公有云中身份基多源 IoT 终端数据 PDP 方案

王化群[1]，刘哲[2]，何德彪[3]，李继国[4]

（1. 南京邮电大学计算机学院，江苏 南京 210023；2. 南京航空航天大学计算机科学与技术学院，江苏 南京 210016；
3. 武汉大学国家网络安全学院，湖北 武汉 430072；4. 福建师范大学数学与信息学院，福建 福州 350007）

摘 要：针对公有云中多源物联网（IoT）数据完整性验证问题，提出了一种身份基多源 IoT 终端数据可证明数据持有（ID-MPDP）方案。首先，给出了 ID-MPDP 方案的系统模型和安全模型的形式化定义。然后，使用 RSA 设计了具体的 ID-MPDP 方案。最后，给出了该方案的性能分析和安全性分析。性能分析和安全性分析结果表明，该方案是可证安全的、高效和可转换的，并具有以下优势：可用于多源 IoT 终端的数据完整性检测；具有较低的块扩展率；使用身份基公钥密码技术，消除了证书管理；满足可转换性。

关键词：云计算；身份基公钥密码体制；可证数据持有；物联网

中图分类号：TP393

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021077

Identity-based provable data possession scheme for multi-source IoT terminal data in public cloud

WANG Huaqun[1], LIU Zhe[2], HE Debiao[3], LI Jiguo[4]

1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
2. College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
3. School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
4. School of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China

Abstract: To solve the problem of multi-source IoT data integrity verification, identity-based provable data possession for multi-source IoT terminal in public cloud (ID-MPDP) was proposed. Firstly, the formal definitions of system model and security model of ID-MPDP were given. Then, the specific ID-MPDP scheme was designed by using RSA. Finally, the performance analysis and security analysis of ID-MPDP were given. Through performance analysis and security analysis, ID-MPDP was provably secure, efficient and convertible. It has the following advantages, such as it can be used for the integrity checking for multi-source IoT terminal data, it has lower block expansion rate, it eliminates the certification management cost by using the identity-based public key cryptography and it is convertible.

Keywords: cloud computing, identity-based public-key cryptography, provable data possession, IoT

1 引言

5G 等新兴技术的出现为物联网环境下数据转发的研究带来了新的机遇和挑战。5G 极大地提升了通信速度，能在无人驾驶、智能家居、智慧城市等方面广泛应用，5G 的诞生能改写物联网领域。云计算能够为远程计算机用户提供按需 IT 服务，是继互联网经济繁荣以来的又一个重要 IT 产业增长点。云平台能够为移动用户提供计算和存储资源，但存在外包存储数据被篡改的风险。

在公有云中，具体数据处理由公有云服务器（PCS，public cloud sever）执行。使用公有云具有

收稿日期：2020−01−14；修回日期：2021−03−08

基金项目：国家自然科学基金资助项目（No.61941116, No.61972294, No.62072104）

Foundation Item: The National Natural Science Foundation of China (No.61941116, No.61972294, No.62072104)

以下优势：因为公有云提供了硬件、应用程序和带宽成本，所以操作简单且成本低廉；提供可扩展性以满足需求；因为用户只需为使用的东西付费，所以不会浪费资源等。在公有云中，数据不受用户控制。PCS 对外包数据的安全性和隐私性负有更多责任。但是，为了保护自己的利益，不诚实的 PCS 可能会拒绝对丢失的数据负责。因此，确保数据所有者的远程数据完整性是重要的。在某些应用程序环境中，外包数据属于多个所有者，并且很敏感。在执行远程数据完整性检测时，必须确保验证者不能获取所存储文件内容的任何信息。同时，验证者必须确保检测的数据属于指定的多个所有者。但是，谁有能力检测远程数据的完整性？为了保护数据隐私，只能由数据所有者授权的验证者执行远程数据完整性检测。PKI（public key infrastructure）需要公钥证书的分发和管理，会产生相当大的消耗，例如证书生成、交付、吊销、更新等。身份基公钥密码体制消除了复杂的证书管理，为了提高效率，研究公有云中多源物联网（IoT, Internet of things）终端数据身份基完整性检测非常必要。

当海量数据存储在 PCS 中时，轻量级远程数据完整性概率检测是一种可行的方法。可证数据持有（PDP, provable data possession）是一个重要的远程数据完整性概率检测模型。基于 RSA 加密技术，Ateniese 等[1]在 2007 年提出了 PDP 模型和设计方法。然后，文献[2-3]提出了 PoR（proof of retrievability）模型。近年来，PDP 模型和方案设计得到了进一步发展。

利用区块链技术，研究人员相继提出了高效的私有 PDP 方案[4]、无可信中心 PDP 方案[5]、动态 PDP 方案[6]等。为解决复杂证书管理问题，基于双线性对或 RSA 假设，一些高效的身份基 PDP 方案被提出[7-8]。当远程数据需要动态处理时，如插入、删除等，动态 PDP 方案是必要的[9-10]。另外，可迁移的远程数据 PDP 方案[11]、匿名 PDP 方案[12]等相继被提出。

在某些特殊情况下，数据属于多个所有者。例如在工业物联网中，需要多个物联网节点协作交互，感知物理空间数据，并将这些数据存储在远程云服务器中。多所有者数据管理已成为重要的研究领域，在公有云中研究其完整性检测非常重要。现有 PDP 方案都不是多源 PDP, 因而不适合本文的场景。本文提出的 PDP 方案不仅适用于多源数据，并且是基于身份的，消除了复杂的证书管理，提高了效率。

