doi:10.11959/j.issn.1000-436x.2015232

移动云计算环境中基于身份代理签名的完整性检测协议

闫莉，石润华，仲红，崔杰，张顺，许艳
（安徽大学 计算机科学与技术学院，安徽 合肥 230601）

摘 要：提出一个新的适用于移动云计算环境的完整性检测模型，在 PDP 模型中引入了拥有较强计算能力的代理签名方，代替移动终端生成数据验证标签。基于该模型设计了一个 IBPS-PDP 协议，该协议利用基于身份的签名方法，减少了系统的公钥证书管理和移动终端的证书认证代价。最后，证明 IBPS-PDP 在随机预言机模型下是安全的。

关键词：移动云计算；数据完整性；基于身份签名；代理签名
中图分类号：TP393 **文献标识码**：A

Integrity checking protocol with identity-based proxy signature in mobile cloud computing

YAN Li, SHI Run-hua, ZHONG Hong, CUI Jie, ZHANG Shun, XU Yan
(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: Based on provable data possession (PDP) model, a more perfect data integrity checking model for mobile cloud computing was proposed, in which there was an additional proxy party with stronger computing power to help the mobile users to calculate the block tags. Furthermore, for the proposed model, an identity-based proxy signature PDP (IBPS-PDP) protocol was presented. By using identity-based signatures, the system did not need to manage public key certificates and further the users did not need to take the additional computations to verify the other's certificates yet. Finally, the security of the proposed IBPS-PDP protocol is proved in the random oracle model.

Key words: mobile cloud computing; data integrity; identity-base signature; proxy signature

1 引言

随着诸如智能手机、平板电脑等终端设备的发展，人们越来越多地利用移动终端处理日常事务。但是，由于终端设备的计算和存储能力有限，而需要处理的数据量却急速增加，将数据处理和存储迁移出移动设备变得非常必要。移动云计算为此提供了一种有效的解决方法：将移动设备的数据处理和存储任务交给拥有强计算能力和集中式数据管理优势的云服务器，从而为终端用户提供更加丰富的移动应用和更加低成本的数据管理服务。但是，上传到云中的数据由云服务提供商(CSP, cloud service provider)而非用户进行管理，因此，考虑到云端数据可能遇到外部篡改、数据丢失等数据安全问题，用户需要确定云服务器是否正确存储数据。

验证远程数据完整性最简单的方法是计算整个数据的消息验证码。为了避免每次检测需要下载远程服务器上的整个文件，用户利用带密钥的MAC函数为数据预计算 k 个消息验证码 MAC（对应 k

收稿日期：2014-10-15；**修回日期**：2015-03-25
基金项目：国家自然科学基金资助项目(61173187, 61173188, 11301002)；高等学校博士学科点专项科研基金资助项目(20133401110004)；安徽省自然科学基金资助项目(11040606M141, 1408085QF107)；安徽大学“211”基金资助项目(33190187, 17110099)
Foundation Items: The National Natural Science Foundation of China (61173187, 61173188, 11301002); Specialized Research Fund for the Doctoral Program of Higher Education (20133401110004); The Natural Science Foundation of Anhui Province (11040606M141, 1408085QF107); The “211” Project of Anhui University (33190187, 17110099)

个密钥)，并存储在本地。在每一次验证过程中，用户需要揭露一个密钥给云服务器，要求其利用该密钥计算整个数据的消息验证码并返回。用户对比本地存储的 MAC，从而判断数据的完整性。但这种方法只能够支持 k 次验证，并且用户需要存储大量的验证信息。为此，Juels 等提出了 POR(proof of retrievability)模型[1]，通过对预处理数据阶段中随机嵌入的哨兵进行挑战，概率性地检测数据的完整性。但这种方法也只能够支持有限次的验证。Ateniese 等首次提出了 PDP(provable data possession)模型[2]，在初始阶段用户为每个数据块生成同态标签，云存储服务器在接收到用户发出的挑战后将数据标签和数据内容进行聚合并发送给用户验证。该模型通过标签聚合减少了认证过程中的计算和通信量，并且支持无限次验证。作者给出了 2 个实际可用的 PDP 协议：EPDP(efficient PDP)和 SPDP (secure PDP)，支持公开验证和数据追加。Shacham 等对 PDP 模型的安全模型进行了完善[3]。进一步地，对数据动态更新的支持是完整性检测非常重要的属性。一些文献利用认证跳跃链表和 RSA 认证树[4]、MHT 认证结构[5,6]、2-3 树[7]、索引列表[8]等动态结构实现了 PDP 模型中对数据以块为单位进行增加、删除、修改操作的支持。考虑到用户端较弱的计算能力，文献[6,9,10]在 PDP 模型中引入了可信但好奇的第三方审计者 TPA(third party auditor)作为代理验证方。文献[11~15]利用云存储服务器副本分布式存储的特点，提出了基于多副本的高效完整性检测方法。为了减少基于 PKI 的完整性检测系统中验证者需要额外承担的证书认证代价以及系统需要进行繁重的公钥证书管理工作，Zhao[16]和 Wang[17]等提出了基于身份签名的完整性检测协议。但这些 PDP 模型都需要用户在初始阶段生成大量用于验证的数据标签，对于计算和存储能力有限的移动设备并不适合。基于以上考虑提出了 IBPS-PDP (identity-based proxy signature PDP)模型，利用一个拥有较强计算能力的代理者来代替移动终端进行标签计算，为移动设备节省大量的计算时间。在实际应用中，代理方可以是用户公司或家中的计算机。同时，使用同态签名技术和 PDP 模型中的概率检测机制来快速地检测代理签名的正确性。进一步地，采用基于身份的签名以减少移动终端证书询问和认证的计算及通信代价。

2　系统与安全模型

2.1　系统模型

基于身份代理签名的完整性检测协议 IBPS-PDP 能够高效地检测存储到远程云服务器中的数据是否完整。不同于传统 PDP 模型，本文引入了代理签名方代替用户生成签名，以减少用户的计算代价。IBPS-PDP 系统模型如图 1 所示。

