计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20150515
52(10):2422-2430,2015

基于部分授权的可证明数据持有性验证

钟 婷 韩 校 赵宇龙
(电子科技大学信息与软件工程学院 成都 610054)
(zhongting@uesct.edu.cn)

Provable Data Possession by Partial Delegation

Zhong Ting, Han Xiao, and Zhao Yulong
(*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu* 610054)

Abstract Provable data possession (PDP) is an important integrity checking technique in cloud storage. By using PDP, the client moves its data to cloud server and checks the possession of the data with constant computation. However, the client sometimes is not available to check its data possession. For example, the client wants to check its data which are stored in cloud server when he is in prison or at sea. In those cases, a convenient way to delegate the power of checking data possession to a proxy is necessary. In order to solve this problem, we propose a new provable data possession based on partial delegation (PDPPD). The PDPPD system model and security model are based on bilinear pairing and partial delegation. And the major feature of the proposed scheme is following: the client can delegate verification power to the proxy by sharing the converted secret key with the proxy, and the client can revoke or delete the proxy in an easy way at any time. Through our security analysis, the proposed scheme is provably secure. Compared with existing PDP schemes, the proposed scheme has less computation and communication overhead with the same level of security and also has wider application scenarios.

Key words provable data possession (PDP); bilinear pairing; partial delegation; provable security; storage security

摘 要 可证明数据持有性验证(provable data possession, PDP)是云存储中重要的完整性验证技术,采用可证明数据持有验证,客户可通过常量级运算验证云服务器是否诚实地持有客户数据.某些情况下,客户无法亲自验证云端的数据持有,此时客户需要授权代理对云端数据进行持有验证.针对上述问题,提出了一种基于部分授权的可证明数据持有验证方案(provable data possession based on partial delegation, PDPPD),新方案基于双线性对及部分授权技术支持数据拥有者直接通过密钥变形方式委任代理方进行数据持有验证,并且数据拥有者可以随时撤销或更换代理方,证明了方案的安全性.与现有数据持有性验证方案相比,新方案在保证相同安全强度的条件下,具有更小的计算量和通信量,且应用场景更加广泛.

关键词 可证明数据持有性验证;双线性对;部分代理签名;可证明安全;存储安全

中图法分类号 TP309.2

收稿日期:2015-06-13;**修回日期**:2015-08-17

基金项目:国家自然科学基金项目(61472064);四川省科技支撑基金项目(2015GZ0095);中央高校基本科研业务费基础研究项目(YGX2013J072)

随着云存储技术的快速发展,作为其重要组成部分的数据外包技术也得到越来越多的关注.数据外包是指客户(即数据拥有者)将数据存储到云服务提供商.相比于传统的数据存储方法,数据外包具有灵活性高、支持动态存储、存储维护成本低等特点.由于数据外包的远端存储方式,客户对数据的机密性、完整性和可用性有更高的要求.

可证明数据持有性验证(provable data possession, PDP)[1]以及数据可恢复性验证(proof of data retrievability, POR)[2]都是近年来研究较多的数据完整性验证技术,其主要目的是保证客户端的数据不被恶意毁坏或删除.通过使用PDP的同态验证标签,客户能够有效地验证其存储的数据是否被更改.

参照验证者身份,PDP方案可以分为2类:私有验证PDP方案及公开验证PDP方案.在私有验证PDP方案中,验证者需要提供私密信息(如私钥等)才能验证数据持有性.反之,公开验证PDP方案允许任何人无需私密信息对数据持有性进行验证.相对于公开验证PDP方案,私有验证PDP方案能更好保护数据隐私.

但在某些情况下,例如客户在监狱或者在海上旅行,客户本身并不能够对数据进行持有性验证.这时客户需要一种能够委托代理检验数据持有性的验证技术.

在现有PDP方案的基础上,我们提出了基于部分授权的可证明数据持有验证方案(provable data possession based on partial delegation, PDPPD).此方案中,客户将变形后的密钥传递给代理,以授权代理进行数据持有验证.本方案属于私有验证方案,拥有私钥的客户能进行数据持有验证.此外,利用双线性对的良好特性,客户通过部分授权使得代理方在无客户私钥的情况下也可进行数据持有验证.相比现有的数据持有验证方案,本方案在应用场景方面优于私有验证,在对数据隐私保护方面优于公开验证.该方案能够保证:1)代理无法获知关于客户密钥的信息;2)客户和代理均可以对数据持有进行有效的检查;3)方案在存储和带宽方面计算开销较小.

本文的主要贡献如下:

1) 基于部分授权,提出了PDPPD方案.该方案无需可信授权机构,客户与代理方均可实现数据持有验证.

2) 分析并证明了PDPPD方案的安全性.

3) 分析并评估了PDPPD方案的性能.

1　相关工作

2007年,Ateniese等人首先提出了数据持有验证的概念.对于客户存储在不可信服务器上的数据,客户只需抽样检查部分数据而不是检查全部数据就能够验证服务器是否完整地持有原始数据.Ateniese等人提出了2个可证明安全的PDP方案[1].他们的方案基于RSA模运算来产生证据,并没有考虑到数据的更新以及串谋攻击.之后,Ateniese等人又提出了基于对称加密技术的可扩展PDP方案[3].这种方案虽然支持动态验证,但却限制了更新和挑战的次数.同一时期,Curtmola等人提出了多副本的PDP方案,这种方案允许客户检查服务器是否诚实地存储一个文件的多个副本[4].与随后提出的多种PDP方案[5-8]类似,该方案扩展了PDP的应用,但存在着前面提到的问题.

