} 进行加密存储到 PNTP 存储池中。

5 性能分析

5.1 安全强度分析及性能评估

(1)数据篡改防护概率

设服务器对每个校验的特定行计算令牌预期，可以发现选定行上的简单策略减少了服务器上的计算，同时提高了预防数据损坏的概率。

设 n_c 个服务器是因各种原因失误的，在以下的分析中，不对 n_c 进行限制，即只要 $n_c \leqslant n$ 即可。设对手篡改了编码文件矩阵 L 行中的 z 行，令 r 是用户挑战中要求的行数，X 是不相关的随机变量，表示用户挑战要求数据中被对手篡改的行数。首先分析用户要求数据中至少一行是对手修改数据的概率：

$$P_m^r = 1 - P\{X=0\} = 1 - \prod_{i=0}^{r-1}\left(1 - \min\left\{\frac{z}{l-i}, 1\right\}\right) \geqslant 1 - \left(\frac{l-z}{l}\right)^r$$

如果第 i 次校验过程中 r 行中全部没有被删除或篡改，则表示对手已逃过了检验。

下一步研究漏检概率，即 r 行中的数据块被篡改，但检验等式仍成立的概率。设第 i 次挑战中数据存储服务器应答 $R_i^{(1)}, \cdots, R_i^{(n)}$ 中的任一应答值 $R_i^{(j)}$，校验服务器的应答 $R^{(m+1)}, \cdots, R^{(n)}$ 的数目是 $k=n-m$。

根据以上推理，跨所有存储服务的数据篡改发现概率为 $P_d = P_m^r \cdot (1 - P_f^r)$。图 2 是 P_d 关于 L, r, z 的曲线，其它参数 $p=8, n_c=10, k=5$。可以看出，即使仅有少部分数据遭到损坏，但只需要挑战很少次数就可以以大概率地发现数据的篡改，例如在 $z=1\%l$ 时每个令牌仅需要覆盖 460 个索引。

(2)失误服务器发现概率

前面已经讨论了如果对手篡改数据存储服务器中的数据块，我们的方案可以很容易地发现攻击。发现数据篡改的同时，用户还可以比较应答值 $R_i^{(j)}$ 和预存的令牌 $v_i^{(j)} j \in \{1, \cdots, n\}$，从而确定发生失误的服务器。错误定位或者失误服务器鉴别的概率同样可以算出，即简单校验匹配概率和非错检概率的乘积。很明显匹配概率为 $\hat{P}_m^r = 1 - \prod_{i=0}^{r-1}\left(1 - \min\left\{\frac{\hat{z}}{l-i}, 1\right\}\right)$，其中，$\hat{z} \leqslant z$。

继续考虑至少 $\hat{z}$ 块中有一个被篡改时 $R_i^{(j)} = v_i^{(j)}$ 的错检概率，$GF(2^p)$ 中两个不同数据向量计算令牌不一致的概率为 $\hat{P}_f^r = 2^{-p}$，因此失误服务器鉴别概率为 $\hat{P}_d = \hat{P}_m^r \cdot (1 - \hat{P}_f^r)$。分析鉴别发现概率时，如果 $z=1\%l$，则每个令牌覆盖 450 个索引，失误服务器鉴别概率至少为 98%。

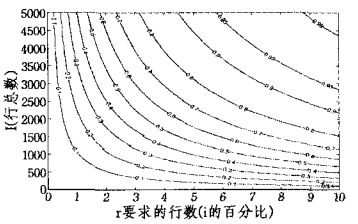

图 2 数据篡改的概率分布

5.2 本策略优点

为确保云数据存储的安全性和可靠性，根据对手模型，在数据动态核实和操作的条件下，本方案和国外最新或成熟的方案进行比较，具有以下特点：①存储正确性：保证用户数据恰当、完整、永久地存储在云中；②错误快速定位：发现数据损毁以后快速定位故障服务器；③动态数据支持：用户修改、删除、添加云中数据时不能降低存储正确性；④可靠性：加强抵御 Byzantine 失效、恶意篡改数据、服务器共谋攻击的能力，降低数据错误或服务器失效带来的影响；⑤占用资源少：将用户用于存储校验的开销降到最低。

结束语 本文研究了云数据存储中的数据安全性问题，以及云安全中数据隐私保护的策略。在一定程度上确保了云数据的准确性及隐私性。相对于传统的基于文件复制的分配技术，此策略降低了需要交换的信息量和数据存储量。利用同型数据分布式检测令牌，保证了数据的准确性并实现了错误定位，在存储数据准确性核查中发现数据损毁时，也可以确定数据损毁位置，即失误服务器鉴定。

物联网环境下的云计算数据安全性研究极其重要且充满

(下转第 90 页)

结束语 本文以文献[10]为基础，提出了一种 OFDMA 协同通信系统保证用户间公平性的子载波和功率联合分配算法。首先，在平均功率下进行子载波分配，保证每个用户获取最小需求速率的同时兼顾用户间的公平性。其次，在子载波分配结束后对中继站剩余功率进行再分配，进一步提升系统容量。仿真结果表明，本文提出的子载波、功率联合分配算法在提升系统容量的同时，能够公平地将系统资源分配给用户。中继站剩余功率分配以最大化系统容量为目标，对用户间的公平性有一定影响，但中继站发射功率调整对系统容量影响较弱。仿真结果表明，联合分配算法依然保证了用户间的公平性。若中继站发射功率较大，则运行本文提出的子载波、功率联合分配算法后，系统容量会有明显上升，但用户间资源分配的公平性也会受到一定程度的影响。

