证明:设存在一个多项式时间 t 内的敌手$\mathcal{A}$可以通过最多 q 次私钥提取查询,以不可忽略优势ε攻破本方案的 IND-sLa-CPA 安全,则可构造概率多项式时间的算法,以概率不小于ε'、时间不多于 t'的优势攻破判定性 l-BDHE.我们可以构造一个模拟器$\mathcal{B}$在敌手$\mathcal{A}$的帮助下,以同样的优势攻破群 G 中判定性 l-BDHE 问题.

首先,模拟器$\mathcal{B}$选定 l-BDHE 问题的相关参数,生成挑战元组$(g,e(g,g),G_1,G_2,h,y_1,\ldots,y_l,y_{l+2},\ldots,y_{2l},T)$,其中,$g$ 是 G_1 的生成元, $x\in\mathbb{Z}_p^*$.定义 $y_i=g^{x^i}$,为兼容一般性,其中, $y_0=g^{x^0}=g$.并且 T 为 l-BDHE 解 $e(g^{x^{l+1}},h)=e(y_{l+1},h)$ 或 G_2 中的随机元素 $e(g,h)^{\gamma},\gamma\in\mathbb{Z}_p^*$.当 T 是 l-BDHE 解时,模拟器$\mathcal{B}$输出 1;否则,T 是随机数,输出 0.因此,模拟器$\mathcal{B}$扮演游戏中的挑战者,并与敌手$\mathcal{A}$进行如下交互.

- 初始化

在游戏开始之前,敌手$\mathcal{A}$首先给出要攻击的主体安全标记 $L^*=\langle A_1^*,...,A_m^*,\{C_1^*,...,C_n^*\},level^*\rangle$ 以及访问控制阈值 t^*.

- 系统建立

模拟器$\mathcal{B}$收到敌手$\mathcal{A}$选取的挑战主体标记 L^*,根据定理 1、定理 2,将$\bigcup_{i=1}^{m}A^*$,$\bigcup_{i=1}^{n}C^*$ 和 $level^*$转化合并为分类属性 $\bigcup_{i=1}^{m+n}C_{k_i}^*$.设所有的深度为$\{k_1,\ldots,k_{m+n}|0\le k_i\le l\}$分类属性树记做空间$\Omega^*$,则$|\Omega^*|=m+n$,且 $\Omega^*=\bigcup_{i=1}^{m+n}C_{k_i}^*$. 相应地,分类属性 c_{h_i,k_i,j_i}^* 是第 h_i 个分类属性树中深度为 k_i 的第 j_i 个属性,其根为 $c_{h_i,0,0}^*$,则该属性的路径定义为

$$p_{h_i,k_i,j_i}^*=(c_{h_i,0,0}^*,...,c_{h_i,k_i-1,j_{i'}}^*,c_{h_i,k_i,j_i}^*).$$

模拟器$\mathcal{B}$首先生成判定性 l-BDHE 问题相关的系统参数,为简化表示,令 $2n+2m+1=l$ 生成伴随属性集 $V=\bigcup_{i=1}^{m+n-1}v_i$, 其中, $v_i\in\mathbb{Z}_p^*$ 且 $v_1,\ldots,v_{m+n-1}$ 互不相同,全属性空间为$\Omega\cup V$.在一次访问请求中,$m+n-t^*$伴随属性被选中参与运算,表示为 $V_t^*=\{v_1,v_2,...,v_{m+n-t^*}\}$. 其次,模拟器$\mathcal{B}$选择随机数 $\alpha'\in\mathbb{Z}_p^*$,隐含地令 $\alpha=\alpha'+x^l$,设置 $g_1=g^{\alpha}=g^{\alpha'}\cdot y_l$,$g_2=g^x$.进一步,$\mathcal{B}$随机选择 $\alpha_i\in\mathbb{Z}_p^*(0\le i\le l)$,计算 $u_0'=g^{\alpha_0}\prod_{i\in\Omega^*\cup V_t^*}u_i'^{-1}$ 和 $u_i'=g^{\alpha_i}\cdot y_{l-i+1}$,$1\le i\le l$.最终,$\mathcal{B}$随机选择 $\theta_{i,j}\in\mathbb{Z}_p^*$,令 $u_{i,j}=g^{\theta_{i,j}}$,其中,$u_{i,0}=1$.

计算上述参数完毕后,模拟器$\mathcal{B}$将模拟生成的系统参数 $(g,e,G_1,G_2,g_1,g_2,\{u_i'\}_{0\le i\le 2n+2m+1},\{u_{i,j}\}_{0\le i\le m+n+1,0\le j\le d_i})$ 作为公钥发送给敌手$\mathcal{A}$.注意:本步中所有参数均为 G 中独立均匀分布.

- 查询阶段 1

在此阶段,敌手$\mathcal{A}$可以进行多项式次数的适应性私钥提取询问,模拟器$\mathcal{B}$回答相应的询问.