与PDP方案相比,POR方案不仅仅确保远程数据的完整性,同时也能够恢复远程数据.Juels等人首先提出了具有强安全性可抵御各类攻击的POR方案[2].他们使用了支持公开验证的BLS签名技术.Dodis等人首先提出POR码的概念并给出理论分析[9].之后,学者又相继提出了一系列的POR方案[10-11].

随后,隐私保护和多用户并行验证成为可证明数据持有性验证的研究热点.Wang等人在文献[12]中首次提出具有隐私保护的数据持有验证方案,该方案在服务回传数据中加入了掩饰参数,有效防止了攻击者通过分析回传数据造成的隐私泄露攻击,同时该方案还支持高效的多任务验证.之后,Yang等人提出了支持动态多用户、多服务器及隐私保护的数据持有验证方案[13],该方案可以对来自多个服务器的多个用户进行高效的数据持有验证,同时支持数据动态更新及数据隐私保护.此外,Wang等人提出了一种支持群组动态更新及隐私保护的可证明数据持有验证方案[14],方案中数据拥有者是一个群组,组内成员拥有相同权限且对外完全匿名,该方案在保证数据隐私的前提下支持群组成员动态变化及数据的动态更新.

针对分布式云存储环境,文献[15]基于层次Hash索引提出了协作式数据持有验证(cooperative provable data possession, CPDP),以支持云服务器上的数据迁移,从而提高云存储服务的可扩展性.然而,后续研究[16]发现,CPDP存在严重的安全问题,因此目前尚无有效的分布式云存储环境下的协作式

数据持有验证方案.最近,针对 PDP 系统的实用性与效率也开展了大量研究工作[17-18].

对于代理加密,已经有了多年的研究.1996 年,Mambo 等人提出了代理签名的概念,并且提出了完全授权(full delegation)、部分授权(partial delegation)、担保授权(delegation by warrant)的代理分类[19].在部分授权技术中,首先由原始密钥产生代理密钥;然后通过授予代理签名者代理密钥,原始签名者将签名权限授予给代理签名者.其中代理密钥是由原始密钥变形并添加安全参数形成的.通过使用代理密钥,代理签名者能够代表原始签名者对消息进行签名.担保授权是另一种形式的代理加密技术[20],代理签名者通过使用授权担保来产生代理签名.担保包含了代理签名者的身份、授权的有效期以及授予代理签名权限的相关限制条件.

1997 年 Kim 等人提出了基于担保的部分授权方案(partial delegation by warrant)[21],该方案结合了部分授权与担保授权,方案安全性更高.此后,Zhang 提出了阈值代理签名方案[22].在该方案中,原始签名者将签名权限授予给 n 个代理签名者.在这 n 个代理签名者中,任意 t 个代理签名者可共同产生一个有效的代理.在 2001 年,Lee 等人用各种不同的攻击场景推翻了以前的一些代理签名方案,同时提出了一种更为安全的代理签名[23]方案.在 2005 年 Lu 等人提出了一种易撤销代理权限的代理签名方案[24].最近,Wang 提出了一种代理的 PDP 方案(proxy provable data possession, PPDP)[25].在这种方案中,数据拥有者能够授权代理对存储在不可信服务器上的远程数据进行验证的权限.此方案中,须先由可信授权机构(third party auditor, TPA)分配公钥和私钥给代理方,之后再由代理利用这对密钥来验证服务器的数据持有性.客户对代理进行授权必须通过可信第三方授权机构,但客户对代理的授权及撤销缺乏灵活度.

在现实的应用场景中,需要一种客户无需依赖可信第三方的授权机制.在这种机制下,客户能够自主地授予代理进行数据持有性验证的权限,代理能够高效地验证服务器端的数据持有性,服务器能够验证代理的合法性.

2 PDPPD 系统模型

2.1 预备知识

2.1.1 双线性对

$\mathbb{G}$,$\mathbb{G}_T$均是阶为 p 的循环乘法群,我们定义 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$是拥有如下属性的双线性映射:

1) 双线性.对于任意的 $g,g_1\in\mathbb{G}$,$a,b\in\mathbb{Z}_p$,有 $e(g^a,g_1^b)=e(g,g_1)^{ab}$.

2) 非退化性.存在 $g,g_1\in\mathbb{G}$ 使得 $e(g,g_1)\neq 1$.

3) 可计算性.对于 $g,g_1\in\mathbb{G}$,存在有效的算法来计算 $e(g,g_1)$.

当 g 是$\mathbb{G}$的生成元时,$e(g,g)$是$\mathbb{G}_T$的生成元.

双线性映射即为双线性对,双线性对可以由 Weil 对或 Tate 对生成[26-27].

2.1.2 Diffie-Hellman 困难问题

1) 计算性 Diffie-Hellman 困难问题(computational Diffie-Hellman assumption, CDH):已知(g,g^a,g^b),其中 $a,b\in\mathbb{Z}_p$,计算 g^{ab}.

2) 判定性 Diffie-Hellman 困难问题(decisional Diffie-Hellman assumption, DDH):已知(g,g^a,g^b,g^c),其中 $a,b,c\in\mathbb{Z}_p$,判断 $ab=c \bmod p$ 是否成立.

3) 双线性 Diffie-Hellman 困难问题(bilinear Diffie-Hellman assumption, BDH):对于$a,b,c\in\mathbb{Z}_p$,给定(g,g^a,g^b,g^c),计算 $e(g,g)^{abc}$.

