DACPCC:一种包含访问权限的云计算数据访问控制方案

王于丁,杨家海
(清华大学网络科学与网络空间研究院,北京 100084)

摘 要: 目前云计算访问控制技术最常用的加密体系是 CP-ABE,但传统的 CP-ABE 加密体系中没有涉及用户的访问权限问题,数据提供者只能让用户去读取数据而不能写数据,访问控制机制不灵活,且效率低. 针对这一不足,本文提出了一种包含访问权限的高效云计算访问控制方案 DACPCC,该方案在 CP-ABE 基础上设置了权限控制密钥来加密云中的数据,数据提供者通过对权限控制密钥的选择来控制数据的访问权限. 文章对 DACPCC 进行了详细的设计,并做了安全性证明和实验验证,结果表明 DACPCC 能够让数据提供者对其数据资源进行权限控制,并且是安全和高效的.

关键词: 云计算; 访问控制; 属性基加密; 访问权限; 属性撤销
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2018)01-0236-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.01.033

DACPCC:A Data Access Control Scheme with Access Permission for Cloud Computing

WANG Yu-ding, YANG Jia-hai
(*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing* 100084, *China*)

Abstract: Currently, the most common encryption scheme of cloud computing access control system is CP-ABE, but the conventional CP-ABE encryption did not deal with the issue of user's access permission; data owners only allow the users to read the data but not to write, such kind of coarse access control mechanism is not flexible and low efficiency. To deal with this issue, the paper proposes a Data Access Control scheme with access Permission for Cloud Computing (DACPCC), it sets permission control keys to encrypt the data in cloud based on CP-ABE; the data owner controls the data's access permission by choosing the permission control keys. The paper illustrates the design details of the proposed scheme, then theoretically proves the security and evaluates the performance through simulated experiments; the results show that DACPCC allows the data owners to control the access permission of the data, and it is safe and more efficient.

Key words: cloud computing; access control; CP-ABE; access permission; attribute revocation

1 引言

随着云计算技术的发展,云平台中数据的访问控制成为越来越关键的问题[1]. 由于云计算环境的开放性和弹性,数据在云存储中面临如下问题:一方面,一旦将数据置于云端,数据提供者将完全失去对数据的控制,数据的安全性和隐私将面临来自云平台内外多方面的威胁;另一方面,由于云平台会根据实时需求进行动态资源供给,网络范围一直处于被动变化之中,导致访问控制的策略动态变化,不易管理[2]. 为此,人们将基于密文的属性基加密(CP-ABE,attribute-based encryption)[3,4]方案用于云计算中,这项技术为云计算数据的访问控制提供了新的思路,它允许数据所有者定义自己需要的访问策略并对数据进行加密,用户只需要符合相应属性条件便可解密,并且 CP-ABE 将加密规则蕴含在加密算法之中,可以免去密文访问控制中频繁出

收稿日期:2016-08-30;修回日期:2016-10-16;责任编辑:诸叶梅
基金项目:国家自然科学基金(No.61432009,No.61462009);教育部博士学科专项基金(No.20130002110058);国家 863 高技术研究发展计划(No.2015AA015601)

现的密钥分发代价[5,6]. CP-ABE 最突出的优点是很适合于云环境下解密方不固定的场景,并且具有很强的灵活性和可扩展性. 但 CP-ABE 运算效率低下,机制不完善等问题也受到了很多关注.

近些年来,将 CP-ABE 运用在云计算环境中的研究工作主要集中在如何进一步提高 CP-ABE 访问控制系统的效率. 主要方法有:(1)通过改进 CP-ABE 的访问结构来提高效率,如文献[7,8];(2)通过改进 CP-ABE 属性层级来提高效率,如文献[9,10];(3)通过改进 CP-ABE 的认证机构来提高效率,如文献[11~13];(4)通过改进 CP-ABE 的属性撤销机制来提高运行效率,如文献[13~15]等等. 而本文将从改进 CP-ABE 的另一个特征——访问权限来提高运行效率. 在云计算系统中,为了提高工作效率,数据提供者不仅可以授权可信的用户来读取数据,还可以授权可信用户来进行写数据,从而大大减轻数据提供者和云平台负担从而提高系统工作效率,比如说企业的 CEO 可以授权秘书来修改企业数据,战场上指挥员可以授权作战参谋来修改战场数据等等,然而 CP-ABE 本身只是通过用户属性来授予读权限,而不能授予写权限,所以本文的目的就是研究一套包含用户细粒度访问权限控制的 CP-ABE 云计算访问控制机制,在运算效率,安全性,属性撤销等方面都较原始 CP-ABE 有很大提高. 这方面的工作目前研究较少,文献[16]支持权限控制,但效率较低,属性撤销不完善.

本文针对这些不足,以 CP-ABE 为基础,提出了一种包含访问权限的高效云计算访问控制方案:DACPCC,该方案设置非对称的权限控制密钥加密云中的数据,数据提供者通过选择不同的权限控制密钥来授予和撤销数据的读写权限,弥补了 CP-ABE 中用户只能读不能写数据的缺陷;该方案采用云平台 token 解密方法将核心的解密工作交给云平台处理,提高了系统的效率,并具有完善的属性撤销机制. 文章的结构是:第二章阐述本文工作将会用到的基本理论,第三章定义相关模型和介绍系统流程,第四章阐述 DACPCC 的完整方案,第五章对方案的安全性进行分析,第六章进行该方案的实验评估. 第七章是结论和进一步工作展望.

2　基本理论

DACPCC 机制通过访问结构表示策略,以双线性对为技术基础,并基于 DBDH(Decision Bilinear Diffie Hellman)假设构建安全性. 下面分别给出本文基本概念的形式化定义[17~19].

2.1　双线性对

设$\mathbb{G}_1$和$\mathbb{G}_2$均为 p 阶循环乘法群,$\mathbb{Z}_p$ 为模 p 的剩余类加群,设 g 为群$\mathbb{G}1$ 的一个随机生成元,又设 e 为双线性映射 $e:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_2$,并定义公开映射 $H:\{0,1\}\rightarrow\mathbb{G}_2$. 则:

(1)双线性:对于所有的 $g_1,g_2\in\mathbb{G}_1$,对于所有的 $a,b\in\mathbb{Z}_p$,均有 $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$成立;

(2)非退化性:存在 g, $g\in\mathbb{G}_1$,满足 $e(g,g)\neq1$;

(3)可计算性:对于所有的 $g_1,g_2\in\mathbb{G}_1$,存在一个有效的多项式算法来计算 $e(g_1,g_2)$.

