doi:10.3969/j.issn.1000-436x.2013.z1.037

基于属性的安全增强云存储访问控制方案

牛德华，马建峰，马卓，李辰楠，王蕾
(西安电子科技大学 计算机学院，陕西 西安 710071)

摘 要：为了保证云存储中用户数据和隐私的安全，提出了一种基于属性的安全增强云存储访问控制方案。通过共用属性集，将基于属性的加密体制(ABE)与 XACML 框架有机结合，在 XACML 框架上实现细粒度的基于属性的访问控制并由 ABE 保证数据的机密性。考虑到数据量很大时 ABE 的效率较低，因此，云存储中海量敏感数据的机密性用对称密码体制实现，ABE 仅用于保护数据量较小的对称密钥。实验分析表明，该方案不仅能保证用户数据和隐私的机密性，而且性能优于其他同类系统。

关键词：云存储；访问控制；XACML 框架；基于属性的加密；共用属性集
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2013)Z1-0276-09

Enhanced cloud storage access control scheme based on attribute

NIU De-hua, MA Jian-feng, MA Zhuo, LI Chen-nan, WANG Lei
(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: In order to ensure the security of data and privacy in cloud storage, an enhanced cloud storage access control solution based on attribute was proposed. By designing a common set of attributes, attribute-based encryption(ABE) was integrated into XACML (eXtensible access control markup language) framework and the goal to ensure the confidentiality of sensitive data and to provide fine-grained access control was achieved. Considering the efficiency of ABE is very low when it is used to a large amount of data, symmetric cryptography was used to ensure the confidentiality of the vast amounts of sensitive data while ABE was used to protect the small number of symmetric keys. Experiments show that the scheme can ensure the confidentiality of the data and privacy and its performance is superior to other similar systems.

Key words: cloud storage; access control; XACML framework; ABE; attribute set

1 引言

近年来，云存储的出现使得用户可随时随地存取数据，因而得到了广泛应用。但是云存储发展的同时也引起了用户对数据安全和隐私保护方面的担忧。2009 年亚马逊、Google、LinkUp 等多家著名云存储服务商都出现用户数据和隐私泄露的安全问题，导致了严重的后果[1]；2011 年索尼“泄密门”再次给云存储安全敲响警钟。2012 年中国区云计算安全调查分析报告显示，79%的用户仍然不愿意将敏感数据存储在云环境下。如何保护用户隐私和敏感数据的机密性已成为云存储亟需解决的问题。

访问控制机制为解决上述问题提供了很好的解决思路，但在云环境下，不同的客户端和云服务提供商(CSP, cloud service provider)频繁交互，这些交互方有时位于不同的安全域内，相互只知道对方的部分信息，传统的基于角色的访问控制已不适用于类似的分布式环境下[2]。基于属性的访问控制(ABAC, attribute-based access control)是伴随着分布式应用的发展而被提出的一种访问控制机制，用于解决分布式环境下的访问控制问题，因而先天对云环境有更好的适应性。ABAC 可以根据客户属性特征并结合访问控制策略判断是否允许客户的访问请求。其基本思想是：访问控制以实体(主体、资源和环境)的属性作为基础进行授权决策，它可以随着实体属性的变化动态地更新访问控制决策，提供一种更加细粒度、灵活的动态访问控制方法。ABAC

收稿日期：2013-07-03

尤其是由 Sahai 和 Waters 提出的 ABE[3]，将解密规则蕴含在加密算法之中，可免去加密过程中频繁的密钥分发代价。由于这一良好特性，有很多用 ABE 实现的密文访问控制来解决云存储中数据安全和隐私保护的研究[4~6]。此类方法可较好地保证敏感数据的机密性，但是依然存在下述问题。1）现实的云存储中既有敏感数据又有非敏感数据，如果非敏感数据也用 ABE 实现的访问控制，系统的性能将会很低。2）用户权限变更时，数据所有者(DO, data owner)需要重新加密敏感数据，当 DO 的计算能力有限时很有可能成为系统瓶颈。3）用户从 CSP 得到密文后，需要用 ABE 解密，但是由于 ABE 算法的效率较低[7]，当用户设备的计算能力有限时(如手机等移动设备)，解密需要很长时间，用户体验较差。因此使用 ABE 实现密文访问控制面临着一个严峻的问题：如何在保证数据机密性的同时，又提供高效、细粒度、动态、可扩展的访问控制。

针对上述问题，本文提出了一种基于属性的安全增强云存储访问控制方案，通过共用属性集，将 ABE 体制与 XACML[8]框架有机结合，使得该方案不仅能保证用户数据和隐私的安全，而且能提供高效、细粒度、动态、可扩展的访问控制。充分考虑到基于属性的加密体制的效率较低，因此，云环境下大规模数据的机密性采用对称密码体制实现，基于属性的加密体制仅用于保护数据量较小的对称密钥。

2　预备知识

2.1　ABE

在 ABE 机制中，密文以及用户的解密密钥与属性集或属性策略相关联，一个用户的密钥跟密文匹配时该用户才能解密。根据属性策略与密文相关还是与密钥相关，ABE 机制分为密钥策略的基于属性的加密技术(KP-ABE, key-policy attribute-based encryption)[9]和密文策略的基于属性的加密技术(CP-ABE, ciphertext-policy attribute-based encryption)[5]。由于 CP-ABE 比 KP-ABE 更适合云存储的访问控制[5,10]，当前密文访问控制大多数是基于 CP-ABE[10~13]，本文也采用 CP-ABE 保证敏感数据的机密性。

