# Security Application Model Based on Identification Management for Cloud Computing

Yazhe Wang, Shunan Ma, Yu Wang

State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing

***Abstract*—Aiming at the challenge of entity identity management technology in cloud computing environments, we propose a security application model based on identification management, and present a security access method based on privacy protection and audit about the entity uniform identity management. In this paper, the multi-entity identification encoded by using the unified structure is introduced, and the method to achieve the secure access of RP by the local identification based on UUID and privacy protection is proposed. We also describe the audit of multi-entity identification association mapping, thus these methods realize the identity management of entities in cloud computing environments. The identification management method we proposed has the advantages of strong universality, convenient storage and inquiry. The access method has higher security.**

***Keywords*—Cloud Computing, Identity Management, Safety Access, Audit, Consistent Hash Ring**

# 云计算环境下基于身份标识管理的安全应用模型

王雅哲　马书南　王瑜

中国科学院信息工程研究所信息安全国家重点实验室，北京，中国

**摘　要**　针对云计算环境所面临的实体身份管理技术的挑战，本文提出了基于身份标识管理的安全应用模型，并针对实体标识提出了其分布式存储、隐私保护和安全审计的方法。本文采用统一标识结构对多元实体编码，通过基于 UUID 局部标识及隐私保护的方法实现与身份依赖方 RP 的安全访问，并描述了多元实体标识关联映射的审计，从而实现云计算环境下多元实体标识身份的管理。本文所提出的实体身份标识方法具有较强的通用性，且方便存储查询；访问方法具有较高的安全性。

**关键词**　云计算，身份管理，安全访问，审计，一致性哈希环

## 1．引言

近年来，云计算以其超大规模、虚拟化、高可靠性的独特优势引发计算机网络变革，同时也带来安全问题。云计算的运作方式是将各种软件资源、存储资源和计算资源等资源联合起来形成规模巨大的虚拟共享资源池，其目的是实现资源共享。对于每个用户来说，在对方身份未知的情况下要求进行协同，存在很大的风险，因此云计算中多元实体身份标识管理已成为亟待解决的问题[1]。同时，身份标识管理也是安全系统中的第一道屏障，在云计算的未来发展中有着举足轻重的作用。

实体的身份标识是确认实体身份信息的基础，采用统一身份标识技术可实现实体在不同网络、不同业务间建立可靠的身份认证以共享服务，避免由异构认证机制带来的复杂性，提高用户对网络和业务的使用效率、提高身份信息的安全性具有重要的意义。

在云计算场景中，多元实体的标识作为数字化的实体形态，其标识可分为主体标识和客体标识。对于身份标识，目前已经展开了相关的研究工作。例如，基于 X.509 目录服务的 PKI、基于 Kerberos[2,3]架构的 CAS[2]等信息化身份管理系统，实现传统意义上的实体身份标识认证管理。OpenID[4,5]是由 LiveJournal 组织提出的一个去中心化的网上身份标识认证系统，它通过统一资源标志符（URI）对网络用户进行唯一标识和身份验证，它的核心思想是将身份管理、验证鉴别功能和具体应用业务剥离并托管给专业的身份服务提供商，从而实现用户可选择的身份登录方案。微软从 Vista 操作系统开始，引入了 CardSpace[6-8]标识元系统，为不同的数字身份系统提供一个统一的抽象表示层，从而在用户体验层面屏蔽了不同的安全令牌和标识信息结

国家自然科学基金项目（61202476），中国科学院信息工程研究所所内创新项目资助（No.Y3Z0071202）

构。上述研究有其自身的特点，但对云计算场景缺乏技术支撑，没有考虑物理设备实体标识、物化对象实体标识、虚拟对象实体标识等多元实体身份的管理架构，在标识管理涉及的信息存储、信息检索、信息使用模式等方面缺乏解决方法。