2.2　访问结构

假定实体集$\mathbb{P}=\{P_1,P_2,\cdots,P_n\}$上共享了一个秘密,若$\mathbb{P}$的一个子集能共享该秘密,则称这个子集为一个授权子集;否则,称其为非授权子集. 所有授权子集构成的集族$\mathbb{A}$,称为对该秘密的一个访问结构. 访问结构$\mathbb{A}$是单调的,是指 $\forall B,C$,若 $B\in\mathbb{A}$ 且 $C\subseteq B$,那么 $C\in\mathbb{A}$.

2.3　DBDH 假设

给定阶为大素数 p 的群$\mathbb{G}_1$和$\mathbb{G}_2$以及双线性映射 $e:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_2$,群$(\mathbb{G}_1,\mathbb{G}_2)$上的判定双线性 Diffie Hellman 问题 DBDH 是指:给定 g,选择 $a,b,c\in\mathbb{Z}_p$, $R\in\mathbb{G}_1$,判断 $R=e(g,g)^{abc}$是否成立. 定义一个算法 P 解决 DBDH 问题的优势为:

$$Adv_P^{DBDH}=|\Pr[P(g,g^a,g^b,g^c,R)=0]-\Pr[P(g,g^a,g^b,g^c,e(g,g)^{abc})=0]|;$$

如果对任意一个多项式时间算法 P,其解决 DBDH 问题的优势小于可忽略的时间 ε,则称 DBDH 假设成立,并称(g,g^a,g^b,g^c,R)为一个 BDH(bilinear Diffie Hellman)元组.

3　DACPCC 模型定义

3.1　系统模型

DACPCC 由 4 部分实体组成如图 1 所示,其名称和作用如下所述:(1)数据所有者(DO,Data Owner),定义其数据的访问策略,将数据进行加密,外包到云服务器中供合法用户使用;(2)用户(U,User),用户可以下载在云中共享的任何加密的数据,如果它的私钥满足访问控制策略可以将其解密使用;(3)云服务提供商(CSP,Cloud Service Provider),即云平台,拥有强大的存储计算功能,存储 DO 提交上来的数据,并供 U 下载;(4)主认证中心(MA,Master Authority),注册 DO 和 U,

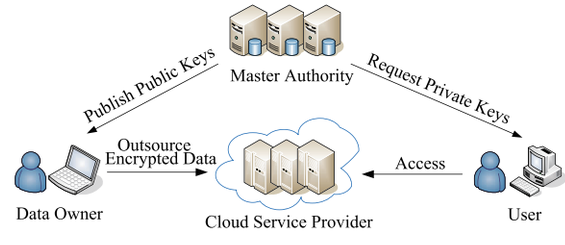

图1　系统模型示意图

并且根据 DO 和 U 的请求来计算和发放密钥,其中,将系统公钥发放给 DO,将私钥发放给 U,同时还负责撤销和更新用户的属性密钥.

在整个系统模型中,为了保证数据有效安全的提交和访问,需要做以下安全假设:(1)MA 是完全可以被信任的,并且不会串谋其他用户进行窃取数据,忽略 DO 和 U 在 MA 上的认证阶段,凡是能参与本系统的 DO 和 U,均已合法注册成功;(2)CSP 可以被信任,其内部不存在对数据盗取和篡改;(3)DO 和 U 不能跨过云平台私自通信,以保证数据的安全;(4)密钥传输是安全的.

3.2 基本思想

在研究云计算系统访问控制模型基础上,我们对该问题有以下分析,(1)云存储系统的数据存储在云端,U 的属性如果满足 DO 的访问结构,U 可以读取数据;要想让 U 同时获得数据的写权限,必须在 CP-ABE 加入相应的控制访问权限的密钥,同时还要保证多个 U 同时写入数据不发生冲突.(2)要想提高系统运行效率,必须降低数据的加解密的运算复杂度. 尤其是当 U 解密数据的时候,如果将复杂的解密运算交给拥有强大计算能力的 CSP 来运行,U 端关于其属性数量的解密时间复杂度则由 $O(n)$ 降低为 $O(1)$,大大节省了系统整体运行时间.(3)一旦 U 的属性发生撤销或者相关权限发生改变,U 的注册机构 MA 就会生成用户新的属性集,然后对用户密钥和密文中涉及变更属性集的项进行更新,未涉及的项不发生变化. 同时还要必须满足系统的安全性.

DACPCC 的主要思路是 DO 生成一对用于控制权限的非对称密钥,用这对非对称密钥加密数据,并通过 CP-ABE 加密权限控制密钥并上传 CSP,通过所加密的权限密钥的不同来控制 U 的读写权限,U 解密时要向 CSP 申请解密 token 来进行解密. U 属性撤销时先更新加密密钥,然后分别更新用户的私钥和密文,分别保证系统的后向安全和前向安全. 整个系统由 5 个阶段构成:其中密钥生成阶段(算法 a,b,c)、加密阶段(算法 d,e)、解密阶段(算法 f,g,h)用于数据的访问控制,属性撤销阶段(算法 i,j)用于用户属性撤销,权限更改阶段用于用户权限的授予与撤销,系统流程图如图 2 所示,由于权限更改阶段没有特殊的算法,故没在流程图上显示.