CP-ABE 中密文与带属性的访问控制策略相关，解密密钥与用户的属性集相关，只有解密密钥关联的属性集满足密文关联的带属性的访问控制策略时才能解密。如图 1 所示。

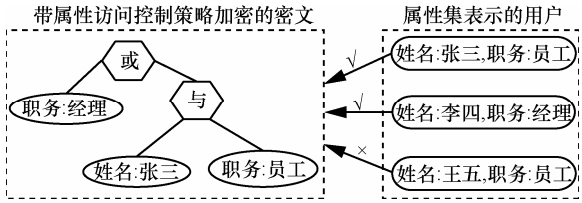


图 1　CP-ABE 访问控制结构示意

CP-ABE 算法主要包括以下 4 部分。

1) *Setup*(λ ,*U*)：利用安全参数 λ 和属性描述 *U* 作为输入生成主密钥 *MK* 和公开参数 *PK*。

2) K_A = *KeyGen*(*MK*,*A*)：使用 *MK* 和用户属性集 *A* 生成用户的私钥 K_A。

3) CT_T = *Encrypt*(*PK*,*M*,*T*)：使用 *PK*、访问控制策略结构 *T* 将明文数据 *M* 加密为密文 CT_T。

4) *M* = *Decrypt*(CT_T, K_A)：使用私钥 K_A 解密密文 CT_T 得到明文 *M*。只有 *A* 满足 *T* 的条件下，*Decrypt*( )操作才能成功。

2.2　XACML

XACML 是在 2003 年 2 月由 OASIS（organization for the advancement of structured information standards）制定的一种基于 XML (extensible markup language)用于决定请求/响应的通用访问控制策略描述语言和执行授权策略的框架。作为一种基于属性的访问控制策略和请求描述语言，与其他语言相比，XACML 作为一种功能强大的标准语言因其通用性、可扩展性而被广泛使用[14]，其中，比较著名的有 SunXACML[15]、NDG_XACML[16]等。

如图 2 所示，XACML 访问控制架构主要由 PEP（policy enforcement point）、上下文处理器、PDP（policy decision point）、PAP（policy administration point）和 PIP（policy information point）组成。首先 PEP 接受原始的访问请求（NAR, original access

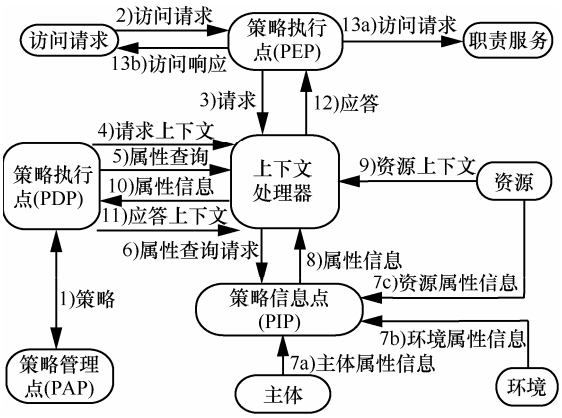


图 2　XACML 访问控制架构

request），然后由上下文处理器把 NAR 解析为 XACML 规范的请求并将其传递给 PDP，PDP 根据从 PAP 所获得的策略以及从 PIP 获取的实体属性对 XACML 规范请求进行判定，并将判定的结果发给 PEP，最后由 PEP 执行访问判定结果。

3 基于属性的安全增强云存储访问控制方案

3.1 总体方案

本文通过 XACML 和 CP-ABE 共用规格化得到通用属性集，将 ABE 体制与 XACML 框架有机结合，保证用户数据和隐私的安全，同时提供高效、细粒度、动态、可扩展的云存储访问控制。总体方案如图 3 所示，主要包括认证、授权和审计 3 个子系统。其中，认证采用令牌方式；审计提供日志记录和违规操作报警功能；授权采用基于属性的安全增强云存储访问控制，由 XACML 授权机制保证安全要求相对不高的非敏感数据的机密性，由 XACML 授权机制和 ABE 机制共同保证安全要求较高的敏感数据的机密性。

本文主要介绍总体方案中的授权部分，即基于属性的安全增强云存储访问控制。

由图 3 可知，为了提高敏感数据的机密性，该方案采用了 ABE。但 ABE 算法效率较低，ABE 的使用会导致系统整体性能下降，本文采用以下措施来缓解系统性能的下降。

1) 对 XACML 和 CP-ABE 各自的属性集进行规格化得到一个通用属性集，XACML 和 CP-ABE 共用该通用属性集，从而减少属性集的管理开销，避免异构属性的产生。

2) 用对称加密算法加密数据，用 CP-ABE 加密对称密钥，并且加解密操作都在云端执行，保证加密算法的效率。

3.2 方案实现

本文基于开源项目 NDG_XACML 和 cpabe Toolkit[17]在 OpenStack 构建的云平台上实现该方案。

3.2.1 构建与调用通用属性集

本文用属性权威（AA, attribute authority）负责实体（主体、资源和环境）属性的创建和管理。首先对 XACML 和 CP-ABE 各自的实体属性集进行规格化得到一个通用实体属性集，然后用 MySQL 数据库集群存储实体属性。其中，主体属性包括姓名、职工编号、公司、部门、职务、权利级别等，资源属性包括名称、主题、所有者、大小、创建日期等，环境属性包括日期和时间、是否有攻击、网络状况等。通过共用属性集，降低了为 XACML 和 CP-ABE 单独维护和管理属性集的复杂度，避免了异构属性的产生。