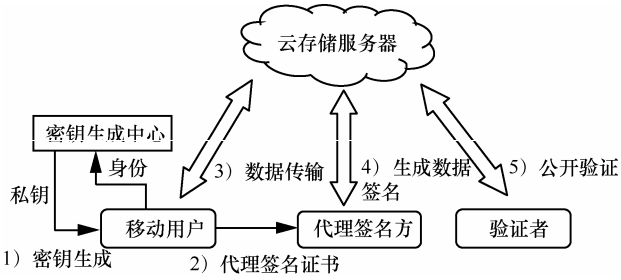


图 1　IBPS-PDP 系统模型

该模型包括移动用户(mobile user)、云存储服务器(cloud storage server)、代理签名方(proxy party)、私钥生成中心(private key generator)，验证者(verifier)5 类参与者，与 PDP 模型相比增加了代理签名方。各参与者具体定义如下。

1) 移动用户：数据拥有者，拥有需要存储到云端的大量数据，利用云存储服务器提供的各种接口执行数据操作。

2) 云存储服务器：拥有大量存储和计算资源的分布式存储系统，为用户提供数据存储、计算、审计等服务。

3) 代理签名方：拥有较强计算能力的代理签名方，为 Mobile User 提供代理签名服务。

4) 私钥生成中心：接受用户发送的身份 ID，计算对应的私钥并通过安全通道传给用户。

5) 验证者：验证云存储服务器上数据的完整性。

在该模型中，移动用户上传数据到云存储服务器，由计算能力较强的代理签名方生成有效数据签名以减少移动用户的计算负担。移动用户成功验证数据签名后删除本地数据。验证者利用数据签名的同态性以较小的通信和计算代价验证远程数据的完整性。

2.2　IBPS-PDP 协议定义

定义 1　IBPS-PDP 由 9 个算法组成，具体定义如下。

1) Setup(1^k)：系统建立算法，由 PKG 运行。输入系统安全参数 1^k，输出系统公共参数 *params*，系统公钥 P_{pub} 和系统主密钥 *s*。

2) Extract(1^k, *params*, *s*, *ID*):输入系统安全参数 1^k，系统公共参数 *params*，系统公钥 P_{pub}，系统主密钥 *s*，用户身份 *ID*，输出用户身份 *ID* 对应的私钥 S_{ID}。

3) Delegate(*ID*,ω)：输入原始签名人身份 *ID*，代理授权文件 ω，输出由用户 *ID* 私钥签名的代理授权证书 *W*。

4) DVerify(*ID*,*W*)：输入原始签名人身份 *ID*，代理授权证书 *W*，判断 *W* 是否是一个有效代理授权证书。如果判断证书有效，则输出 Accept，否则输出 Reject。

5) PKGen(S_{ID},*W*)：输入代理签名人的私钥 S_{ID}，代理授权证书 *W*，输出代理签名私钥 *SP* 和公钥 *QP*。

6) PSign(*SP*,ω,*F*)：输入代理签名私钥 *SP*，代理授权文件 ω，待签名数据 *F*，输出 *F* 的代理签名 σ。

7) PVeri(ω,Φ,*F*,*chal*)：由原始签名人执行，输入数据 *F*，代理授权文件 ω，验证挑战 *chal*，由云存储服务器返回的聚合签名 Φ。如果判断代理签名有效，则输出 Accept，否则输出 Reject。

8) GenProof(*F*,*chal*, σ)：由云存储服务器运行，输入数据 *F*，数据块签名 σ，挑战 *chal*，输出完整性证据 *Proof*。

9) VerifyProof(*chal*,*Proof*)：输入挑战 *chal*，完整性证据 *Proof*。如果验证者判断数据是完整的，则输出 Accept，否则输出 Reject。

2.3 安全模型

本文将在随机预言机模型下证明 IBPS-PDP 协议中签名方案的安全性，并通过 2 个游戏证明 IBPS-PDP 能够正确检测数据的完整性。

定义 2 令 G_1 表示具有素阶数 *q* 的加法循环群，G_2 表示具有同样阶的乘法循环群。双线性映射 $e:(G_1,G_1)\rightarrow G_2$ 是一个具有下面性质的映射。

1)双线性性：$\forall Q_1$，$Q_2\in G_1$ 和 *a*，$b\in Z_q^*$，有 $e(aQ_1,bQ_2)=e(Q_1,Q_2)^{ab}$；

2) 非退化性：$\exists Q_1$，$Q_2\in G_1$ 使 $e(Q_1,Q_2)\neq 1$；

3) 可计算性：存在一个高效的算法 E 能够计算 *e*。

从 1)可以得到双线性映射的下述性质

$\forall Q_1$，Q_2，$Q_3\in G_1$，$e(Q_1,Q_2+Q_3)=e(Q_1,Q_2)e(Q_1,Q_3)$

定义 3 CDH(computational Diffie-Hellman)问题：给定 *a*，$b\in Z_q^*$，*P*，*aP*，*bP*$\in G$，计算 *abP*$\in G_1$。其中，G_1 是大素阶数 *q* 的加法循环群。

在本文中，选择 G_1 使在该群中 CDH 问题是困难的。

根据文献[17]中安全模型的定义，认为完整性检测协议是安全的，如果满足：不存在攻击者在多项式时间内能够以不可忽略的概率在某个数据块集合上欺骗验证者。相应地，针对新模型进行如下安全定义。

定义 4 如果任何一个概率多项式攻击者 *O* 能够赢得 IBPS-PDP 游戏 Game1 的概率是可忽略的，那么认为 IBPS-PDP 中基于身份的代理签名协议存在性不可伪造。

定义 5 在代理签名存在性不可伪造的情况下，如果任何一个概率多项式攻击者 *O* 能够赢得 IBPS-PDP 游戏 Game2 的概率是可忽略的，那么认为 IBPS-PDP 是安全的。

其中，攻击者 *O* 拥有适应性选择消息，适应性选择授权文件，适应性选择身份的能力。

Game1 攻击者 *O* 和挑战者 *C* 之间的游戏 Game1 描述如下。

1) 初始化：挑战者 *C* 生成系统参数并将其发送给攻击者 *O*。

2) 询问：攻击者 *O* 向挑战者 *C* 进行如下询问。

Hash 询问：攻击者 *O* 询问散列函数，挑战者 *C* 返回相应的散列值，并保存该散列值。

Extract 询问：攻击者 *O* 可以向挑战者询问任何一个身份 *ID* 的私钥，挑战者 *C* 返回对应的私钥 S_{ID}。

Delegate 询问：攻击者 *O* 要求获得授权文件 ω 的代理授权证书，其中授权者身份为 *ID*。挑战者 *C* 返回一个有效的代理授权证书 *W*，并将(*ID*,ω,*W*)添加到列表 *DList* 中。

