doi:10.3969/j.issn.1000-436x.2013.z1.033

基于散列链的云存储资源使用度量机制研究

刘玫[1]，付戈[2]，李奕希[2]，张鸿[2]，刘欣然[2]，杜翠兰[2]
(1. 中国科学院 信息工程研究所，北京 100093；2. 国家计算机网络应急技术处理协调中心，北京 100029)

摘 要：云存储服务的一种主要收费模式为依据服务提供商度量的客户资源实际使用量进行计费。因此，支付方和服务提供商之间的信任问题成为这种商业计费模式的关键因素，并可能引发安全问题。一方面，云存储服务提供商或者内部人员可能声称更多的客户资源使用量而多收取服务费用；另一方面，支付方可能否认已使用的资源从而减少应支付的费用。提出了一种基于散列链的资源使用度量机制，对不同资源分别产生可验证的证据。对于多数资源，现有云存储计费机制可以依据资源使用总量产生证据，但是考虑到存储量随时间不断波动并且资源的计费不仅与存储量相关还与时间因素相关，因此现有机制不能完全适用。提出的存储资源使用度量机制同时考虑时间和存储量 2 个因素，利用与计费方式关联的散列链产生证据，实现了原有机制的改进，达到了资源使用的可验证度量目标。

关键词：云存储；资源度量；散列链
中图分类号：TP393 **文献标识码**：B **文章编号**：1000-436X(2013)Z1-0246-10

Secure resource metering and accounting with hash chain in cloud storage

LIU Mei[1], FU Ge[2], LI Yi-xi[2], ZHANG Hong[2], LIU Xin-ran[2], DU Cui-lan[2]
(1.Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

Abstract: Cloud storage is a pay-per-use service of which the billing plan is typically based on users' resource consumption metered by server side. Therefore, the trustworthiness between payer and service provider becomes a key factor for the business, and triggers security concerns. On the one hand, service providers or insiders may inflate the amount of resource consumed to get more service charge. On the other hand, payers may deny the resource consumption to pay less. Thus, a resource metering and accounting scheme based on hash chain was proposed. It generated verifiable proof for different types of resources. For most resources, proof can be generated based on total amount of resource incurred. However, similar scheme cannot be applied to storage space usage, since it fluctuates and the billing relies on not only the utilized storage space but also the storage duration. The proposed storage usage metering scheme considers time factor together with storage space, and generates proof according to the diverse storage billing plans. It improves previous schemes, and achieves the goal of verifiable metering and accounting.

Key words: cloud storage; resource metering; hash chain

1 引言

云存储[1,2]模式下，企业或者个人将数据存储在服务提供商的虚拟数据容器中，其他用户则可以通过网络访问。这种方式在提供便捷的数据存储及共享的同时，也带来了安全问题。其中一个就是资源使用度量和计费安全的问题。云存储服务主要依据实际使用的资源量计算费用。例如，Amazon S3 中默认由数据桶拥有者根据数据桶的请求数量、网络通信流量和周期存储量来支付费用[1]。HP 公司也采用了类似的方式，服务提供商根据硬盘的使用量和带宽进行计费[3]。由于支付方对服务提供商的资源使用计费过程控制不足，因此其可能会担心服务提供商进行了不正确的使用量统计和计费[3]。一方面，

收稿日期：2013-08-05

即使服务提供商不会故意欺骗支付方增多资源的使用量和费用，其也可能由于程序故障或者其他原因而被错误地统计；另一方面，支付方可能对提供商度量的资源使用量产生疑问，否认已使用资源而减少支付费用。在目前商用云存储服务中，付费方需要信任服务提供商对其资源使用的统计，而且一般很难发现不正确的统计并验证统计结果。支付方不能简单地通过记录本地资源使用日志并对比服务提供商给出的资源使用情况来检验其正确性。因为一般数据容器拥有者的本地日志只包含了其自身访问数据容器的信息，但是其作为支付方，除了需要为其自身资源的使用支付费用，还可能需要为其他用户访问其数据容器使用的资源付费。此外，当支付方和服务提供商对资源的使用情况产生纠纷时，这种本地日志的方式也无法提供证据供第三方验证。因而在云存储服务中缺少一种资源使用度量的可验证机制[3]，使得当产生分歧时，可以依据服务过程中产生的证据在服务结束后判定争论的结果。

云计算出现以前，电子支付和移动通信领域就存在对安全计费问题的研究，希望提供高效、支付额可验证的安全计费方案。近几年随着云计算的发展，该领域也开始出现对上述安全问题的研究[3,4]，解决方案一般基于散列链。其中，Saksha 系统[3]提供的安全资源使用计费机制要求用户产生数据传输量和存储增长量的可验证收据，服务提供商产生存储下降量的可验证收据。类似地，机制还可以保证存储的增长量和下降量可验证。这种方法虽然解决了传输资源使用量可验证的问题，但是对存储资源的度量仍存在一些问题。

目前解决方案[3,4]根据存储量和传输量产生证据，但是这 2 种资源的计费方式并不相同。数据传输资源的计费依赖于计费周期结束时的传输总量，直接对传输量采用散列链的方法可以保证整个计费周期的传输量可验证。但是对于存储资源来说，使用的存储空间在整个计费周期内随时间上下浮动，这就使得直接利用存储量构造散列链收据的方法只能保证计费周期结束时的存储量可验证。但是在存储资源的通用计费方式中，存储使用的资源量与存储量随时间的变化及存储的时间相关，而并非取决于某个特定时间点（比如计费周期结束时）的存储总量。为了解决上述问题，本文提出了一种改进的基于散列链的资源使用度量机制，对不同的资源特别是存储资源的使用分别提供可验证、付费方不可否认的证据。

2　背景和相关工作

对交互式资源使用量的证明，最原始的方式需要每次对当前已使用的资源量进行数字签名并作为证据，但是在频繁使用小量资源时，大量数字签名会引入大量计算开销。为此，一些电子支付机制和移动通信系统[5~11]采用了基于散列链的方法。这种方式仅在首次产生证据时进行数字签名，降低了证据产生的开销。