本文主要贡献如下：提出了公有云中身份基多源 IoT 终端数据 PDP 概念，给出了形式化定义、系统模型和安全模型；使用 RSA 和一些相应的困难问题，设计了一种有效且安全的身份基多源 IoT 终端数据可证明数据持有（ID-MPDP，identity-based multi-source IoT terminals provable data possession）方案。安全性分析和性能比较表明，本文的 ID-MPDP 方案是可证安全和有效的。

2　系统模型、安全模型和预备知识

2.1　系统模型、安全假设和形式化定义

系统模型。本文系统由以下网络实体组成：系统管理器、私钥生成中心（PKG, private key generator）、PCS、数据所有者、验证者。

1) 系统管理器：生成系统参数。

2) PKG：可信第三方，为用户生成相应的私钥，也有助于验证者完成远程数据完整性检测。

3) PCS：具有大量存储空间和计算资源来处理用户数据。

4) 数据所有者：即多源 IoT 终端，将大量数据上传到 PCS 进行存储和计算。本文中，用户指 IoT 终端。

5) 验证者：通过与 PKG 交互检测远程数据完整性。

信任假设。PKG 得到所有其他实体的信任。PKG 不能与所有其他实体（包括 PCS、数据所有者、验证者）合谋。验证者不能与 PCS、数据所有者合谋。对于属于 n_1 个所有者的已处理文件 F，允许 PCS 与最多 n_1-1 个所有者合谋。

ID-MPDP 方案形式化定义如定义 1 所示。

定义 1　ID-MPDP 方案。其包括 5 个算法：SetUp、KeyGen、TagBlock、GenProof 和 CheckProof，具体如下。

1) $\mathrm{SetUp}(1^k)\rightarrow$系统参数。输入安全参数 k，输出系统参数。

2) $\mathrm{KeyGen(ID)}\rightarrow x_{\mathrm{ID}}$。输入用户身份 ID，输出用户私钥 x_{ID}。

3) $\mathrm{TagBlock}(F_j, x_{\mathrm{ID}_i}, \mathrm{ID}_i \in U)\rightarrow T_j$。输入数据块 F_j，U 中用户交互生成相应的标签 T_j，并将 (F_j, T_j) 上传到 PCS。其中，U 表示参与协议的用

户集。

4) GenProof(chal) $\rightarrow V$。输入挑战 chal，PCS 生成完整性检测证明 V，并将 V 发送给验证者。

5) CheckProof(chal,V) $\rightarrow$ 成功/失败。输入挑战 chal 和响应 V，如果有效，则返回“成功”；否则，返回“失败”。

2.2 安全模型

ID-MPDP 方案必须满足以下安全要求。

1) GenProof 只能由数据所有者授权的验证者执行。

2) 授权验证者不需要文件和标签的完整副本。

3) 修改某些存储的数据后，恶意证明者（即 PCS）只能以很小的概率通过验证者的检测。即使 PCS 与部分数据所有者串通，它仍然只能以很小的概率通过验证者的检测。

定义 2 不可伪造性。对于公有云中的多所有者数据，如果任何 PPT 敌手 A（即恶意 PCS 和部分所有者）都无法伪造 ID-MPDP 方案，A 赢得游戏的概率可以忽略不计。游戏来自挑战者 S 与敌手 A 的交互，游戏如下。

1) 初始化。生成系统参数和数据所有者的公私钥对，并将系统参数和数据所有者的公钥发送给 A。如果某个文件 F 存在 n_1 个数据所有者，S 将 n_1-1 个数据所有者的私钥发送给 A。

2) 请求。A 将预言机请求自适应地发送给 S，S 响应这些请求。

① 哈希预言机。收到哈希请求后，S 用相应的哈希值响应 A。

② KeyGen 预言机。收到身份后，S 会生成相应的私钥并将其发送给 A。

③ TagBlock 预言机。收到 TagBlock 请求 F_j 后，S 计算数据块标签 T_j 并将 T_j 发送给 A。

3) 挑战。S 向 A 发送挑战 chal，限制条件是，至少有一个挑战数据块没有发给 TagBlock 预言机。

4) 伪造。根据 chal，A 计算并返回证明 V 给 S。

在上面的游戏中，如果响应 V 以不可忽略的概率通过 S 的检测，则 A 获胜。

定义 3 (ρ,δ) 安全[8]。如果 PCS 篡改了全部数据块−标签对的 ρ 部分，验证者以至少 δ 的概率检测到该篡改，则 ID-MPDP 方案是 (ρ,δ) 安全的。

2.3 预备知识

1) RSA。RSA 密码系统由 Rivest、Shamir 和 Adleman 于 1978 年发明。在标准 RSA 中，$n=pq$ 是 2 个同样大小的大素数的乘积，公钥 e 满足 $1\leqslant e\leqslant\phi(n)$ 的整数，且 e 和 $\phi(n)$ 互素，即 $\gcd(e,\phi(n))=1$，其中 $\phi(n)=(p-1)(q-1)$ 是欧拉函数。求解方程 $ed=1\bmod\phi(n)$ 得到私钥 d。

定义 4 RSA 问题[13]。设 $n=pq$ 是 2 个同样大小的大素数的乘积，给定公钥 $n,e,y\in Z_n^*$，RSA 问题是计算的模数 y 第 e 个根 x，使 $x^e=y\bmod n$。

A 解决 RSA 问题的成功概率为

$$\mathrm{Succ}_{Z_n^*}^{\mathrm{RSA}}(A)=\Pr\left[y=x^e \bmod n \mid A(n,e,y)=x\in Z_n^*\right]$$