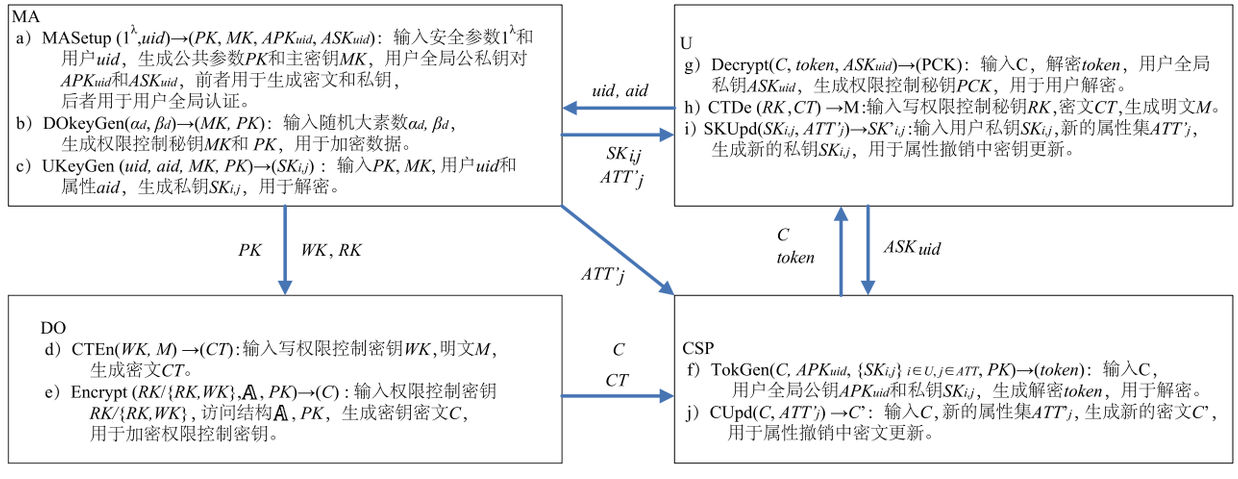

图2 系统流程图

3.3 安全模型

DACPCC 的安全模型是基于不可区分的选择明文攻击(IND-CPA)[20]建立的. 假设存在敌手 A 和挑战者 C,A 要向 C 进行一系列的询问和挑战. 由于私钥被用户属性集生成,密文由访问结构生成,所以 A 将向 C 挑战的内容是密文中的访问结构$\mathbb{A}$.

Init:敌手 A 公开宣布他要挑战访问结构$\mathbb{A}$;

Setup:挑战者 C 运行 MASetup 算法,输出系统公钥 PK,并交于敌手 A;

Phase 1:敌手 A 向挑战者 C 询问用户 $i(i\in U)$ 关于满足属性集 ATT_j^* (ATT_j^* ($\mathbb{A}^*$) = 0) 的私钥,要求用户的属性集 ATT_j^* 不能满足$\mathbb{A}^*$,即所询问的私钥不能直接解密密文. 然后挑战者 C 将私钥 $SK_{i,j}$ 返回给敌手 A;

Challenge:敌手 A 选取两个等长度的明文 M_0 和 M_1,挑战者 C 随机选一个明文 M_η ($\eta\in\{0,1\}$) 进行加密,将计算出来的密文返回给敌手 A;

Phase 2:与 Phase 1 相同,敌手继续提交用户私钥的询问;

Guess:敌手回答对 η 的猜测结果 η',如果 $\eta'=\eta$,则敌手 A 获胜;

定义 1 敌手 A 攻击优势为 $Adv_{\text{DACPCC}}^{\text{IND-CPA}}=\Pr[\eta'=\eta]$

$-\frac{1}{2}$.

定义 2　DACPCC 是 IND-CPA 安全的,当且仅当上述的攻击游戏中,任何多项式时间内敌手 A 的攻击优势是可忽略的. 也就是说如果 DACPCC 安全,敌手 A 是不能通过计算的方式得到正确 η,要想获胜必须猜测,对于 $\eta \in \{0,1\}$ 来说,η 取值概率均等,故猜对的概率最大是 1/2,根据定义 1 得出敌手的攻击优势可忽略.

4　DACPCC 详细设计

(1) 密钥生成阶段: MA 生成系统公钥 PK 和主密钥 MK,每个数据所有者向 MA 申请生成非对称的权限控制密钥对;每个用户向 MA 申请,生成用于全局认证的公私钥对 APK/ASK,用于解密的私钥 SK,并根据该用户的权限授予权限控制密钥. 选取两个阶均为素数 p 的乘法群 $\mathbb{G}_1$ 和 $\mathbb{G}_2$, 设 g 为群 $\mathbb{G}_1$ 的一个随机生成元,又设 e 为双线性映射 $e:\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$,并定义公开映射 $H:\{0,1\} \to \mathbb{G}_2$.

(a) MASetup $(1^\lambda, uid) \to (PK, MK, APK_{uid}, ASK_{uid})$ 此算法由 MA 来执行,MA 选择随机数 $\alpha,\beta,\gamma \in \mathbb{Z}_p$,生成 PK 和 MK,

$PK = \{Y = e(g,g)^\alpha, D = g^\beta\}$, $MK = \{\alpha,\beta\}$

对于每一个需要注册的合法 U,MA 分配给其一个全局的 $uid \in U = \{1,2,\cdots,u\}$,对于每一个 uid,需要生成全局认证公钥 $APK_{uid} = g^\gamma$,以及全局认证私钥 $ASK_{uid} = \gamma$.

(b) DOkeyGen $(\alpha_d, \beta_d) \to (WK, RK)$ 此算法由 MA 执行,MA 选择随机大素数 $\alpha_d, \beta_d \in \mathbb{Z}_p$($\alpha_d$ 和 β_d 的乘积在 1024bit 数量级),针对每一个合法注册的 DO,由 α_d, β_d 生成一对权限控制非对称密钥对(WK,RK),并通过安全通道传给 DO. WK 用于控制读写权限和加密,RK 用于控制只读权限和解密.

(c) UKeyGen $(uid, aid, MK, PK) \to (SK_{i,j})$ 此算法由 MA 来执行,对于每一个 user,$uid = i (i \in U)$,每一个属性 $aid = j (j \in ATT)$,选取随机数 $t_j \in \mathbb{Z}_p$,计算用户私钥 $SK_{i,j}$

$$SK_{i,j} = \left(K_0 = \mathrm{H}(uid)^{\frac{1}{\beta}} \cdot g^{\frac{\alpha}{\beta}},\ K_{1,j} = \mathrm{H}(uid)^{t_j}_{(\forall j \in ATT)}\right)$$

(2) 加密阶段:首先 DO 用权限控制密钥 WK 对要分享的数据进行加密生成 CT,然后 DO 把权限密钥做为明文进行 CP-ABE 加密生成 C,将 CT 和 C 一并存入 CSP. 这样做一方面减轻了加密开销,另一方面可以通过权限密钥对用户进行权限控制.

(d) CTEn $(WK, M) \to (CT)$,此算法由 DO 进行,输入要分享的数据明文 M,通过权限控制密钥 WK 对 M 进行加密得到 $CT = \mathrm{E}_{WK}(M)$.