2.1　计费系统散列链理论介绍

考虑一个计费系统[5]。定义 h 为一个单向散列函数，定义 T 为轮数的最大值。散列链的 $T+1$ 个值由对随机输入 x 反复作用 h 函数产生，并且满足 $h^i(x)=h(h^{i-1}(x))(i=1,2,\cdots,T)$，其中 $h^0(x)=x$。x 的随机性和单向散列函数的性质使得除了散列链产生者外其他人对于某 i 值($1\leqslant i\leqslant T$)，由 $h^i(x)$ 找到 z 使得 $h(z)=h^i(x)$ 在计算上不可行。安全支付中可利用此理论产生服务提供的证据，B 为 A 提供服务的过程如下：

1) 服务开始前，A 随机选择 x，计算散列链的根 $a_0=h^T(x)$ 并签名发送给 B;

2) A 将散列链按产生顺序的逆序发送，即在第 i 轮将 $a_i=h^{T-i}(x)(i=1,2,\cdots)$ 发送给 B（说明 B 已为 A 提供了 i 轮服务），B 收到后验证 $h(a_i)=a_{i-1}$ $(i=1,2,\cdots)$，并用 a_i 覆盖 a_{i-1}。此过程一直重复，直到 A、B 之间停止此阶段服务或者达到 T 轮。t 轮后，B 最终验证并保留散列值 a_t。

由于 B 不能从已收到的散列值推出未收到的散列值并且无法伪造 A 签名的 a_0，所以通过保留 A 签名的 $a_0=h^T(x)$ 和 A 最后发送的 $a_t=h^{T-t}(x)$ 并且验证 $h^t(a_t)=a_0$，即可说明 B 为 A 提供了 t 轮的服务。

2.2　相关工作

在云概念出现以前，电子支付和移动通信领域已经提供了可验证的计费方案[5~11]。文献[5]提出了一个利用散列链的小额支付系统。这项技术也普遍使用于移动系统的安全计费机制[8,9,12~14]。文献[8]和文献[9]将安全支付融入第三代移动通信系统(UMTS)，实现了用户和增值服务提供商间的安全计费机制。文献[14]指出移动通信网络中通常需要相信网络运营商根据通信量进行了正确的计费，但是在产生纠纷时却没有可验证的证据，因此也基于

散列链提出了其安全计费机制。随着云计算的出现和发展，散列链机制被引入作为存储服务中存储使用量和通信量的安全度量证明[3,4]。此外也有一些工作着重对资源度量的架构、模型及存在的问题进行了讨论[15~17]。文献[3]指出在外包的服务方式下安全计费的研究当时仍处于缺失状态，因此提出了针对存储服务的可验证资源使用度量系统 Saksha。该方案利用散列链理论由用户构造带宽使用量和存储空间增长量的收据并由服务提供商构造存储空间下降量的收据，保证数据传输资源和存储空间的使用量可验证。资源使用统计系统(THEMIS)[4]引入公证机构监管资源计费。系统要求用户、服务提供商和公证机构分别产生散列链。计费阶段，服务提供商发送其散列链相应元素给用户表明用户资源的使用量，用户用自己表明资源使用量的散列链相应元素和收到的服务提供商的散列值一起生成计费请求发送给公证机构。公证机构验证后发送其散列链相应元素对用户使用的资源量进行确认。本文前期工作也基于散列链产生资源使用的证据，但没有对存储资源采用与传输资源不同的处理方式。因此本文的工作主要研究针对存储资源使用的安全度量机制，依据不同计费方式并同时考虑存储量和时间因素产生基于散列链的证据信息。

3 模型和目标

为了讨论资源使用的安全度量方案，本节首先定义问题的模型和安全度量的目标。本模型参考了 Amazon S3[1]、文献[3]及云计算使用实例标准[18]。总体模型如图 1 所示，模型涉及 3 个实体。

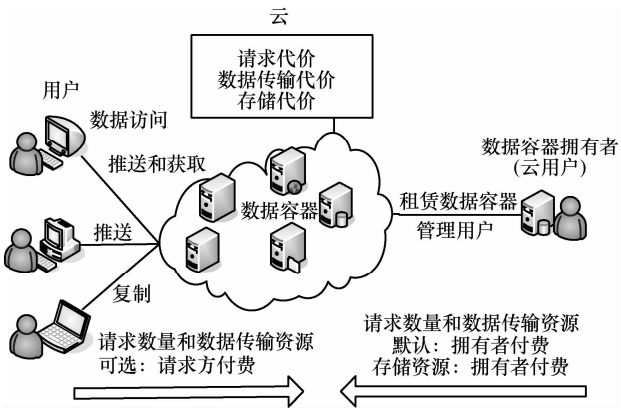


图 1 总体模型

云存储服务提供商：该实体提供数据容器的实际存储平台并根据资源使用情况收取服务费用。

数据容器拥有者：该实体租用服务提供商的数据容器来存储数据并支付费用，是云存储服务的客户。

用户：该实体通过网络访问数据容器。用户可以是数据容器拥有者、拥有数据容器的企业或者部门的员工或者授予访问权限的其他数据使用者等。

模型的收费资源包括请求数量、数据传输资源和存储资源[1]，并且采用类似 Amazon S3 的付费模式。服务提供商依据客户（数据容器拥有者）或者用户实际使用的资源量计费。

请求数量和数据传输资源：一般默认由数据容器拥有者根据自身和其他用户访问数据容器时的资源使用量向服务提供商支付费用（情形 1.1）。它存在于数据容器拥有者为公司或者公司部门，用户为公司的雇员或合作伙伴的雇员以及用户间存在某些联系并由一方付费的情况。特殊情况下，数据容器也可以配置成请求方付费[19]，即每个请求方根据其资源使用量向服务提供商支付费用(情形 1.2)。它存在于数据容器拥有者共享数据但是不愿为访问用户下载等操作消耗的资源付费的情况。

存储资源：一般由数据容器拥有者为数据容器中所有存储数据向服务提供商付费[19]（情形 2）。

模型中，由于数据容器拥有者可以设定用户的访问权限，因此假定此两者之间存在一定的预先信任关系。文献[3]也采用了类似的方式。此外，假定数据容器拥有者和用户不相信服务提供商给出的资源使用报告。同样，服务提供商也不相信数据容器拥有者和用户，因为他们可能否认消耗的资源。

云存储资源使用度量机制希望达到的目标是：服务过程产生的资源使用证据是可验证并不可否认的。证据可以被服务提供商、数据容器拥有者、用户和第三方验证。同时，对于上述 3 种情形服务提供商不能增加支付方需要付费的资源量，提供比实际更多的资源消耗的证据;支付方也不能否认此资源使用量。