对于任何概率多项式时间敌手 A，如果 $\mathrm{Succ}_{Z_n^*}^{\mathrm{RSA}}(A)$ 可忽略不计，则 RSA 假设成立。

2) 二次剩余[13]。如果存在一个整数 x 使 $x^2\equiv z\bmod n$，则整数 z 为二次剩余；否则，z 为模 n 的非二次剩余。

3 ID-MPDP 方案与安全分析

3.1 具体方案

假设上传的数据块−标签对的最大数量为 $\hat{n}$，文件 F 可能通过使用纠错码（例如 Reed-Solomon 码）进行编码将上传到 PCS。F 分为 $\hat{n}$ 个块 $(F_1,\cdots,F_{\hat{n}})$。本文中所用符号和说明如表 1 所示。

表 1 符号和说明

符号	说明
p,q	2 个相同比特大小的大素数
n	$n=pq$
$\phi(n)$	$\phi(n)=(p-1)(q-1)$
Z_n^*	模 n 乘法群
QR_n	模 n 二次剩余群
主公钥 mpk	n,e,e'
主私钥 msk	d
f	伪随机函数
π	伪随机置换
H_1,H_2,H_3	密码哈希函数
$(\mathrm{ID}_i,x_{\mathrm{ID}_i})$	用户身份和私钥
n_1	数据所有者集合 U 的基数
$\hat{n}$	块的数量
$F=(F_1,\cdots,F_{\hat{n}})$	存储的文件 F 被分为 $\hat{n}$ 个块
T_j	对应于 F_j 的标签

SetUp(1^k)。n_1表示数据所有者的数目，k表示安全性参数。选取 2 个素数e和e'，其中$2^k \leqslant e,e' \leqslant 2^{2k}$且$2^k n_1 < e < e'/n_1$。计算$(n,d)$对，其中，$n=pq$使$p=2p'+1$，$q=2q'+1$，$p$、$q$、$p'$和$q'$都是素数。$d=e^{-1} \bmod \phi(n)$，其中$\phi(n)=4p'q'$。设主公钥$\text{mpk}=(n,e,e')$，主私钥$\text{msk}=d$，$f$为伪随机函数，$\pi$为伪随机置换，$H_1$、$H_2$、$H_3$为密码哈希函数，具体如下。

$$f: Z_n^*\{1,2,\cdots,\hat{n}\} \to Z_n^*$$

$$\pi: Z_n^*\{1,2,\cdots,\hat{n}\} \to \{1,2,\cdots,\hat{n}\}$$

$$H_1: \{0,1\}^* \to \text{QR}_n$$

$$H_2: Z_n \to \{0,1\}^k$$

$$H_3: \{0,1\}^* \to \text{QR}_n$$

KeyGen(ID) $\to x_{\text{ID}}$。输入第i个用户的身份ID_i，PKG 计算其私钥$x_{\text{ID}_i}=H_1(\text{ID}_i)^{2d} \bmod n$，并通过安全通道发送$x_{\text{ID}_i}$给用户$\text{ID}_i$。

TagBlock $(F_j, x_{\text{ID}_i}, \text{ID}_i \in U) \to T_j$。$U$为参与协议的用户集，基数为$n_1=|U|$，$U=\{\text{ID}_1,\text{ID}_2,\cdots,\text{ID}_{n_1}\}$。对于数据块$F_j$。选定用户$\text{ID}_i$作为数据的上传者，$\text{ID}_i$计算如下。

1) 记$W_j=U \| j$，选择一个随机的$k_{j,i} \in QR_n$并计算$h_j=H_3(W_j)$，$a_{j,i}=(k_{j,i})^e \bmod n$。

2) 选取$r_{j,i} \in [0,e']$并计算$C_{j,i}=h_j^{r_{j,i}} a_{j,i}^{e'} \bmod n$，令$D_{j,i}=r_{j,i}$并广播$(\text{ID}_i, C_{j,i})$。

3) 收到$(\text{ID}_s, C_{j,s}), j \neq i$后，用户$\text{ID}_i$计算

$$C_j=\prod_{s=1}^{n_1} C_{j,s} \bmod n$$

$$z_{j,i}=k_{j,i}\left(x_{\text{ID}_i}\right)^{F_j+H_2(C_j)} \bmod n$$

并广播$\left(z_{j,i}, D_{j,i}\right)$。

4) 计算$z_j=\prod_{i=1}^{n_1} z_{j,i} \bmod n$和$D_j=\prod_{i=1}^{n_1} D_{j,i} \bmod n$。

5) 对于块F_j，输出标签$T_j=(z_j, C_j)$。然后，通过安全通道将D_j发送到验证者。

GenProof(chal) $\to V$。收到挑战 chal=(c,k_1,k_2)后，其中$1 \leqslant c \leqslant \hat{n}, k_1, k_2 \in Z_n^*$，PCS 计算如下。

1) 对于$1 \leqslant j \leqslant c$，PCS 计算

$i_j=\pi_{k_1}(j), \alpha_j=f_{k_2}(j)$

2) PCS 计算

$$\bar{z}=\prod_{j=1}^{c} z_{i_j}^{\alpha_j} \bmod n$$

$$\bar{z}=\prod_{s=1}^{n_1}\prod_{j=1}^{c} z_{i_j,s}^{\alpha_j} \bmod n$$

$$\bar{C}=\prod_{j=1}^{c} C_{i_j}^{\alpha_j} \bmod n$$

3) PCS 计算$\hat{\sigma}=\sum_{j=1}^{c} \alpha_j (F_{i_j}+H_2(C_{i_j}))$并将$V=(\bar{z}, \bar{C}, \hat{\sigma})$发送给验证者。