设敌手$\mathcal{A}$对属性集Ω提交不超过 q 次私钥查询,且提交的属性集Ω^*不能通过访问控制结构,即$|\Omega\cap\Omega^*|<t^*$.模拟器$\mathcal{B}$构造一个私钥 SK 通过 $KeyGen$ 过程后传输给$\mathcal{A}$.当模拟器接受到一次私钥查询时,$\mathcal{B}$构造Ω的一个属性子集Γ,使得Γ中的属性支配Ω^*中的属性,即 $\Gamma=(\Omega\cap\Omega^*)\cup V_t^*$.同样,定义Γ',使得 $\Gamma\subseteq\Gamma'\subseteq\Omega^*\cup V_t^*$,且$|\Gamma'|=m+n$,令 $S=\Gamma'\cup\{0\}$.对于每个属性 $c\in\Gamma'$,模拟器随机选择 w,令 $q(H(c))=w$.当 $q(0)=\alpha=\alpha'+x^l$ 时,利用插值公式可唯一确定 $m+n$ 次多项式函数 $q(z)$,于是,模拟器可以对每一个分类属性 $c\in\Omega\cup V$ 按照如下公式计算其对应的私钥 sk_c:

(1) 对于每个属性 $c_i=c_{h_i,k_i,j_i}\in\Gamma'$,$\mathcal{B}$随机选取 $r_i'\in\mathbb{Z}_p^*$,令 $r_i=x^i+r_i'$,结合 $q(H(c_i))=w_i$,计算私钥如下:

$$sk_{c_i}=\langle a_i,b_i,\{d_{i,j}\}_{j\ne i,0\le j\le 2m+2n+1},\{e_{h,j}\}_{h\ne i,0\le j\le d_h;h=i,k_i<j\le d_h}\rangle \tag{11}$$

其中,第 1 部分:

$$\left.\begin{aligned}a_i&=g_2^{q(H(c_i))}(u_0'u_i'\prod_{\delta=1}^{k_i}u_{i,\delta}^{c_{h_j,\delta,\xi}})^{r_i}=(g_2^{w_i})(g^{\alpha_0}\prod_{i\in\Omega^*\cup V_t^*}u_i'^{-1}\cdot u_i'\cdot\prod_{\delta=1}^{k_i}(u_{i,\delta})^{c_{h_i,\delta,\xi}})^{x^i+r_i'}\\&=(g_2^{w_i})(u_0'u_i')^{r_i'}\left(g^{\alpha_0+\sum_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\prod_{j\in\Omega^*\cup V_t^*,j\ne i}u_i'^{-1}\right)^{x^i}\end{aligned}\right\} \tag{12}$$

第 2 部分: $b_i=g^{r_i}=g^{r_i'+x^i}=y_i\cdot g^{r_i'}$;第 3 和第 4 部分易得: $d_{i,j}=u_j''^{r_i}=(u_j')^{r_i'+x^i}$,$e_{h,j}=u_{h,j}^{r_i}=(g^{r_i'}\cdot y_i)^{\theta_{h,j}}$.

注意,模拟构造 sk_{c_i} 的困难之处在于其包含模拟器未知的 $g^{x^{l+1}}$,由于划分了 Γ,Γ' 和 S 这 3 个集合,对于 $c_i\in\Gamma$, a_i 中的因子 $(u_0'u_i')^{x^i}$ 可以消掉未知的 $g^{x^{l+1}}$.

(2) 对于每个属性 $c_i=c_{h_i,k_i,j_i}\notin\Gamma'$,也就是说 $c\notin\Omega^*\cup V_t^*$,可以通过拉格朗日插值公式计算:

$$q(H(c_i))=\sum_{c'\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot q(H(c'))+\Delta_{0,S}(H(c_i))\cdot q(0) \tag{13}$$

$\mathcal{B}$随机选取 $r_i'\in\mathbb{Z}_p^*$,令 $r_i=r_i'-\Delta_{0,S}(H(c_i))\cdot x^i$,计算私钥如下.

其中,第 1 部分(注意,此时 $u_j'=g^{\theta_j}$):

$$\left.\begin{aligned}a_i&=g_2^{\sum_{c_j\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot w_j+\Delta_{0,S}(H(c_i))\cdot q(0)}\cdot(u_0'u_i'\prod_{\delta=1}^{k_i}u_{i,\delta}^{c_{h_i,\delta,\xi}})^{r_i'-\Delta_{0,S}(H(c_i))\cdot x^i}\\&=g_2^{\sum_{c_j\in\Gamma'}\Delta_{c',S}(H(c_i))\cdot w_j+\Delta_{0,S}(H(c_i))\cdot\alpha'}(u_0'u_i')^{r'}(u_0')^{-\Delta_{0,S}(H(c_i))\cdot x^i}(y_i)^{-\Delta_{0,S}(H(c_i))\alpha_i}\end{aligned}\right\} \tag{14}$$

注意:此时对于 $c_i\notin\Gamma$,a_i 中的因子 $(g_2)^{q(0)}(u_i')^{x^i}$,可以消掉未知的 $g^{x^{l+1}}$.

第 2 部分: $b_i=g^{r_i}=g^{r_i'+x^i}=y_i\cdot g^{r_i'}$;第 3 和第 4 部分易得: $d_{i,j}=u_j''^{r_i}=(u_j')^{r_i'+x^i}$,$e_{h,j}=u_{h,j}^{r_i}=(u_j)^{r_i'+x^i}$.