4 请求数量和数据传输资源使用的度量

4.1 资源使用度量机制概述

机制以文献[3]的方法为基础，并补充讨论了证据内容及分析。机制要求用户使用一定量服务后发送证据给服务提供商[12]，提供商对其进行验证并在周期结束时将证据和账单发送给支付方，支付方验证并支付费用。

4.2　证据产生和验证

机制利用 2 个技术实现轻量级的证据产生。首先，证据产生的基本单位为若干请求组成的请求组或者一定大小的数据块而不是每个请求和数据传输的每个字节[3]；此外，机制采用散列链的方式[3,4]，此方式只需在初始时对证据进行签名，后续过程不再需要签名操作。机制的每个计费周期包括 2 个阶段：初始化阶段和服务阶段[3,8]。

初始化阶段：每个用户选择随机 x、散列族 H 中确定单向散列函数的初始向量 IV 和最大轮数 T，产生散列链：$a_T = x$，$a_{T-1} = H_{IV}(x)$，$\cdots$，$a_0 = H_{IV}^T(x)$。用户对散列链的根 a_0 签名并发送给服务提供商。

服务阶段：在服务阶段的第 i 轮，用户通过发送 $a_i = H_{IV}^{T-i}(x)(i = 1,2,3\cdots)$ 表示从计费周期开始此用户确认服务提供商为其累计处理了 i 组请求或者传输了 i 个数据块（如果 $i \leqslant T$ 不成立，可以通过重新构造新散列链来重新初始化，因此下面的讨论均认为 $i \leqslant T$ 。）。服务提供商验证收到的散列值。假定上次收到的为 a_j，此次为 $a_i (i > j)$，那么服务提供商计算，并验证 $a_{j'}$ 是否与 a_j 相等。如果相等说明自上次发送散列值后用户确认云服务提供商又为其处理了 $i - j$ 组请求或者传输了 $i - j$ 个数据块，即累计提供了 i 个单位的服务。验证通过后，服务提供商用新散列值覆盖原来的并作为证据缓存[14]。

为了确定需要发送的散列值，每个用户维护了 2 个计数器，计数器中包含了除去发送散列值确认的资源使用外服务提供商累计为此用户处理的请求数或者传输的数据量[3]。用户将当前计数器的值 X 换算为请求组或者数据块的数量 k（$k = \lfloor X/b \rfloor$，b 为组请求数或者数据块的大小），并发送上次发送的散列值后的第 k 个给服务提供商，然后将计数器值更新。

4.3　讨论和分析

在初始化阶段，用户签名信息需要包含 *Cloud _ ID*（服务提供商）和 *Service _ ID*（目标服务）。服务与数据容器相关联，*Cloud _ ID* 和 *Service _ ID* 可以唯一确定一个数据容器和容器拥有者。不包含 *Cloud _ ID* 会使任何监听到散列值的人都可以冒充收费方。而不包含 *Service _ ID* 会使数据容器拥有者可以否认用户访问的容器是属于他的，从而拒绝交费。因此，收到用户签名以及提交的服务请求时，服务提供商需要验证这些信息。此外，签名内容还应包括时间戳 *TS* 和 $ch\arg ing\_\inf o$。$ch\arg ing\_\inf o$ 包含其他收费相关的信息，它可能涉及收费时间区间、资源类型、产生证据的资源使用基本单位及用户请求中的其他信息。*Cloud _ ID*、*Service _ ID* 和时间戳 *TS* 组合每次均不同，保证了用户签名不能被重放。

在服务过程中，用户不会产生多于实际资源使用量的证据，这是由于用户与支付方存在信任关系（情形 1.1）或者其本身就是支付方（情形 1.2）。此外，用户也不能在消耗大量资源后不提供证据。如果服务提供商没有收到证据，可以不再为用户提供后续的服务。资源使用度量机制可以实现第 2 节中的目标。假定用户对 a_0 进行签名，服务提供商最后收到并验证通过的是 a_m，那么如果 a_0 与 $H_{IV}^m(a_m)$ 相等，则用户不能否认其使用了 m 个单位的资源，服务提供商也不能声称此用户使用了多于 m 个单位的资源。进而数据容器拥有者不能否认所有用户的请求总量和数据传输总量，云服务提供商也不能增加此资源总量。同时，用户、服务提供商、数据容器拥有者或者第三方可以根据证据验证。

按照上述方式，机制可能存在文献[17]中提到的网络传输和操作延时造成的用户和服务提供商在同一时间点对资源使用量判断不一致的问题。比如，用户在足够接近计费周期结束时发送了请求并将其计入用户此周期的资源使用量，而可能由于延时其在下一个计费周期才到达服务提供商，因此提供商将其计入了下一个计费周期[17]。因此，双方认为需要发送的散列值就可能存在差异。对于这种差异，一种解决方案就是在服务提供商给用户的操作响应信息中加入操作发生的具体时间。用户收到响应后验证此时间是否在请求和响应之间，如果是，则将其作为双方达成共识的操作执行时刻。

5　存储资源使用的度量

对于存储资源，目前方案[3]在处理时主要考虑存储量的大小。因此，只能证明计费周期结束时的存储总量。而这与存储资源的通用计费方式并不相符，一般情况下其计费并不完全取决于计费周期结束时的存储总量，而与存储量随时间的变化及存储的时间相关。

5.1 存储资源的计费

云服务提供商通常基于存储量存储一定时间（比如可以依据字节小时数 ByteHrs[17]）来计费，其根据策略可能采用不同的具体计费方式。其中，一种方式需要首先计算本计费周期的计费存储量，然后按照计费存储量存储了整个计费周期来计算费用，计费存储量可以是周期内存储量的最大值也可以采用其他方法得到；另一种方式则根据存储量对时间的分布来计算[2,17]，即通过计算计费周期内存储量对时间的积分来计算费用。第一种计费方式利用了对存储资源使用的粗略估计进行计费，比较简单；第二种计费方式更能体现真实的资源消耗量。