CheckProof(chal, V) $\to$ 成功/失败。根据挑战 chal=(c,k_1,k_2)和响应$V=(\bar{z}, \bar{C}, \hat{\sigma})$，验证者计算如下。

1) 对于$1 \leqslant j \leqslant c$，计算

$$i_j=\pi_{k_1}(j)$$

$$\alpha_j=f_{k_2}(j)$$

$$W_j=U \| j$$

$$y=\prod_{\text{ID}_i \in U} H_1(\text{ID}_i)^2 \bmod n$$

2) 验证者将$\hat{\sigma}$发送给 PKG。

3) PKG 计算$v=y^{-\hat{\sigma} e'}=y^{-(\hat{\sigma} \bmod \phi(n)) e' \bmod n}$，并将$v$发送给验证者。

4) 验证者检测以下等式是否成立。

$$\bar{C}=\bar{z} e e' \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j D_{i_j}} \bmod n$$

如果等式成立，验证者输出“成功”；否则，验证者输出“失败”。

3.2　安全分析

ID-MPDP 方案的正确性分析和安全性分析如下。

定理 1　如果上传的数据块–标签对有效，即数据所有者和验证者是诚实的，并且数据没有被篡改，那么上传的数据块–标签对能够通过验证者的完整性检测，即 CheckProof 满足正确性。

证明　正确性来自以下推导。

$$\bar{C}=\prod_{j=1}^{c} C_{i_j}^{\alpha_j} \bmod n = \prod_{j=1}^{c}\prod_{s=1}^{n_1} C_{i_j,s}^{\alpha_j} \bmod n =$$

$$\prod_{j=1}^{c}\prod_{s=1}^{n_1} h_{i_j}^{r_{i_j,s}\alpha_j} a_{i_j,s}^{e'\alpha_j} \bmod n =$$

$$\prod_{j=1}^{c}\prod_{s=1}^{n_1} k_{i_j,s}^{ee'\alpha_j} \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j \sum_{s=1}^{n_1} r_{i_j,s}} \bmod n=$$

$$\prod_{j=1}^{c}\prod_{s=1}^{n_1} k_{i_j,s}^{ee'\alpha_j} \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j D_{i_j}} \bmod n =$$

$$\prod_{j=1}^{c}\prod_{s=1}^{n_1} (z_{i_j,s} x_{ID_s}^{-(F_{i_j}+H_2(C_{i_j}))})^{ee'\alpha_j} \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j D_{i_j}} \bmod n =$$

$$\bar{z}^{ee'} y^{-\hat{\sigma}e'} \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j D_{i_j}} \bmod n =$$

$$\bar{z}^{ee'} v \prod_{j=1}^{c} H_3(W_{i_j})^{\alpha_j D_{i_j}} \bmod n$$

定理 2 基于 RSA(e)问题困难性假设，ID-MPDP 方案满足不可伪造性。即如果敌手可以在 t_1 时间内以 ε' 的概率破坏 ID-MPDP 方案，那么 RSA(e)问题可以在 $t < t_1 + t_2 + O(N_{H_1} + N_{H_2} + N_{H_3} + N_{\mathrm{KG}} + N_{\mathrm{TB}})$ 时间内以 $(1-\rho^{n_1})\varepsilon$ 的概率被解决，其中，$\rho \in (0,1)$，t_2 是常数，N_{H_1}、N_{H_2}、N_{H_3}、N_{KG}、N_{TB} 分别表示请求不同预言的时间，即 H_1-预言机、H_2-预言机、H_3-预言机、KeyGen-预言机、TagBlock-预言机。

证明 不失一般性，令存储的索引–数据块集合为 $\{(1,F_1),(2,F_2),\cdots,(n,F_n)\}$。$n_1$ 个数据所有者表示为 $O_1,O_2,\cdots,O_{n_1}$。设 S 为挑战者，A 为敌手（即 PCS 和 n_1-1 个数据所有者）。假设 n_1-1 个数据所有者拥有私钥集 $\{x_{\mathrm{ID}_j}, 1 \leqslant j \leqslant n_1-1\}$，公钥是其身份。存在一个不参与合谋的数据所有者 O_{n_1}。O_{n_1} 的身份为 ID_{n_1}，私钥未知，令 $L=\{\mathrm{ID}_1,\mathrm{ID}_2,\cdots,\mathrm{ID}_{n_1}\}$。所有数据所有者的私钥都是基于模数 n 实现的，给出 $(n,e,\tilde{y})$。S 的目标是找到 $b=\tilde{y}^{\frac{1}{e}} \bmod n$。$S$ 与 A 交互如下。

Setup。S 选择一个素数 e' 满足 $2^k \leqslant e' \leqslant 2^{2k}$ 和 $2^k n_1 < e < e'/n_1$。初始化 Tab_1、Tab_2、Tab_3。Tab_1 存储哈希函数 H_1 的请求响应，Tab_2 存储哈希函数 H_2 的请求响应，Tab_3 存储哈希函数 H_3 的请求响应。

Query。A 自适应地请求哈希函数 H_1、H_2、H_3 和标签预言机。

1) H_1-预言机。收到请求 $\mathrm{ID} \in \{0,1\}^*$ 时，S 检测 Tab_1。如果已对 ID 进行了 H_1 请求，则 S 检索记录的元组 $(\mathrm{ID},b,\delta_{\mathrm{ID}},y)$ 并返回 $H_1(\mathrm{ID})=y$。否则，S 抛掷硬币 b，$b=0$ 的概率为 ρ，$b=1$ 的概率为 $1-\rho$。根据不同的结果，S 执行以下过程。