(e) Encrypt $(RK/\{RK, WK\}, \mathbb{A}, PK) \to (C)$,此算法由 DO 来执行,设权限控制密钥 $PCK = RK/\{RK, WK\}$,DO 加密 PCK 时要输入系统参数 PK 和访问结构 $\mathbb{A}(\boldsymbol{M}, \rho)$[14],$\boldsymbol{M}$ 为 $l \times n$ 的矩阵,l 为属性的总数,函数 ρ 为矩阵 $\boldsymbol{M}$ 的第 i 行到属性之间的 1 对 1 的单射. 首先选择随机的秘密数 s, $r \in \mathbb{Z}_p$ 和随机向量 $\boldsymbol{v} = (s, y_2, \cdots, y_n) \in \mathbb{Z}_p^n$,$y_2, \cdots, y_n$用于共享秘密 s,A_C 是密文的属性集. 计算 $\lambda_x = \boldsymbol{v} \cdot \boldsymbol{M}_x$,然后计算密钥密文,并上传 CSP:

$C_0 = PCK \cdot \left(\prod_{j \in A_C} e(g,g)^\alpha\right)^s$, $C_{1,j} = g^{\beta s} (j \in A_C)$,

$C_{2,x} = g^{\lambda_x} \cdot g^{t_{\rho(x)} r} (\forall x \in (1, \cdots, l))$, $C_3 = g^r$, $C_4 = g$

$C = (C_0, C_{1,j}, C_{2,x}, C_3, C_4)_{j \in A_C, x \in (1,\cdots,l)}$

数据在 CSP 中存储格式如图 3 所示. 其中,除 DataID、C 和 CT 外,可写数据比只读数据多出两项,$timestamp$ 作为数据修改的时间记录,来保证数据的一致性;U_{uid}记录获得此可写数据的 U,用于认证.

●只读数据 ORData( ):

DataID	C = Encrypt(RK, $\mathbb{A}$, PK)	$CT = \mathrm{E}_{WK}(M)$
数据 ID 号	Header	Body

●可写数据 RWData( ):

DataID	C = Encrypt ({RK, WK}, $\mathbb{A}$, PK)	timestamp	U^{uid}	$CT = \mathrm{E}_{WK}(M)$
数据 ID 号	Header			Body

图 3　数据存储格式

权限控制有三种场景,其解决方法如下:①数据只读,将只读权限密钥 RK 进行加密;②数据对所有属性满足$\mathbb{A}$用户开放读写权限,将只读权限密钥 RK 和读写权限密钥 WK 同时进行加密;③数据对于某个具体的用户开放 s 读写权限,将$\mathbb{A}$定向到该用户,同时对 RK 和 WK 同时加密.

(3)解密阶段:用户通过全局认证密钥向 CSP 提出申请并进行认证,CSP 确定用户身份以及所拥有的权限,生成解密的 $token$ 并和 C、CT 一起传给用户,这样 CSP 将承担大量的解密运算,减轻了 U 解密,从而提高了效率.

(f) TokGen $(C, APK_{uid}, \{SK_{i,j}\}_{i \in U, j \in ATT}, PK) \to (token)$,此算法在 CSP 进行,为了降低用户端的开销,我们设计了基于 $token$ 的解密模式,将繁重的核心解密工作由云端来完成. 对于每一个 $user$,$U_i (i \in U)$,发送其密

钥$\{SK_{i,j}\}$到 CSP,CSP 执行 TokGen 来生成解密的 *token*,仅当用户 U_i 的属性满足 C 中的访问结构$\mathbb{A}$,CSP 才能计算出正确的 *token*. 选取常量 $w_x \in \mathbb{Z}_p$,使得 $\sum_x w_x\lambda_x = s$,计算 *token* 如下

$$\begin{aligned}token &= \prod_{j\in A_c}\frac{e(C_{1,j},K_0)\cdot e(C_4,APK_{uid})^{-1}}{\prod_{j\in A_c}\left(\frac{e(C_{2,x},\mathrm{H}(uid))}{e(C_3,K_{1,\rho(x)})}\right)^{w_x}}\\ &= \prod_{j\in A_c}\frac{e(C_{1,j},K_0)\cdot e(C_4,APK_{uid})^{-1}}{\prod_{j\in A_c}\left(\frac{e(g^{\lambda_x}g^{t_{\rho(x)}r}\mathrm{H}(uid))}{e(g^r,\mathrm{H}(uid)^{t_{\rho(x)}})}\right)^{w_x}}\\ &= \prod_{j\in A_c}\frac{e(C_{1,j},K_0)\cdot e(C_4,APK_{uid})^{-1}}{\prod_{j\in A_c}e(g,\mathrm{H}(uid))^{\lambda_x w_x}}\\ &= \prod_{j\in A_c}\frac{e(g^{\beta s},\mathrm{H}(uid)^{\frac{1}{\beta}}\cdot g^{\frac{\alpha}{\beta}})\cdot e(g,g^{\gamma})^{-1}}{\prod_{j\in A_c}e(g,\mathrm{H}(uid))^{\lambda_x w_x}}\\ &= \prod_{j\in A_c}\frac{e(g^{\beta s},\mathrm{H}(uid)^{\frac{1}{\beta}}\cdot g^{\frac{\alpha}{\beta}})^{\frac{1}{\gamma}}}{\prod_{j\in A_c}e(g,\mathrm{H}(uid))^{s}}\\ &= \prod_{j\in A_c}e(g,g)^{\frac{\alpha s}{\gamma}}\end{aligned}$$

(g) Decrypt$(C,token,ASK_{uid})\rightarrow(PCK)$,此算法在 U 进行,用户端利用全局私钥 ASK_{uid} 和 *token*,计算得到 *PCK*

$$\frac{C_0}{token^{ASK_{uid}}} = \frac{PCK\cdot\left(\prod_{j\in A_c}e(g,g)^{\alpha}\right)^{s}}{token^{\gamma}} = PCK$$

(h) CTDe $(RK,CT)\rightarrow M$:得到 *PCK* 之后,用户可以用 *RK* 解密 *CT* 得到

$$M = \mathrm{D}_{RK}(CT) = \mathrm{D}_{RK}(\mathrm{E}_{WK}(M)).$$

当具有读写权限的用户读取 M 后,如果该用户需要写数据时,重复(d) $CT = \mathrm{E}_{WK}(M)$得到密文,并将密文上传 CSP,CSP 先通过用户的全局公私钥对进行签名和身份认证,然后通过 DataID 查找到该原始存储,并核对此用户的 *uid*,如果一致,CSP 将更新所有的 *CT*,并记录时间戳,保证数据的一致性.

(4) 属性撤销阶段:属性撤销主要是为了让不具有该属性的用户无法再访问数据. 在属性撤销阶段,首先 MA 会更新用户的相关属性集,然后分别对用户密钥和密文中涉及变更属性集的项进行更新,然后和未发生更新的项组合成新的密钥和密文.