5.2 存储量变化的描述

本节讨论存储量变化的描述，并作为后续对机制讨论的基础。每个用户对应一个累积存储增量(CSSI, cumulative storage space increment)曲线。曲线记录了该用户作为操作执行者对容器数据进行操作时带来的存储数据量的变化。此种方式使用户的 *CSSI* 可能为负值(比如，用户可能只执行了一个删除文件的操作)。在本节到第 6 节中，图 2~图 6 描述了存储总量和 *CSSI* 的变化，其采用了类似的表示方法。图中的 x 轴表示时间，t_0 为计费周期的起始时间，t_n 为计费周期的结束时间，t_1、t_2、t_3…为存储量发生变化的时刻；y 轴为存储总量或者 *CSSI* 的值。

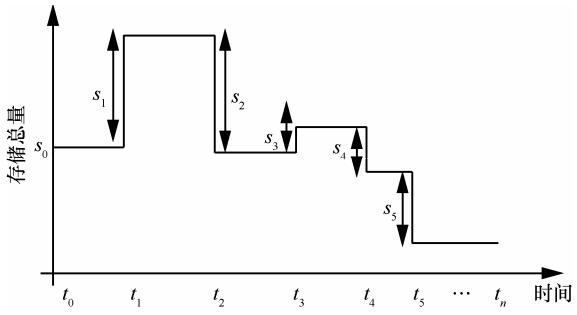
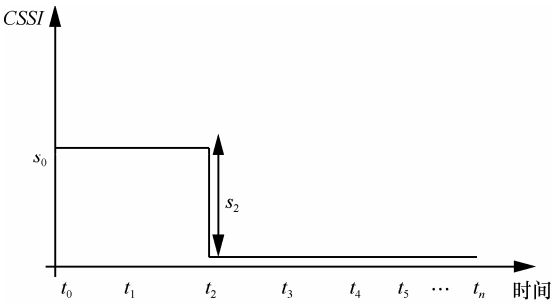
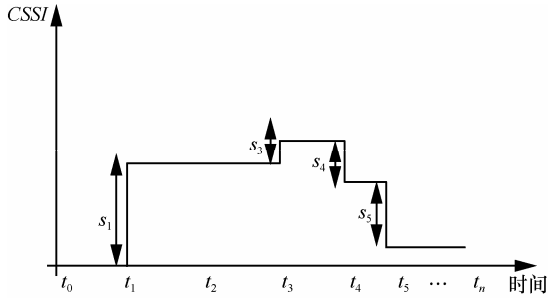


图 2 存储总量曲线

图 2 展示了一个计费周期中存储总量的变化。如果假定数据仅被 2 个用户操作，那么用户 1 和用户 2 的 *CSSI* 曲线可能如图 3(a)和图 3(b)所示，其对应的操作为用户 1 在 t_0 时刻增加了 s_0 的数据量，并在 t_2 时刻减少了 s_2 的数据量；用户 2 在 t_1、t_3 时刻分别增加了 s_1 和 s_3 的数据量，在 t_4、t_5 时刻分别减少了 s_4、s_5 的数据量。一般地，如果数据容器存储总量随时间的变化表达式为 $f(t)$，l 个用户的 CSSI 随时间变化曲线的表达式分别为 $f_1(t)$，$f_2(t)$，…，$f_l(t)$，则 $f(t)=f_1(t)+f_2(t)+\cdots+f_l(t)$。