因此,模拟器$\mathcal{B}$可以计算构造$|\Omega\cap\Omega^*|<t^*$的身份私钥,其分发过程与原有系统模式相同.

- 挑战阶段

敌手$\mathcal{A}$提交两个相同长度的挑战明文 m_0 和 m_1,以及属性空间Ω.模拟器$\mathcal{B}$随机抛一枚硬币,即,随机选择 $\beta\in\{0,1\}$,并构造返回 m_β的密文给敌手$\mathcal{A}$:

$$CT\to\left(m_\beta\cdot T\cdot e(y_1,h^{\alpha'}),h,h^{\alpha_0+\sum\limits_{j\in\Omega^*\cup V_t^*}\sum\limits_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\right) \tag{15}$$

我们将讨论:当 T 是 l-BDHE 的挑战时,CT 是 m_β的有效加密;当 T 是随机时,CT 是一个随机信息的加密.

首先,注意:由于 h 是 l-BDHE 问题中的均匀分布,故第 2 部分的随机性是均匀分布的.敌手$\mathcal{A}$只得到与 h 相关的 CT.然后计算当 $h=g^c$ 的第 3 部分的正确形式:

$$h^{\alpha_0+\sum\limits_{j\in\Omega^*\cup V_t^*}\sum\limits_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}=\left(g^{\alpha_0}\cdot\prod_{j\in\Omega^*\cup V_t^*}(u_j'^{-1})\prod_{j\in\Omega^*\cup V_t^*}(u_j')g^{\sum\limits_{j\in\Omega^*\cup V_t^*}\sum\limits_{\delta=1}^{k_j}\theta_{j,\delta}c_{j,\delta,\xi}}\right)^c=\left(u_0'\cdot\prod_{j\in\Omega^*\cup V_t^*}\left(u_j'\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{j,\delta,\xi}}\right)\right)^c \tag{16}$$

最后,针对同样的 c,可以得到:

$$e(g,h)^{x^{l+1}}\cdot e(y_1,h^{\alpha'})=(e(y_1,y_l)\cdot e(y_1,g^{\alpha'}))^c=e(y_1,y_l\cdot g^{\alpha'})^c=e(g_1,g_2)^c \tag{17}$$

因此,对比上述 CT 中的未知元素 T,对于选定的属性集Ω^*,当 $T=e(g,h)^{x^{l+1}}$ 时,$\mathcal{B}$可以构造 m_β的有效密文:

$$CT=\left(m_\beta\cdot e(g_1,g_2)^c,g^c,\left(u_0'\cdot\prod_{j\in\Omega^*\cup V_t^*}\left(u_j'\cdot\prod_{\delta=0}^{k_j}u_{j,\delta}^{c_{j,\delta,\xi}}\right)\right)^c\right) \tag{18}$$

当 T 是随机数时,则 CT 是一个随机信息的加密.

- 查找阶段 2

与查找阶段 1 类同,模拟器$\mathcal{B}$相应的响应敌手$\mathcal{A}$的查询.

- 猜测

最终,敌手$\mathcal{A}$输出猜测的β',若$\beta=\beta'$模拟器$\mathcal{B}$输出 1,即,猜测 $T=e(g,h)^{x^{l+1}}$;否则,模拟器$\mathcal{B}$输出 0,表明它认为 T 是 G_2 中的随机元素.

- 概率分析

上述挑战-应答游戏成功,即,挑战者解决判定性 l-BDHE 问题的成功概率分析如下.

(1) 如果模拟器$\mathcal{B}$的输出为 1,即 $T=e(g,h)^{x^{l+1}}$,挑战密文 CT 是对 m_β的有效密文,敌手$\mathcal{A}$的环境被完美模拟,

因此,敌手有ε概率成功解密,$\left|\Pr[\beta=\beta']-\frac{1}{2}\right|\geq\varepsilon$;

(2) 如果模拟器$\mathcal{B}$的输出为 0,即 T 是 G_2 中的随即元素,敌手.$\mathcal{A}$不能得到有关β的任何有效信息,因此,敌手有ε概率成功解密,$\left|\Pr[\beta\neq\beta']-\frac{1}{2}\right|=0$.

综上,模拟器$\mathcal{B}$可以以不可忽略的优势解决 *l-BDHE* 问题:

$$|\Pr[\mathcal{B}=1|T=e(g,h)^{x^{l+1}}]-\Pr[\mathcal{B}=0|T\in Random]|\geq\left|\left(\frac{1}{2}+\varepsilon\right)-\frac{1}{2}\right|=\varepsilon \tag{19}$$

证毕. □

6.3 效率分析

本节中,我们将分别将 CGAC 算法与已有的使用层次化结构的属性基加密算法、已有的定长密文属性加密算法进行效率和安全性对比.其中,计算开销主要由加密运算和解密运算组成,而通信开销以及存储空间长度需要分析密文长度、私钥空间.除此之外,本文还给出了方案之间的访问控制结构和安全性对比,具体见表 2.