同时在新一代网络场景中，企业和组织一般使用不同云资源服务商所提供的不同标识模式下的多元化资源，使多元化资源缺少统一标识，由此对资源的管理和访问控制产生很多问题，即云资源统一标识问题。例如，基于 DOI 的数字对象标识、基于 UUDI[9]的企业资源标识等资源标识储存和查询方案在虚拟资源管理规模、虚拟资源的动态性和可扩展性、虚拟资源业务模式演化等方面已不能满足信息安全对虚拟资源管理的要求。基于传统的统一资源定位符 URL(Universal Resource Locator)进行资源定位，已不能很好的解决虚拟空间环境对资源多样式、自主性、动态可扩展性带来的访问问题。Ex Libris 公司和比利时 Ghent 大学提出了一种基于 Web 学术信息环境实现开放链接的框架——OpenURL[10,11]，它通过 OpenURL 的相关规则对网络数字对象进行唯一标识和管理，其核心思想是提供传输书目元数据的语法和信息服务间的对象识别符，从而实现用户在异质系统之间数字对象的互操作。联机计算机图书馆中心 OCLC 提出了一个永久性名称解析系统——PURL(Persistent URL)[12]，其设计思想是通过名称而不是 URL 来标识网络资源。它的具体实施方案是利用重定向的功能保持迁移中网络资源的标识，即通过用 PURL 注册用户维护和修改网络资源的名称与 URL 的对应关系来实现标识的唯一性。上述研究工作虽然在资源标识方面提出了相应地解决方法，但在解决云资源的动态性、虚拟化、资源可管理性等特点上仍存在欠缺。

针对上述问题，本文提出了一个基于身份标识管理的安全应用模型。其中包括：多元化实体的统一身份标识方法，并针对实体标识实现了其分布式存储、隐私保护和安全审计的方法。模型如图 1 所示。

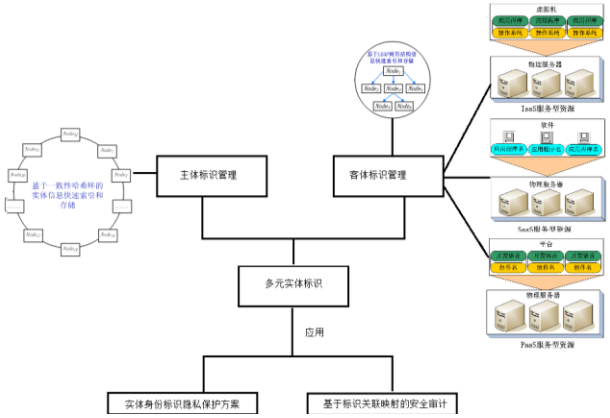


图 1 基于身份标识管理的安全应用模型

## 2. 多元实体标识管理

为了应对新一代网络场景的安全挑战，多元实体的身份管理应提供一种多元实体标识的统一描述规则。依据云计算提供不同层次的服务模式，该方法涉及的主体包括：用户实体、终端设备实体、虚拟对象实体和物化实体，涉及的客体包括：IaaS 型资源实体、PaaS 型资源实体和 SaaS 型资源实体。

### 2.1 实体身份标识的统一描述

基于统一资源标识符 URI 的分层结构和标识唯一特性，对用户实体、终端设备实体、虚拟对象实体等身份标识实体类型进行编码，具体方案如下：URI 标识体系的结构主要由管理域标识、类型标识、实体标识等组成，管理域标识表示为 Domain ID，类型标识表示为 Class ID，实体标识表示为 Entity ID，完整的标识结构为 URI://Domain ID/Class ID/Entity ID，其中 Domain ID 可以根据管理域的内部组织结构进行细化分层，例如 Domain A/Organization B/Group C/；类型标识在同一父类型下可以划分若干子类型，例如 Class A/Subclass B/；实体标识 Entity ID 编码结构在不与 Domain ID 所蕴含语义冲突的前提下，可采用自定义的编码规则进行描述，例如字符串和数字串等，如图 2 所示。

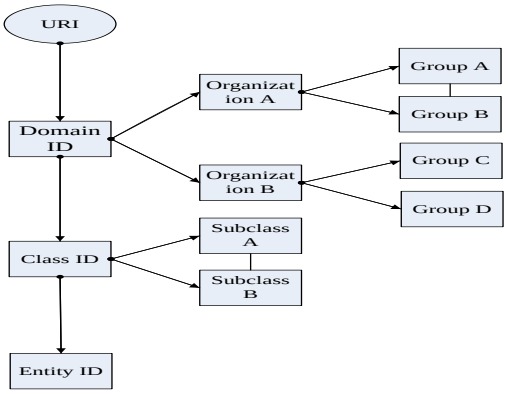


图2 基于URI的身份标识的结构示意图

### 2.2 资源标识的统一描述

基于统一资源名 URN 的分层结构和标识的唯一性，对新一代网络场景下提供服务的资源实体进行编码，即对服务类型对应的资源进行统一标识，具体编码方案如下。该编码体系的完整结构为 URN：Service ID：Resource ID，其中 Service ID 代表 IaaS、PaaS 和 SaaS，Resource ID 代表云计算服务模式对应的统一资源描述。