4) Gap Diffie-Hellman 困难问题(gap Diffie-Hellman assumption, GDH):已知(g,g^a,g^b,g^c),$a,b,c\in\mathbb{Z}_p$,存在有效算法解决判定性 Diffie-Hellman 困难问题,但不存在算法解决计算性 Diffie-Hellman 困难问题.

2.2 定 义

在图 1 中我们给出了 PDPPD 方案的系统模型图,在表 1 中我们给出 PDPPD 方案涉及的符号表以及方案的定义,然后我们再介绍 PDPPD 的敌手游戏.

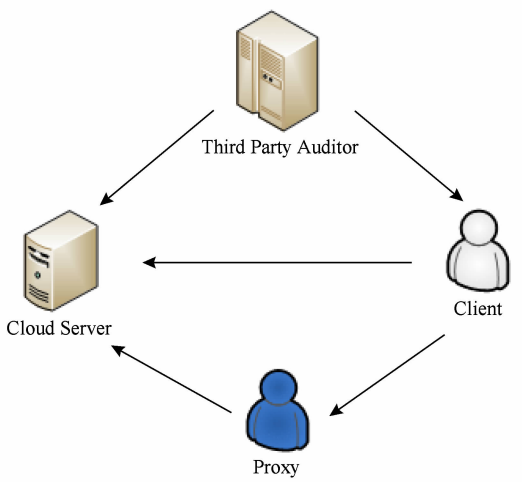

Fig. 1 System scheme of PDPPD.

图 1 PDPPD 系统模型

Table 1　Symbol Table and Definition of PDPPD Scheme

表 1　PDPPD 方案涉及的符号表以及定义

Symbol	Definition
$((u,X),x)$	Public key and secret key of client
(Y,y)	Public key and secret key of cloud server
$F=(m_1,m_2,\cdots,m_n)$	File and its sub block
$\Sigma=(T_{m_1},T_{m_2},\cdots,T_{m_n})$	Set of verification label
m_ω	Guarantee for the condition of the agency, and it is generated by client
(z,K)	Partial authorized information which is generated by client, $K=g^k$, $z=x+kK$
(r,s,K)	Verification information of proxy
V	Proof which is generated by cloud server

下面给出 PDPPD 方法中的多项式算法定义.

定义 1. PDPPD 方案是由 9 个多项式算法组成:

1) 密钥生成 $KeyGen(1^\kappa)\rightarrow(sk,pk)$:$\kappa$ 为安全参数,算法在安全参数 κ 下产生服务器与客户的公私钥对.

2) 客户生成标签 $TagBlock(x,Y,F)\rightarrow(\Sigma)$:已知文件 $F=(m_1,m_2,\cdots,m_n)$、用户私钥 x 以及服务器公钥 Y,计算验证标签 T_{m_i} 的集合,即 $\Sigma=(T_{m_1},T_{m_2},\cdots,T_{m_n})$.

3) 服务器检查验证标签 $CheckTag(\{(T_{m_i},m_i),1\leqslant i\leqslant n\})\rightarrow$("success","failure"):对所有的文件分块 m_i,服务器检查验证标签 T_{m_i} 的合法性,返回成功或失败.

4) 客户端生成授权信息 $PreProxy(x)\rightarrow(m_\omega,sign_x(m_\omega),(z,K))$:客户端产生描述代理限制条件的担保 m_ω,并用用户私钥 x 对担保进行签名 $sign_x(m_\omega)$,客户端最后计算授权代理的部分授权信息(z,K).

5) 代理验证授权信息 $ProxySelfVer(z,K,m_\omega,sign_x(m_\omega))\rightarrow$("success","failure"):代理收到客户产生的担保 m_ω,检查其签名 $sign_x(m_\omega)$的有效性以及检查自己是否满足代理的限制条件.代理同时检查客户产生的部分授权信息(z,K)的合法性,返回成功或失败.

6) 代理生成验证信息 $GenProxyInfo(z,m_\omega)\rightarrow(r,s,K)$:对担保 m_ω,代理利用代理私钥 z 生成代理验证信息(r,s,K)发送给服务器.

7) 服务器验证代理 $VerProxy(m_\omega,r,s,K)\rightarrow$("success","failure"):服务器检查担保 m_ω 及代理验证信息(r,s,K),返回成功或失败.

8) 服务器生成证据 $GenProof(F,chal,\Sigma)\rightarrow V$:服务器收到挑战请求 $chal$,根据文件信息 F 及验证标签信息 Σ 生成证据 V.

9) 代理对服务器进行持有性验证 $CheckProof(X,chal,V)\rightarrow$("success","failure"):代理方收到服务器返回的挑战 $chal$ 的证据 V,利用用户公钥 X 验证服务器是否以极大概率诚实地存放了客户所有的文件信息.

一个由 PDPPD 方案构成的 PDPPD 系统可以分 5 个阶段:

1) 系统初始化.系统的初始化时运行 $KeyGen(1^\kappa)\rightarrow(sk,pk)$,为服务器端、客户端生成公私钥对.

2) 客户端对文件进行分块处理,并运行 $TagBlock(x,Y,F)\rightarrow(\Sigma)$为分块后的文件生成验证标签,客户将文件块与标签$\{(T_{m_i},m_i),1\leqslant i\leqslant n\}$发送给服务器.服务器运行 $CheckTag(\{(T_{m_i},m_i),1\leqslant i\leqslant n\})\rightarrow$("success","failure")检查验证标签的合法性.