Table 2 Comparison with the proposed scheme and existing schemes

表 2 本文算法与已有方案之间的对比

方案	密文长度	最长私钥长度	加密	解密	安全	特点	控制结构
AL[25]	$2G_1+G_2$	$(2n+5)G_1$	$4E$	$3P+(n-1)E$	CPA	定长	固定
HLR[26]	$2G_1+G_2$	$(2n-1)G_1$	$(n+t+1)E$	$3P+O(t^2)E$	CPA	定长	$(t-n)$阈值
HASBE[12]	$(2n+3)G_1+G_2$	$n(l+n+3)G_1$	$(2n+4)E$	$(2P+E)n$	CPA	层次	与或门
HABE[27]	$(n+1)G_1+G_2$	$(l+1)nG_1$	$(n+2)E$	$2n(2P+E)$	CPA	层次	$(t-n)$阈值
本文 CGAC	$2G_1+G_2$	$((2+l)(n+1)+3)G_1$	$3E$	$2P+2nE$	CPA	层次+定长	$(t-n)$阈值

在上述性能比较中,加密、解密分别表示加密解密的计算复杂度,密文长度和最长私钥长度表示存储的空间复杂度,安全表示论文中所给出的方案安全性证明,特点及控制结构分别表示论文方案的特点和其访问控制的结构.设 P 表示最耗时的一个双线性对运算时间,E 表示相对次耗时的一个幂指运算时间,G_1和 G_2分别表示所在群中元素的长度.同时,在表格中我们定义空间的所有属性个数为 n,层次属性的最大深度为 l,用户可以解密数据的属性个数阈值 t,则$(t-n)$阈值表示访问控制结构为 n 个属性中存在 t 个属性满足即可访问.

通过对比本文所提出的方案,相较于文献[26,27],其密文长度、加解密运算所需的双线性对运算与系统属性个数相关,本文方案所带来的计算开销及存储空间都将大大缩小.同时,与其他定长方案[25,26]相比,本文的访问控制结构比较灵活,同时,加解密的计算开销较小.此外,本文方案同样满足 CPA 安全性.但需要指出的是:与其他方案相比,由于本方案将属性空间相关参数嵌入私钥,使计算后的密文空间达到固定长度,因而产生的私钥长度较大,带来相应的存储空间开销.尽管如此,本方案通过牺牲用户的私钥存储空间,达到了降低访问通信带宽开销、提高计算效率的目的.因为在实际运用的网络环境中,系统间通信带宽通常成为制约的瓶颈,提高维护通信带宽的成本代价又非常昂贵,而相对的存储的增加和维护十分容易,以 SS512 的群为例,100 个分类属性和最大深度为 30 的私钥存储开销最大仅为 20M,因此,本文方案是可行的.

7 系统实现与结果分析

7.1 系统设计

Openstack Swift 是一种典型并且开源的对象存储,其作为云基础服务 Openstack 的核心子项目之一,为其他子项目提供存储服务.Swift 利用便宜的基础硬件存储,通过软件层面的算法,引入一致性散列技术实现数据冗余性和均衡分布,同时支持多租户模式、容器和对象读写操作,适用于存储互联网应用场景下的非结构化数据.

7.1.1 访问控制中间件

Swift 利用 Proxy Server 模块对外提供标准的基于 HTTP 的 REST 接口,对账户、容器和对象进行 CRUD

操作.而在 Swift 内部,它利用 Python 的 WSGI 模型(Web services gateway interface)和 Python Paste 框架构建,根据 Pipeline 配置中的调用顺序,依次通过中间件处理 Swift 的请求链.中间件类似于洋葱结构包裹在 Swift 核心模块之上,请求会依次通过各个加载中间件,我们可以定制自己的中间件组件,处理进出中间件的响应请求,在到达核心 Swift 之前修改其中的请求数据,或者直接交给下层中间件处理,也可以在本层直接响应结果.

如图 4 所示,本文的方案是将访问控制策略通过 Swift 中间件实现,用户通过 REST 接口向 proxyserver 提交访问请求.其后,请求被交给访问控制中间件处理.访问控制中间件从请求中获取 Token,从而根据第 4.2 节中访问控制模型中的流程验证用户身份并获取主体标记;同时,从请求中获取客体标记,利用第 4.1 节访问控制规则 1 和规则 2 进行判定:如果不满足规则要求,则直接返回响应状态码“403 Forbidden”;否则,交给下层中间件进一步进行下个逻辑的处理.

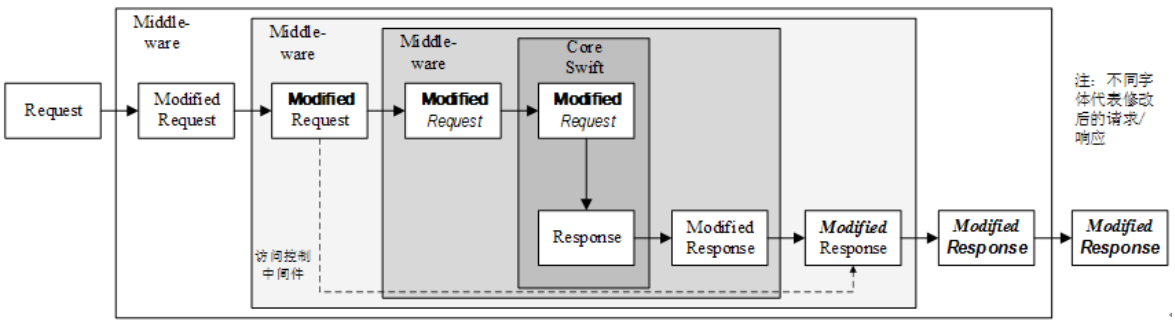


Fig.4 Process flows of Swift middle wares

图 4 Swift 中间件处理流程

7.1.2 系统实现

本节主要基于 Openstack Swift 介绍第 4.2 节中访问控制模型中的对象云存储的具体实现,分别对模型中的 3 个参与者进行详细描述.