①基于 URN 的 IaaS 型资源标识

URN：IaaS：Domain ID-VM ，其中 IaaS 代表提供的

是基础设施即服务类资源；Domain ID 代表服务提供商；VM 代表着云计算中心提供给用户的虚拟机资源，该虚拟机资源的元数据集由四部分组成，分别为 ID、计算资源 CompRes、存储资源 StoRes 和网络带宽 NetRes。

②基于 URN 的 SaaS 型资源标识

URN：SaaS：Domain ID-Software，其中 SaaS 代表提供的是软件即服务类资源；Domain ID 代表软件服务提供商；Software 代表着云计算中心提供给用户的软件类资源，主要分为应用程序类和 API 接口类。应用程序类软件资源的元数据集由五部分组成，分别为 ID，应用程序名 AppName、应用程序开发商 AppDev、运行环境 RunEnv(如 windows、Linux 等操作系统)、应用程序语言 AppLan(如中文、英文等可设定的语言)；API 接口类软件资源的元数据集由四部分组成，分别为 ID，API 接口名 ApiName、输入值类型 ApiInput（如 String、Boolean 等类型值）、输出值类型 ApiOutput (如 String、Boolean 等类型值)。例如应用程序类 URN 标识描述为：URN：SaaS：Domain ID-Software.ID||AppName||AppDev||RunEnv||AppLan；API 接口类 URN 标识描述为：URN：SaaS：Domain ID-Software.ID||ApiName||ApiInput||ApiOutput；

③基于 URN 的 PaaS 型资源标识

URN：PaaS：Domain ID-Platform，其中 PaaS 代表提供的是平台即服务类资源；Domain ID 代表平台服务提供商；Platform 代表着云计算中心提供给用户的平台服务类资源，主要是指软件研发平台。该平台服务类资源的元数据集由五部分组成，分别为 ID，开发语言 DevLan、组件名 ComName、组件开发商 ComDev、组件语言 ComLan。例如 URN：PaaS：  .

实体向标识服务提供者（IdP）提交标识申请，由 IdP 验证实体身份标识实体在各自管理体系下的身份有效性之后，完成实体标识注册。

## 3. 分布式实体标识信息存储

本节对分布式存储领域新技术进行研究，并结合云计算的数据动态管理的应用场景，提出两种面向不同多元实体标识的存储方法，以满足海量的实体标识信息存储方便，查询效率高的要求。

### 3.1 基于一致性哈希环 CHR 身份标识的存储

多元实体的身份凭证信息是海量的，同时需要配合业务系统满足对身份信息的快速检索，基于这种需求，本节详细阐述基于一致性哈希环 CHR 建立针对身份标识信息分布式存储与索引。具体过程如下：

①预先设定 $0 \sim 2^{32}$ 为一致性哈希环 CHR 的数值区间，分布式存储节点 K 的标识表示为 $Identification_K$，基于哈希算法 $Hash_M$ $Hash_M$ (具体实现可采用 SHA-2 系列算法、SHA-3 算法等)计算 $Hash_M(Identification_K)$，从而确定节点 K 在 CHR 中的位置，如图 3 所示；

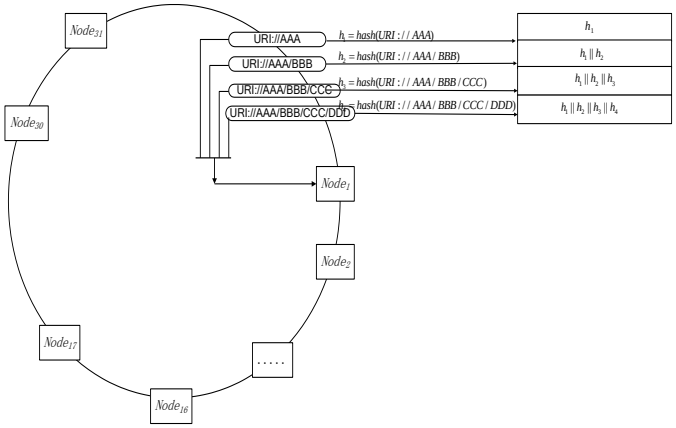


图3 基于CHR的多元实体身份标识的存储索引结构

②设定实体 E 的 URI 表示为 URI://AAA/BBB/CCC/DDD, 计算 $h_1 = hash_m(AAA)$，$h_2 = hash_m(AAA/BBB)$，$h_3 = hash_m(AAA/BBB/CCC)$，$h_4 = hash_m(AAA/BBB/CCC/DDD)$，$h_E = h_1 \| h_2 \| h_3 \| h_4$；通过这个结构计算出的哈希值来确定实体信息在一致性哈希环 CHR 中存储位置，$h_1$ 值首先确定存储节点，$h_2$，$h_3$，$h_4$ 值确定在该节点的存储位置，实现一种目录结构树的存储；