属性构建完成后，DO 从主体、资源和环境属性仓库中获取属性，生成对应资源的访问控制策略并存储到策略仓库中，如图 4(a)所示。

CSP 对 User 的访问请求进行决策时，XACML 需要从属性仓库中获取相应的属性帮助 PDP 做决

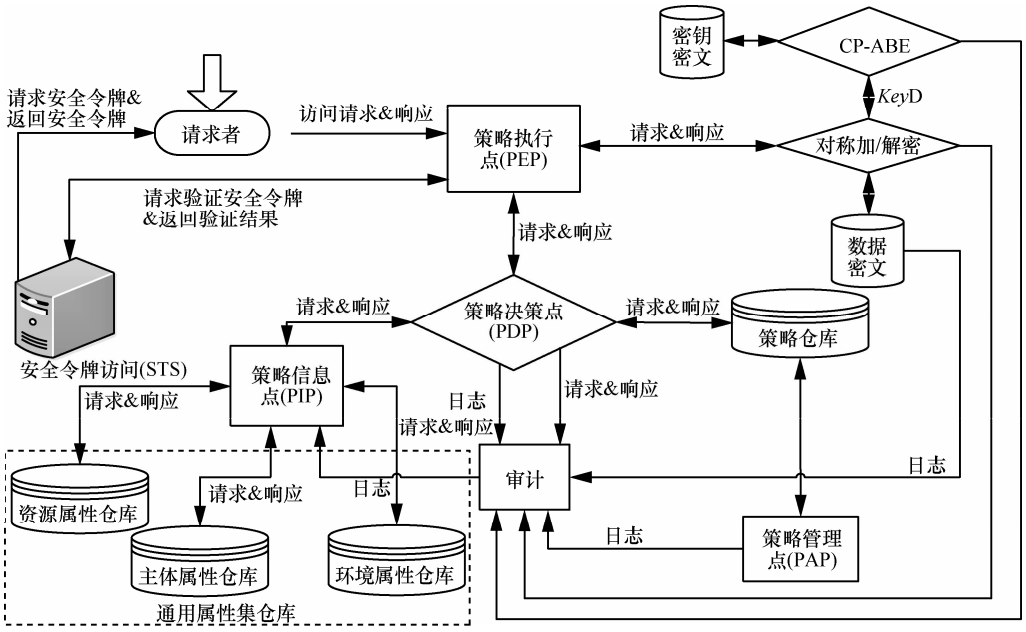
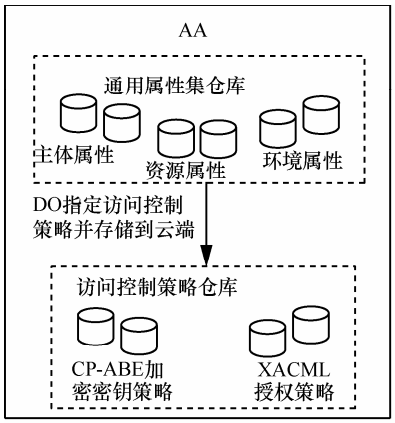


图 3 云存储访问控制方案



(a) 利用通用属性构建访问控制策略

```
public EvaluationResult frndAttribute(String contextPath,Node namespaceNode,URI
attribute Type,EvaluationCtx context,String xpathVersion){
……
while(ithasNext()){
 AttributeFinderModule module=(AttributeFinderModule)(it.ntxt());
 //查找指定类型的属性
 EvaluatonResult result=module.frndAttribute(contextPath,namespaceNode,attributeType,
                                    context,xpathVersion):
……}
……}
```

(b) XACML 框架提取通用属性模块部分代码

图 4　通用属性集调用

策，为了提高 PDP 的决策效率，本文提供了属性提取模块，部分代码如图 4(b)所示。若 PDP 决策结果是允许，CSP 从主体属性仓库中获取员工的部分属性作为用户属性集 A，并调用 $KeyGen(MK, A)$函数为该用户生成带属性的私钥 K_A。

3.2.2　DO 申请数据托管服务与 CSP 加密数据

DO 上传数据的同时需要指定访问控制策略 T 及是否需要加密存储。CSP 接到 DO 数据上传请求后，首先判断 DO 是否申请加密，如果是，则按图 5 所示云存储访问控制的加解密模块进行加密操作，否则直接存储。为简单起见，只描述单个文件 F 的加密过程。