(i) SKUpd$(SK_{i,j},ATT'_j)\rightarrow SK'_{i,j}$此算法在 U 进行,MA 根据新属性集的 ATT'_j 选择新的随机数 $t'_j \in \mathbb{Z}_p$,与 ATT'_j 相对应,计算新的属性更新密钥 $g^{t'j}$和密钥更新参数 $d = t'_j/t_j$,然后将新的属性更新密钥发送给 DO,将 d 发送给所有撤销属性的用户,用户更新私钥如下:

$$K'_{1,j} = K'^{d}_{1,j} = \mathrm{H}(uid)^{t'_j}_{(\forall j\in ATT_{new})}$$

得到新的私钥:

$$SK'_{i,j} = (K_0, K'_{1,j} = \mathrm{H}(uid)^{t'_j}_{(\forall j\in ATT_{new})})$$

(j) CUpd$(C,ATT'_j)\rightarrow C'$ 此算法在 CSP 进行,DO 收到属性更新密钥 $g^{t'j}$,然后选择新的随机向量 $\vec{v}' = (s',y'_2,\cdots,y'_n)\in\mathbb{Z}_p^n$,计算 $\lambda'_x = \vec{v}'\cdot\boldsymbol{M}_x$,计算一组密文更新密钥 *CUK*:

$$CUK_0 = \left(\prod_{j\in A_c}e(g,g)^{\alpha}\right)^{s'-s}, CUK_{1,j} = s'/s,$$

$$CUK_{2,x} = g^{\lambda'_x-\lambda_x}g^{t'_{\rho(x)}r-t_{\rho(x)}r}_{\forall x\in(1,\cdots,l)}, if\rho(x) = j,$$

将 *CUK* 发送给 CSP,CSP 完成对密文的更新:

$$C'_0 = C_0\cdot CUK_0 = PCK\cdot\left(\prod_{j\in A_c}e(g,g)^{\alpha}\right)^{s'},$$

$$C'_{1,j} = C_{1,j}^{CUK_{1,j}} = g^{\beta s'}_{(j\in A_c)},$$

$$C'_{2,x} = \begin{cases}g^{\lambda'_x}g^{t'_{\rho(x)}r}_{\forall x\in(1,\cdots,l)}, \text{if } \rho(x)\neq j\\ C_{2,x}\cdot CUK_{2,x} = g^{\lambda'_x}g^{t'_{\rho(x)}r}_{\forall x\in(1,\cdots,l)}, \text{if } \rho(x) = j,\end{cases}$$

得到新的密文:$C' = (C'_0, C'_{1,j}, C'_{2,x}, C_3, C_4)_{j\in A_{c_-},x\in(1,\cdots,l)}$

(5) 权限更改阶段:

如果一个用户需要授权写权限,首先令 $PCK = \{WK,RK\}$,根据用户的属性更新属性集 ATT'_j,执行(j)得到新的密文,同时授予写权限,然后更新 CSP 中写权限数据存储结构的相关数据项,记录 U_{uid}和 timestamp.

如果一个用户需要撤销写权限,首先重新执行(b) DOkeyGen $(\alpha'_d,\beta'_d)\rightarrow(WK',RK')$,生成新的 *PCK*',令 *PCK*' = *RK*',根据用户的属性更新属性集 ATT'_j,执行(j)得到新的密文,由于是撤销用户权限所以不需要更新用户密钥,然后更新 CSP 中写权限数据存储结构的相关数据项,记录 U_{uid}和 timestamp. 权限更改算法如算法 1 所示.

算法 1 PermissionUpd (wperm, U_{uid}, ATT'_j)

```
1: if wperm = 1 then
2:   C = Encrypt ({WK,RK}, 𝔸, PK);
3:   CT' = E_WK'(M);
4:   if RWData.DataID(M) = ∅ then
5:     Insert into RWData(all);
6:   else C' = CUpd(C, ATT'_j);
7:     updata RWData(.C←C',.CT←CT',.U_uid←U'_uid,
       timestap);
8:   end if
9: end if
10: if wperm = 0 then
11:   random pick (α'_d, β'_d) ←Z_p;
12:   (WK',RK') = DOkeyGen(α'_d, β'_d);
13:   C = Encrypt(RK', 𝔸, PK);
14:   CT' = E_WK'(M);'
15:   if num(RWData.U_uid) = 1 then
16:     delete from RWData(U_uid);
17:   else C' = CUpd(C, ATT'_j);
18:     updata RWData(.C←C',.CT←CT',.U_uid←U'_uid,
```

timestap);
19: end if
20: updata ORData(. $C \leftarrow C'$,. $CT \leftarrow CT'$);
21:end if

5 安全性证明

这一章从数据保密性,前向和后向安全,IND-CPA 安全三方面对 DACPCC 进行安全性分析与证明.

5.1 数据保密性

DACPCC 的数据保密性可以通过用户的属性不能满足密文的访问结构来保证,由于用户的属性集不能满足访问结构$\mathbb{A}(\boldsymbol{M},\rho)$,用户就不能计算出 $e(g,\mathrm{H}(uid))^{\lambda x}$,从而就不能够解密出密文. 当用户被撤销属性后,用户同样不能满足访问结构,所以同样不能访问数据,直到用户的属性重新满足. 另一方面,数据提供者可以授权用户来修改数据,但非法用户假冒授权用户来修改数据是不可能的,因为只有授权用户才能解密得到读写权限控制密钥 WK,在上传到 CSP 之后,首先要通过用户全局密钥进行认证,其次 CSP 要核实写权限用户的 uid 和时间戳,满足后才能更新数据,所以 DACPCC 的机密性得以保证.

5.2 前向安全和后向安全

属性撤销要求系统要做到两方面的安全保障,第一是前向安全,即新加入的用户只要满足属性,就能够解密其加入之前发布的密文;第二是后向安全,即其属性被撤销的用户不能解密基于这些属性结构加密的密文. 当一个用户的属性被撤销,其和此属性相关的解密的公钥和私钥都会发生更新,并且密文也要通过数据所有者重新加密,因此被撤销属性的用户不能通过旧的密钥计算出新的参数$\left(\prod_{j\in A_C} e(g,g)^{\alpha}\right)^{s'}$,因此能够保证后向安全. 当一个新用户加入到 DACPCC 后,系统会重加密与该属性及相关的密文,更新相关的密钥,新用户依照策略就可以访问数据,保证了前向安全.