PKGen 询问：攻击者 *O* 要求获得代理签名人身份为 *ID*，代理授权证书为 *W* 的代理签名私钥。挑战者 *C* 根据 *ID*、*W*，输出代理签名私钥 *SP*，代理签名公钥 *QP*，发送给攻击者 *O*，并将(*ID*,*W*,*SP*,*QP*)添加到 *GList* 中。

PSign 询问：攻击者 *O* 询问(*W*, *m*)的代理签名，挑战者 *C* 对 *m* 进行签名并返回一个有效的代理签名 σ，将 σ 发送给 *O* 并添加(*W*,*m*,σ)到列表 *PSList* 中。

3) 响应：当攻击者 O 认为上述过程可以结束时，O 将给出一个输出，如果 O 的输出满足下列条件之一，则认为 O 在游戏 Game1 中获胜。

Case1 O 输出伪造(ID^*,W^*)，使 DVerify(ID^*, W^*) = Accept，且 $(ID^*,\omega^*,\cdot)\notin DList$，$(ID^*,W^*,\cdot,\cdot)\notin GList$，并且 O 没有对 ID^*进行私钥提取询问。

Case2 O 输出伪造(W^*,F^*,σ^*)，使 PVeri(ω^*, $\sigma^*,F^*,\cdot$) = Accept，且 $(W^*,m^*,\cdot)\notin PSList$，$(ID_j^*,W^*,\cdot,\cdot)\notin GList$，并且 O 没有对 ID_j^*进行私钥提取询问。D_j^*为从 ω^*中提取的代理签名方的身份 ID。

Game2 攻击者 O 和挑战者 C 之间游戏 Game2 描述如下。

在 Game2 中，攻击者 O 拥有所有数据块的有效数据签名。如果攻击者 O 能够输出关于挑战 $chal$($k1$,$k2$,$Name$)一个完整性证据 $Proof$，其中至少有一个被挑战的数据块 F_i 已经被修改，并且使 VerifyProof($chal$, $Proof$) = Accept，则攻击者在该游戏中获胜。

3　基于代理身份签名的完整性检测协议

IBPS-PDP 由 3 个阶段构成：初始化阶段、代理签名生成阶段和完整性证据验证阶段。

3.1　初始化阶段

在该阶段进行系统参数和用户私钥的生成。PKG 执行 Setup(1^k)算法生成公共参数 $params$ 及系统主密钥 s。在基于身份的签名系统中，移动用户需要向 PKG 询问用户签名私钥。对于输入用户 ID，PKG 执行 Extract(1^k,$params$,s,ID)算法为用户生成签名私钥 S_{ID}。用户公钥由其身份信息 ID 计算得到。初始化算法的实现如图 2 所示。

Setup(1^k)

1) 生成加法循环群 G_1，乘法循环群 G_2，双线性对 e: $(G_1,G_1)\rightarrow G_2$。P 为 G_1 的生成元，q 为 G_1，G_2 的素数阶。定义加密散列函数：$H_1:\{0,1\}^*\rightarrow G_1$；$H_2:\{0,1\}^*\times G_1\rightarrow Z_q^*$；$H_3:\{0,1\}^*\times G_1\rightarrow Z_q^*$。

2) 定义 f 为伪随机函数，π 为伪随机置换函数：$f:Z_q^*\times\{1,2,\cdots,n\}\rightarrow Z_q^*$；$\pi:Z_q^*\times\{1,2,\cdots,n\}\rightarrow\{1,2,\cdots,n\}$。

3) 选择随机数 $s\in Z_q^*$，输出系统公钥为 $P_{\text{pub}}=sP$，$params=\{G_1,G_2,e,q,P,P_{\text{pub}},H_1,H_2,H_3,f,\pi\}$。PKG 保存私钥 s。

Extract(1^k,$params$,s,ID)

1) PKG 计算 $S_{ID}=sH_1(ID)$，通过安全通道将私钥 S_{ID} 发送给用户。

2) 用户验证私钥有效性：如果 $e(P_{\text{pub}},H_1(ID))\stackrel{?}{=}e(P,S_{ID})$ 成立，则接受 S_{ID}。否则拒绝。用户 ID 的公钥为 $Q_{ID}=H_1(ID)$

图 2　初始化阶段描述

3.2　代理签名生成阶段

1) 在该阶段进行数据签名的生成与验证。为了验证数据的完整性，移动数据拥有者需要为数据生成签名。在 IBPS-PDP 模型中，由于移动设备计算能力有限，数据拥有者委托给代理签名方生成数据签名，以减少计算代价。首先，数据拥有者选择代理签名方并生成代理授权证书。数据拥有者执行 Delegate(A,ω)算法生成代理授权证书 $W_{A\rightarrow B}$，其中，数据拥有者为原始签名人，身份为 A；代理签名方身份为 B；ω 为代理授权文件，描述如下授权关系：原始签名者和代理签名方的身份信息，允许签署的消息范围、期限以及签名参数 μ 等。然后，数据拥有者将 $W_{A\rightarrow B}$ 发送给 B，并上传数据 F 到云存储服务器中。

2) 代理签名方为数据生成代理签名。过程描述如下：代理签名方 B 接收到 $W_{A\rightarrow B}$ 并执行 DVerify(A,$W_{A\rightarrow B}$)算法以判断其是否有效。如果验证成功，则 B 接受代理委派。B 在接受代理委派后，执行 PKGen(S_{ID},W)算法以生成代理签名私钥 SP 和代理签名公钥 QP。接着，执行 PSign(SP,ω,F)算法为数据 F 生成代理数据签名 σ。B 将代理签名上传到云存储服务器。