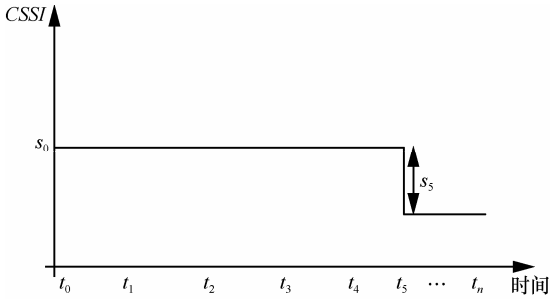


(a) 用户 1 的 *CSSI* 曲线

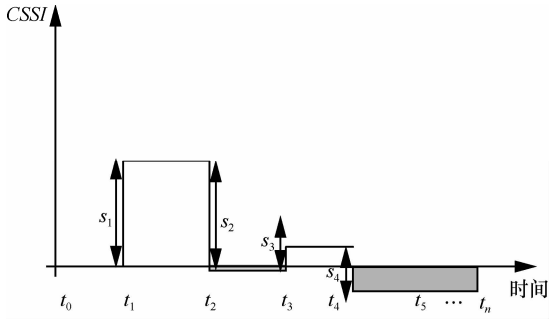


(b) 用户 2 的 *CSSI* 曲线

图 3 用户的 *CSSI* 曲线



(a) 例 1 中用户 1 的 *CSSI* 曲线



(b) 例 1 中用户 2 的 *CSSI* 曲线

图 4 例 1 中用户 1 和用户 2 的 *CSSI* 曲线

6 基于存储量峰值的存储资源使用度量

第一种计费方式按照计费存储量存储了整个计费周期来计算费用。计费存储量可以根据策略设

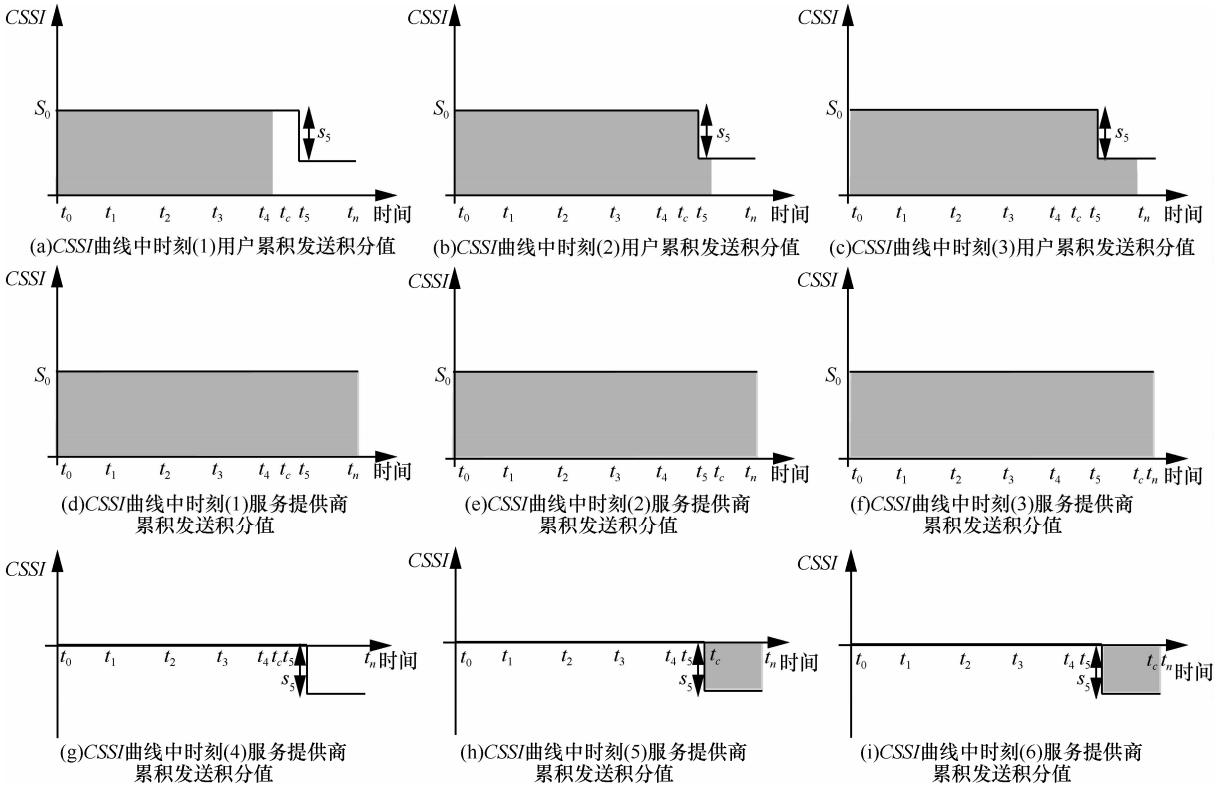


图 5　图 4(a)中 CSSI 曲线的证据积分值

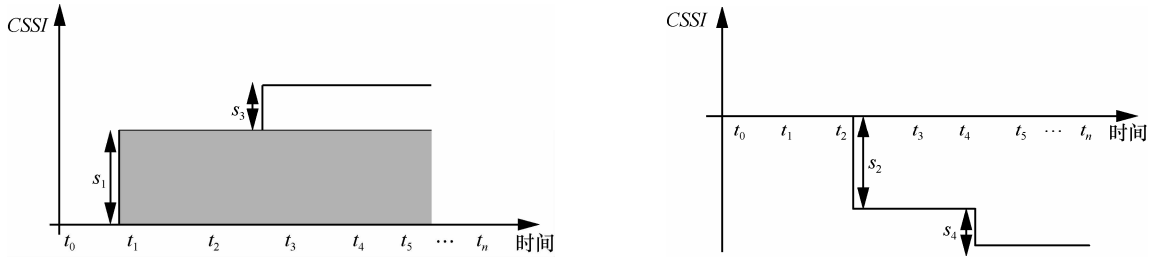


图 6　例 1 中用户 2 的 $f_2'(t)$ 和 $f_2''(t)$ 曲线

定为周期内存储量的最大值等。文献[3]中的方法可以应用于按照计费周期结束时的存储量存储了整个计费周期来计费的方式。在这里，考虑按照周期内数据容器的最大存储总量（峰值）计算费用的情况。

6.1　资源使用度量机制概述

机制要求数据容器拥有者（或授权用户）产生最大存储总量使用证据，并在服务中交给云服务提供商验证。

6.2　证据产生和验证

服务提供商按照计费周期中存储总量 $f(t)$ 的最大值对数据容器拥有者收费。但用户 j 无法通过 $f_j(t)$ 知道 $f(t)$ 何时达到最大值。因此，方案要求在服务阶段用户都将到当前时间的 $f_j(t)$ 通知证据产生者，证据产生者获得所有用户的 *CSSI* 曲线后就可以得到 $f(t)$。如果到当前时间为止 $f(t)$ 最大值比上次发送证据的最大值大，则发送新的证据。具体的，证据产生者对 $a_0 = H_{IV}^{T}(x)$ 签名发送给服务提供商。继而通过发送 $a_i = H_{IV}^{T-i}(x)$ $(i = 1,2,3\cdots)$ 表示从计费周期开始 $f(t)$ 的最大值达到 i 个数据块。

这种发送 $f_j(t)$的方式使所有用户都需要与证据产生者进行通信。用户可以在每次操作完成并从操作响应信息中获得操作发生时刻后将其 *CSSI* 的变化及变化的时刻发送给证据产生者。因此，用户与证据产生者的通信开销取决于该用户执行操作的数量。同时，服务提供商出于效率考虑可能采用检

查点采样的方式，其只需检查证据产生者发送证据证明的最大值是否大于等于服务提供商采样的各个检查点的存储总量，如果是则认为证据产生者如实地发送了证据。

7 基于存储量对时间分布的存储资源使用度量

第二种计费方式根据存储总量对时间的分布进行计费。收费金额仅仅依赖于计费周期内存储总量对时间的积分，即 $\int_{t_0}^{t_n} f(t)\mathrm{d}t$。因此，机制用存储量对时间的积分代替存储量作为可验证的资源量。

7.1 资源使用度量机制概述

服务提供商依据 $\int_{t_0}^{t_n} f(t)\mathrm{d}t$ 对数据容器拥有者收取费用，因此收费资源总量为 $\sum_{j=1}^{l}\int_{t_0}^{t_n} f_j(t)\mathrm{d}t$。对于 $\int_{t_0}^{t_n} f_j(t)\mathrm{d}t$，机制要求用户和服务提供商在服务过程中分别产生此积分不同部分的证据并发送给对方。然后在计费周期结束时， 数据容器拥有者通过同时验证两方产生的证据就可以获得 $\int_{t_0}^{t_n} f(t)\mathrm{d}t$。证据使服务提供商不能增加 $\int_{t_0}^{t_n} f(t)\mathrm{d}t$ 的值，数据容器拥有者也不能抵赖。与基于峰值的方式不同，此度量机制不再需要用户和证据产生者进行通信。

7.2 证据产生和验证

由概述可知，机制可以通过首先保证数据容器拥有者不能提供证据减少用户 j 的 $\int_{t_0}^{t_n} f_j(t)\mathrm{d}t$，并且服务提供商也不能提供证据增加此值来保证 $\int_{t_0}^{t_n} f(t)\mathrm{d}t$ 的值。