(1) 对于访问控制模型中的云存储服务 CSP,主要实现对象存储如图 5 所示,其中:访问决策模块进行访问控制策略的判定由 Swift 中间件实现,其编写规则参照 WSGI 标准;标记解析模块负责主客体标记的解析,通过 Token 从 Keystone 获取用户的主体标记,从元数据管理模块获取客体的标记,交给访问控制决策模块分析;当用户满足访问控制规则,由访问决策模块交给后端控制模块 Controller,通过一致性散列技术完成相应的对象数据或元数据的 CURD 操作.

需要特别指出的是:由于 Swift 的元数据最大长度默认为 256B,元数据越短,服务器可以缓存更多的元数据,保持较高的响应速度;而当元数据长度增长时,会消耗大量硬件计算资源和存储资源,使云存储服务的性能急剧降低,因此,固定长度的密文可以作为元数据存储.当用户进行上传操作时,Proxy Server 通过 REST 接口获取到用户 POST 的数据客体标记,及通过 CGAC 算法加密的固定长度的密文后,将上述信息作为对象数据密文的元数据存储.

另外,由于采用无状态 REST 协议,代理服务 proxy server 和存储结点都可以横向扩展来实现负载均衡,避免单点故障,此时,访问控制中间件都需要配置并加载在 proxy server 的 pipeline 中.同时还需要修改 Swift 的 Cache 集群结构,利用一致性散列分配地址空间,缓存 Token 的验证和主体标记.

(2) 访问控制模型中的主从 KGC 则由 Keystone 身份认证模块负责实现,进行用户的身份认证管理及 CGAC 算法中的 *Setup*,*KeyGen* 和 *Delegate* 算法实现,完成 CGAC 系统的初始化和用户、子 KGC 的私钥产生、授权等,具体实现通过修改 Openstack 的认证组件 Keystone 来完成;认证模块与访问控制决策模块的交互,包括 Token 同步等,通过 Fernet 机制进行.

(3) 访问控制模型中的用户 User 端,需要实现的模块主要包括提供用户身份信息进行认证并获取用户私

钥进行存储,通过调用对象存储接口实现对象数据的上传、下载,以及实现 CGAC 算法中的 *Encrypt* 和 *Decrypt* 进行用户数据的加密或云数据的解密等操作.

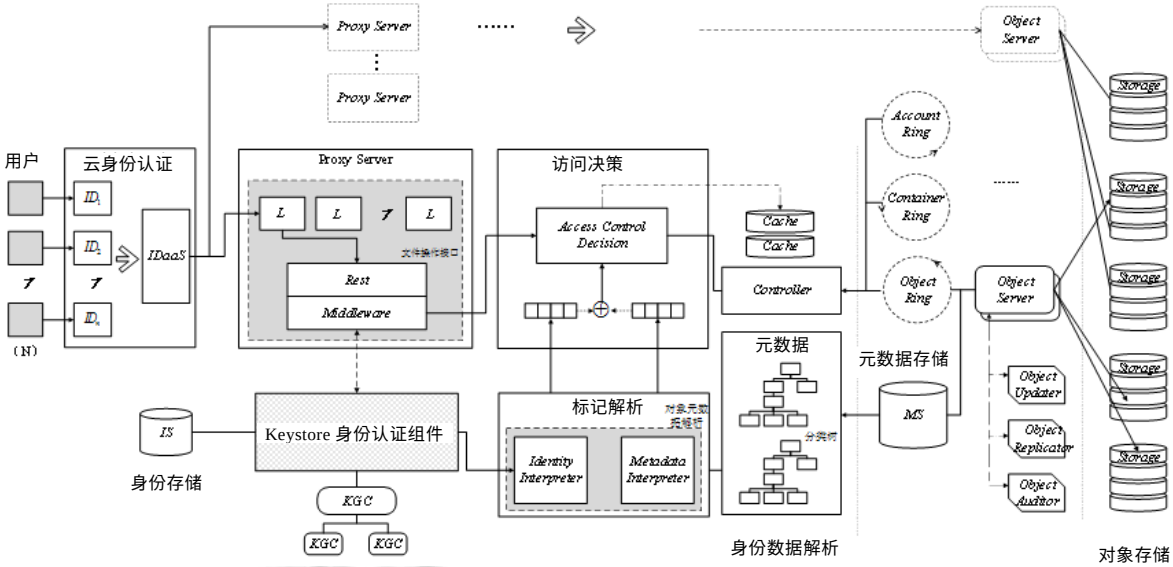


Fig.5 Implementations of access control system for objectstorage

图 5 对象云存储访问控制系统实现

7.2 实验环境

根据上述的系统结构,在 2.7GHZ inteli5 CPU,4GB DDR3RAM 的电脑上通过 VMWARE 建立一个 4 核 CPU 和 4GB 内存的虚拟机,运行 Debian Linux8.2jessie 系统.并在该系统中建立 Swift 对象存储,其中,其配置上只有 1 个代理节点和利用本地回环建立的 4 个存储结点.而在用户客户端的加解密模块和用户私钥生成模块的实现中,我们使用了 Charm-Crypto[28]作为双线性密码计算的库,并选择群中元素 g 的大小$|g|$为 512 比特,利用 python 语言实现了对象云存储中分类分级数据的访问控制系统.限于篇幅,本文系统的部分核心实现可在 GITHUB[29] 上获取.