① 当 $b=0$ 时，选择一个随机的 $\delta_{\mathrm{ID}} \in Z_n$ 并计算 $y=\delta_{\mathrm{ID}}^e$。令 $H_1(\mathrm{ID})=y$，S 将 $(\mathrm{ID},b,\delta_{\mathrm{ID}},y)$ 存储在 Tab_1 中。

② 当 $b=1$ 时，选择一个随机的 $\delta_{\mathrm{ID}} \in Z_n$ 并计算 $y=\tilde{\delta}_{\mathrm{ID}}^e$。令 $H_1(\mathrm{ID})=y$，S 将 $(\mathrm{ID},b,\delta_{\mathrm{ID}},y)$ 存储在 Tab_1 中。

2) H_2-预言机。收到请求 $C \in Z_n$ 后，检测 Tab_2。如果已对 C 进行了 H_2 请求，则 S 检索记录元组 (C,s_C) 并返回 $H_2(C)=s_C$。否则，S 选择一个随机值 $s_C \in \{0,1\}^k$ 并返回 $H_2(C)=s_C$。S 将记录 (C,s_C) 存储在 Tab_2 中。

3) H_3-预言机。对于请求 $W_j=U \| j$（j 是块 F_j 的索引），S 检测 Tab_3。如果已对 W_j 进行了 H_3 请求，则 S 检索记录的元组 (W_j,h_{W_j}) 并返回 $H_3(W_j)=h_{W_j}$。否则，S 选择一个随机的 $h_{W_j} \in \mathrm{QR}_n$ 并返回 $H_3(W_j)=h_{W_j}$。S 将记录 (W_j,h_{W_j}) 存储在 Tab_3 中。

4) KeyGen 请求。S 收到身份 ID 后，S 在 Tab_1 中查找条目 ID。如果不存在这样的条目，则 S 对身份 ID 执行 H_1 请求。否则，存在记录 $(\mathrm{ID},b,\delta_{\mathrm{ID}},y)$。基于不同的 b，S 执行以下过程。

① 当 $b=0$ 时，S 以 δ_{ID}^2 响应 A，即 $x_{\mathrm{ID}}=\delta_{\mathrm{ID}}^2$。

② 当 $b=1$ 时，S 中止游戏。

5) TagBlock-预言机。收到 (F_j,j) 和用户组 L 的 TagBlock 请求时，S 执行以下过程。对于 L，如果 S 可以从 KeyGen-预言机获得所有私钥，则 S 执行 TagBlock。否则，S 计算 $h=\tilde{y}^{e'}$。当 $\mathrm{ID}_i \in L$，S 执行以下过程。

①当 $b_i=0$ 时，选择 $k_{i,j} \in \mathrm{QR}_n$，$r_{i,j} \in Z_e$ 并计算 $a_{i,j}=k_{i,j}^e$，$C_{i,j}=h^{r_{i,j}} a_{i,j}^{e'}$，令 $D_{i,j}=r_{i,j}$；否则，选择 $s_{i,j} \in Z_e$ 并计算 $C_{i,j}=\tilde{y}^{e's_{i,j}}$。

② 对于 $\mathrm{ID}_i \in L$，计算 $C_j=\prod_{\mathrm{ID}_i \in L} C_{i,j}$ 和 $c_j=H_2(C_j,L)+F_j$。

③当 $b_i=0$ 时，计算 $z_{i,j}=k_{i,j}x_{\mathrm{ID}_i}$；否则，计算

如下。

$$r_{i,j}=s_{i,j}+c_j \bmod e$$

$$l_{i,j}=(s_{i,j}+c_j-r_{i,j})e^{-1}$$

$$z_{i,j}=\tilde{y}^{l_{i,j}}\delta_j^{c_j}$$

令 $D_{i,j}=r_{i,j}$。

④ 对于 $\mathrm{ID}_i\in L$，计算

$$z_j=\prod_{\mathrm{ID}_i\in L}z_{i,j}\ ,D_j=\prod_{\mathrm{ID}_i\in L}D_{i,j}$$

⑤ 输出 (z_j,C_j,D_j)。集合 $\bar{S}$ 表示{ $j\,|\,(F_j,j)$ 已查询过 TagBlock 预言机}。

挑战。A 向 S 提出挑战 chal=(c,k_1,k_2)。对于 $1\leqslant j\leqslant c$，A 计算 $i_j=\pi_{k_1}(j)$，则 $i_1,i_2,\cdots,i_c$ 是 A 请求对其进行完整性检测的块索引。限制条件是 $\{i_1,i_2,\cdots,i_c\}\notin\bar{S}$。

伪造。为了响应 chal，A 生成有效证明 V=$(\bar{z},\bar{C},\bar{\sigma})$。通过使用分叉定理，$A$ 能够伪造另一个有效证明 $\hat{V}$=$(\hat{z},\bar{C},\hat{\sigma})$，该证明基于相同的挑战和随机性。

