基于数字证书的 openstack 身份认证协议

朱智强[1,2]，林韧昊[1]，胡翠云[1]

（1. 解放军战略支援部队信息工程大学密码工程学院，河南 郑州 450001；
2. 郑州信大先进技术研究院，河南 郑州 450001）

摘 要：openstack 作为开源云平台的行业标准，其身份认证机制采用的是 keystone 组件提供的基于用户名/口令的单因素认证方式，不适用于对安全等级需求较高的应用场景。因此，设计出一种基于数字证书的身份认证协议，该协议包括云用户身份标识协议和云用户身份鉴别协议，来满足高安全性应用场景的安全需求。通过对 keystone 组件进行扩展实现了基于数字证书的身份认证系统，该系统综合运用了密码认证服务器、UKey、加密、完善的密钥管理等技术。经分析，该系统能够有效抵抗多种网络攻击，提高了云用户在登录云平台时的安全性。

关键词：云计算；数字证书；身份认证系统；身份认证协议

中图分类号：TP309

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019030

Openstack authentication protocol based on digital certificate

ZHU Zhiqiang[1,2], LIN Renhao[1], HU Cuiyun[1]

1. Institute of Cryptography Engineering, Information Engineering University, Zhengzhou 450001, China
2. Zhengzhou Xinda Institute of Advanced Technology, Zhengzhou 450001, China

Abstract: As the industry standard for open source cloud platforms, openstack uses the single-factor authentication method based on username and password that provides by keystone components to identity authentication mechanism, while it is not suitable for application scenarios with high security level requirements. A digital certificate-based identity authentication protocol which had cloud user identification protocol and authentication protocol was designed to meet the requirements. With expending the keystone component to achieve a digital certificate-based identity authentication system, a combination of authentication server, UKey technology, encryption technology and well-established key management and so on was used. According to the research, the system can effectively resist multiple cyber-attacks and improve the security of cloud users when they log in to the cloud platform.

Key words: cloud computing, digital certificate, authentication system, authentication protocol

1 引言

随着互联网和计算机技术的不断发展，产生了一种新的计算模式——云计算，云计算技术通过整合分布式资源，构建出灵活、可扩展的虚拟计算环境[1]。随着云计算的不断发展，大量的云用户将自己的隐私数据保存在云端[2]，若云用户的合法身份被冒充，就会造成隐私数据的泄露，身份认证技术是解决这一问题的可靠技术。目前，云计算已经广泛应用于政务、金融、邮政、军事等对安全性需求较高的领域，与其相关的云基础设施和云平台均应按照信息系统安全要求的最高等级进行建设。应用最为广泛的基于用户名/口令的认证方式存在诸多安全性问题，不能适用于这些应用场景。数字证书是一个经证书授权中心（CA, certificate authority）签名的用来标识通信双方身份信息的一串数字，主

收稿日期：2018−01−11；**修回日期**：2018−06−30

基金项目：国家重点研发计划基金资助项目（No.2016YFB0501900）

Foundation Item: The National Key Research and Development Program of China (No.2016YFB0501900)

要包含了公钥及公钥拥有者的信息[3]，具有唯一性和不可否认性。数字证书的主要作用是确保互联网上通信双方进行安全且相互信任的通信。

openstack 是一个由美国国家航空航天局和 Rackspace 合作研制的开源项目，是一种旨在提供基础设施即服务（IaaS，infrastructure as a service）的虚拟化管理平台，能够提供可靠的云部署方案[4]。目前，许多国内外的云行业领军企业都参与到了 openstack 的项目研发中，并将其作为云计算架构的基石，这标志着 openstack 已经是事实上的开源云平台的行业标准。

2　openstack 云平台认证机制分析

2.1　开源云平台 openstack

openstack 云平台由多种组件共同组成，各组件分工明确、相互独立，构成云平台的实质就是组合这些组件[5]。openstack 采用无共享、基于消息的架构，既可以将每个组件分别部署在单独服务器上，也可以部署在同一台服务器上。各组件统一采用 Rest-full API（application programming interface）接口规范，具有模块松耦合、组件配置灵活、易于二次开发等优点。

openstack 的安全问题一直是国内外各大企业、学者关注的焦点。官方成立了 openstack 安全项目组并编写了《openstack 安全指南》，致力于解决 openstack 中存在的安全问题，并通过安全漏洞管理团队及时对漏洞进行修复。他们于 2015 年创建了 Security 项目，提供安全工具解决 openstack 中存在的安全问题，并在 2018 年 2 月 28 日发布的 Queens 版本中对 Security 项目进行更新，通过 Anchor 提供轻量级 PKI（public key infrastructure）服务，自动验证和签发短期证书。通过该项目中 Bandit 等工具检测代码、API、各组件中存在的安全问题和漏洞。

2.2　基于 keystone 组件的认证机制

openstack 相较于其他虚拟化管理软件，采用了灵活的低耦合分布式 SOA（service-oriented architecture）架构模式，所以需要一个类似服务总线的模块对 openstack 中的各组件进行统一的授权认证及服务规则管理。

openstack 在 essex 版本增加了 keystone 这一核心组件[6]并使其作为串联 openstack 中各组件的认证授权中心，keystone 组件通过其特有的插件化结构为 openstack 云平台用户提供身份管理（account）、访问控制（authentication）以及统一授权（authorization）服务。

keystone 组件作为 openstack 云平台的身份认证核心[7]，可以与其他后端授权系统进行集成，其身份认证机制的特点主要通过令牌（token）来实现，主要包括了 UUID token、PKI token 以及 fernet token。云用户在与 openstack 其他组件交互之前，需要从 keystone 获得令牌，然后将该令牌用于与 openstack 云平台中各组件交互时的身份验证，其认证工作具体流程如图 1 所示，其中项目（project）