7.3 结果分析

通过在上述环境中实现了图 5 架构下的整个系统,并获取了一些系统运行截图:

如图 6 左所示,展示用户通过 REST 接口获取 Token 的运行图;而图 6 右则展示了当用户请求下载数据时,由于主体标记中的安全级别小于客体的安全级别,下载失败.

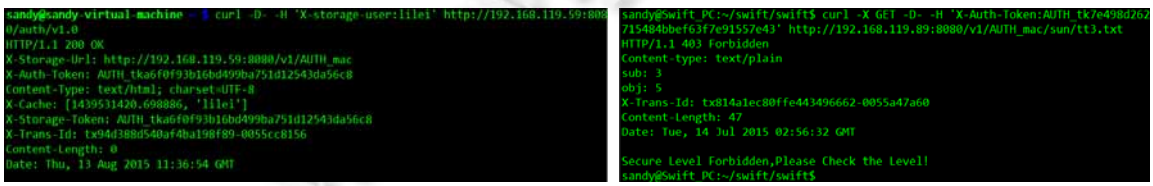


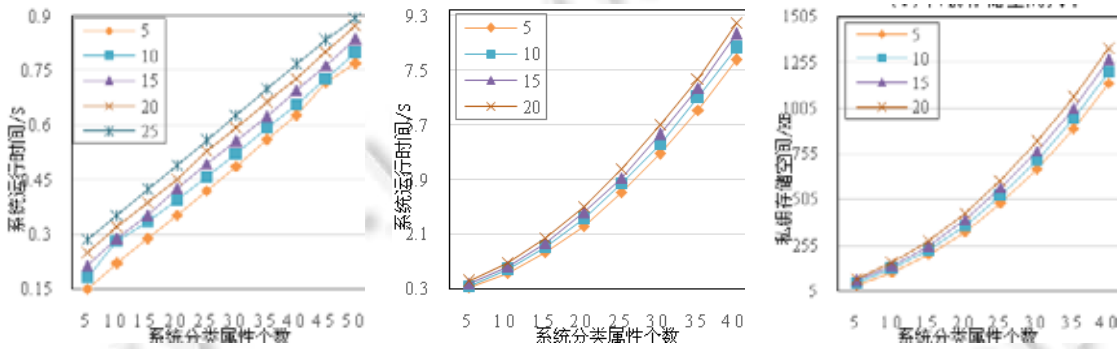
Fig.6 Screen shots of system

图 6 系统运行截图

通过对整个对象存储访问控制系统进行测试,得出下面的整个系统运行结果图,系统运行时间包含了 CGAC 算法执行时间、加密上传或下载解密、访问控制决策时间、对象存储检索时间、元数据管理时间等系

统操作时间.由于在本地测试,统计时间不包含网络延时.时间结果也反映了用户客户端数据加密解密效率,即 CGAC 算法的 *Encrypt* 和 *Decrypt* 效率.同时,私钥生成时间也反映了分布式 keystone 的 *Keygen* 执行效率.

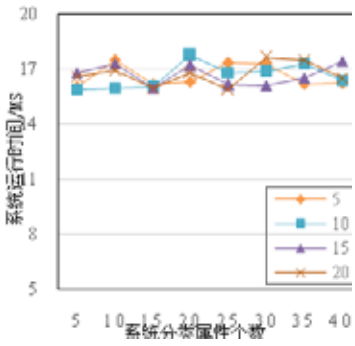
随机生成 t=2 的一个分类拓扑进行模拟访问,通过实际实验的结果,不同的曲线表示不同的分类树深度在系统分类属性个数下的运行效果.图 7(a)表明,系统建立时间与系统属性个数及分类树深度成正比.图 7(b)表明:私钥的生成时间与系统属性个数及分类树深度同样成正比关系,且增长速度变快.图 7(c)表明:私钥的存储空间与系统属性个数及分类树深度同样成正比关系,且与私钥生成时间的曲线相符.图 7(d)表明:加密算法的执行时间与系统属性个数及分类树深度相关性不大,且波动较小.还可以发现,加密时间很短平均只有 17ms,因此加密算法的效率是很高的.图 7(e)表明:对称密钥密文存储空间与系统属性个数及最大分类树深度相关性不大,且波动较小,平均对称密钥密文长度为 0.47KB,因此只要设置对象存储中单个元数据的大小大于 0.5KB 即可.图 7(f)表明:解密算法的执行时间,由于参与运算的分类树深度是固定的,解密时间所以与分类树深度相关性不大,而与系统属性个数程正相关性.