1) 为 F 选择唯一的标号 ID_F，随机选取对称密钥 K_F 加密 F 得到 C_F。

2) 以 DO 指定的 T 为参数调用 CP-ABE 加密算法 $Encrypt(PK,M,T)$加密 K_F 得到 CT_T 。具体过程：算法根据 T 构建一棵访问结构树，T 中的属性作为叶节点，逻辑关系作为分支节点。设节点 x 的门限值为 k_x，为节点 x 生成一个(k_x-1)次随机多项式 q_x, $q_x(0)$就代表了该节点的秘密。从 Z_P 中随机选择一个数 s，令 $q_R(0)=s$(R 为根节点)。对于其他节点 x，令 $q_x(0)=q_{parent(x)}(index(x))$。假设 Y 是叶子节点的集合，则加密后的密钥密文 CT_T 如下。

$$CT_T=(T,\tilde{C}=M\cdot e(g,g)^{\alpha s},C=g^{\beta s},$$
$$\forall y\in Y:C_y^{(1)}=g^{q_y(0)},C_y^{(2)}=H(att(y))^{q_y(0)}) \quad (1)$$

说明　g 是阶为 P 的双线性群 G_0 的生成元，$parent(x)$返回节点 x 的父节点，$index(x)$返回 x 的编号，$att(y)$返回叶子节点对应的属性，α、β 是从 Z_P 中选择的随机数。

3) 将 C_F 存储到密文数据库，CT_T 存储到密钥数据库，$\{ID_F, C_F, CT_T\}$存储到索引表中。

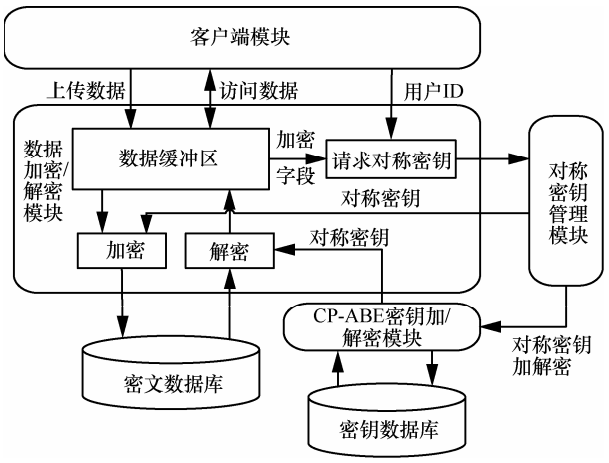


图 5　云存储访问控制加解密流程

3.2.3　CSP 构建访问控制策略

CSP 根据 DO 上传数据时指定的访问控制策略 T，利用相应的策略构建模块将 T 转换为 XACML 和 CP-ABE 能够使用的策略文件，并存储到策略仓库中。以图 1 所示的 T 为例，具体转换流程如下。

1) CSP 利用字符串转换模块将 T 转换为 $Encrypt()$ 加密函数使用的带属性访问控制策略，转换结果为：“*title=manager*” 或 (“*title=staff*” 和 “*name=zhangsan*”)。

2) CSP 利用策略构建模块将 T 转换为 XACML 形式的访问控制策略。策略构建模块部分代码如图 6 所示，生成的策略文件如图 7 所示。

```
public class SamplePolicyBuilder
{ …
//根据D0指定的访问限制条件“只有职务是经理或者职务为员工并且姓名为张三的人才能读数据F”构建XACML的访问控制策略
public static Target createRuleTarget()throws URISyntaxException{
…
//构建主体策略：职务是经理或者职务为员工并且姓名为张三的人
List subject=new ArrayList();
String subjectMatchId="utn.oasisnames.tc.xacml1.0.function string-ins-in"
URI subjectDesignatorType=new URI("http://WWW.W3.org/2001/XMLSchema#string")
URI subjectDesignatorId=new URI("un oasis names.tc.xacml.1.0:subject:subject-attnbutes")
AttributeDesignator subjectDesignator=new AttnbuteDesignator(AttnbuteDesignator.SUBJECT_TARGET,subjectDesignatorType,subjectDesignatorID,false);
StringAttribute subjectValue1=new StringAttribute("tutke=manager")
StringAttribute subjectValue2=new StringAttribute("tutke=staff and name=zhangsan")
subject.add(createTargetMatch(TargetMatch.SUBJECT,subjectMatchId,subjectDesignator,subjectValue1))
subject.add(createTargetMatch(TargetMatch.SUBJECT,subjectMatchId,subjectDesignator,subjectValue2))
//构建资源策略：文件名为F的数据
…
AnyURIAttribute resowceValue=new AnyURIAttribute(new URI("F"))
resource.add(createTargetMatch(TargetMatch.RESOURCE,resourceMatchId,resourceDesignator,resourceValue));
//构建操作策略：读权限
…
StringAttribute action Value=new StringAttribute("read")
action.add(createTargetMatch(TargeMatch.ACTION,actionMatchId,actionDesignator,actionValue));
…
return new Target(subjects,resources,actions);
}
…
}
```