另一方面,如果一个用户被授权了写权限,数据提供者要更新 CSP 中写权限数据存储结构的相关数据项,用户可下载写权限数据相关的密文,解密后可以上传新的数据,保证了数据正常读写. 如果用户的写权限被撤销,数据提供者要告知 CSP 取消写权限数据相关的数据项,同时更新权限控制密钥 PCK,保证用户不能更新数据.

5.3 IND-CPA 安全证明

定理 1　如果在群$(\mathbb{G}_1,\mathbb{G}_2)$上的 DBDH 假设成立,则 DACPCC 是 IND-CPA 安全的.

证明:假定一个多项式时间内敌手 A 以优势 ε 解决 DBDH 假设,我们构建一个模拟器 B,以概率 $\varepsilon/2$ 来区分一个 BDH 元组 T_{BDH}和一个随机元组 T_{rand}.

(1)挑战者 C 产生双线性对映射 $e:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_2$,$g$ 为群$\mathbb{G}_1$的一个随机生成元.

(2)挑战者 C 随机选择 $a,b,c,z\in\mathbb{Z}_p$,掷硬币 $\theta\in\{0,1\}$,如果 $\theta=0$,则设定$(g,A,B,C,Z)=(g,g^a,g^b,g^c,e(g,g)^{abc})$为 T_{BDH},如果 $\theta=1$,则设定$(g,A,B,C,Z)=(g,g^a,g^b,g^c,e(g,g)^z)$为 T_{rand}.

(3)挑战者 C 把(g,A,B,C,Z)发送给模拟器 B,在以下的交互中,模拟器 B 将作为挑战者.

Init:敌手 A 选择一个访问结构$\mathbb{A}^*$公开宣布挑战模拟器 B.

Setup:模拟器 B 运行 MASetup 算法,选择随机数 $\alpha,\beta\in\mathbb{Z}_p$,生成公钥 $PK=\{Y=e(g,g)^{\alpha},D=g^{\beta}\}$,并将 PK 发送给敌手 A.

Phase 1:该阶段 A 要向模拟器 B 进行一系列的询问,

(1)敌手 A 向模拟器 B 询问用户 $i(i\in U)$关于满足属性集 $ATT_j^*(ATT_j^*(\mathbb{A}^*)=0)$的私钥.

(2)模拟器 B 接收到敌手 A 的私钥询问后,对于每一个属性 $j(j\in ATT_j^*)$,选取随机数 $t_j\in\mathbb{Z}_p$,计算:

$K_0=\mathrm{H}(uid)^{\frac{1}{\beta}}\cdot g^{\frac{\alpha}{\beta}}$和 $K_{1,j}=\mathrm{H}(uid)^{t_j}{}_{(\forall j\in ATT)}$,并将私钥 $SK_{i,j}=(K_0,K_{1,j})$发送给敌手 A.

Challenge:(1)敌手 A 选取两个等长度的明文 M_0 和 M_1 并提交给模拟器 B,模拟器 B 掷硬币 $\eta\in\{0,1\}$.

(2)模拟器 B 选择随机的秘密数 $s,r\in\mathbb{Z}_p$ 和随机向量 $\boldsymbol{v}=(s,y_2,\cdots,y_n)\in\mathbb{Z}_p^n$,对于访问结构$\mathbb{A}^*(\boldsymbol{M}^*,\rho)$,计算 $\lambda_x^*=\boldsymbol{v}\cdot\boldsymbol{M}^*_{x(\forall x\in(1,\cdots,l^*))}$,并运行 Encrypt 算法对 M_η 进行加密,A_C^* 是访问结构$\mathbb{A}^*$的属性集,计算密文 C:

$C_0=M_\eta\cdot Z, C_{1,j}=g^{\beta s}(j\in A_C^*)$,

$C_{2,x}=g^{\lambda_x^*}g^{t_{\rho(x)}r}{}_{(\forall x\in(1,\cdots,l^*))}, C_3=g^r, C_4=g$

最后将 $C^*=(C_0,C_{1,j},C_{2,x},C_3,C_4)$返回给敌手 A.

(3)如果 $\theta=0$,则 $Z=e(g,g)^{abc}$,令 $ab=\alpha,c=s$,则 $C_0=M_\eta\cdot Z=M_\eta\cdot e(g,g)^{abc}=M_\eta\cdot e(g,g)^{\alpha s}$,所以 C^* 是密文 M_η 的随机加密,即 C^* 是 M_η 的合法密文.

(4)如果 $\theta=1$,则 $Z=e(g,g)^z$,$C_0=M_\eta\cdot Z=M_\eta\cdot e(g,g)^z=M_\eta\cdot e(g,g)^z$,因为 $z\in\mathbb{Z}_p$,所以 C_0 也是一个随机元素,即 C^* 不包含 M_η.

Phase 2:与 Phase 1 相同.

Guess:敌手 A 提交对 η 的猜测结果 η'.

(1)如果 $\eta'=\eta$,模拟器 B 输出 $\theta=0$,表明敌手 A 给出了有效的 BDH 元组 $T_{\mathrm{BDH}}:(g,A,S,Z)=(g,g^{\alpha},g^s,e(g,g)^{\alpha s})$.

(2)如果 $\eta'\neq\eta$,模拟器 B 输出 $\theta=1$,表明敌手 A 给出了无效的随机 5 元组

$T_{\mathrm{rand}}:(g,A,B,C,Z)=(g,g^a,g^b,g^c,e(g,g)^{abc})$.

(3)计算敌手 A 攻击优势:

在 $\eta' \neq \eta$ 条件下,当 $\theta = 1$ 时,敌手 A 给出的是一个无效的密文,敌手 A 只能随机猜测,概率为 1/2,用条件概率表示为:$Adv_1 = \Pr[\theta = 1 | \eta' \neq \eta] = \frac{1}{2}$;

因为 $\theta = 1$ 和 $\eta' \neq \eta$ 两个事件独立,即只要 $\theta = 1$,A 就只能给出一个无效的密文,根据独立事件的条件概率性质,

$$Adv_1 = \Pr[\theta = 1 | \eta' \neq \eta] = \Pr[\theta = 1 | \eta' = \eta] = \frac{1}{2};$$

根据 DBDH 假设定义,

$$\begin{aligned} Adv_{\mathrm{P}}^{\mathrm{DBDH}} &= \Pr[\mathrm{P}(g, g^a, g^b, g^c, R) = 0] \\ &\quad - \Pr[\mathrm{P}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] \\ &= \Pr[\theta = 0 | \eta' = \eta] - \Pr[\theta = 1 | \eta' = \eta] \leqslant \varepsilon \end{aligned}$$

推出 $\Pr[\theta = 0 | \eta' = \eta] \leqslant \frac{1}{2} + \varepsilon$.