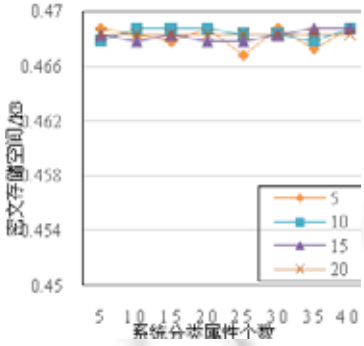
(a) 系统建立时间

(b) 私钥直接生成时间

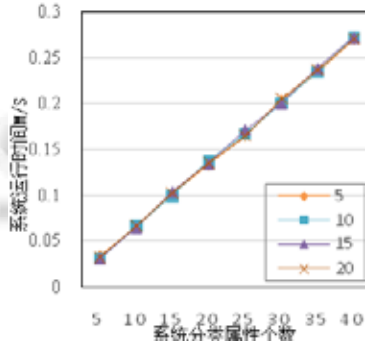
(c) 私钥存储空间大小



(d) 加密算法执行时间



(e) 密文存储空间大小



(f) 解密算法执行时间

Fig.7 Run time and communications of system

图 7 系统运行时间及通信量

通过实际运行结果的效率对比图,发现其结果与算法分析中相符,随着属性个数的增多,私钥的存储空间大将会增长很快.但是这种开销在存储廉价的现代是可以忽略的,因而,本文提出的 CGAC 可以将对称密钥加解密计算量和空间限定,使其更好的与对象云存储相结合,实现了对分类分级特征对象数据的细粒度访问控制.在真实云计算环境中,用户私钥开销可以接受,且当用户数目和系统属性个数增加时,更能体现本方案优势.此外,现有的 ABE 密文访问控制系统中都存在访问权限的更改,包括策略和属性变化时,尤其是用户属性的撤销设计难度大的难题,通常需要进行重加密,导致效率不高.CGAC 算法虽然同样具有以上问题,但是由于对象存储的场景下,存储的多为图片音频等静态数据,更新的频率很小,因此此处的性能损耗同样是在可以接受的范围内.

8 结束语

在云计算越来越普及的环境下,云存储利用网络对存储资源整合利用所面临的数据安全问题越来越多.本文针对分类分级特点的对象存储服务,提出了一套事实可行的访问控制方案和模型;同时,借助 ABE 机制,设计出一种可靠的基于分类分级属性的属性加密算法.该算法将强制访问控制、定长密文的属性加密、对象存储与分类分级特性的优势相结合,不仅提高了数据的安全性,解决了细粒度访问控制问题,同时使得计算开销和通信开销大大减少,提高了系统效率.本文同时给出了基于 OpenstackSwift 对象存储的具体实现,验证了本方案的可行性.在下一步的工作中,将研究更高效的算法降低系统复杂度,同时对阈值访问控制结构进行扩展;另外,研究基于代理重加密的撤销机制,降低撤销时的开销.

References:

[1] Factor M, Meth K, Naor D, Rodeh O, Satran J. Object storage: The future building block for storage systems. In: Proc. of the Local to Global Data Interoperability-Challenges and Technologies. IEEE, 2005. 119−123. [doi: 10.1109/LGDI.2005.1612479]

[2] Mesnier M, Ganger GR, Riedel E. Object-Based storage. IEEE Communications Magazine, 2003,41(8):84−90. [doi: 10.1109/MCOM.2003.1222722]

[3] Committee AIT. Project t10/1355-d working draft: Information technology—SCSI objectbased storage device commands. 2004.

[4] Arnold J. OpenStack Swift: Using, Administering, and Developing for Swift Object Storage. O'Reilly Media, Inc., 2014.

[5] Hamlen K, Kantarcioglu M, Khan L, Thuraisingham B. Security issues for cloud computing. International Journal of Information Security and Privacy 2010,4(2):39−51.

[6] Wang YD, Yang JH, Xu C, Ling X, Yang Y. Survey on access control technologies for cloud computing. Ruan Jian Xue Bao/Journal of Software, 2015,26(5):1129−1150 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[7] Bethencourt J, Sahai A, Waters B. Ciphertext-Policy attribute-based encryption. In: Proc. of the IEEE Symp. on Security and Privacy (SP 2007). IEEE, 2007. 321−334. [doi: 10.1109/SP.2007.11]

[8] Bell DE, Padula LJL. Secure computer system: Unified exposition and multics interpretation. In: Proc. of the Secure Computer System Unified Exposition & Multics Interpretation. 1976. 161.

[9] Shen C. Application analysis of BLP model in cloud storage. Computer & Digital Engineering, 2012,40:65−66 (in Chinese with English abstract). [doi: 10.3969/j.issn.1672-9722.2012.06.021]

[10] Lin GY, He S, Huang H, Wu JY, Wei C. Access control security model based on behavior in cloud computing environment. Journal on Communications, 2012,33(3):59−66 (in Chinese with English abstract).