图 6　XACML 策略构建模块部分代码

```
<Policy PolicyId="TimeRangePolicy"RuleCombiningAlgId="ordered-permit-overrides">
 <Description>只有职务是经理或者职务为员工并且姓名为张三的人才能读数据F</Description>
  <Rule RuleId="OnlyThisPeoplesAlways"Effect="Permit">
   <Target>
    <Subjects>
     <Subject>
      <SubjectMatchMatchId="urn:oasis:names.tc.xacml:1.0:function:string-is-in">
     <AttributeValue DataType="http://WWW.W3.org/2001/XMLSchema#string">title=manager</Attribute Value>
    <AttributeValue DataType="http://WWW.W3.org/2001/XMLSchema#string">title=staff and name=zhangsan</Attribute Value>
   <SubjectAttributeDesignator Data Type="… #string"AttnbuteId="urn:oasis:names:tc:xacml:1.0;subject:subject-attnibutes"/>
  </SubjectMatch>
 </Subject>
</Subjects>
<Resources>
  <ResourceMatch MatchId="um:oasis:names.tc:xacml:1.0function:string-is-in">
   <Attribute Value DataType="http://WWW.W3.org/2001/XML Schema#string">F<Attnbute Value>
    <ResourceAttributeDesignator DataType="… #string" AttnbuteId="um:oasis:names.tc:xacml:1.0:resoutce:attnbutes"/>
     </ResourceMatch>
      </Resources>
        <Actions>
       <Action>
      <ActionMatch MatchId="urn:oasis:names.tc.xacml:1.0:function:string-equal">
     <Attribute Value DataType−"http://WWW.W3.org/2001/XMLSchema#string">read</Attribute Value>
    <Attribute Value Designator DataType="… #string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
   </ActionMatch>
  </Action>
  </Actions>
 </Target>
 </Rule>
</Policy>
```

图 7　生成的 XACML 策略文件

3.2.4　User 申请访问数据

User 访问数据的流程如图 8 所示，具体步骤如下。

1) 令牌服务器验证 User 的合法性，若验证失败则拒绝请求，否则继续执行。

2) XACML 访问控制流程对用户的数据访问请求进行决策，具体流程如图 8 虚线框所示。其中，PDP 是访问控制的核心，它需要根据访问请求从

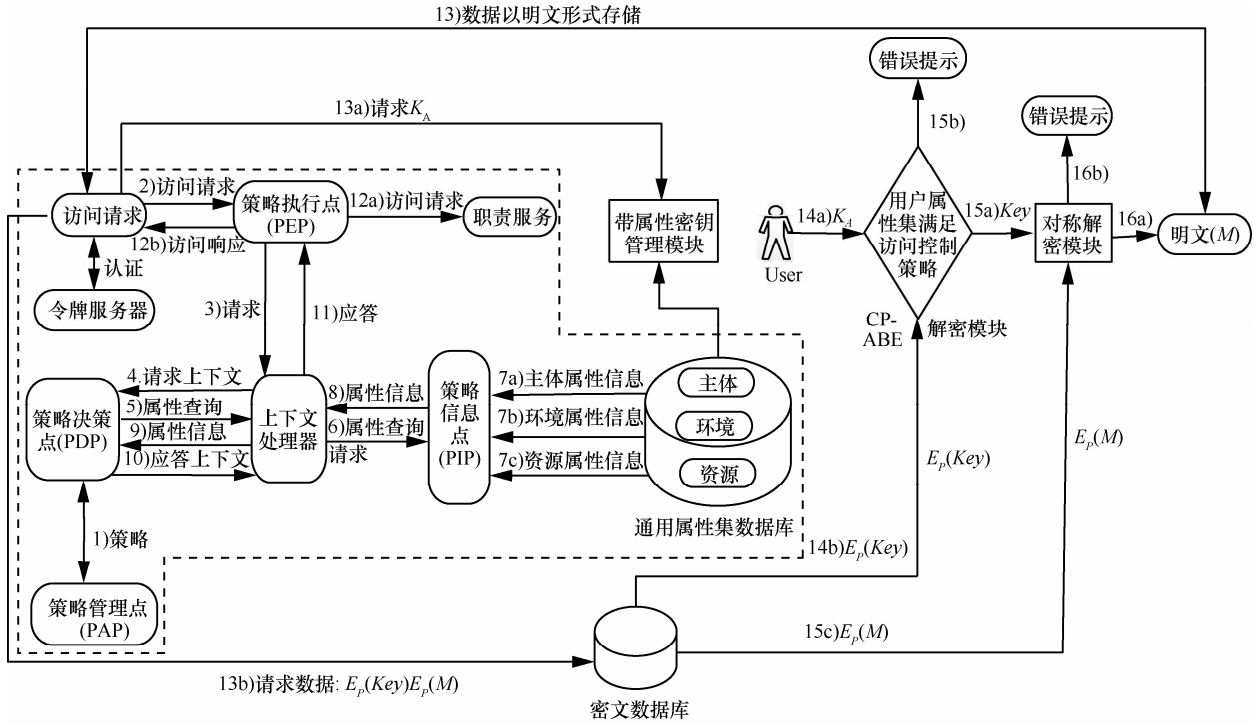


图 8　User 访问流程

PIP 提取相关的实体属性，并从 PAP 提取对应的访问控制策略文件，然后将实体属性与访问控制策略文件中指定的实体属性进行比较，若不匹配则拒绝访问，否则继续执行。

3) 判断数据是否以明文形式存储，若是则把明文数据发给 User，否则继续执行。

4) 根据 User 的属性集生成对应的私钥；设 User 的属性集是 A（CSP 主体属性仓库中对应 User 属性集的子集），先生成随机数 $\gamma\in Z_P$，然后为每个属性 $j\in A$ 选择随机数 $\gamma_j\in Z_P$，最后生成的私钥为

$$K_A=(D=g^{(\alpha+\gamma)/\beta},\ j\in A: D_j=g^{\gamma}H(j)^{\gamma_j},D_j^{'}=g^{\gamma}) \tag{2}$$