3) 客户端运行 $PreProxy(x)\rightarrow(m_\omega,sign_x(m_\omega),(z,K))$生成授权信息.并将担保 m_ω、对担保的签名 $sign_x(m_\omega)$和部分授权信息(z,K)发送给代理.代理收到授权信息后,运行 $ProxySelfVer(z,K,m_\omega,sign_x(m_\omega))\rightarrow$("success","failure")验证客户传来的授权信息有效性且在验证通过后接受客户的授权.

4) 代理向服务器验证其身份,代理运行 $GenProxyInfo(z,m_\omega)\rightarrow(m_\omega,(r,s,K))$生成验证信息,并将验证信息$(m_\omega,(r,s,K))$发送给服务器.服务器接收后,并运行 $VerProxy(m_\omega,r,s,K)\rightarrow$("success","failure")以验证代理的合法性.

5) 代理在需要验证服务器数据持有性时,向服务器发起挑战 $chal$.代理生成挑战 $chal$,并向服务器发送请求,要求服务器从总体 n 个数据块中随机抽取 c 个数据块,并生成这 c 个数据块的持有证据.服务器接收挑战 $chal$ 之后,运行 $GenProof(F,$

$chal, \Sigma) \rightarrow V$ 生成验证证据发送给代理. 代理运行 $CheckProof(X, chal, V) \rightarrow$ ("success","failure")验证服务器提供的持有性证据.

现在让我们来考虑 PDPPD 系统的安全模型. 在这个系统当中,服务器是不可信的,只有客户和授权的代理具有检验数据持有证据的权限,所以系统仅支持私有验证.

PDPPD 系统安全分为 2 部分:授权代理的安全性及数据持有性验证的安全性. 对于授权代理的安全性,文献[19]已给出详细的安全性证明,在此不再赘述. 对于数据持有性验证安全性,我们使用一个敌手游戏来证明其安全性,敌手游戏的特性如下:

1) 设定. 可信授权机构运行 $KeyGen(1^{\kappa}) \rightarrow (sk, pk)$产生客户的公私钥对$(X, x)$以及服务器的公私钥对$(Y, y)$,然后发送公钥 X, Y 给敌手,私钥不公开.

2) 查询. 敌手从 $m_1, m_2, \cdots, m_n$ 随机选择部分数据块并把它们发送给挑战者. 挑战者通过运行算法 $TagBlock(x, Y, F) \rightarrow (\Sigma)$计算产生验证元数据 $\Sigma = (T_{m_1}, T_{m_2}, \cdots, T_{m_n})$,并把它们发送给敌手,然后敌手一并存储 $m_1, m_2, \cdots, m_n$ 和 $T_{m_1}, T_{m_2}, \cdots, T_{m_n}$.

3) 挑战. 挑战者生成一个挑战 $chal$,并请求敌手提供与有序集合 $m_{i_1}, m_{i_2}, \cdots, m_{i_c}$ 相关的持有证据 $(1 \leqslant i_j \leqslant n, 1 \leqslant i_c \leqslant n, 1 \leqslant j \leqslant c, 1 \leqslant c \leqslant n)$.

4) 伪造. 敌手为 $chal$ 所提出的数据块计算持有证据 V,并且返回 V.

在 PDPPD 系统中,我们定义敌手赢得游戏,若:

$$Pr[Checkproof = \text{"success"}] \geqslant \frac{1}{p(\tau)}, \quad (1)$$

其中,$p(\tau)$是安全参数 τ 的多项式,它意味者敌手成功的可能性是不可忽略的(non-negligible).

定义 2. 如果对于任意概率的多项式敌手来说,敌手赢得敌手游戏的概率是可以忽略的(negligible),那么 PDPPD 系统中数据持有性验证就是安全的.

2.3 PDPPD 方案

本节我们将详细构造 PDPPD 方案,表 2 给出了不同完整性验证方案的对比. 首先介绍 PDPPD 方案构造中所使用的参数:κ 是安全参数,$f_k(x)$表示输入为 x,k 的函数 f;h, h_1, h_2 为密码学杂凑函数;假设文件 F 被分成了 n 个块:$(m_1, m_2, \cdots, m_n)$,$m_i \in \mathbb{Z}_q^*$;f, φ 是伪随机函数(pseudo-random function);π 是伪随机置换(pseudo-random permutation);$\mathbb{G}$,$\mathbb{G}_T$均是阶为 p 的循环乘法群,g 是循环乘法群$\mathbb{G}$的生成元;定义 e 为$\mathbb{G} \times \mathbb{G}$ 到$\mathbb{G}_T$双线性映射. 以上函数具体定义如下:

$$f: \mathbb{Z}_p^* \times \{1, 2, \cdots, n\} \rightarrow \mathbb{Z}_p^*, \quad (2)$$

$$\varphi: \mathbb{Z}_p^* \times \{1, 2, \cdots, n\} \rightarrow \mathbb{Z}_p^*, \quad (3)$$

$$\pi: \mathbb{Z}_p^* \times \{1, 2, \cdots, n\} \rightarrow \{1, 2, \cdots, n\}, \quad (4)$$

$$e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T. \quad (5)$$

Table 2 Comparisons of Different Remote Data Integrity Checking Schemes

表 2 不同完整性验证方案复杂度对比

Scheme	Server. comp	Client. comp	Communication	Privacy	Prob. of Detection
PDP[1]	$O(c)$	$O(c)$	$O(1)$	Yes	$1-(1-\rho)^c$
SPDP[3]	$O(c)$	$O(c)$	$O(c)$	Yes	$1-(1-\rho)^{cs}$
CPOR-1[11]	$O(c)$	$O(c)$	$O(1)$	No	$1-(1-\rho)^c$
CPOR-2[11]	$O(c+s)$	$O(c+s)$	$O(c+s)$	No	$1-(1-\rho)^{cs}$
Audit[12]	$O(c\log n)$	$O(c\log n)$	$O(c\log n)$	Yes	$1-(1-\rho)^c$
Our Scheme	$O(c\log n)$	$O(c\log n)$	$O(c\log n)$	Yes	$1-(1-\rho)^c$

n is the total number of data blocks which are generated by the files that to be verified; c is the number of random-choosing data blocks which are to be verified in challenge phase; s is the number of segments of each data block (it is needed to block the data blocks to form segments in some methods); ρ is the probability of data block corruption.