3) 数据拥有者验证代理签名的有效性。移动用户 A 生成挑战 $chal$ 发送给云存储服务器,并执行 PVeri(ω,Φ,F,$chal$)算法验证代理签名是否有效，其中，Φ 为云存储服务器返回的聚合签名。如果验证成功，移动用户 A 对代理授权证书 $W_{A\rightarrow B}$、代理签名公钥 QP、数据名称 $Name$ 计算签名 T，上传 $\{W_{A\rightarrow B},QP,Name,T\}$ 到云存储服务器并删除本地数据。

这里用 F 表示要存储到云中的数据，并将数据划分为 n 个数据块，即 $F=(F_1,F_2,\cdots,F_n)$。同时，为了减少签名的计算和存储代价，将每个数据块 F_i 划分为 c 个子块，使每个子块的长度能够满足安全参数，即 $F_{i,j}\in Z_q^*$，则 $F_i=(F_{i,1},F_{i,2},\cdots,F_{i,c})$。$(N_i,i)$ 对应着不同的数据块 F_i，其中，N_i 表示 F_i 的名称，i 表示 F_i 的数据块序号。代理签名生成阶段的实现如图 3 所示。

A 对代理授权证书 $W_{A\rightarrow B}$、代理签名公钥 QP、数据名称 $Name$ 签名方法描述如下：A 随机选择 $r\in Z_q^*$，并计算 $U_F=rP$，$h=H_2((W_{A\rightarrow B}||QP||Name),U_F)$，$V_F=hS_A+rP_{\text{pub}}$，生成签名 $T=(U_F,V_F)$。

3.3　完整性证据验证阶段

在该阶段，验证者向云服务器发出挑战并

Delegate(A,ω)

1) 移动用户 A 私钥为 S_A，公钥为 Q_A。A 随机选取 $\mu_i \in Z_q^*$，$\mu=\{\mu_1,\mu_2,\cdots,\mu_c\}$。随机选择 $r \in Z_q^*$，计算 $U=rP$，$h=H_2(\omega,U)$，$V=hS_A+rP_{\text{pub}}$。

2) 输出 $W_{A\to B}=(\omega,U,V)$。

DVerify(A, $W_{A\to B}$)

B 计算 $h=H_2(\omega,U)$，如果 $e(P,V)\overset{?}{=}e(P_{pub},U+hQ_A)$ 成立输出 Accept。否则，输出 Reject。

PKGen(S_{ID},W)

1) B 的私钥为 S_B，公钥为 Q_B。B 从 $W_{A\to B}$ 中提取 U、V、ω。计算 $h=H_2(\omega,U)$，$SP=hS_B+V$，$QP=h(Q_A+Q_B)+U$。

2) 输出 SP，QP。

PSign(SP,ω,F)

1) B 随机选择 $r_0 \in Z_q^*$，并计算 $UP=r_0P_{\text{pub}}$。

2) 对每一个数据块：B 向云存储服务器发送数据块序号 i，由云存储服务器查询对应数据，并返回 $\tilde{F}_i=\sum_{j=1}^{c}\mu_j F_{i,j}$。$B$ 计算 $h_i=H_3(N_i\|i,UP)$，$VP_i=h_iSP+\tilde{F}_i r_0P_{\text{pub}}+r_0P_{\text{pub}}$。

3) 输出 $\sigma=\{\sigma_i|\sigma_i=(VP_i,UP)\}_{1\leqslant i\leqslant n}$

PVeri(ω, Φ,F,$chal$)

采用概率检测的方式以减少验证计算量，挑战 l 个数据块的代理签名。

1) A 随机选择 $k0 \in Z_q^*$，计算 $I=\{b_i\,|\,b_i=\pi_{k0}(i),1\leqslant i\leqslant l\}$ 作为挑战的数据块序号，$chal$=($k0$,$Name$)。

2) 云存储服务器计算 $I=\{b_i\,|\,b_i=\pi_{k0}(i),1\leqslant i\leqslant l\}$，$VP'=\sum_{b_i\in I}VP_{b_i}$；$UP'=\sum_{b_i\in I}UP$。输出 Φ={VP',UP'}并发送给用户 A。

3) A 计算：$TP=\sum_{j=1}^{c}\mu_j\sum_{b_i\in I}F_{b_i,j}$；$h=\sum_{b_i\in I}H_3(N_{b_i}\|b_i,UP)$。如果 $e(P,VP')\overset{?}{=}e(P_{pub},UP'+TP\cdot UP+hQP)$ 成立，输出 Accept。否则，输出 Reject。

图 3　代理签名生成阶段描述

验证者 D 随机生成($k1$,$k2$)，$k1$，$k2 \in Z_q^*$。$I=\{b_i\,|\,b_i=\pi_{k1}(i),1\leqslant i\leqslant c\}$ 表示将要挑战的数据块序号的集合，$COE=\{a_i\,|\,a_i=f_{k2}(i),b_i\in I\}$ 表示随机挑战系数集合，$chal$=($k1$,$k2$,$Name$)。

GenProof(F,$chal$, σ)

该算法由云服务器执行。

1) 计算 $I=\{b_i\,|\,b_i=\pi_{k1}(i),1\leqslant i\leqslant c\}$，$COE=\{a_i\,|\,a_i=f_{k2}(i),b_i\in I\}$。

2) 计算 $\tilde{F}_j=\sum_{b_i\in I}a_iF_{b_i,j}$，$VP'=\sum_{b_i\in I}a_iVP_i$，$UP'=\sum_{b_i\in I}a_iUP$。

3) 输出 $Proof=(W_{A\to B},QP,Name,T,VP',UP',UP,\{\tilde{F}_j\}_{1\leqslant j\leqslant c})$。

VerifyProof($chal$,$Proof$)

1) D 验证 T 是否为($W_{A\to B}$||QP||$Name$)的有效签名。计算 $h=H_2((W_{A\to B}\|QP\|Name),U_F)$，如果 $e(P,V_F)\overset{?}{=}e(P_{pub},U_F+hQ_A)$ 成立，计算：$h=\sum_{b_i\in I}a_iH_3(N_{b_i}\|b_i,UP)$。

2) D 判断：

$e(P,VP')\overset{?}{=}e(P_{\text{pub}},UP'+(\sum_{j=1}^{c}\mu_j\tilde{F}_j)UP+hQP)$