5) 以 K_A 为参数调用 $Decrypt(CT_T, K_A)$解密密钥密文，解密成功后用得到的对称密钥解密数据密文，并将得到的明文发给 User；否则提示无权访问。具体解密过程如下。

定义递归运算 $DecryptNode(CT_T, K_A, x)$，令 $i=att(y)$，对于每个叶子节点 x 计算：

$$DecryptNode(CT_T,K_A,x)=\begin{cases}\dfrac{e(D_i^{(1)},C_x^{(1)})}{e(D_i^{(2)},C_x^{(2)})}=e(g,g)^{\gamma q_x^{(0)}}, & i\in A\\ \perp, & i\notin A\end{cases} \tag{3}$$

对于每个非叶子节点 x，至少利用 k_x 个 $e(g,g)^{\gamma q_x^{(0)}}$ 作为拉格朗日插值定理的插值节点，计算得到 $e(g,g)^{\gamma q_x^{(0)}}$，这里 $e(g,g)^{\gamma q_x^{(0)}}$ 是从节点的孩子节点 $\{z_j\}$ 计算得到的。令 $B=e(g,g)^{\gamma q_R^{(0)}}=e(g,g)^{\gamma s}$，则明文 $M=\tilde{C}/(e(C,D)/B)$。

3.2.5　撤销权限和更新策略

当 DO 要撤销 User a 对文件 F 的访问权限时，CSP 只需要更新 F 文件对应的 XACML 访问控制策略，添加一条拒绝访问规则，拒绝访问规则示例如图 9 所示。DO 还可修改 CP-ABE 的 T 并要求 CSP 重新生成对称密钥并重新加密数据，并用修改后的 T 加密密钥，由于数据和密钥加密都由 CSP 执行，效率较高。

4　分析与实验

4.1　安全性分析

本节对基于属性的安全增强云存储访问控制方案的安全性进行分析。具体内容分为 3 部分：对数据机密性分析、对访问控制策略的机密性分析和对操作机密性的分析。

4.1.1　数据机密性分析

数据机密性包括数据密文的机密性和密钥密文的机密性。本文采用对称加密算法（AES-128）

```
<Rule RuleId="DenyAllAction"Effect="Deny">
 <Target>
  <Subjects>
   <Subject>
    <SubjectMatch MatchId="urn:oasis:names.tc.xacml:1.0:function:string-equal">
    <Attribute Value DataType="http://WWW.W3.org/2001/XMLSchema#string">a</Attribute Value>
    <SubjectAttributeDesignator DataType="…#string"|AttributeId="urn:oasis:names.tc.xacml:1.0:subject:subject-name"/>
   </SubjectMatch>
  </Subject>
 </Subjects>
<Resources>
  <ResourceMatch MatchId="urn:oasis:names:tc.xacml:1.0:function:string-equal">
  <AttributeValue DataType="http://WWW.W3.org/2001/XML Schema#string">F</Attribute Value>
  <ResourceAttributeDesigrator DataType="…#string"AttnbuteId="urn:oasis:names:tc:xacml:1.0:resource.name"/>
  </ResourceMatch>
 </Resources>
 </Target>
</Rule>
```

图 9 拒绝访问规则示例

对用户敏感数据加密，从而保证数据的机密性；用 CP-ABE 加密对称密钥，文献[5]已经证明 CP-ABE 是安全的，从而保证密钥的机密性。并用 XACML 细粒度的授权架构保证只有合法用户才能访问密钥密文、数据密文以及非敏感数据。

4.1.2 访问控制策略机密性分析

在该方案中，访问控制策略由 DO 制定，并且策略中包含的主体、客体和环境的属性只有 DO 知道，其他人无法获取这些属性信息，访问控制策略的机密性得到了保障。此外，策略中可以包含多个属性，尤其是实时的环境属性信息，使得该方案与基于角色的访问控制等传统访问控制相比具有更细的访问控制粒度，提高了系统的安全性。

4.1.3 操作机密性分析

本文用到的对称密钥都是随机生成并用 CP-ABE 加密后存储在云端，并没有发给用户，不会出现由于 User 的失误导致密钥泄露的情况。用户申请访问密文数据不但要通过 XACML 授权，而且用户的带属性密钥必须能解密密钥密文。此外，本文还用 DH 密钥交换和公共密钥加密等加密技术保护 CSP 和 DO、CSP 和 User 以及 DO 和 User 间的通信，防止人为的中间攻击，保证数据传输过程的安全性。

4.2 性能分析

实验环境：Inter(R) Core(TM) 3.10GHz CPU、4GB DDR3 内存，操作系统为 Ubuntu12.04 64 bit，实验平台是 OpenStack 构建的云环境，Swift 作为云存储服务。实验代码基于开源项目 NDG_XACML、cpabe-0.11 库编写，对称加密用基于 openssl-1.0.1b 库的 128 bit AES 加密算法，需要保护的用户数据是大小为 150 MB 的公司客户花名册 customers。