1) 系统的初始化包括服务器端、客户端的密钥生成,具体步骤为:

步骤 1. 执行 $KeyGen(1^{\kappa})$生成服务器密钥:首先产生服务器的公钥和私钥对(Y, y),其中 $y \in \mathbb{Z}_p$是一个随机数,$Y = g^y \in \mathbb{G}$.

步骤 2. 执行 $KeyGen(1^{\kappa})$生成客户端密钥:客户端选择一个随机值 $u \in \mathbb{G}$ 及一个随机数 $x \in \mathbb{Z}_p$,计算 $X = g^x \in \mathbb{G}$. 客户的公钥为(u, X),客户的私钥为 x.

2) 客户端对文件进行分块处理,并为分块后的文件生成验证标签,发送给服务器,服务器验证标签. 具体步骤包括:

步骤 1. 标签生成 $TagBlock(x,Y,F)$:给定文件 $F=(m_1,m_2,\cdots,m_n)$,客户为文件块 m_i 计算标签 T_{m_i};计算步骤如式(6)(7)所示:

$$t=h_2(e(Z,Y)^x)\ ,\quad W_i=\varphi_t(i),\tag{6}$$

$$T_{m_i}=(h(W_i)u^{m_i})^x.\tag{7}$$

我们用 Σ 表示 T_{m_i} 的集合,即 $\Sigma=(T_{m_1},T_{m_2},\cdots,T_{m_n})$;客户将$\{(T_{m_i},m_i),1\leqslant i\leqslant n\}$发送给服务器.

步骤 2. 服务器运行 $CheckTag(\{(T_{m_i},m_i),1\leqslant i\leqslant n\})$检查验证标签的合法性:对 $i,1\leqslant i\leqslant n$,服务器计算 $\tilde{t}=h_2(e(X,Z)^y)$和 $\widetilde{W}_i=\varphi_{\tilde{t}}(i)$,并验证等式 $e(T_{m_i},g)=e(h(\widetilde{W}_i)u^{m_i},X)$是否成立.如果成立,则服务器存放$((T_{m_i},m_i),1\leqslant i\leqslant n)$,客户在本地删除$((T_{m_i},m_i),1\leqslant i\leqslant n)$;如果不成立,服务器向客户端返回出错信息.

3) 客户端基于自己的私钥生成部分授权信息,并将授权信息传递给代理进行授权.代理收到授权信息后,验证客户传来的授权信息有效性且在验证通过后接受客户的授权.具体步骤为:

步骤 1. 客户端生成授权信息 $PreProxy(x)$:客户产生一个随机数 $k\in\mathbb{Z}_p$并且计算 $K=g^k$.客户端产生一个表明代理拥有检验远程数据持有权限的担保 m_ω 以及用用户私钥 x 对担保进行签名 $sign_x(m_\omega)$.担保描述了代理代表客户执行验证的限制条件.客户端最后计算 $z=x+kK$ 并将担保 m_ω、对担保的签名 $sign_x(m_\omega)$和部分授权信息(z,K)发送给代理.

步骤 2. 代理验证授权信息 $ProxySelfVer(z,K,m_\omega,sign_x(m_\omega))$:代理收到客户产生的担保 m_ω,检查其签名 $sign_x(m_\omega)$的有效性以及检查自己是否满足代理的限制条件.代理收到客户产生的(z,K),代理检查其是否满足:

$$g^z=XK^K.\tag{8}$$

如果满足式(8),那么证明代理是有效的.然后代理把 z 作为自己的私钥,把 $Z=g^z\in\mathbb{G}$ 作为自己的公钥.否则,代理拒绝客户端授权.

4) 代理向服务器验证其身份,具体步骤包括:

步骤 1. 代理生成验证信息 $GenProxyInfo(z,m_\omega)$:代理选择一个随机数 $\theta\in\mathbb{Z}_p$,计算 $r=g^\theta$.对代理的担保 m_ω,计算 $s=\theta^{-1}(h_1(m_\omega)-rz)$.将$(m_\omega,(r,s,K))$发送给服务器.

步骤 2. 服务器验证代理 $VerProxy(m_\omega,r,s,K)$:服务器检查 m_ω 以确认代理是否满足 m_ω 描述的限制条件,同时服务器检查$(m_\omega,(r,s,K))$是否满足:

$$g^{h_1(m_\omega)}=(XK^K)^r r^s.\tag{9}$$

如果符合,服务器接受代理方并将其作为客户的代理;否则,服务器拒绝接受代理方作为客户的代理.

5) 代理在需要验证服务器端数据持有性时,向服务器发起挑战.服务器接收挑战之后,生成验证证据发送给代理.代理验证服务器提供的持有性证据.具体步骤包括:

步骤 1. 代理生成挑战 $chal=(c,k_1,k_2)$,其中 $1\leqslant c\leqslant n,k_1\in\mathbb{Z}_p,k_2\in\mathbb{Z}_p$.代理方向服务器发送该挑战请求,要求服务器从总体 n 个数据块中随机抽取 c 个数据块,并生成这 c 个数据块的持有证据.