破解。假设 V 和 $\hat{V}$ 是有效伪造，以及 $y=\prod_{i=1}^{n_1}H_1(\mathrm{ID}_i)^2$，可得

$$\bar{C}=\bar{z}^{ee'}y^{-\bar{\sigma}e'}\prod_{j=1}^{c}H_3(W_{i_j})^{\alpha_jD_{i_j}}$$

$$\bar{C}=\hat{z}^{ee'}y^{-\hat{\sigma}e'}\prod_{j=1}^{c}H_3(W_{i_j})^{\alpha_jD_{i_j}}$$

$$(\bar{z}\hat{z}^{-1})^e=y^{\bar{\sigma}-\hat{\sigma}}$$

对于用户组 $L=\left\{\mathrm{ID}_1,\mathrm{ID}_2,\cdots,\mathrm{ID}_{n_1-1}\right\}$，可以分为 2 个独立的用户组 L_1 和 L_2，其定义如下。

$$L_1=\{\,\mathrm{ID}_i\in L\,|\,b=0\text{在}T_1\text{中对应的记录}\,\}$$

$$L_2=\{\,\mathrm{ID}_i\in L\,|\,b=1\text{在}T_1\text{中对应的记录}\,\}$$

设 L_2 的基数为 $|L_2|$，得到

$$(\bar{z}\hat{z}^{-1})^e=(\prod_{\mathrm{ID}_i\in L_1}\delta_{ID_i}^{2e}\prod_{\mathrm{ID}_i\in L_2}(\tilde{y}\delta_{\mathrm{ID}_i}^{e})^2)^{\bar{\sigma}-\hat{\sigma}}$$

$$(\bar{z}\hat{z}^{-1}\prod_{\mathrm{ID}_i\in L}\delta_{\mathrm{ID}_i}^{2(\bar{\sigma}-\hat{\sigma})})^e=\tilde{y}^{2|L_2|(\bar{\sigma}-\hat{\sigma})}$$

由于 $2^{k+1}n_1<e$，因此 $\gcd(e,|L_2|(\bar{\sigma}-\hat{\sigma}))=1$，使用扩展的欧几里得算法得到 $\tilde{y}$ 的 e 次根。

根据假设，A 在时间 t_1 内以概率 ε 获得伪造的 V。在上面的模拟中，必须完成 A 和 S 之间的交互。在交互中，假设 A 请求 H_1-预言机 N_{H_1} 次，请求 H_2-预言机 N_{H_2} 次，请求 H_3-预言机 N_{H_3} 次，KeyGen-预言机 N_{KG} 次，以及 TagBlock-预言机 N_{TB} 次。基于相同的块和随机性产生第二次伪造时，A 将花费时间 t_2。在模拟中，为了继续游戏，至少一个私钥对应于 $b=1$，概率为 $1-\rho^{n_1}$。这样，S 在时间 $t_1+t_2+O(N_{H_1}+N_{H_2}+N_{H_3}+N_{\mathrm{KG}}+N_{\mathrm{TB}})$ 内以 $(1-\rho^{n_1})\varepsilon$ 的概率破解 RSA 问题。基于 RSA 问题困难性假设，定理 2 得证。

定理 3　假设 PCS 已存储 $\hat{n}$ 个块标签对 $\left((F_1,T_1),(F_2,T_2),\cdots,(F_{\hat{n}},T_{\hat{n}})\right)$，并且修改了 d 个数据块-标签对。设挑战为 chal=(c,k_1,k_2)，所提的 ID-MPDP 方案是 $\left(\frac{d}{\hat{n}},1-\left(\frac{\hat{n}-d}{\hat{n}}\right)^c\right)$ 安全的。令 P_R 为检测到修改的概率，则

$$1-\left(\frac{\hat{n}-d}{\hat{n}}\right)^c\leqslant P_R\leqslant 1-\left(\frac{\hat{n}-c+1-d}{\hat{n}-c+1}\right)^c$$

证明　令 R 为验证者选择并已由 PCS 修改的数据块–标签对的数目，P_R 定义为 $P_R=\Pr\{R\geqslant 1\}$，则

$$P_R=\Pr\{R\geqslant 1\}=1-\Pr\{R=0\}=$$
$$1-\frac{\hat{n}-d}{\hat{n}}\frac{\hat{n}-1-d}{\hat{n}-1}\cdots\frac{\hat{n}-c+1-d}{\hat{n}-c+1}$$

所以有

$$1-\left(\frac{\hat{n}-d}{\hat{n}}\right)^c\leqslant P_R\leqslant 1-\left(\frac{\hat{n}-c+1-d}{\hat{n}-c+1}\right)^c$$

证毕。

4　性能分析

4.1　计算消耗

假设存在一个文件 F，它属于 n_1 个数据所有者。将文件 F 分成 $\hat{n}$ 个数据块。在 KeyGen 阶段，对于身份 ID，PKG 需要执行哈希函数 H_1 和以整数 n 为模的指数算法。在 TagBlock 阶段中，对于单个数据块，每个所有者将计算一次哈希函数 H_3、4 次模幂 n 计算和 $2n_1-1$ 次乘运算。对于挑战 chal=(c,k_1,k_2)，PCS 将执行 $2c$ 次幂运算和 $2(c-1)$ 次乘运算，执行 $2c-1$ 次加法和 c 次乘法。在 CheckProof 阶段中，验证者将执行 $c+n_1+1$ 次幂运

算和 $c+n_1-1$ 次乘运算。另外，PKG 将计算 $\hat{\sigma} \bmod \phi(n)$ 和一次幂运算。

4.2 通信消耗

RSA 的有效性取决于整数因子分解的计算困难性。1024 bit 的非对称密钥长度通常被认为是 RSA 加密算法所需的最小长度。

数据上传一次可以全部完成，仅考虑完整性请求和响应引起的通信消耗。在完整性请求中，验证者将挑战 chal=(c,k_1,k_2) 发送到 PCS。换句话说，验证者将 Z_n^* 中的 2 个元素和一个小整数 c 发送到 PCS。当 n 的长度为 1 024 bit 时，挑战的长度为 $1\,024\times 2+\text{lb}\hat{n}=2\,048+\text{lb}\hat{n}$ bit。为了响应验证者的挑战，PCS 生成证明 $V=(\bar{z},\bar{C},\hat{\sigma})$，其长度小于 $1\,024\times 3=3\,072$ bit。