实验涉及的评估引擎包括 Sun XACML、Enterprise XACML[18]、XEngine[19]和本方案。为了检验本方案在多规则、多策略组合不同场景下的实际性能，实验测试采用 XACML 官方测试分组[20]，并对其进行修改和扩展，得到 2 组测试样本 1 和 2。其中，样本 1 由 2 000 条规则构成单一策略，样本 2 由 500 条策略组成，平均每条策略包含 4 条规则。根据策略中包含的属性信息，分别随机生成 200 次不同的访问请求（请求中已包含完整属性信息），每个请求发送 50 次，各评估引擎评估访问请求并计算响应时间。

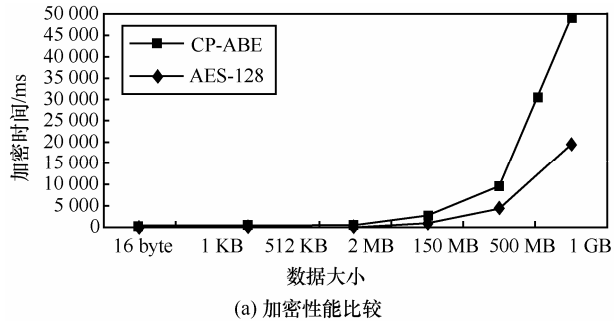
4.2.1 CP-ABE 与 AES-128 性能分析

图 10 展示了随着数据大小的增加，使用 CP-ABE 和 AES 算法加解密数据的时间代价对比。可以看出，AES 算法的加解密性能明显优于 CP-ABE 算法，而且随着数据大小的增加，这种优势越来越明显。因此，在云存储中本方案采用对称密码体制保证海量数据的机密性，采用基于属性的加密体制保护数据量较小的对称密钥的机密性。

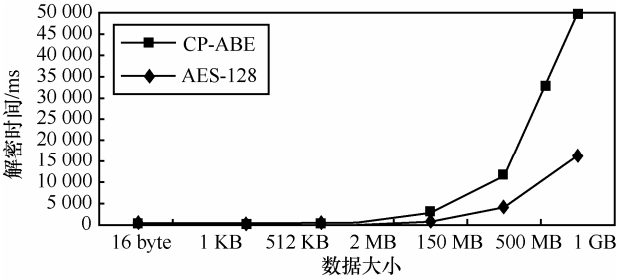
4.2.2 非敏感数据访问控制性能分析

公司客户花名册 customers 以明文的形式存储在云环境中，用户只要通过 XACML 授权即可访问该数据。图 11 展示了各评估引擎对测试用例访问请求做出响应所花费的平均时间。

对于相同样本由于各评估引擎对 XACML 策略进行评估时采用的优化机制不同，所以评估效率不完全相同；其中，Sun 公司的 XACML 评估引擎没有采用任何优化机制，Enterprise XACML 采用了索引机制，XEngine 采用数值化和标准化机制，本方案采用了多级缓存机制和两级索引机制。



(a) 加密性能比较



(b) 解密性能比较

图 10　CP-ABE 和 AES 加解密性能比较

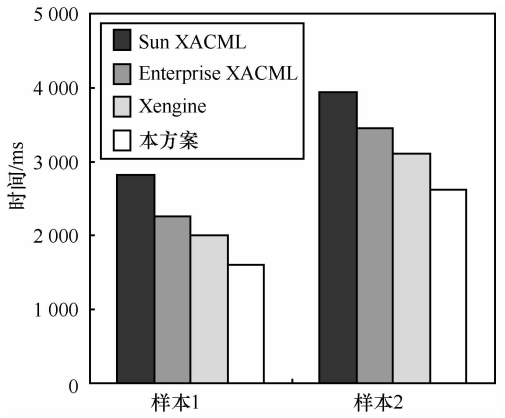


图 11　明文访问控制性能比较

虽然 2 组样本的规则总体规模相等，但各种引擎对多规则组合样本的评估速度普遍快于多策略组合样本，这主要是由于后者的策略结构比前者的策略结构复杂，引擎在策略解析匹配时需要更多的处理时间。

可以看出，无论在多规则组合场景下，还是多策略组合场景下，本方案的评估性能都优于其他同类引擎。

4.2.3　敏感数据访问控制性能分析

公司客户花名册 customers 以密文的形式存储在云环境中，由于 4 种访问控制评估引擎中，只有本方案提供了密文存储机制，所以只对本方案进行了密文访问控制性能的分析测试，结果如图 12 所示。

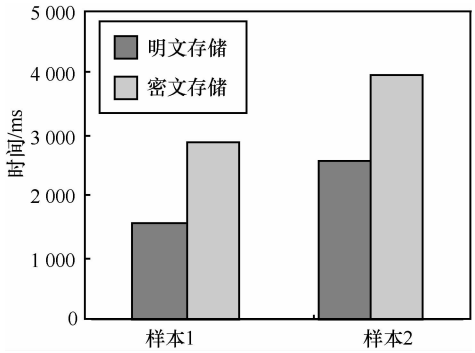


图 12　明文和密文访问控制性能比较

由图 12 可知，本方案对密文访问请求做出响应所花费的时间要多于对明文访问请求做出响应所花费的时间。这是由于密文请求响应过程比明文请求响应过程复杂，用户要先通过 XACML 授权并得到密钥密文和数据密文，然后用 CP-ABE 解密密钥密文得到对称密钥，最后用对称密钥解密数据密文得到明文。