7.2.1 度量机制方案

在讨论存储资源的度量机制前，本节首先对例 1 进行讨论和分析。

例 1 用户 1 在 t_0 时刻增加了 s_0 的数据量，t_5 时刻减少了 s_5 的数据量，用户 2 在 t_1、t_3 时刻分别增加了 s_1 和 s_3 的数据量，并在 t_2 和 t_4 时刻减少了 s_2 和 s_4 的数据量。

例 1 中用户 1 和用户 2 的 $f_j(t)(j=1,2)$ 如图 4(a)和图 4(b)所示，存储总量的曲线 $f(t)$ 仍然如图 2 所示。可以看出，每个用户的 *CSSI* 值可正可负。如果假定 t_c 为当前时刻，用户 j 可以获知 $f_j(t)(t_0 \leqslant t \leqslant t_c)$ 的曲线并依此发送积分 $\int_{t_0}^{t_c} f_j(t)\mathrm{d}t$ 的证据，但是对于 $t_c \leqslant t_n$ 此积分不一定非负，且没有随 t_c 的增加递增，所以散列链机制不能直接适用。因此，首先考虑此种情形下的一种特殊情况对问题进行简化。

例 2 用户 1 在 t_0 时刻增加了 s_0 的数据量，并在 t_2 时刻减少了 s_2 的数据量，用户 2 在 t_1 和 t_3 时刻分别增加了 s_1 和 s_3 的数据量，在 t_4 和 t_5 时刻分别减少了 s_4 和 s_5 的数据量。

例 2 中用户 1 和用户 2 的 CSSI 曲线如图 3(a)和图 3(b)所示。可以看出，此种情况下每个用户 *CSSI* 曲线的 y 值一直为非负。这一特征使 $\int_{t_0}^{t_c} f_j(t)\mathrm{d}t$ 对于 $t_c \leqslant t_n$ 非负且随 t_c 的增加而递增。因此，机制可以直接采用基于散列链的方法。在服务阶段，用户发送 $a_i = H_{IV}^{T-i}(x)(i=1,2,3,\cdots)$ 表示 $\int_{t_0}^{t_c} f_j(t)\mathrm{d}t$ 达到 i 个单位。

回到例 1 的情况，可以注意到，当 *CSSI* 始终为非负时，*CSSI* 对时间的积分值非负并且随着 t_c 的增加而增长。而当 *CSSI* 始终为非正时，*CSSI* 对时间的积分值为非正，并且随着 t_c 增加而下降。因此，机制要求当 *CSSI* 对时间的积分值增长时由用户产生证据，并且当其下降时由云服务提供商产生证据。对于用户 2(如图 4(b)所示)，用户依据底色为白色（x 轴上方）的积分部分产生证据，服务提供商依据底色为深色（x 轴下方）的积分部分提供证据，就可以保证各自积分的单调性。对于一般的情况，在 t_c 时刻用户基于 x 轴上方 t_c（$t_c \leqslant t_n$）时刻前的 *CSSI* 对时间的积分 $\int_{t_0}^{t_c} \frac{|f_j(t)|+f_j(t)}{2}\mathrm{d}t$ 产生证据，此积分值非负且随 t_c 的增加单调递增。服务提供商产生的证据则基于 x 轴下方 t_c 时刻以前的 *CSSI* 的负值对时间的积分 $\int_{t_0}^{t_c} \frac{|f_j(t)|-f_j(t)}{2}\mathrm{d}t$，此积分值也非负且随 t_c 单调递增。而 $\int_{t_0}^{t_c} f_j(t)\mathrm{d}t$ 为这 2 部分的差。至此，用户和服务提供商产生的证据依然可以依赖于散列链机制。具体地，在初始化阶段，每个用户产生不同的散列链，服务提供商对每个用户产生一个散列链。散列链产生后，用户和服务提供商分别对自己散列链的根签名并将签名结果发送给对方。在后续的服务阶段，用户发送 $a_i = H_{IV}^{T-i}(x)(i=1,2,3,\cdots)$ 表示 $\int_{t_0}^{t_c} \frac{|f_j(t)|+f_j(t)}{2}\mathrm{d}t$ 达

到 i 个单位。服务提供商也发送相应的散列值表示 $\int_{t_0}^{t_c}\frac{|f_j(t)|-f_j(t)}{2}\mathrm{d}t$ 达到了一定量。

7.2.2 *度量机制改进方案*

如图 5 所示，深色部分表示证据产生方（用户或者服务提供商）随 t_c 时刻的增加发送证据依据的不同积分值。如果对图 4(a)利用 7.2.1 节中的方式产生证据，则用户发送证据的积分值变化如图中方案 1 所示。其中图 5(b)和图 5(c)说明虽然从时刻 t_5 到计费周期结束 *CSSI* 都不再变化（用户不再对数据进行操作），但是仍然需要用户在 t_5 后每隔一段时间在线发送 $\int_{t_0}^{t_c}\frac{|f_j(t)|+f_j(t)}{2}\mathrm{d}t$ 积分增长的证据。为了解决这个问题，提出了改进方案。改进方案从计费周期开始到当前时刻 t_c 按照 *CSSI* 实际变化进行积分而从当前时刻到计费周期结束按照当前时刻存储状态不变（*CSSI* 不变）来计算积分，这两部分积分的和作为当前时刻的积分值证据，也就是说当前时刻证据还考虑了当前时刻到计费周期结束按当前 *CSSI* 预存储的部分。图 5 给出了改进后机制(方案 2)对图 4(a)的证据产生方式。图 5(a)~图 5(c)和图 5(d)~图 5(f)分别为用户和服务提供商发送证据依据的积分值随 t_c 的变化，两方积分值的和为上述考虑预存储后 t_c 时刻的积分值。图 5 中，改进方案只需要用户和服务提供商各发送一次证据。对比两方案可以看出改进方式具有用户在不进行操作时无需一直在线发送证据的优势。

为了讨论预存储时证据的产生，首先定义如下几个表达式。

$\overline{f_j(t)}$：当前时刻 t_c 的 $\overline{f_j(t)}$ 由 $f_j(t)$ $(t_0\leqslant t\leqslant t_c)$ 扩展而成。具体来说，从 t_0 到 t_c，$\overline{f_j(t)}$ 与 $f_j(t)$ 相等，从 t_c 到 t_n *CSSI* 始终为 $f_j(t_c)$。可以看出，$\overline{f_j(t)}$ 随 t_c 变化，并且当 $t_c=t_n$ 时，$\overline{f_j(t)}$ 与 $f_j(t)$ 相同。