[11] Horwitz J, Lynn B. Toward hierarchical identity-based encryption. In: Proc. of the Advances in Cryptology—EUROCRYPT. Springer-Verlag, 2002. 466−481. [doi: 10.1007/3-540-46035-7_31]

[12] Wan Z, Liu JE, Deng RH. HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. on Information Forensics and Security, 2012,7:743−754. [doi: 10.1109/TIFS.2011.2172209]

[13] Deng H, Wu Q, Qin B, Domingo-Ferrer J, Zhang L, Liu J, Shi WC. Ciphertext-Policy hierarchical attribute-based encryption with short ciphertexts. Information Sciences, 2014,275:370−384. [doi: 10.1016/j.ins.2014.01.035]

[14] You L, Wang L. Hierarchical authority key-policy attribute-based encryption. In: Proc. of the 2015 IEEE 16th Int'l Conf. on Communication Technology (ICCT). IEEE, 2015. 868−872. [doi: 10.1109/ICCT.2015.7399963]

[15] Wang S, Zhou J, Liu JK, Yu J, Chen J, Xie W. An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans. on Information Forensics and Security, 2016,11:1265−1277. [doi: 10.1109/TIFS.2016.2523941]

[16] Liu Z, Yan H, Lin Z, Xu L. An improved cloud data sharing scheme with hierarchical attribute structure. Journal of Universal Computerence, 2015,21(3): 454−472. [doi: 10.3217/jucs-021-03-0454]

[17] Ge A, Zhang R, Chen C, Ma C, Zhang Z. Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In: Proc. of the Information Security and Privacy. Springer-Verlag, 2012. 336−349. [doi: 10.1007/978-3-642-31448-3_25]

[18] Zhang XC, Yang G. Attribute-Based access control model with constant-size ciphertext in Hadoop cloud environment. Computer Engineering and Applications, 2015,51(23):87−93 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.1311-0372]

[19] Biswas P, Patwa F, Sandhu R. Content level access control for openstack swift storage. In: Proc. of the 5th ACM Conf. on Data and Application Security and Privacy. ACM Press, 2015. 123−126. [doi: 10.1145/2699026.2699124]

[20] Boneh D. Identity-Based encryption from the Weil pairing. In: Proc. of the Advances in Cryptology—CRYPTO 2001. Springer-Verlag, 2001. 213−229. [doi: 10.1007/3-540-44647-8_13]

[21] Boneh D, Boyen X, Goh EJ. Hierarchical identity based encryption with constant size ciphertext. In: Proc. of the Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Springer-Verlag, 2005. 440−456. [doi: 10.1007/11426639_26]

[22] Ran C, Halevi S, Katz J. Chosen-Ciphertext security from identity-based encryption. Siam Journal on Computing, 2007,36: 1301−1328.

[23] Fujisaki E, Okamoto T. Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology, 2013,26: 80−101. [doi: 10.1007/s00145-011-9114-1]

[24] Shamir A. How to share a secret. Communications of the ACM, 1979,22:612−613. [doi: 10.1145/359168.359176]

[25] Agrawal S, Freeman DM, Vaikuntanathan V. Functional encryption for inner product predicates from learning with errors. In: Proc. of the Advances in Cryptology—ASIACRYPT 2011, Int'l Conf. on the Theory and Application of Cryptology and Information Security. Seoul, 2011. 21−40. [doi: 10.1007/978-3-642-25385-0_2]

[26] Herranz J, Laguillaumie F, Ràfols C. Constant size ciphertexts in threshold attribute-based encryption. In: Proc. of the Int'l Conf. on Practice and Theory in Public Key Cryptography. 2010. 19−34. [doi: 10.1007/978-3-642-13013-7_2]

[27] Li J, Wang Q, Wang C, Ren K. Enhancing attribute-based encryption with attribute hierarchy. Mobile Networks and Applications, 2011,16:553−561. [doi: 10.1007/s11036-010-0233-y]

[28] Akinyele JA, Green M, Rubin A. Charm: A framework for rapidly prototyping cryptosystems. Cryptology ePrint Archive, Report. 2011/617. 2011.

[29] Yang T. Sample code of "an access control mechanism for classified and graded object storage in cloud computing". 2016. https://github.com/hbhdytf

附中文参考文献:

[6] 王于丁,杨家海,徐聪,凌晓,杨洋.云计算访问控制技术研究综述.软件学报,2015,26(5):1129−1150. http://www.jos.org.cn/1000-9825/4820.htm [doi: 10.13328/j.cnki.jos.004820]

[9] 沈承东,严明向.BLP 模型在云存储中应用分析.计算机与数字工程,2012,40:65−66. [doi: 10.3969/j.issn.1672-9722.2012.06.021]

[10] 林果园,贺珊,黄皓,等.基于行为的云计算访问控制安全模型.通信学报,2012,33(3):59−66.

[18] 张欣晨,杨庚.Hadoop 环境中基于属性和定长密文的访问控制方法.计算机工程与应用,2015,51(23):87−93. [doi: 10.3778/j.issn.1002-8331.1311-0372]

杨腾飞(1990－),男,河北邯郸人,博士生,主要研究领域为云计算安全,网络与系统安全.

田雪(1986－),女,助理研究员,主要研究领域为云计算安全.

申培松(1993－),男,博士生,主要研究领域为云计算安全,系统安全.

冯荣权(1966－),男,博士,教授,博士生导师,主要研究领域为密码学,信息安全.