由图 12 可知，引入密文机制在提供数据安全性的同时，必然会带来系统整体性能的下降；但是本方案采用加密效率相对较高的对称加密机制来保护数据的机密性，并且加解密操作都由计算效率很高的 CSP 执行，所以系统性能下降幅度并不大。

实验结果表明，本方案不仅能保证用户数据和隐私的机密性，而且性能优于其他同类系统。此外，由于本方案通过共用属性集将 CP-ABE 与 XACML 授权框架有机结合，因而既有 XACML 作为授权方式具有的细粒度、实时、可扩展、动态的优点[2]，也有用 CP-ABE 保密敏感数据机密性的优点[5,10]。

5　结束语

近年来，云存储泄密事件层出不穷，本文提出了一种基于属性的安全增强云存储访问控制方案来保证用户数据和隐私的机密性。通过共用属性集，将 CP-ABE 与 XACML 框架有机结合，使得该方案既有 XACML 作为授权方式具有的细粒度、实时、可扩展、动态的优点，又有 CP-ABE 保密敏感数据机密性的优点。充分考虑到 CP-ABE 的效率较低，因此，云存储中海量数据的机密性采用对称密码体制实现，CP-ABE 仅用于保护数据量较小的对称密钥。实验分析表明，该方案不仅能保证用户数据和隐私的机密性，而且性能优于其他同类系统。

参考文献：

[1] CHRISTIAN C, IDIT K, ALEXANDER S T. Trusting the cloud[J]. ACM SIGACT News Archive, 2009, 40(2):81-86.

[2] ERIC Y，JIN T, HAMILTON B A. Attributed based access control (ABAC) for web services[A]. Proceedings of the IEEE International Conference on Web Services[C]. Orlando Florida,USA, 2005.7.

[3] SAHAI A, WATERS B. Fuzzy identity based encryption[J]. LNCS, 2005, 3494(1):457-473.

[4] YU S C, WANG C, REN K. Achieving secure, scalable, and fine-grained data access control in cloud computing[A]. Proceedings of the 2010 IEEE INFOCOM[C]. Piscataway, NJ, USA, 2010. 534-542.

[5] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[A]. Proceedings of 2007 IEEE Symposium on Security and Privacy[C]. Berkeley, USA, 2007. 321-334.

[6] MALEK B, MIRI A. Combining attribute-based and access systems[A]. Proceedings of 12th IEEE Int'l Conf on Computational Science and Engineering[C]. Vancouver, Canada, 2009. 305-312.

[7] 洪澄, 张敏, 冯登国. 面向云存储的高效动态密文访问控制[J].通信学报, 2011, 32(7):90-98.
HONG C, ZHANG M, FENG D G. Achieving effitient dynamic cryptographic access control in cloud storage[J]. Joumal on Communications, 2011, 32(7):90-98.

[8] OASIS eXtensible access control markup language (XACML)[EB/OL]. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev-xacml.

[9] GOYAL V, PANDEY O, SAHAI A. Attibute-based encryption for fine-grained access control of encrypted data[A]. Proceedings of the 13th ACM Conference on Computer and Communications Security[C]. New York, USA, 2006. 89-98.

[10] WAN Z G, LIU J, DENG R H. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2):743-754.

[11] 洪澄, 张敏, 冯登国. AB-ACCS:一种云存储密文访问控制方法[A]. 计算机研究与发展, 2010, 47:259-265.
HONG C, ZHANG M, FENG D G. AB-ACCS: a cryptographic access control scheme for cloud storage[J]. Journal of Computer Research and Development, 2010, 47:259-265.

[12] HUR J B, NOH D K. Attribute-based access control with efficient revocation in data outsourcing systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(7):1214-1221.

[13] WANG G J, LIU Q, WU J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services[A]. Proceedings of the 17th ACM Conference on Computer and Communications security[C]. New York, USA, 2010. 735-737.

[14] Brief introduction to XACML[EB/OL].https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html.

[15] XACML open source project SUNXACML[EB/OL]. http://sunxacml.sourceforge.net.

[16] XACML open source project NDG-XACML[EB/OL]. http://python-hosted.org/ndg-xacml/ndg-module.html.

[17] Advanced crypto software collection[EB/OL]. http://acsc.cs.utexas.edu/cpabe.

[18] XACML open source project enterprise XACML[EB/OL]. http://code.google.com/p/enterprise-java-xacml/.

[19] XACML open source project XEngine[EB/OL]. http://xacmlpdp.sourceforge.net/.

[20] XACML 2.0 conformance tests[EB/OL]. http://www.oasis-open.org/committees/ download.php/14846/xacml2.0-ct-v.0.4.zip.

作者简介：

牛德华（1989-），男，山西吕梁人，西安电子科技大学硕士生，主要研究方向为云存储访问控制、云存储安全、移动终端安全等。

马建峰（1963-），男，陕西西安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为密码学、计算机网络与信息安全。

马卓（1980-），男，陕西延安人，博士，西安电子科技大学副教授，主要研究方向为无线网络安全、安全协议的形式化分析与设计理论与方法、可信计算理论与技术等。

李辰楠（1988-），男，陕西西安人，西安电子科技大学硕士生，主要研究方向为网络安全、Web 访问控制、云计算安全存储等。

王蕾（1988-），女，陕西西安人，西安电子科技大学硕士生，主要研究方向为网络安全、密钥管理、云计算安全存储等。