所以根据定义 1,

$$\begin{aligned} Adv_{\mathrm{DACPCC}}^{\mathrm{IND\text{-}CPA}} &= \Pr[\eta' = \eta] - \frac{1}{2} \\ &= \frac{1}{2}(\Pr[\theta = 0 | \eta' = \eta]) + \frac{1}{2}(\Pr[\theta = 1 | \eta' = \eta]) - \frac{1}{2} \\ &\leqslant \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \leqslant \frac{\varepsilon}{2} \end{aligned}$$

由上得知,如果一个多项式时间内敌手 A 解决了 DBDH 假设,则他的优势不大于可忽略的值 ε,所以 $\varepsilon/2$ 也可忽略,那么根据定义 2 得:以可忽略的优势在 IND-CPA 攻击游戏中获胜,该方案是 IND-CPA 安全的.

证明完毕

6 性能评价分析

6.1 理论分析

这一节我们将 DACPCC 和 DACC[13]、ABACCS[16]进行性能对比分析,从综合性能分析、运算能力和属性撤销与权限修改 3 个方面进行分析,如表 1 ~ 3 所示. 设 $|p|$ 为群 $\mathbb{G}_1$、$\mathbb{G}_2$ 和空间 $\mathbb{Z}_p$ 阶 p 的大小,n_{ac} 是密文属性的总数,n_{au} 是用户属性的总数,n_u 是用户的数量,n_{nru} 是没有被撤销属性的用户的数量,n_{rac} 是包含撤销属性的密文数量,n_{aa} 是多属性机构中属性的总数. $|M|$ 是明文的大小,$|K_s|$ 是用于加密明文的对称密钥的大小,$|K_a|$ 是用于加密明文的非对称密钥的大小.

表 1 综合性能评价

名称	权限控制	加解密		属性撤销开销($\|p\|$)	安全性能		密文更新	权限修改	认证中心
		加密秘钥	解密方		后向安全	前向安全			
DACC	否	ABE	U	$O(n_{nru} n_{rau})$	是	否	DO	否	多
ABACCS	是	K_{data}	U	$O(n_{ac+}\ n_{au})$	否	否	DO	DO	单一
DACPCC	是	PCK	CSP,U	$O(n_{nru}+1)$	是	是	CSP	CSP	单一

表 2 运算能力

名称	公钥长度	私钥长度	密文长度	加密开销	解密开销
DACC	$(n_{ac}+2n_{aa})\ \|p\|$	$(n_{nru}+n_{aa})\ \|p\|$	$(3n_{ac}+1)\ \|p\|$	$O(n_{ac}\|M\|)$	$O(n_{au})$
ABACCS	$\|K_a\|+\|K_s\|+(n_{ac}+1)\ \|p\|$	$(2n_{nru}+1)\ \|p\|$	$(3n_{ac}+1)\ \|p\|$	$O(n_{ac}\|K_s\|+\|K_a\|+\|K_s\|)$	$O(n_{au}+\|K_s\|)$
DACPCC	$\|K_a\|+(n_{ac}+1)\ \|p\|$	$(2n_{nru}+1)\ \|p\|$	$(3n_{ac}+1)\ \|p\|$	$O(n_{ac}\|K_a\|+\|K_a\|)$	$O(1+\|K_a\|)$

表 3 属性撤销与权限修改

名称	密钥更新	密文更新	权限更改
DACC	否	$n_{rac}n_{nru}\|p\|+\|p\|$	否
ABACCS	否	$n_{ac}\|p\|$	$(n_{ac}+\ n_{au})\|p\|$
DACPCC	$n_{nru}\|p\|$	$\|p\|$	$(n_{nru}+1)\|p\|$

由上述表格表明,DACPCC 无论从功能、运算能力、存储开销、属性撤销和权限修改等方面都明显强于 DACC 和 ABACCS,综上所述,由于 DACPCC 的权限控制能力突出、加解密、属性撤销、权限修改等方面良好的设计使其能够成为更加全面、更加高效的云计算访问控制技术.

6.2 仿真实验

为了进一步评估本文方案在实际云计算系统中的效率,我们对 ABACCS 和 DACPCC 做了对比仿真实验. 仿真实验的操作系统是 windows7,硬件为 Intel Core i5 - 2430M, 2.4GCPU, 4GB 内存,平台采用了 Pairing-Based Cryptography library 来仿真访问控制系统,加密算法为 128b AES 和 1024b RSA. 实验对象是拥有 1 至 10 个属性的单个用户,数据大小为 256K,实验过程是在不同属性数目的用户访问数据时,分别对其加密、解密和权限修改的时耗进行仿真,结果如图 4 所示.

通过实验可以看出,首先在加密和解密阶段:DACPCC 有绝对优势,尤其是在解密阶段中,由于采用

token 方式,大量的解密过程通过 CSP 来处理,用户的解密的运算量非常小,体现了 DACPCC 的高效. 其次在权限修改阶段,我们通过对用户的一次授权和一次撤销写权限的过程来进行实验,可以看出 DACPCC 也要明显优于 ABACCS,权限修改的过程大部分时间消耗在修改密文的过程,DACPCC 能够通过提取未撤销用户的属性直接对密文进行修改,所以在权限更改时节省了大量的时间. 综上所述,DACPCC 在加密、解密和权限更改方面都有着更小的时延,更偏向于用户操作.