③将数值区间划分为 32 个子区间，每一子区间对应一个节点，设哈希环 CHR 分布 32 个存储控制节点，为 CHR 建立子区间到存储节点的映射列表，形如图 4，若实体 E 的 URI 标识按照步骤②的方法计算哈希 $h_1 \in [0, 2^{27}-1]$，则实体 E 的身份信息由节点 $Node_1$ 负责存储或管理；

| 映射节点 | 映射区间 |
|---|---|
| $Node_1$ | $[0 \sim 2^{27}-1]$ |
| $Node_2$ | $[2^{27} \sim 2\times 2^{27}-1]$ |
| $Node_3$ | $[2\times 2^{27} \sim 3\times 2^{27}-1]$ |
| $Node_4$ | $[3\times 2^{27} \sim 4\times 2^{27}-1]$ |
| ..... | ..... |
| $Node_n$ | $[(n-1)\times 2^{27} \sim n\times 2^{27}-1]$ |
| ..... | ..... |
| $Node_{32}$ | $[31\times 2^{27} \sim 32\times 2^{27}-1]$ |

图4 CHR的存储索引结构数值区间映射示意图

④若 URI://AAA 对应实体 E 的相关信息存储在节点 K 上，则其他以 URI://AAA 开头的实体信息也都存储在节点 K 上（或通过节点 K 控制管理）；

⑤节点 K 对应 $h_E$ 为 $h_1$ 开头的所有实体信息，若经过统计 $h_E$ 的结构相对简单，即下层路径分支少于某一阈值的结构，则节点 K 对应的物理主机负责实现实体身份信息的本地存储；若经过统计 $h_E$ 的结构相对复杂，即下层路径分支较多，则节点 K 对应的物理主机可以也采用一致性哈希环对下一级路径不同的实体信息进行分布式定位存储。

⑥出于信息冗余考虑，若实体 E 的信息定位在节点 $Node_i$，则相关信息可以在节点 $Node_{i+1}$ 和 $Node_{i+2}$ 上进行备份存储，保证每个实体信息在 CHR 有至少 2 个备份。3.2 基于轻量级目录访问协议 LDAP 云资源标识的存储

利用 LDAP 活动目录的树形结构建立云资源统一 URN 标识的分布式存储与索引，以应对云资源标识的海量性、可扩展性，管理上的方便性、可控性，同时满足业务系统对云资源标识的快速检索的要求。具体优化过程如下：

①分析云资源标识的注册流程中相关数据信息，设定纳入 LADP 中的数据集，主要包括 IaaS 型资源数据集、SaaS 型资源数据集和 PaaS 型资源数据集。IaaS 型资源数据集包含服务提供商信息、IaaS-R 资源的信息（如计算资源量、存储资源量、网络资源量、资源访问 URL 和资源访问的安全策略等）；SaaS 型资源（SaaS-R）数据集包含服务提供商信息、SaaS-R 资源的信息分为两类，一类信息包含应用程序名、应用程序开发商、应用程序语言、访问安全策略等，另一类信息包含 API 接口名、输入输出类型、访问安全策略等；PaaS 型资源数据集包含服务提供商信息、PaaS-R 资源的信息（如组件名称、组件语言、开发语言、资源访问 URL 和资源的访问安全策略等）。

②借助①设定的相关数据集，设计基于统一资源名 URN 标识的云资源属性和属性聚类的语法规范 schema，相应的 schema 设计简单表示如下：

IaaS-R 资源信息类定义，类名：IaaS-Res，父类：top，属性：通用名 CN、CompRes、StoRes、NetRes、URL、Policy；

SaaS-R-App 资源信息类定义，类名：SaaS-Res-App，父类：top，属性：通用名 CN、AppName、AppDev、AppLan、RunEnv、URL、Policy；