参 考 文 献

[1] 沈嘉. B3G 无线通信技术的发展趋势[J]. 现代电信科技，2007，37(9)：5-8

[2] Zhao Yi，Adve R，Lim T J. Improving Amplify and Forward Relay Networks：Optimal Power Allocation Versus Selection[J]. IEEE Transactions on Wireless Communications，2007，6(8)：3114-3123

[3] Zhang Xing，Chen Shu-ping，Wang Wen-bo. Multi-user Radio Resource Allocation for Multi-service Transmission in OFDMA-based Cooperative Relay Networks[J]. EURASIP on Wireless Communications and Networking，2009，11(1)：1-14

[4] Amin O，Uysal M. Optimal Bit and Power Loading for Amplify-and-Forward Cooperative OFDM Systems[J]. IEEE Transactions on Wireless Communications，2011，10(3)：772-781

[5] Pan Yu-wen，Nix A，Beach M. Resource Allocation Techniques for OFDMA-based Decode-and-Forward Relaying Networks[C]// Vehicular Technology Conference. 2008：1717-1721

[6] Jee L，Wang H，Seo W，et al. QoS-Guaranteed Transmission Mode Selection for Efficient Resource Utilization in Multi-hop Cellular Networks[J]. IEEE Transactions on Wireless Communications，2008，7(10)：3697-3701

[7] Awad M K，Shen Xue-min. OFDMA Based Two-Hop Cooperative Relay Network Resource Allocation[C]// Proceedings of ICC'08. 2008：4414-4418

[8] Zhang Dan-hua，Tao Xiao-ming，Lu Jian-hua. Dynamic Resource Allocation for Real-time Services in Cooperative OFDMA Systems[J]. Communications Letters，IEEE，2011，15(5)：497-499

[9] Salah A A，Ali B M，Saqer A. An efficient resource allocation algorithm for OFDMA cooperative relay networks with fairness and QoS guaranteed[C]//2010 Second International Conference on Network Applications Protocols and Services(NETAPPS). 2010：188-192

[10] Chen Kai，Zhang Bi-ling，Liu Dan-pu，et al. Fair Resource Allocation in OFDMA Two-hop Cooperative Relaying Cellular Networks[C]// Vehicular Technology Conference Fall(VTC 2009-Fall). 2009：1-5

[11] 彭木根，王文博. 协同无线通信原理与应用[M]. 北京：机械工业出版社，2008

[12] Zhang Dan-hua，Wang You-zheng，Lu Jian-hua. QoS Aware Resource Allocation in Cooperative OFDMA Systems with Service Differentiation[C]// IEEE International Conference on Communications. 2010：1-5

(上接第 65 页)

挑战，现在仅仅处于开始阶段，还有很多问题需要解决。我们设想了一些未来的研究方向，最有希望的是强制公开检验模型。公开检验[4,6]，允许 TPA 审计云数据存储而不占用用户资源和时间。在模型中能否找到一个既能公开检验，又能保证动态数据存储准确性的方法很值得研究。另外除了研究动态云数据存储外，我们还计划解决更细致的数据错误定位问题。

参 考 文 献

[1] Bowers K D，Juels A，Oprea A. HAIL：A High-Availability and Integrity Layer for Cloud Storage[R]. 2008/489. Cryptology ePrint Archive，2008. http://eprint.iacr.org/

[2] Cloud Security Alliance. Security Guidance for Critical Areas of Focus in Cloud Computing[M]. April 2009

[3] Juels A，Burton J，Kaliski S. PORs：Proofs of Retrievability for Large Files[C]// Proc. of CCS '07. 2007：584-597

[4] Shacham H，Waters B. Compact Proofs of Retrievability[C]// Proc. of Asiacrypt '08. Dec. 2008

[5] Bowers K D，Juels A，Oprea A. Proofs of Retrievability：Theory and Implementation[R]. 2008/175. Cryptology ePrint Archive，2008. http://eprint.iacr.org/

[6] Ateniese G，Burns R，Curtmola R，et al. Provable Data Possession at Untrusted Stores[C]// Proc. Of CCS '07. 2007：598-609

[7] Ateniese G，Pietro R D，Mancini L V，et al. Scalable and Efficient Provable Data Possession[C]// Proc. of SecureComm '08. 2008：1-10

[8] Schwarz T S J，Miller E L. Store，Forget，and Check：Using Algebraic Signatures to Check Remotely Administered Storage[C]// Proc. of ICDCS '06. 2006：12

[9] Lillibridge M，Elnikety S，Birrell A，et al. A Cooperative Internet Backup Scheme[C]// Proc. of the 2003 USENIX Annual Technical Conference(General Track). 2003：29-41

[10] Hendricks J，Ganger G，Reiter M. Verifying Distributed Erasurecoded Data[C]// Proc. 26th ACM Symposium on Principles of Distributed Computing. 2007：139-146

[11] Plank J S，Ding Y. Note：Correction to the 1997 Tutorial on Reed-Solomon Coding[R]. CS-03-504. University of Tennessee，2003

[12] Wang Q，Ren K，Lou W，et al. Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance[C]// Proc. of IEEE INFOCOM. 2009