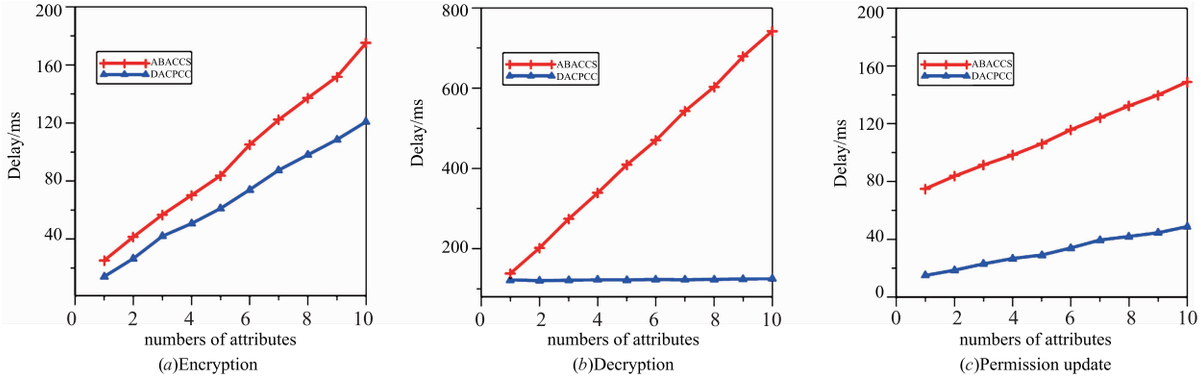

图4　不同属性数目的加密、解密、权限更改时耗对比

7　结束语

本文针对传统 CP-ABE 在云计算访问控制中没有涉及用户的访问权限这一问题,提出了一种包含访问权限的高效云计算访问控制系统 DACPCC. 文章分析了云计算访问控制系统存在的不足,引入了权限控制密钥来进行权限的授予与撤销,并设计解密 token 将繁重的解密运算分配给云平台,文章最后从理论的角度对 DACPCC 进行了安全性分析,同时基于开源的加解密函数库进行了仿真性能实验和评价,结果表明该系统安全且高效. 后续的工作中,一方面将基于 Openstack 开源云平台设计和实现一个 DACPCC 原型系统,从实现层面对所设计的方案进行安全和性能方面更全面的评估验证,另一方面,尝试将 DACPCC 与 RBAC(Role-based Access Control)结合起来,设计更完善的云计算访问控制系统.

参考文献

[1] 王于丁,杨家海,等. 云计算访问控制研究综述[J]. 软件学报,2015,26(5):1129 - 1150.
Wang YD, Yang JH, et al. Survey on access control technologies for cloud computing[J]. Journal of Software, 2015,26(5):1129 - 1150. (in Chinese)

[2] 冯朝胜,秦志光,等. 云计算环境下访问控制关键技术[J]. 电子学报,2015. 42(2):3125 - 319.
Feng C, Qin Z, et al. Key techniques access control for cloud computing[J]. Acta Electronica Sinica, 2015, 42(2):312 - 319. (in Chinese)

[3] Sahai A, Waters B. Fuzzy identity-based encryption[A]. Advances in Cryptology-EUROCRYPT 2005[C]. Berlin, Heidelberg: Springer-Verlag, 2005. 457 - 473.

[4] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[A]. Proc of the 2007 IEEE Symp on Security and Privacy[C]. Washington: IEEE Computer Society, 2007. 321 - 334.

[5] 李凤华,苏芒,史国振,马建峰. 访问控制模型研究进展及发展趋势[J]. 电子学报,2012,40(4):805 - 813.
Li FH, Su M, Shi GZ, Ma JF. Research status and development trends of access control model[J]. Acta Electronica Sinica, 2012,40(4):805 - 813. (in Chinese)

[6] 俞能海,郝卓,等. 云安全研究进展综述[J]. 电子学报,2013,41(2):371 - 381.
Yu NH, Hao Z, et al. Review of cloud computing security[J]. Acta Electronica Sinica, 2013,41(2): 371 - 381. (in Chinese)

[7] Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption[A]. Proc of the ICALP 2008[C]. Berlin, Heidelberg: Springer-Verlag, 2008. 579 - 591.

[8] Liang XH, Cao ZF, Lin H, Xing DS. Provably secure and efficient bounded ciphertext policy attribute based encryption[A]. Proc of the ASIAN ACM Symp on Information, Computer and Communications Security (ASIACCS 2009)[C]. New York: ACM Press, 2009. 343 - 352.

[9] Liu X, Ma J, Xiong J, et al. Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption Data[J]. International Journal of Network Security, 2014,16(4):351 - 357.

[10] Wan Z, Liu J, Deng RH. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing[J]. Information Forensics and Security,

IEEE Transactions,2012,7(2):743 – 754.
[11] Kan Y, Xiaohua J, Kui R, Bo Z. DAC-MACS: Effective data access control for multi-authority cloud storage systems[A]. 2013 Proceedings IEEE INFOCOM[C]. Turn, Italy: IEEE, 2013. 2895 – 2903.
[12] Jianwei C, Huadong M. Efficient decentralized attribute-based access control for cloud storage with user revocation[A]. IEEE ICC 2014[C]. Sydney: IEEE, 2014. 3782 – 3787.
[13] Sushmita R, Amiya. N, Ivan. S, DACC: Distributed access control in clouds[A]. IEEE Trust Com'11[C]. Changsha: IEEE, 2011. 91 – 98.
[14] Attrapadung N, Imai H. Attribute-Based encryption supporting direct/indirect revocation modes[A]. Proc of the Cryptography and Coding 2009[C]. Berlin, Heidelberg: Springer-Verlag, 2009. 278 – 300.
[15] Yu SC, Wang C, Ren K, Lou WJ. Attribute based data sharing with attribute revocation[A]. Proc of the ASIAN ACM Conf. on Computer and Communications Security (ASIACCS 2010)[C]. New York: ACM Press, 2010. 261 – 270.
[16] 洪澄,江敏,冯登国. AB-ACCS:一种云存储密文访问控制方法[J]. 计算机研究与发展,2010,47:259 – 265.
Hong C, Zhang M, Feng D. MAB-ACCS: A cryptographic access control scheme for cloud storage[J]. Journal of Computer Research and Development, 2010, 47: 259 – 265. (in Chinese)
[17] Boneh D, Goh E, Nissim K. Evaluating 2-DNF formulas on ciphertexts[A]. Proc of the Theory of Cryptography (TCC2005)[C]. Berlin: Springer-Verlag, 2005. 325 – 341.
[18] Dan B, Matthew KF. Identity-based encryption from the weil pairing[A]. Advances in Cryptology-CRYPTO 2011[C]. Santa Barbara, USA: Springer, 2001. 213 – 229.
[19] Beimel A. Secure schemes for secret sharing and key distribution[D]. Haifa: Israel Institute of Technology, 1996.
[20] Taeho J, XiangYang L, Zhiguo W, Meng W. Privacy preserving cloud data access with multi-authorities[A]. 2013 Proceedings IEEE INFOCOM[C]. Turn, Italy: IEEE, 2013. 2625 – 2633.

作者简介

王于丁 男,1984 年生于河北石家庄,清华大学网络科学与网络空间研究院博士生. 研究方向为云计算安全.

杨家海(通信作者) 男,1966 年生于浙江丽水,清华大学网络科学与网络空间研究院研究员,博士生导师. 研究方向为计算机网络,网络管理与测量,云计算.
E-mail: yang@ cernet. edu. cn