SaaS-R-Api 资源信息类定义，类名：SaaS-Res-Api，父类：top，属性：通用名 CN、ApiName、ApiInput、ApiOutput、URL、Policy；

PaaS-R 资源信息类定义，类名：IaaS-Res，父类：top，属性：通用名 CN，ComName、ComLan、ComDev、DevLan、URL、Policy；

以上的 schema 结构是可以统一表示，如：类，类名，父类，属性。

③设定基于统一资源名 URN 标识云资源的数据组织结构 DN，数据组织结构 DN 结构中，常用的属性有 DC（组织域名）、OU（组织单元）、CN（通用名）。DN 的设计按照云资源标识 URN 的分层结构，建立 LDAP 树形索引结构。包括：

(1)构造多元化资源统一资源名 URN 标识的基准 DN，该步骤是在系统初始化时进行，通过 LDAP 服务器按照固定的协议来构造。

Dn: DC=URN
objectClass: Top
objectClass: Dcobject
DC=URN

(2)构造多元化资源统一资源名 URN 标识的云服务模式组织结构，如图 5 所示。

Dn: OU=IaaS, DC=URN
objectClass: Dcobject
objectClass: Organization
OU=IaaS
DC=URN
Dn: OU=SaaS, DC=URN
objectClass: Dcobject
objectClass: Organization
OU=SaaS
DC=URN
Dn: OU=PaaS, DC=URN
objectClass: Dcobject
objectClass: Organization
OU=PaaS
DC=URN

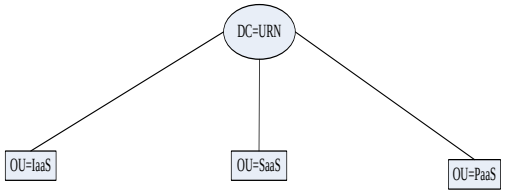


图5 云资源URN标识的LDAP基础树形结构

(3)构造 IaaS-R 资源统一资源名 URN 标识的云服务模式组织结构，如图 6 所示。

dn：CN=$VM_A$，OU=IaaS，DC=URN
objectClass：Dcobject
objectClass：Organization
objectClass：IaaS-Res
CN=$VM_A$

OU=IaaS

DC=URN

CompRes= $CompRes_1$

StoRes= $StoRes_1$

NetRes= $NetRes_1$

URL= $URL_1$

Policy= $Policy_1$

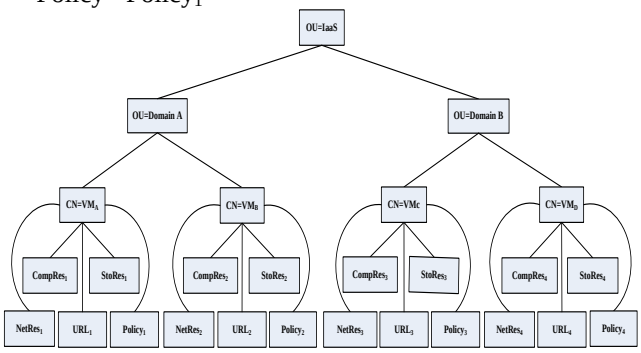


图6　IaaS-R资源URN标识的LDAP目录树形结构

## 4. 多元实体身份标识隐私保护方案

多元实体通过注册获取的 URI 标识（即实体的身份标识）具备一定程度的语义可读性。在与身份依赖方 RP 的业务交互中，除去强制审计要求之外，应提供用户针对 RP 的身份隐私保护出示方案。本节论述了基于 128bit 登录随机通用唯一识别码 UUID，拼接 DVB 同步字节反转算法处理后的 128bit 登录随机码，生成 256bit 随机登录码。再结合 SHA-256 算法处理的 256bit 实体 URI 标识哈希摘要进行相关操作实现用户的身份匿名出示。该保护方案实施期间涉及 RP、URI-IdP 和 User 三方，其中 URI-IdP 应具备颁发可信时间戳(timestamp)的能力，过程如下：