步骤 2. 服务器接受调整生成证据 $GenProof(F,chal,\Sigma)$.服务器收到请求后,首先对 $1\leqslant j\leqslant c$,计算其索引 $i_j=\pi_{k_1}(j)$和对应的系数 $a_j=f_{k_2}(j)$;接着计算 $T=\prod_{j=1}^{c}T_{m_{i_j}}^{a_j}$ 和$\rho=\sum_{j=1}^{c}a_jm_{i_j}$;最后服务器输出 $V=(\rho,T)$,并将 $V=(\rho,T)$发送给代理方完成对挑战的回复.

步骤 3. 代理对服务器进行持有性验证 $CheckProof(X,chal,V)$:代理方收到服务器回复 V 后,验证如下等式:

$$e(T,g)=e\Big(\prod_{j=1}^{c}h(\varphi_{\tilde{t}}(\pi_{k_1}(i_j)))^{f_{k_2}(j)}u^\rho,X\Big),\tag{10}$$

如果相等,则代理可确信服务器以极大概率诚实地存放了客户所有的文件信息.

2.4　正确性和安全性证明

本节将对 PDPPD 方案的正确性和验证安全性进行证明.PDPPD 正确性证明如下:

$$t=e(Z,K)^x=e(k,z)^{xkz}=e(X,K)^z=\hat{t},\tag{11}$$

$$\begin{aligned}e(T,g)&=e\Big(\prod_{j=1}^{c}T_{m_{i_j}}^{a_j},g\Big)=\\&\Big(\prod_{j=1}^{c}(h(W_{i_j})^{a_j})^x u^{x\sum_{j=1}^{c}a_jm_{i_j}},g\Big)=\\&\Big(\prod_{j=1}^{c}h(W_{i_j})^{a_j}u^{\sum_{j=1}^{c}a_jm_{i_j}},X\Big)=\\&\Big(\prod_{j=1}^{c}h(\varphi_t(\pi_{k_1}(i_j)))^{f_{k_2}(j)}u^\rho,X\Big)=\\&\Big(\prod_{j=1}^{c}h(\varphi_{\hat{t}}(\pi_{k_1}(i_j)))^{f_{k_2}(j)}u^\rho,X\Big).\end{aligned}\tag{12}$$

对于 PDPPD 方案的验证安全性,我们给出如下定理及其证明:

定理 1. 基于双线性对和 Diffie-Hellman 问题,PDPPD 方案在随机预言模型中可以保证数据持有性.

即是要证明基于双线性对和 Diffie-Hellman 困难问题的随机预言模型中,除了回复正确的验证信息,否则敌手只有可忽略的概率赢得敌手游戏.

证明. 假设可以通过验证的真实信息为 T 和 ρ,敌手伪造通过验证的伪造信息为 T'和ρ',由于 $T\neq T'$,有 $\rho\neq\rho'$,令 $\Delta\rho=\rho'-\rho$. 根据前面提到的计算性 Diffie-Hellman 问题,如果已知 $g, g^x, h\in\mathbb{G}$,得到 h^x 则说明计算性 Diffie-Hellman 问题在$\mathbb{G}$中可解决.

敌手任选 $\lambda_i\in\mathbb{Z}_q$,有 $g^\lambda\in\mathbb{G}$,以及 $\gamma,\eta\in\mathbb{Z}_q$,令:

$$u=g^\gamma\times h^\eta, \tag{13}$$

$$(h(W_i))^{a_i}=g^{\lambda_i}/g^{\gamma a_i m_i}\times h^{\eta a_i m_i}. \tag{14}$$

那么对于数据块标签,则有:

$$T_{m_i}^{a_i}=(u^{a_i m_i}\times g^{\lambda_i}/g^{\gamma a_i m_i}\times h^{\eta a_i m_i})^x=$$
$$((g^\gamma\times h^\eta)^{a_i m_i}\times g^{\lambda_i}/g^{\gamma a_i m_i}\times h^{\eta a_i m_i})^x=(g^{\lambda_i})^x. \tag{15}$$

若敌手成功用 T'伪造 T 并通过验证,计算:

$$e(T'/T,g)=e(u^{\Delta\rho},X)=e((g^\gamma\times h^\eta)^{\Delta\rho},X) \tag{16}$$

$$e(T'\times T^{-1}\times X^{-\gamma\Delta\rho},g)=e(h,X)^{\eta\Delta\rho}. \tag{17}$$

通过式(10)(11),可以在$\mathbb{G}$中得到计算性 Diffie-Hellman 问题的解:

$$h^x=(T'\times T^{-1}\times X^{-\gamma\Delta\rho})^{\frac{1}{\eta\Delta\rho}}. \tag{18}$$

这与计算性 Diffie-Hellman 问题是困难性问题矛盾. 证毕.

3 分析与评估

本节将进行 PDPPD 方案的性能评估,表 3 中所示为方案所涉及的各原子操作. 评估将重点关注引入代理的开销及 PDPPD 方案的验证开销和效率. 实验在 Linux 系统下用 C 语言开发完成. 实验所用电脑如下:英特尔奔腾双核 2.50 GHz 处理器、DDR2 800 MHz, 2 GB 内存、7200 转硬盘 250 GB 机械硬盘. 实验代码使用的 Pairing-Based Cryptography (PBC)库版本为 0.5.14. 实验中椭圆曲线形式为 MNT 曲线,其基础字段大小为 159 b,嵌入度 6. 实验中安全等级选择 80 b,即安全参数$|a_i|=80$ b,$|p|=160$. 实验结果为 50 次实验取平均值.