$f_j'(t)$ 和 $f_j''(t)$：将 $f_j(t)$ 拆分成 2 个子表达式，$f_j'(t)$ 只记录 $f_j(t)$ 中 *CSSI*(y 值)的增加，$f_j''(t)$ 只记录 $f_j(t)$ 中 *CSSI*(y 值)的减少。对于例 1 中的用户 2 如图 4(b)所示，其对应的 $f_2'(t)$ 如图 6(a)所示，$f_2''(t)$ 如图 6(b)所示。$f_j(t)$、$f_j'(t)$ 和 $f_j''(t)$ 满足 $f_j(t)=f_j'(t)+f_j''(t)$。

$\overline{f_j'(t)}$ 和 $\overline{f_j''(t)}$：类似于 $\overline{f_j(t)}$ 的产生方式，当前时刻 t_c 的 $\overline{f_j'(t)}$ 和 $\overline{f_j''(t)}$ 分别由 $f_j'(t)$ 和 $f_j''(t)$ $(t_0\leqslant t\leqslant t_c)$ 扩展而成。如图 6(a)所示，t_c 在时间 t_1 和 t_3 之间时，$\overline{f_2'(t)}$ 为图中深色底色上方与其紧邻的曲线。由于 $f_j(t)=f_j'(t)+f_j''(t)$，所以对某个 t_c 也满足 $\overline{f_j(t)}=\overline{f_j'(t)}+\overline{f_j''(t)}$。此外，可以看出 $f_j'(t)$ 和 $-f_j''(t)$ 非负，同时对任意时刻的 t_c $(t_0\leqslant t_c\leqslant t_n)$，$\overline{f_j'(t)}$ 和 $-\overline{f_j''(t)}$ 也非负。

机制仍需保证用户 j 的 $\int_{t_0}^{t_n}f_j(t)\mathrm{d}t$ 不能被数据容器拥有者任意减少，也不能被服务提供商任意增加。而 $\int_{t_0}^{t_n}f_j(t)\mathrm{d}t$ 为 $\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 与 $\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$ 的和，$\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 非负，$\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$ 非正，因此问题转化为需要保证 $\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 不能被数据容器拥有者任意减少，也不能被服务提供商任意增加，并且 $-\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$ 不能被数据容器拥有者任意增加，也不能被服务提供商任意减少。

当 $t_c=t_n$ 时，$\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 等于 $\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 并且 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 等于 $-\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$。同时，对 t_c 时刻（$t_c\leqslant t_n$）的 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 和 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 都非负且随 t_c 的增加单调递增。因此，用户和云服务提供商可以利用散列链产生证据。用户在 t_c 时刻利用 t_c 时刻的 $\overline{f_j'(t)}$ 计算出的 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 产生证据。服务提供商利用 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 产生证据。对同一个 t_c，满足 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 与 $\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 的和为 $\int_{t_0}^{t_n}\overline{f_j(t)}\mathrm{d}t$。具体来讲，在服务阶段用户发送 $a_i=H_{IV}^{T-i}(x)(i=1,2,3,\cdots)$ 表示 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 达到 i 个单位。同理，服务提供商发送相应散列值表明 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 达到一定量。并且每个用户维护一个计数器，其记录了 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 中未发送证据的累计积分值。类似的，服务提供商为每个用户维护一个计数器，其记录了 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 中未发送证据的累计积分值。

7.3 讨论和分析

在服务过程中，用户和服务提供商都不会产生多于实际的证据也不能不提供或者少提供证据。如果用户不发送证据，服务提供商可以不再为用户提供服务。而如果云服务提供商不发送证据，那么将会损害其声誉，数据容器拥有者可以不再使用此服务提供商的服务。

对于改进方案，用户不能否认、服务提供商不能增加 $\int_{t_0}^{t_n}\overline{f_j'(t)}\mathrm{d}t$ 的值。同时，服务提供商不能否认、用户不能增加 $-\int_{t_0}^{t_n}\overline{f_j''(t)}\mathrm{d}t$ 的值。因此，在计费周期结束后可以验证 $\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 和 $-\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$ 的值。而 $\int_{t_0}^{t_n}f_j(t)\mathrm{d}t$ 为 $\int_{t_0}^{t_n}f_j'(t)\mathrm{d}t$ 和 $-\int_{t_0}^{t_n}f_j''(t)\mathrm{d}t$ 的差，所以数据容器拥有者不能提供证据减少 $\int_{t_0}^{t_n}f_j(t)\mathrm{d}t$ 的实际值，服务提供商也不能提供证据增加。进而数据容器拥有者不能减少 $\int_{t_0}^{t_n}f(t)\mathrm{d}t$ 的值，服务提供商也不能声称用户使用了多于实际的存储资源。因此，该机制达到了第 2 节中目标的要求。

上述存储资源使用度量机制实际上考虑了 *CSSI* 的变化连续记录并用于计算资源使用量的情况，在实际中服务提供商还可能会对某些检查时间点的存储量采样，并将采样值作为计费的依据。比如，Amazon S3 每天至少对数据容器的存储量采样 2 次并且检查点时间的选取是随机的[17]。它将采样结果乘以上次检查点后经过的时间并在每个月（计费周期）结束时将字节小时数加和到一起[17]进行计费。这种计费方式实际上也是将积分作为存储需要度量的资源量，因此上述机制稍加变化即可应用于这种利用多检查点的存储量来计费的情况。

用户和服务提供商需要根据已知检查点的 *CSSI* 按服务提供商事先约定的方式将其扩展成扩展 *CSSI* 曲线（比如假定两个检查点间的 *CSSI* 不变或者将某一时刻的 *CSSI* 作为当天整天的 *CSSI* 值）。此时，扩展存储总量对时间的积分仍然为每个用户的扩展 *CSSI* 对时间积分的和。因此，仍然可以将扩展 *CSSI* 曲线作为原机制中的 *CSSI* 曲线由用户和服务提供商一起对积分产生可验证证据。