4.3 降低数据块扩展率

不失一般性，假设上传的文件为 F，将其分为 $\hat{n}$ 个块 $F=(F_1,\cdots,F_{\hat{n}})$。为了实现远程数据完整性检测，必须为每个块生成相应的标签。在标签 $T_j=\left(z_j,C_j\right)=\left(\prod_{i=1}^{n_1} z_{j,i} \bmod n,\prod_{s=1}^{n_1} C_{j,s} \bmod n\right)$ 的生成过程中，如果已知生成器 x_{ID_i} 的阶数 $\hat{q}$，则 F_j 的长度必须小于 $\hat{q}$ 的长度。由于 x_{ID_i} 和 $\hat{q}$ 都是未知的，因此 F_j 的长度是任意的。因此，ID-MPDP 方案可以降低块扩展率。

4.4 可转换性

在 CheckProof 阶段，秘密值 D_i（$i=1,2,\cdots,n_1$）是必不可少的。当验证者获得授权并获得秘密值 D_i 时，它可以执行 GheckProof。否则，它将无法执行 CheckProof。因此，当数据所有者将 D_i 保密时，ID-MPDP 方案是私有远程数据完整性检测。当数据所有者将 D_i 公开时，ID-MPDP 方案就是公共远程数据完整性检测。当数据所有者将 D_i 发送给某些特殊的验证者时，ID-MPDP 方案就是指定验证者远程数据完整性检测。因此，该方案是可转换的。

4.5 方案比较

本节将 ID-MPDP 方案的计算和通信消耗与文献[14-15]方案进行对比。本文方案和文献[14-15]方案都是使用 RSA 设计的。由于哈希函数和加法效率较高，因此在比较中将它们省略。表 2 给出了 3 种方案的计算和安全性比较。在表 2 中，n_1 表示数据所有者数，Exp 和 Mul 分别表示乘运算和幂运算的时间消耗，Sig 和 VeriCert 分别表示生成签名和验证证书的时间消耗。在 PKI 中，验证来自证书颁发机构的证书是必不可少的。同时，本文还比较了它们的其他属性。文献[14-15]方案是在 PKI 和基于身份的密码学中设计的，需要认证管理。本文方案完全基于身份的密码学设计，不需要认证管理。同时，本文方案满足可转换性，文献[14-15]方案不满足可转换性。另一方面，本文方案可用于处理属于多个所有者的数据，文献[14-15]方案不能用于处理属于多个所有者的数据。

表 3 将本文方案的通信消耗与文献[14-15]方案进行了比较，包括以下三部分：标签大小、挑战大小和响应大小。在表 3 中，n 表示 RSA 中的安全大整数，$\hat{n}$ 表示总块数，n_1 表示所有者数。在文献[14-15]方案中，签名是必要的，用 $|\text{Sig}|$ 表示签名长度，用 $|\mu|$ 表示合计块数量。

从表 2 和表 3 可知，ID-MPDP 方案具有以下优点。

1) 该方案可用于多所有者数据完整性检测，而

表 2　3 种方案的计算和安全性比较（数据所有者数量为 n_1）

方案	TagBlock	CheckProof	PKI	基于身份	可转换性	多所有者
文献[14]方案	$5n_1\text{Exp}+3n_1\text{ Mul}+n_1\text{Sig}$	$n_1(c+2)\text{Exp}+cn_1\text{Mul}+n_1\text{VeriCert}$	需要	部分	否	否
文献[15]方案	$6n_1\text{Exp}+n_1\text{Mul}+n_1\text{Sig}$	$4n_1\text{ Exp}+n_1\text{Mul}+n_1\text{VeriCert}$	需要	部分	否	否
本文方案	$4n_1\text{ Exp}+(n_1+n_1^2)\text{Mul}$	$(n_1+c+2)\text{Exp}+(n_1+c)\text{Mul}$	不需要	是	是	是

表 3　3 种方案的通信消耗比较（数据所有者数量为 n_1）

方案	标签大小（数据所有者→PCS）	挑战大小（验证者→PCS）	响应大小（PCS→验证者）
文献[14]方案	$n_1(3n+2\|\text{Sig}\|)$	$n_1(\hat{n}+2n)$	$n_1(3n+2\|\text{Sig}\|+\|\mu\|)$
文献[15]方案	$n_1(2n+\|\text{Sig}\|)$	$n_1(\hat{n}+2n)$	$n_1(2n+\|\mu\|)$
本文方案	$2n$	$\hat{n}+2n$	$3n$

其他方案则不能。

2) 相对于文献[14-15]方案，该方案具有较低的数据块扩展率。

3) 利用基于身份的公钥密码学，消除了证书管理成本。

4) 该方案是可转换的，高效且满足更多安全性要求。

4.6 仿真实验

仿真实验采用 C 语言和 Miracl 库。PCS 和 PKG 运行在 DELL PowerEdge R420 服务器，该服务器性能如下。

① CPU：Intel R Xeon R processor E5-2400，E5-2400 v2 product families

② Physical Memory：8 GB DDR3 1 600 MHz

③ OS：Ubuntu 13.04 Linux 3.8.0-19-generic SMP i686

验证者运行在 PC 平台，性能如下。

① CPU：Intel(R) Core(TM) i5-3470 CPU@ 3.20 GHz

② Physical Memory：4.00 GB (3.36 GB available)