①用户登录 RP 门户，在未提供身份证明的情况下，RP 通过用户选择的方式将其跳转至 URI-IdP 门户；

②用户利用身份凭证通过 URI-IdP 的鉴别，URI-IdP 基于 UUID 通用计算方法生成用户本次登录随机的 128bit 位 UUID 标识（表示为 $UUID_{128}(User)$），然后采用 DVB 同步字节反转算法对该次登录的随机标识 $UUID_{128}(User)$进行反转生成 128bit 位的反转 UUID 标识（表示为 $UUID_{128}(User)_{反}$），并将 $UUID_{128}(User)_{反}$和 $UUID_{128}(User)$拼接组成本次登录的 256bit 位 UUID 标识（表示为 $UUID_{256}(User)= UUID_{128}(User)_{反}||UUID_{128}(User)$），再利用 SHA-256 算法生成用户 URI 标识的 256bit 位哈希摘要（表示为 $SHA\text{-}256_{256}(URI)$）；

③将 $UUID_{256}(User)$和 $SHA\text{-}256_{256}(URI)$进行异或运算 $UUID_{256}(User)\oplus SHA\text{-}256_{256}(URI)$生产用户临时身份凭证，拼接 URI 标识服务提供者 URI-IdP 生成时间戳 timestamp，用 URI-IdP 的私钥 Pri 对其进行签名，表示为 $Sig_{pri}[UUID_{256}(User)\oplus SHA\text{-}256_{256}(URI)||timestamp]$；

④URI-IdP 门户将用户重定向到 RP 方，同时将 $Sig_{pri}[UUID_{256}(User)\oplus SHA\text{-}256_{256}(URI)||timestamp]$ 作为实体临时访问凭证 Access Token 一起传递到 RP；

⑤RP 用 URI-IdP 的公钥验证签名有效性，并提取 timestamp 验证实体临时访问凭证 Access TokenID 是否过期，如果过期，则拒绝该用户的访问。$UUID_{256}(User)$保证了每次用户每次登陆的凭证新鲜性，可以有效预防重放攻击，$SHA\text{-}256_{256}(URI)$为了事后用户实体行为的身份取证，确认用户的唯一 URI 标识。

上述隐私保护方案，实现了对多元实体身份标识的匿名保护，达到了一种访问请求以临时凭证出示的目的，显著提高了访问请求的安全性。

## 5. 基于标识关联映射的安全审计

在具体的业务过程中，多元实体可能会涉及多种类型的标识信息（身份标识和云资源标识）。例如针对用移动设备连接企业资源的公司用户，其自身具备组织机构的身份标识，其使用的设备具有 IMEI 标识，其访问的公司资源具备虚拟资产标识，通过多元实体映射关系可以建立标识间的关联，从而实现基于多元实体标识的综合审计。

流程如下：

①审计方(Auditor)无条件的访问由 URI-IdP 创建与维护的基于 URI 的实体身份标识映射表（表 1），取得该映射表中任意一条记录的 URI 和 UUID256 属性值。该映射表由四个属性字段组成，分别为实体身份标识 URI、256 位登录随机通用唯一识别码 UUID256、实体访问时间戳 Timestamp 和实体 URI 身份标识 256 位哈希摘要 SHA256，保证 URI 和 SHA256 的一致性以及 URI、UUID256 的唯一性；

表 1 基于 URI 多元实体身份标识映射表

| URI | UUID256 | Timestamp | SHA256 |
|---|---|---|---|
| $uri_1$ | $UUID_{256}(_{user1})$ | $times_1$ | $sha_1$ |
| $uri_2$ | $UUID_{256}(_{user2})$ | $times_2$ | $sha_2$ |

②审计方 (Auditor)无条件的访问由RP创建与维护的访问请求映射表，该映射表包括如下属性字段：访问凭证Access Token、访问资源标识Resource ID、访问类型标识Access ID、当前访问时间CurrentTime以及访问结果Result。该映射表如表2所示。其中，Access Token指实体在第四部分生成的实体临时访问凭证，且具有唯一性。审计方 (Auditor)用URI-IdP的公钥验证每条记录 Access Token 有效性，并从中取得这条记录的

UUID256(User)⊕SHA-256256(URI)||timestamp，除去时间戳timestamp形成UUID256(User)⊕SHA-256256(URI)与①取得UUID256进行异或运算UUID256(User)⊕SHA-256256(URI) ⊕UUID256(User)获得SHA-256256(URI)摘要，通过基于URI实体身份标识映射表中URI和SHA256的一致性，确定访问实体URI标识在访问请求映射表对应的记录。

表 2 访问请求记录映射表

| Access Token | Resource ID | Access ID | CurrentTime | Result |
|---|---|---|---|---|
| $AccessT_1$ | $urn_1$ | Update | $CurrentT_1$ | Permit |
| $AccessT_2$ | $urn_2$ | Delete | $CurrentT_2$ | Deny |