是否成立。如果验证成功，输出 Accept。否则输出 Reject。

图 4　完整性验证阶段描述

验证数据完整性。IBPS-PDP 支持数据完整性的公开验证，任何用户都可以挑战云存储服务器，检测数据是否被修改。验证者生成挑战信息 $chal$ 并发送给云存储服务器。云存储服务器执行 GenProof(F,$chal$, σ) 算法以计算完整性证据 $Proof$，并返回 $Proof$ 给 D 作为完整性证据。D 执行完整性证据验证算法 VerifyProof($chal$, $Proof$)，如果验证成功，则 D 认为数据 $Name$ 在云存储服务器上是完整的。完整性证据验证阶段的实现如图 4 所示。

4　方案证明和性能分析

4.1　方案正确性

如果 PKG、Proxy Party、CSS、Mobile User 以及 Verifier 是诚实的，则对于挑战 $chal$=($k1$,$k2$, $Name$)，CSS 返回的完整性证据 $Proof$ 就能够通过完整性检验式 $e(P,VP')\overset{?}{=}e(P_{\text{pub}},UP'+(\sum_{j=1}^{c}\mu_j\tilde{F}_j)\ UP+hQP)$，在这种情况下认为 IBPS-PDP 协议是正确的。正确性证明如下

$$
\begin{aligned}
&e(P_{\text{pub}}, UP' + (\sum_{j=1}^{c} \mu_j \tilde{F}_j)UP + hQP) \\
&= e(P_{\text{pub}}, \sum_{b_i \in I} a_i (UP + (\sum_{j=1}^{c} \mu_j \tilde{F}_j)UP) + \\
&\sum_{b_i \in I} a_i H_3(N_{b_i} \| b_i, UP) QP) \\
&= e(sP, \sum_{b_i \in I} a_i r_0 P + (\sum_{j=1}^{c} \mu_j \sum_{b_i \in I} a_i F_{b_i,j}) r_0 P + \\
&(\sum_{b_i \in I} a_i H_3(N_{b_i} \| b_i, UP)) QP) \\
&= e(P, \sum_{b_i \in I} a_i (r_0 sP + (\sum_{j=1}^{c} \mu_j F_{b_i,j}) r_0 sP + \\
&(H_3(N_{b_i} \| b_i, UP)(h(sQ_A + sQ_B) + srP)) \\
&= e(P, \sum_{b_i \in I} a_i (r_0 P_{\text{pub}} + (\sum_{j=1}^{c} \mu_j F_{b_i,j}) r_0 P_{\text{pub}} + \\
&(N_{b_i} \| b_i, UP) SP)) \\
&= e(P, \sum_{b_i \in I} a_i VP_{b_i}) = e(P, VP')
\end{aligned}
$$

4.2　方案安全性证明

本文从 3 个方面证明 IBPS-PDP 是安全的：1)单个签名是不可伪造的，包括存在性授权不可伪造性和存在性代理签名不可伪造性；2)在代理签名过程中保护数据隐私；3)完整性证据的不可伪造性。在单个签名不可伪造的情况下，攻击者不能够伪造一个有效的证据以不可忽略的概率通过验证者的验证。

定义 6　一个基于身份的代理签名协议 IBPS 在适应性选择消息攻击，适应性选择授权文件攻击以及适应性选择身份攻击下是$(t, \varepsilon, q_{H_1}, q_{H_2}, q_{H_3}, q_{\text{Ext}}, q_{\text{D}}, q_{\text{PG}}, q_{\text{PS}})$安全的（如果不存在一个攻击者$(t, \varepsilon, q_{H_1}, q_{H_2}, q_{H_3}, q_{\text{Ext}}, q_{\text{D}}, q_{\text{PG}}, q_{\text{PS}})$能够攻破它）。

定理 1　设在随机预言机模型下，若存在概率多项式时间的攻击者 A_0 执行 2.3 节中描述的游戏 Game1，并在多项式时间 t 内以不可忽略的概率 ε 得到 Case1 或 Case2 的结果，则存在概率多项式时间的攻击者 A_1，能够在 t'时间内以不低于 ε'的概率解决 CDH 难题。令 t_{mul} 为 G_1 上的标量乘法和求逆的时间，$t_{Z_q^*}$ 为在 Z_q*上执行乘法的时间，则 $\varepsilon' = \frac{\varepsilon}{q_{H_1}}$，$t' \leqslant t + (q_{H_1} + q_{H_2} + q_{\text{Ext}} + 3q_{\text{D}} + 2q_{\text{PG}} + 4q_{\text{PS}})t_{\text{mul}} + (cq_{H_3})t_{Z_q^*}$。

证明　不失一般性，假设 A_0 以某个 ID 作为输入进行查询时，一定是该 ID 访问过 H_1 查询。攻击者 A_1 为 A_0 模拟运行环境，并且 A_1 可以访问存放在云存储服务器上的所有数据。

1) 挑战者生成加法循环群 G_1，乘法循环群 G_2，并构造双线性映射 e：$(G_1, G_1) \rightarrow G_2$。其中 G_1 的生成元为 P，阶为素数 q。挑战者随机选取 x，$y \in Z_q$*，输出$(G_1, G_2, e, P, q, xP, yP)$作为 A_1 的输入。

2) A_1 设置系统公共参数为$(G_1, G_2, e, P, q, P_{\text{pub}}, H_1, H_2, H_3)$，其中，$P_{\text{pub}} = xP$。

3) A_1 随机选取 t，$1 \leqslant t \leqslant q_{H_1}$。

4) A_1 维护列表 $H_1List(ID, b)$。A_1 令 $\gamma=1$，并为 A_0 模拟以下各种查询。

H_1 查询。对输入 ID，A_1 检查 $H_1(ID)$是否已经定义，如果没有，则定义如下：如果 $\gamma=t$，则 $H_1(ID)=yP$，令 $ID_t=ID$；否则 A_1 随机选择 $b \in Z_q$*，计算 $H_1(ID)= bP$。A_1 计算 $\gamma=\gamma+1$，添加记录(ID, b)到 H_1List，并返回 $H_1(ID)$给 A_0。

H_2 查询。对于询问(m, U)，A_1 检查 $H_2(m, U)$是否已经定义，如果没有，则随机选取 $b \in Z_q$*，定义 $H_2(m, U)=b$，返回 $H_2(m, U)$给 A_0。