③ OS：Windows 7

TagBlock 阶段的计算时间如图 1 所示。

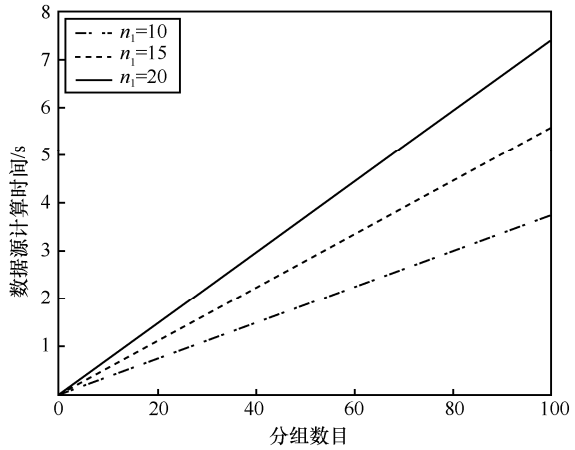


图 1　TagBlock 阶段的计算时间

GenProof 阶段和 CheckProof 阶段的计算时间如图 2 所示。在 GenProof 阶段，计算时间来自 PCS 和 PKG。在 CheckProof 阶段，计算时间来自验证者，n_1 表示数据源数目。

5 结束语

在公有云中，本文提出了一种基于身份的多所有者数据完整性检测的 ID-MPDP 方案，基于 RSA 构造了具体方案。安全性分析和性能分析表明，所提的 ID-MPDP 方案是可证明安全的和高效的。

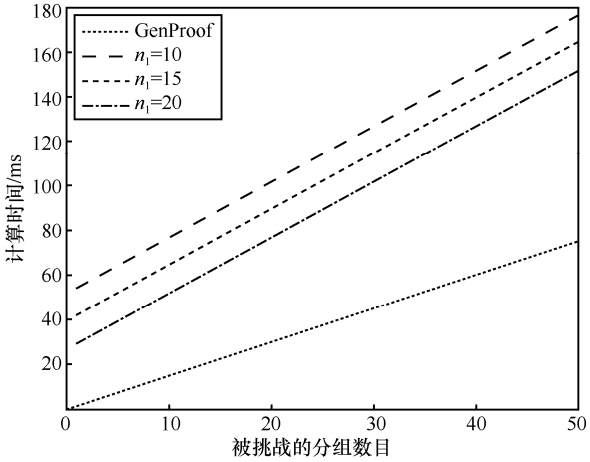


图 2　GenProof 阶段和 CheckProof 阶段的计算时间

参考文献：

[1] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 598-607.

[2] JUELS A, KALISKI B S J. PoRs: proofs of retrievability for large files[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 584-597.

[3] SHACHAM H, WATERS B. Compact proofs of retrievability[J]. Journal of Cryptology, 2013, 26: 442-483.

[4] WANG H Q, WANG Q H, HE D B. Blockchain-based private provable data possession[J]. IEEE Transactions on Dependable and Secure Computing, 2019, PP(99): 1.

[5] LI Y N, YU Y, CHEN R N, et al. IntegrityChain: provable data possession for decentralized storage[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(36): 1205-1217.

[6] CHEN R N, LI Y N, YU Y, et al. Blockchain-based dynamic provable data possession for smart cities[J]. IEEE Internet of Things Journal, 2020, 7(35): 4143-4154.

[7] NI J B, ZHANG K, YU Y, et al. Identity-based provable data possession from RSA assumption for secure cloud storage[J]. IEEE Transactions on Dependable and Secure Computing, 2020, PP(99): 1.

[8] WANG H Q, WU Q H, QIN B, et al. Identity-based remote data possession checking in public clouds[J]. IET Information Security, 2014, 8(32): 114-121.

[9] GUO W, ZHANG H, QIN S J, et al. Outsourced dynamic provable data possession with batch update for secure cloud storage[J]. Future Generation Computer Systems, 2019, 95: 309-322.

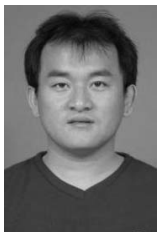
[10] ZHOU L, FU A M, YANG G M, et al. Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics[J]. IEEE

Transactions on Dependable and Secure Computing, 2020, PP(99): 1.
[11] WANG H Q, HE D B, FU A M, et al. Provable data possession with outsourced data transfer[J]. IEEE Transactions on Services Computing, 2019, PP(99): 1.
[12] WANG H Q, HE D B, YU J, et al. Incentive and unconditionally anonymous identity-based public provable data possession[J]. IEEE Transactions on Services Computing, 2019, 12(35): 824-835.
[13] 裴定一, 祝跃飞. 算法数论[M]. 第 2 版. 北京: 科学出版社, 2015.
PEI D Y, ZHU Y F. Algorithmic number theory. second edition[M]. Beijing: Science Press, 2015.
[14] YU Y, XUE L, AU M H, et al. Cloud data integrity checking with an identity-based auditing mechanism from RSA[J]. Future Generation Computer Systems, 2016, 62: 85-91.
[15] ZHANG J H, LI P Y, SUN Z, et al. ID-based data integrity auditing scheme from RSA with resisting key exposure[C]//Provable Security. Berlin: Springer, 2016: 83-100.

[作者简介]

王化群（1974− ），男，江苏南京人，博士，南京邮电大学教授、博士生导师，主要研究方向为应用密码学、区块链技术、未来移动通信安全等。

刘哲（1986− ），男，江苏南京人，博士，南京航空航天大学教授、博士生导师，主要研究方向为密码学、区块链技术、人工智能安全与应用等。



何德彪（1980− ），男，湖北武汉人，博士，武汉大学教授、博士生导师，主要研究方向为密码协议、信息安全、区块链技术与应用等。

李继国（1970− ），男，黑龙江富裕人，博士，福建师范大学教授、博士生导师，主要研究方向为密码学理论与技术、信息安全、云计算安全。