然而，多检查点的情况下需要服务提供商告知用户检查点的时间，但提前告知有可能被用户或者数据容器拥有者恶意利用，比如在到达检查点前他们可能删除大量数据从而降低资源的消耗量[17]。这就需要在检查点过后再通知用户检查的时间[17]。如果服务提供商将检查点按时间顺序告知用户，那么用户一般需要将最近一次已知检查点作为当前时间（即 t_c 代表最近一次检查点时间）来计算相应积分值。此外，类似于数据传输，存储资源也存在文献[17]中提出的网络传输和操作延时造成的用户和服务提供商在同一检查点对资源使用量判断不一致的问题。这是因为 *CSSI* 曲线依赖于双方各自认为的操作发生的时刻，时刻不同会造成这两方观察到的 *CSSI* 曲线的不同。这种差异可以采用与传输类似的方式解决。

8 结束语

本文在散列链理论的基础上提出一种资源使用度量机制并对其安全性进行了讨论。这种机制对云存储服务中资源消耗量提供可验证不可否认的证据。与请求数量和数据传输量不同，存储量随时间波动并且存储资源的计费不仅与存储量相关还与时间因素相关，因此提出的存储资源度量机制同时考虑了这 2 个因素，根据不同的存储资源计费方式，利用与计费模式关联的散列链在服务过程中产生证据。此机制提供了一种新的存储资源的可验证度量方式。在将来的工作中，还将考虑资源使用度量机制中的一些扩展问题。首先是基于峰值的存储资源度量中如何收集用户的资源使用信息的问题，一种方案是文献[15]中提到的在用户平台部署应用程序，但此方案依赖于对部署程序的控制程度等因素[15]，而且需要用户与数据容器拥有者进行通信；第二，用户和服务提供商需要独立地进行资源使用的度量，两方度量的资源变化量可能存在偏差[17]，如何解决这种偏差也是进一步的工作。

参考文献：

[1] Amazon simple storage service (Amazon S3)[EB/OL]. http://aws. amazon. com/s3/, 2011.

[2] Windows azure storage[EB/OL]. http://www.microsoft.com/windowsazure/features/storage/, 2011.

[3] KHER V, KIM Y. Building trust in storage outsourcing: Secure accounting of utility storage[A]. Proceedings of IEEE Symposium on Reliable Distributed Systems[C]. Washington, 2007. 55-64.

[4] PARK K, PARK S K, HAN J, *et al*. Themis: towards mutually verifiable billing transactions in the cloud computing environment[A]. Pro-

ceedings of the 2010 IEEE 3rd International Conference on Cloud Computing[C]. Washington, 2010. 139-147.

[5] PEDERSEN T P. Electronic payments of small amounts[A]. Proceedings of Security Protocols Workshop[C]. Berlin Heidelberg: Springer-Verlag, 1996. 59-68.

[6] DAI X, GRUNDY J C. Netpay: an off-line, decentralized micro-payment system for thin-client applications[J]. Electronic Commerce Research and Applications, 2007,6(1):91-101.

[7] HERZBERG A, YOCHAI H. Minipay: charging per click on the web[J]. Computer Networks, 1997,29(8-13): 939-951.

[8] HORN G, PRENEEL B. Authentication and payment in future mobile systems[A]. Proceedings of European Symposium on Research in Computer Security[C]. Berlin Heidelberg: Springer-Verlag,1998. 277-293.

[9] MARTIN K M, PRENEEL B, MITCHELL C J, *et al*. Secure billing for mobile information services in UMTS[A]. Proceedings of Intelligence in Services and Networks[C]. Berlin Heidelberg: Springer -Verlag, 1998. 535-548.

[10] PATIL V, SHYAMASUNDAR R K. E-coupons: an efficient, secure and delegable micro-payment system[J]. Information Systems Frontiers, 2005,7(4-5): 371-389.

[11] RIVEST R L, SHAMIR A. Payword and micromint: Two simple micropayment schemes[A]. Proceedings of Security Protocols Workshop[C]. Berlin Heidelberg: Springer-Verlag, 1996.69-87.

[12] BUTTYÁN L, HUBAUX J P. Accountable anonymous access to services in mobile communication systems[A]. Proceedings of Symposium on Reliable Distributed Systems[C]. Washington, 1999. 384-389.

[13] BUTTYÁN L, HUBAUX J P. Accountable Anonymous Service Usage in Mobile Communication Systems[R]. 1999.

[14] ZHOU J, LAM K Y. Undeniable billing in mobile communication[A]. Proceedings of Fourth Annual ACM/ IEEE International Conference on Mobile Computing and Networking[C]. New York, 1998. 284-290.

[15] MOLINA C, COOK N, SHRIVASTAVA S. On the feasibility of bilaterally agreed accounting of resource consumption[A]. Proceedings of ICSOC Workshops[C]. Berlin Heidelberg: Springer-Verlag, 2008. 270-283.

[16] MIHOOB A, MOLINA C, SHRIVASTAVA S. A Case for Consumer–centric Resource Accounting Models[A]. Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing[C]. Washington, 2010. 506-512.

[17] MIHOOB A, MOLINA C, SHRIVASTAVA S. Consumer Side Resource Accounting in the Cloud[R]. 2011.

[18] Important actors for public clouds[EB/OL]. http://www.nist.gov/itl/cloud/ actors. cfm, 2011.

[19] Requester pays buckets[EB/OL]. http://docs.amazonwebservices.com/AmazonS3/latest/dev/index.html?RequesterPaysBuckets.html, 2011

作者简介：

刘玫（1984-），女，北京人，博士，中国科学院信息工程研究所工程师，主要研究方向为信息安全、网络安全、云安全等。

付戈（1983-），男，河南郑州人，博士，国家计算机网络应急技术处理协调中心工程师，主要研究方向为数据库、信息安全、云存储等。

李奕希（1985-），男，北京人，硕士，国家计算机网络应急技术处理协调中心工程师，主要研究方向为网络安全。

张鸿（1976-），男，陕西西安人，博士，国家计算机网络应急技术处理协调中心高级工程师，主要研究方向为网络安全。

刘欣然（1971-），男，黑龙江人，博士，国家计算机网络应急技术处理协调中心副主任，主要研究方向为网络安全。

杜翠兰（1969-），女，湖北武汉人，博士，国家计算机网络应急技术处理协调中心高级工程师，主要研究方向为网络安全。