H_3 查询。A_1 维护列表 $H_3List(N, i, UP, h)$。对于 A_0 的询问(N, i, UP)，A_1 查询$(N, i, UP, \bullet)$是否已经定义，如果定义，则返回 $H_3(N||I, UP)=h$。否则，A_1 随机选取 $h \in Z_q$*，添加记录(N, i, UP, h)到 H_3List 中，并返回 $H_3(N||i, UP)=h$。

Extract 查询。对输入 ID，如果 $ID=ID_t$，则 A_1 失败终止。否则，A_1 查询列表 $H_1List(ID, b)$中 ID 对应的 b 的值，计算 $S_{ID}=bP_{\text{pub}}$ 作为用户 ID 的私钥并返回给 A_0。

Delegate 查询。对输入的授权者身份 ID 和授权文件 ω，如果 $ID \neq ID_t$，A_1 利用 ID 的私钥 S_{ID} 为 ω 生成签名 $\sigma=(U, V)$。否则，A_1 模拟 ID_t 的授权查询。

A_1 随机选取 r，$h \in Z_q$*，并计算 $V = rP_{\text{pub}}$，$U = rP - hQ_{ID}$，则 ω 的签名为 $\sigma=(U, V)$，并定义 $H_2(\omega, U)= h$。如果 $H_2(\omega, U)= h$ 已经定义，A_1 因为发生冲突而失败终止。否则，A_1 将记录(ID, ω, W)添加到 $DList$ 中，并将 $W=(\omega, U, V)$返回给 A_0。

PKGen 查询。对于输入的代理者身份 ID_j 和代理信息 $W=(\omega, U, V)$，如果 $ID_j=ID_t$，则 A_1 失败终止。否则，A_1 计算代理签名私钥 $SP = H_2(\omega, U)S_{ID_j} + V$，公钥 $QP = H_2(\omega, U)(Q_{ID_j} + Q_{ID_i}) + U$。其中，$ID_i$ 为原始签名者的身份。A_1 将记录(ID_j, W, SP, QP)添加到

GList 中，并返回 *SP* 给 A_0。

PSign 查询。对于输入为 $W=(\omega,U,V)$和数据块 F，原始签名者和代理签名方身份分别为 ID_i、ID_j，数据块 F 的名称为 N，数据块序号为 b。如果 $ID_j \neq ID_t$，A_1 计算代理签名私钥 *SP* 并对 F 进行签名，生成代理签名(*UP*,*VP*)；否则，A_1 为 A_0 模拟代理签名。

A_1 查询 $H_2(m,U)$，并计算 $QP=H_2(\omega,U)(Q_{ID_t}+Q_{ID_i})+U$。$A_1$ 随机选取 $h, r\in Z_q^*$，并计算 $VP=rP_{\text{pub}}$，$UP=(1+F)^{-1}(rP-hQP)$。如果查询列表 H_3*List* 中已定义 $H_3(N\|b,UP)$，则 A_1 因为发生冲突而失败终止，否则 $\sigma=(UP,VP)$为一个有效的代理签名。A_1 添加记录(W,F,σ)到 *PSList* 中，添加 $H_3(N\|b,UP)=h$，并返回 σ 给 A_0。

1)A_0 与 A_1 进行交互直到 A_0 认为上述过程可以结束。A_0 输出(*ID**,ω*,*W**)或者(*W**,*F**,σ*)满足如下。

Case1 A_0 输出伪造(*ID**,ω*,*W**)，使 DVerify(*ID**, *W**)=Accept，且 $(ID^*,\omega^*,\cdot)\notin DList$，$(ID^*,W^*,\cdot,\cdot)\notin GList$，$ID^*=ID_t$，并且 A_0 没有对 *ID**进行私钥提取询问，则 A_1 能够得到一个有效的代理授权证书 $W^*=(\omega^*,U^*,V^*,h^*)$，其中，$h^*=H_2(\omega^*,U^*)$。同时，通过重置 H_2 查询，A_1 能够获得另一个有效的代理授权证书 $W_1^*=(\omega^*,U^*,(V^*)',(h^*)')$。这里 $(h^*)'\neq h^*$。

Case2 A_0 输出伪造$(W^*,F^*,\sigma^*)=((\omega^*,U^*,V^*),F^*,(VP^*,UP^*))$，使对于单个签名 PVeri$((\omega^*,\sigma^*),F^*,ID_i^*,ID_j^*)$=Accept，且$(W^*,F^*,\cdot)\notin PSList$，$(ID_j^*,W^*,\cdot,\cdot)\notin GList$，$ID_j^*=ID_t$，并且 A_0 没有对 ID_j^*进行私钥提取询问。其中，ID_i^*，ID_j^*分别为原始签名者和代理签名方的身份，数据块 F 的名称为 N，数据块序号为 b。这时，A_1 能够得到一个有效的代理签名$(VP^*,UP^*,(\omega^*,U^*,V^*),h^*,h_p^*)$，其中，$h^*=H_2(\omega^*,U^*)$，$h_p^*=H_3(N\|b,UP^*)$。同时，通过重置 H_3 查询，A_1 能够获得另一个有效的代理签名 $((VP^*)',UP^*,(\omega^*,U^*,V^*),h^*,(h_p^*)')$。这里 $(h_p^*)'\neq h_p^*$。

2) A_1 计算 *xyP* 如下。

Case1 $xyP=\left(V^*-(V^*)'\right)(h^*-(h^*)')^{-1}$

Case2 $xyP=\left(\left(VP^*-(VP^*)'\right)\left(h_p^*-(h_p^*)'\right)^{-1}-V^*\right)\cdot(h^*)^{-1}$

在 A_1 执行过程中，由于 U^*是随机的，所以 A_1 因为 $H_2(\omega^*,U^*)$定义冲突而终止的概率是可忽略的。A_1 的运行时间为 A_0 的运行时间 t 加上 A_1 对 A_0 各种查询访问的应答时间，因此 A_1 在预期时间 $t'\leqslant t+(q_{H_1}+q_{H_2}+q_{\text{Ext}}+3q_{\text{D}}+2q_{\text{PG}}+4q_{\text{PS}})t_{\text{mul}}+(cq_{H_3})t_{Z_p^*}$ 内以不可忽略的概率 $\varepsilon'=\dfrac{\varepsilon}{q_{H_1}}$ 成功求解 CDH 问题。