Table 3 Notation Summary of Cryptography Operations

表 3 方案涉及原子操作总结

Cryptography Operation	Definition
$Add_{\mathbf{G}}^t$	In the multiplicative group $\mathbb{G}$, the additive operation is performed t times.
$Mult_{\mathbf{G}}^t$	In the multiplicative group $\mathbb{G}$, the multiplication operation is performed t times.
$Exp_{\mathbf{G}}^t$	In the multiplicative group $\mathbb{G}$, the exponential operation g^a is performed t times.
$Hash_{\mathbf{G}}^t$	In the multiplicative group $\mathbb{G}$, the Hash operation is performed t times.

1) 从计算量方面分析 PDPPD 方案. 假设数据被分成 n 个数据块,挑战随机选取 c 个数据块. 在标签生成 *TagBlock* 步骤中,客户计算量为 $Mult_{\mathbf{G}}^n+Exp_{\mathbf{G}}^{2n+1}+Hash_{\mathbf{Z}_p^*}^n$. 客户端生成授权信息的 *PreProxy* 步骤中,客户计算量为 $Add_{\mathbf{Z}_p^*}^1+Add_{\mathbf{Z}_p^*}^1+Mult_{\mathbf{Z}_p^*}^1+Exp_{\mathbf{G}}^1$. 在代理自验证 *ProxySelfVer* 步骤中,代理计算量为 $Mult_{\mathbf{G}}^1+Exp_{\mathbf{G}}^2$. 代理生成验证代理信息 *GenProxyInfo* 步骤中,代理计算量为 $Exp_{\mathbf{G}}^1+Mult_{\mathbf{Z}_p^*}^2+Add_{\mathbf{Z}_p^*}^1+Hash_{\mathbf{Z}_p^*}^1$. 在验证代理 *VerProxy* 步骤中,服务器需要解密验证签名. 解密担保,该步骤计算量为 $Exp_{\mathbf{G}}^4+Mult_{\mathbf{G}}^2$. 在证据生成 *GenProof* 步骤中,服务器计算量为 $Exp_{\mathbf{G}}^c+Mult_{\mathbf{G}}^{c-1}+Add_{\mathbf{Z}_p^*}^{c-1}$. 在证据检验 *CheckProof* 步骤中,客户或代理方计算量为 $Exp_{\mathbf{G}}^{c+1}+Mult_{\mathbf{G}}^c+Hash_{\mathbf{Z}_p^*}^c$. 其他计算如伪随机数生成等,由于相比上述计算运算量微小,所以忽略不计.

2) 分析引入代理对于数据持有性验证中客户端和服务器端带来的额外计算开销. 对于客户端,需要生成 $K=g^k$ 和 $z=x+kK$ 进行代理验证,相应地计算消耗为 $Add_{\mathbf{Z}_p^*}^1+Mult_{\mathbf{Z}_p^*}^1+Exp_{\mathbf{G}}^1$. 这是一个常量,挑战阶段随机选取的数据块数量 c 对其没有影响. 同时,相比数据持有性验证中客户端生成的总计算量,这部分计算量可以忽略不计.

相似地,对于服务器端需要生成 $g^{h_1(m_\omega)}=(XK^K)^r r^s$ 进行代理验证. 相应的计算消耗为 $Mult_{\mathbf{G}}^2+Exp_{\mathbf{G}}^4$,这也是一个常量,不随挑战阶段随机选取数据块 c 变化. 相比数据持有性验证中服务器端生成的总计算量,这部分计算量可以忽略不计.

3) 分析本方案验证时间开销及效率. 当进行随机选取数据块 c 为 300 或者 460 的高概率数据持有性验证时,如表 4 所示,相比文献[11],PDPPD 方案验证和服务器端生成签名和证据消耗时间(单位 ms)更少,则验证效率更高. 此外 PDPPD 方案还可委托代理进行数据持有性验证,具有更广泛的应用场景.

Table 4　Performance Comparison under Different Number of Sampled Blocks for PDPPD Scheme and CPOR-2 Scheme

表 4　在 PDPPD 方案和 CPOR-2 方案中选取不同数量数据块下实验表现的对比

Scheme	Number of Sampled Blocks	Computing Time on Cloud Server Side/ms	Computing Time on Verification Side/ms
PDPPD	300	239.43	317.48
	460	358.42	473.83
CPOR-2[11]	300	257.31	330.84
	460	390.34	495.61

从表 4 实验结果可以看出,相较于文献[11]中的 CPOR-2 方案,PDPPD 方案服务器和验证端计算时间更短.这是因为:1)添加代理功能产生的额外计算量微小,对方案计算时间基本没有影响;2)在保证验证具有同等安全等级的前提下,本方案简化了验证参数,缩减了服务器和验证端的计算时间.

4　总　　结

本文提出了基于部分授权的可证明数据持有性证明的概念,并建立了 PDPPD 的系统模型和安全模型;同时给出具体的 PDPPD 方案,并证明了 PDPPD 方案是可证明安全和高效率的.相比现有的可证明数据持有性验证方案,PDPPD 方案具有应用方向广泛、数据保护性强等诸多优点.