③将②中确定的实体访问记录保存在审计方创建与维护的审计访问请求记录映射表。审计方依据要审查的多元实体访问情况的记录，获知实体 URI 身份标识、访问的资源标识、对访问资源的处理、实体访问的当前时间及访问结果，以维护审计访问请求记录映射表。该映射表由以下属性字段构成：URI、Resource ID、Access ID、CurrentTime 和 Result，如表 3 所示。

表3 审计访问请求记录映射表

| URI | Resource ID | Access ID | CurrentTime | Result |
|---|---|---|---|---|
| $uri_1$ | $urn_1$ | Update | $CurrentT_1$ | Permit |
| $uri_2$ | $urn_2$ | Delete | $CurrentT_2$ | Deny |

④重复操作，直到被审计实体 URI 所有记录检索完毕。

## 6. 结束语

本文提出了一个基于身份标识管理的安全应用模型。其中包括：多元化实体的统一身份标识方法，实现了针对实体标识的分布式存储、隐私保护和安全审计的方法。多元化实体的统一身份标识方法，适用于云计算场景中实体在各自体系下标识信息的统一，通用性、实用性较强。基于通用唯一识别码 UUID、DVB 同步字节反转和 SHA-256 算法解决了实体的统一身份标识匿名隐私保护，达到一种访问请求以临时凭证出示的目的，保证了访问请求的安全。多元实体的统一身份标识的存储检索依赖于高效简易的一致性哈希环CHR，达到一种理想的分布式存储查询的目的，因此存储方便、查询效率高。此外，云资源的统一标识采用了基于 LDAP 树形结构的存储索引方法，更好的适应了云资源可扩展、虚拟化、动态性的要求。基于多元实体标识关联映射的审计方法，使得访问请求对审计方的完全透明，具有较强的审计性。所有统一标识信息的传输，利用安全通道机制来实现，因此显著提高了信息传输的安全性。

## 参考文献(References)


[1] Suho Jeong, Seong Hoon Kim, and Minkeun Ha. Enabling Transparent Communication with Global ID for the Internet of Things. Proceedings of Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo, Italy, 4-6 July 2012, 695-701.

[2] Yong Xu, Ling Zhang and Xiaobin Zhou, Research on open identity authentication model for PKI, Journal of National University of Defense Technology, 2013, 35(1):169-174.

[3] Jie Xu, Huayun Yin, Jian Sun, CAS-Based Key Management Scheme for Immeasurable Information Sharing Network, 2012 IEEE 14th International Conference on Communication Technology, Chengdu, 9-11 Nov. 2012, 654-658.

[4] Philip J. Riesch, Xiaojiang Du, Audit Based Privacy Preservation for the OpenID Authentication Protocol, 2012 IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, 13-15 Nov. 2012, 348-352.

[5] Jesse Leskinen, Evaluation Criteria for Future Identity Management, 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, United Kingdom, 25-27 Jun. 2012, 801-806.

[6] Haitham S. Al-Sinani and Chris J. Mitchell. Client-based CardSpace-OpenID Interoperation. Proceedings of the 26th International Symposium on Computer and Information Sciences, London, UK, 26-28 Sep. 2011, 387-393.

[7] Haitham S. Al-Sinani and Chris J. Mitchell. Extending the Scope of Cardspace. Proceedings of the 4th international conference on Security of information and networks, Sydney, Australia, 14-19 Nov. 2011, 235-238.

[8] Waleed A. Alrodhan, Chris J. Mitchell. Improving the security of CardSpace. EURASIP Journal on Information Security, 2009, 3(2):1-18.

[9] T. Holmes, U.Zdun, and S.Dustdar, Automating the Management and Versioning of Service Models at Runtime to Support Service Monitoring, 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, Beijing, 10-14 Sep. 2012, 211-218.

[10] Herbert Van de Sompel and Oren Beit-Arie. Open linking in the scholarly information environment using the OpenURL Framework. New Review of Information Networking, 2001, 7(1): 59-76.

[11] Mark Needleman. The OpenURL: An Emerging Standard for Linking. Serials Review, 2002, 28(1):74-76.

[12] Hui Li and Xin Zhou. Study on Security Architecture for Internet of Things. Proceedings of the International Conference of Applied Informatics and Communication, Xi'an, 20-21 Aug. 2011, 404-411.