代理签名过程中数据隐私保护证明如下：由于在生成代理签名时，代理签名方只能够从云服务器中获得 $\tilde{F}_i=\sum_{j=1}^{c}\mu_j F_{i,j}$，该多项式有 q^{c-1} 个有效解集 $\{F_{i,j}|1\leqslant j\leqslant c\}$,因此代理签名方获得正确数据块 $F_{i,j}$ 的概率可以忽略。

通过 Game2 证明在单个代理签名不可伪造的情况下，攻击者能够伪造完整性证据 *Proof* 使 VerifyProof(*chal*,*Proof*)=Accept 的概率是可忽略的。

定理 2 如果验证者挑战的数据块中一些数据块已经被修改，那么证明者(云存储服务器)伪造一个完整性证据 *Proof* 使对于挑战 *chal*(*k*1,*k*2, *Name*)，VerifyProof(*chal*, *Proof*)=Accept 的概率是可以忽略的。

证明 假设攻击者 A_0 关于挑战 *chal*(*k*1, *k*2, *Name*)返回的完整性证据为 $Proof=(\omega,QP,Name,T,VP,UP,\{\tilde{F}_j\}_{1\leqslant j\leqslant c})$，如果 *Proof* 能够通过验证，则式(1)成立。

$$\prod_{b_i\in I}e(P,VP_{b_i})=\prod_{b_i\in I}e(P_{\text{pub}},UP+(\sum_{j=1}^{c}\mu_j F'_{b_i,j})UP+(H_3(N_{b_i}\|b_i,UP))QP) \tag{1}$$

设 G 为 G_2 的生成元，则存在 x_i，$y_i\in Z_q^*$满足

$$e(P,VP_{b_i})=G^{x_i} \tag{2}$$

$$e(P_{\text{pub}},UP+(\sum_{j=1}^{c}\mu_j F'_{b_i,j})UP+(H_3(N_{b_i}\|b_i,UP))QP)=G^{y_i} \tag{3}$$

由式(2)及式(3)得到

$$G^{\sum_{b_i\in I}a_i x_i}=G^{\sum_{b_i\in I}a_i y_i}$$

$$\sum_{b_i\in I}a_i(x_i-y_i)=0$$

由于单个签名是不可伪造的，因此，至少存在一个 i 使 $x_i\neq y_i$。假设存在 k 这样的(x_i,y_i)，则满足上述等式的挑战系数集合$\{a_i\}$共有 q^{k-1} 个。但由于所有的$\{a_i\}$都是随机选取的，则上述等式成立的概率为 $\dfrac{q^{k-1}}{q^c}<q^{-1}$，是可以忽略的。因此，攻击者输出一个可以通过验证的完整性证据 *Proof* 的概率是可以忽略的。

定理 3 证明者(云存储服务器)通过替换挑战

表 1　　**计算代价比较**

协议	数据拥有者	代理签名方	验证者	证书验证
文献[8]	$n((c+1)t_{\exp}+ct'_{\mathrm{mul}}+t_H)$	无	$(l+2)t_{\exp}+2t_{\mathrm{e}}+lt_{Z_q^*}+lt'_{\mathrm{mul}}$	是
ID-DPDP[17]	$n((c+1)t_{\exp}+ct'_{\mathrm{mul}}+ct_H)$	无	$(l+c+1)t_{\exp}+(l+c)t'_{\mathrm{mul}}+2t_{\mathrm{e}}$	否
IBPS-PDP	$ct_{Z_q^*}+2t_{\mathrm{e}}+2t_{\mathrm{add}}+2t_{\mathrm{mul}}+lt_H$	$n(t_{\mathrm{H}}+3t_{\mathrm{mul}}+t_{\mathrm{add}})$	$(l+c)t_{Z_q^*}+2t_{\mathrm{e}}+2t_{\mathrm{mul}}+2t_{\mathrm{add}}$	否

chal(*k*1, *k*2, *Name*)中的一些数据块序号伪造一个聚合数据标签，从而生成完整性证据 *Proof* 使对于挑战 *chal*，VerifyProof(*chal*,*Proof*)=Accept 的概率是可以忽略的。

证明　攻击者通过将挑战的数据块中被修改的部分替换为其他正确的数据块，生成一个有效的聚合标签。假设挑战为 *chal*(*k*1,*k*2,*Name*)，攻击者将 $I=\{b_i \mid b_i=\pi_{k1}(i),1\leqslant i\leqslant c\}$ 中的子集 $S_1\{d_1,d_2,\cdots,d_k\}$ 替换为 $S_2\{m_1,m_2,\cdots,m_k\}$，则由聚合签名正确性得到

$$e(P,\sum_{m_i\in S_2}a_iVP_{m_i})=e(P_{\mathrm{pub}},\sum_{m_i\in S_2}a_iUP+$$

$$\sum_{j=1}^{c}\mu_j\sum_{m_i\in S_2}a_iF_{m_i,j}UP+\sum_{m_i\in S_2}a_iH_3(N_{m_i}\|m_i,UP)QP) \quad (4)$$

如果 *Proof* 能够通过验证，则有

$$e(P,\sum_{m_i\in S_2}a_iVP_{m_i})=e(P_{\mathrm{pub}},\sum_{m_i\in S_2}a_iUP+$$

$$(\sum_{j=1}^{c}\mu_j\sum_{m_i\in S_2}a_iF_{m_i,j})UP+\sum_{d_i\in S_1}a_iH_3(N_{d_i}\|d_i,UP)QP) \quad (5)$$

由式(4)和式(5)得到 $\sum_{d_i\in S_1}a_iH_3(N_{d_i}\|d_i,UP)=\sum_{m_i\in S_2}a_iH_3(N_{m_i}\|m_i,UP)$。由于 H_3 是抗碰撞的，所以该式成立的概率为 $\frac{q^{k-1}}{q^k}=q^{-1}$。