参　考　文　献

[1] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores [C] //Proc of the 14th ACM Conf on Computer and Communications Security. New York: ACM, 2007: 1165-1182

[2] Juels A, Kaliski Jr B S. Pors: Proofs of retrievability for large files [C] //Proc of the 14th ACM Conf on Computer and Communications Security. New York: ACM, 2007: 584-597

[3] Ateniese G, Di Pietro R, Mancini L V, et al. Scalable and efficient provable data possession [C] //Proc of the 4th Int Conf on Security and Privacy in Communication Networks. New York: ACM, 2008: 1-10

[4] Curtmola R, Khan O, Burns R, et al. MR-PDP: Multiple-replica provable data possession [C] //Proc of the 28th Int Conf on Distributed Computing Systems (ICDCS'08). 2008, 68(7): 411-420

[5] Barsoum A F, Hasan M A. Provable possession and replication of data over cloud servers [R/OL]. Waterloo, Ontario, Canada: Centre for Applied Cryptographic Research (CACR), University of Waterloo, 2010 [2015-06-01]. http://www.researchgate.net/publication/267853007_Provable_Possession_and_Replication_of_Data_over_Cloud_Servers

[6] Erway C, Küpçü A, Papamanthou C, et al. Dynamic provable data possession [C] //Proc of the 16th ACM Conf on Computer and Communications Security. New York: ACM, 2009: 213-222

[7] Wang Qian, Wang Cong, Kui Ren, et al. Enabling public verifiability and data dynamics for storage security in cloud computing [C] //Proc of the 14th European Conf on Research in Computer Security. Berlin: Springer, 2009: 355-370

[8] Zhu Yan, Wang Huaixi, Hu Zexin. Efficient provable data possession for hybrid clouds [C] //Proc of the 17th ACM Conf on Computer and Communications Security. New York: ACM, 2010: 881-88

[9] Dodis Y, Vadhan S, Wichs D. Proofs of Retrievability via Hardness Amplification [M]. Berlin: Springer, 2009: 109-127

[10] Bowers K D, Juels A, Oprea A. Hail: A high-availability and integrity layer for cloud storage [C] //Proc of the 16th ACM Conf on Computer and Communications Security. New York: ACM, 2009: 187-198

[11] Shacham H, Waters B. Compact Proofs of Retrievability [M]. Berlin: Springer, 2008: 442-483

[12] Wang Cong, Sherman C, Wang Qian, et al. Privacy-preserving public auditing for secure cloud storage [J]. IEEE Trans on Compters, 2013, 62(2): 362-375

[13] Yang Kan, Jia Xiaohua. An efficient and secure dynamic auditing protocol for data storage in cloud computing [J]. IEEE Trans on Parallel & Distributed Systems, 2013, 24(9): 1717-1726

[14] Wang Boyang, Li Hui, Li Ming. Privacy-preserving public auditing for shared cloud data supporting group dynamics [C] //Proc of 2013 IEEE Int Conf on Communications. Piscataway, NJ: IEEE, 2013: 1946-1950

[15] Patil M, Rao G R. Integrity verification in multi-cloud storage using cooperative provable data possession [J]. International Journal of Computer Science & Information Technology, 2014, 5(2): 982-985

[16] Wang Huaqun, Zhang Yuqing. On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage [J]. IEEE Trans on Parallel & Distributed Systems, 2014, 25(1): 264-267

[17] Karthikeyan S, Praveen J. Provable data possession for securing the data from untrusted server [J]. International Journal of Engineering Research & Applications, 2015, 5(3): 34-37

[18] Esiner E, Küp04ü A, Özkasap Ö. Analysis and optimization on FlexDPDP: A practical solution for dynamic provable data possession [G] //LNCS: 8993. Berlin: Springer, 2015: 65-83

[19] Mambo M, Usuda K, Okamoto E. Proxy signatures for delegating signing operation [C] //Proc of the 3rd ACM Conf on Computer and Communications Security. New York: ACM, 1996: 48-57

[20] Neuman B C. Proxy-based authorization and accounting for distributed systems [C] //Proc of the 13th Int Conf on Distributed Computing Systems. Piscataway, NJ: IEEE, 1993: 283-291

[21] Kim S, Park S, Won D. Proxy signature, revisited [J]. First International Conf on Information & Communication Security, 1997, 11(11): 223-232

[22] Zhang Kan. Threshold Proxy Signature Schemes [M]. Berlin: Springer, 1998: 191-197

[23] Lee B, Kim H, Kim K. Strong proxy signature and its applications [C] //Proc of the 2001 Symp on Cryptography and Information Security. Oiso, Japan: Institute of Electronics, Information and Communication Engineers (IEICE), 2001: 603-608

[24] Lu Jui-Lin Eric, Hwang Min-Shiang, Huang Chenjian. A new proxy signature scheme with revocation [J]. Applied Mathematics and Computation, 2005 (161): 799-806

[25] Wang Huaqun. Proxy provable data possession in public clouds [J]. IEEE Trans on Services Computing, 2013, 6(4): 551-559

[26] Boneh D, Franklin M. Identity-based encryption from the Weil pairing [G] //Advances in Cryptology (CRYPTO 2001). Berlin: Springer, 2001: 213-229

[27] Miyaji A, Nakabayashi M, Takano S. New explicit conditions of elliptic curve traces for fr-reduction [J]. IEICE Trans on Fundamentals of Electronics, Communications and Computer Sciences, 2001, 84(5): 1234-1243

Zhong Ting, born in 1977. Associate professor. Her main research interests include security issues in cloud computing, smart transportation.

Han Xiao, born in 1991. Postgraduate student. Her main research interests include security issues in cloud computing (822794830@qq.com).

Zhao Yulong, born in 1989. Postgraduate student. His main research interests include security issues in cloud computing (232555341@qq.com).