因此，攻击者伪造一个可以通过验证的完整性证据 *Proof* 的概率是可以忽略的。

综上所述，本文提出 IBPS-PDP 中的代理签名协议是存在性不可伪造的，在此前提下，对于被挑战的数据块，如果数据块被修改或者丢失，云存储服务器输出一个可以通过验证的完整性证据的概率是可忽略的。

4.3　方案性能分析

本文将 IBPS-PDP 协议与文献[8]中提出的协议以及基于身份签名的 ID-DPDP[17]协议在计算代价上进行了详细的比较，结果如表 1 所示。

在表 1 中，t_{add} 为加法循环群上加法运算的时间，t_{e} 为双线性对的计算时间，$t_{\exp}$ 为乘法循环群上的指数运算时间，t'_{mul} 为乘法循环群上乘法运算时间。在 IBPS-PDP 协议中，移动终端将数据标签的生成过程交予代理签名方，因此初始阶段移动终端仅需要承担验证代理签名正确性的计算代价。由于采用概率检测的方法，移动终端抽样检查 *l* 个数据块的数据标签聚合值的正确性，故需要的计算时间为：*l* 次 H_3 的计算时间，*c* 次 Z_q*上运行乘法的时间，2 次双线性对运算时间，2 次 G_1 上标量乘运算和加法运算时间，故总的计算为 $ct_{Z_q^*}+2t_{\mathrm{e}}+2t_{\mathrm{add}}+2t_{\mathrm{mul}}+lt_{\mathrm{H}}$，这里忽略了在 Z_q*执行加法的运算时间。由表 1 中可以看出，相比于 ID-DPDP[17]和文献[8]中提出的协议，IBPS-PDP 协议将大量的标签计算工作交付给代理签名方，而只需要承担少量的标签验证工作，为移动终端节省了大量的时间。

5　结束语

本文提出了一种新的适用于移动云计算环境的完整性检测模型，并在该模型上采用基于身份的签名的方法设计了 IBPS-PDP 协议。IBPS-PDP 协议利用代理签名方代替移动终端为数据生成签名标签，保证了移动设备的性能。同时，由于采用基于身份的签名方法，IBPS-PDP 协议减少了系统证书的生成、管理工作以及移动终端设备的证书询问和认证代价，使协议的执行具有更高的效率。综上所述，在移动设备日益普及的今天，该协议具有很好的应用前景。

参考文献：

[1] JUELS A, KALISKI JR B S. PORs: proofs of retrievability for large files[A]. Proceedings of the 14th ACM conference on Computer and Communications Security[C]. ACM, 2007. 584-597.

[2] ATENIESE G，BURNS R，CURTMOLA R，*et al*. Provable data possession at untrusted stores[J]. Proceedings of CCS, 2007, 10: 598-609.

[3] SHACHAM H, WATERS B. Compact proofs of retrievability[A].Advances in Cryptology-ASIACRYPT[C]. Springer Berlin Heidelberg, 2008. 90-107.

[4] ERWAY C, KÜPÇÜ A, PAPAMANTHOU C, *et al.* Dynamic provable data possession[A].Proceedings of the 16th ACM Conference on Computer and Communications Security[C]. ACM, 2009. 213-222.

[5] WANG Q, WANG C, REN K, *et al*. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems,2011, 22(5): 847-859.

[6] WANG C, CHOW S S M, WANG Q, *et al*. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers,2013, 62(2): 362-375.

[7] ZHENG Q, XU S. Fair and dynamic proofs of retrievability[A]. Proceedings of the First ACM Conference on Data and Application Security and Privacy[C]. ACM, 2011. 237-248.

[8] YANG K, JIA X. An efficient and secure dynamic auditing protocol for data storage in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems , 2013, 24(9): 1717-1726.

[9] HAO Z, ZHONG S, YU N. A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(9): 1432-1437.

[10] WANG C, WANG Q, REN K, *et al*. Privacy-preserving public auditing for data storage security in cloud computing[A]. Proceedings IEEE INFOCOM[C]. IEEE, 2010. 1-9.

[11] ZHU Y, HU H, AHN G J, *et al*. Cooperative provable data possession for integrity verification in multicloud storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(12): 2231-2244.

[12] CURTMOLA R, KHAN O, BURNS R, *et al*. MR-PDP: multiple-replica provable data possession[A]. The 28th International Conference on Distributed Computing Systems[C]. IEEE, 2008. 411-420.

[13] BOWERS K D, JUELS A, OPREA A. HAIL: a high-availability and integrity layer for cloud storage[A]. Proceedings of the 16th ACM conference on Computer and Communications Security[C]. ACM, 2009. 187-198.

[14] HAO Z, YU N. A multiple-replica remote data possession checking protocol with public verifiability[A]. 2010 Second International Symposium on Data, Privacy and E-Commerce (ISDPE)[C]. IEEE, 2010. 84-89.

[15] HE J, ZHANG Y, HUANG G, *et al*. Distributed data possession checking for securing multiple replicas in geographically-dispersed clouds[J]. Journal of Computer and System Sciences, 2012, 78(5): 1345-1358.

[16] ZHAO J, XU C, LI F, *et al*. Identity-based public verification with privacy-preserving for data storage security in cloud computing[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2013, 96(12): 2709-2716.

[17] WANG H. Identity-based distributed provable data possession in multicloud storage[J]. Services Computing, IEEE Transactions on, 2015, 8(2): 328-340.

作者简介：

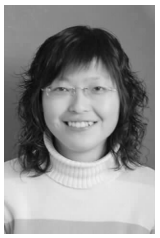
闫莉（1992-），女，安徽阜阳人，安徽大学硕士生，主要研究方向为信息安全、云计算安全。

石润华（1974-），男，安徽安庆人，安徽大学教授、博士生导师，主要研究方向为可证明安全的量子密码、保护隐私的多方协作计算、无线网络安全。

仲红（1965-），女，安徽宿州人，安徽大学教授、博士生导师，主要研究方向为网络与信息安全。

崔杰（1980-），男，河南淮阳人，安徽大学副教授、硕士生导师，主要研究方向为网络与信息安全。

张顺（1982-），男，安徽安庆人，安徽大学副教授、硕士生导师，主要研究方向为信息基复杂度与易处理性、量子框架信息与计算。



许艳（1982-），女，江苏泗洪人，博士，主要研究方向为公钥密码学、数字签名。