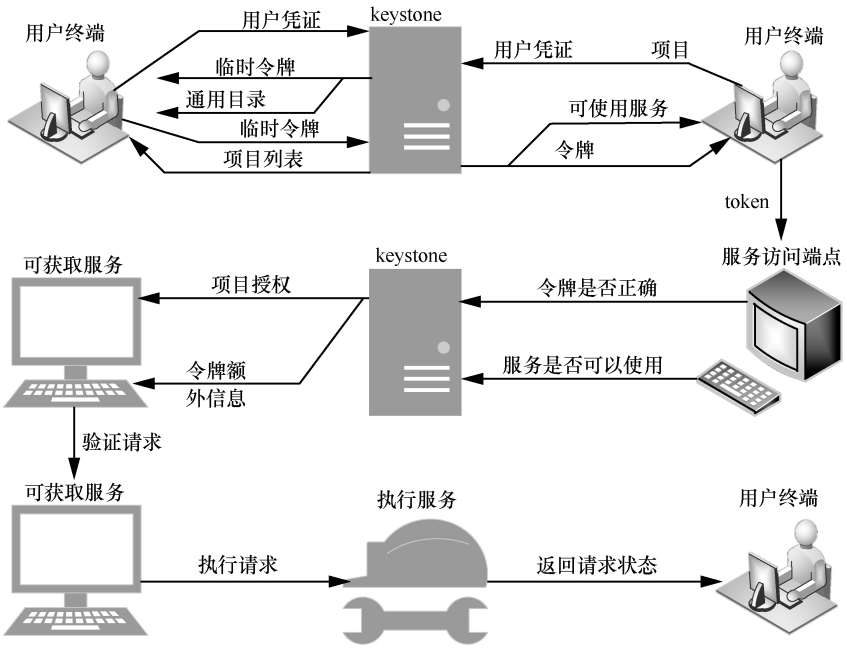

图 1　keystone 组件工作流程

为 keystone 中的基本概念，表示各个服务中可以访问的资源集合。

openstack 云平台基于 keystone 组件的认证机制存在以下问题。

1) 基于 PKI token 的认证方式可以在一定程度上保证云用户在访问其他应用组件时的安全，但是用户在登录云平台时目前仅支持基于用户名/口令认证方式。这种认证方式存在口令泄露、易遭受口令猜测攻击、字典攻击等安全隐患。

2) 认证过程中，用户名/口令都是以明文形式发送，默认使用 HTTP 协议进行认证[8]，攻击者可能通过协议分析器窃听用户的认证信息或对认证请求数据分组进行截取，进而发动重放攻击。

3) 即使将 keystone 组件配置成支持 HTTPS 加密访问环境，攻击者仍能够利用 man-in-the-middle 机制截取网络通信数据并进行数据篡改进而发动中间人攻击。

4) 没有提供重复登录失败后限制登录的方法[9]，这可能会遭受暴力攻击以及 DoS（denial of service）攻击。

5) 文献[10]通过定量安全评价对 openstack 安全性进行分析，根据漏洞数据库（NVD，national vulnerability database）的经验数据得出 keystone 是最脆弱的组件，存在大量漏洞。攻击者可能通过这些漏洞对用户的隐私数据进行窃取。

6) keystone 组件是部署 openstack 云平台的唯一强制服务[11]，这使得 keystone 成为攻击者的首要攻击目标。集成第三方身份验证和授权协议是一种更安全的部署方法。

综上所述，基于 keystone 组件的认证方式存在诸多安全问题，不能有效抵抗多种网络攻击，不能适用于政务、军事等对安全性需求较高的应用场景，因此需要一种高安全性的认证方式来保护 openstack 云用户的隐私数据安全。

2.3 密钥管理组件 Barbican

为了在云中提供强大的数据保护，加密技术通常用于保护传输中的数据以及 rest 中的数据。密钥管理是所有使用加密的云组件面临的巨大挑战。

Barbican 是为 openstack 云平台提供 key 的管理存储服务的组件[12]，通过 rest API 来提供和管理机密信息并进行安全存储。Barbican 组件通过加密密钥对用户机密信息进行加密，成为 Barbican 的秘密（secret）：包括密码、公私钥、加密密钥，并存储到 Barbican 的后端数据库中。Barbican 组件提供了外部密钥管理设备的接口，具有良好的扩展性，可以快速存储和检索传递给 openstack 组件的密钥[13]，能够提供灵活、可靠的密钥周期管理服务。

Barbican 组件的核心是数据的加解密以及秘密的存储，在组件内部生成一个密钥，存储在用于加密处理的硬件安全模块中（HSM, hardware security module），在秘密存储到数据库前对其进行加密。读取数据时，先从数据库读取密文，再通过密钥解密后获取明文，这可以有效防止数据库中的数据泄露。

3 基于数字证书的 openstack 身份认证系统

3.1 认证系统架构

针对 openstack 云平台基于 keystone 的认证机制在政务、军事等领域中安全性不足的问题，本文设计并实现了基于数字证书的 openstack 身份认证系统来保护 openstack 云平台用户隐私数据的安全。针对不同云用户对云平台身份认证安全等级的不同需求，结合 openstack 基于 keystone 组件的认证特点并通过对 keystone 组件进行扩展，从而支持云计算环境下基于数字证书的认证方式。

认证系统主要包括云密码管理子系统和云身份认证子系统两部分。云密码管理子系统主要为云平台用户提供证书的管理、签发服务，由云平台证书分发服务器、证书密钥分发插件以及用户证书下载插件组成。云身份认证子系统是证书认证服务的提供者，由密码认证服务器、云认证客户端插件和认证令牌处理插件组成。认证系统组成如图 2 所示。

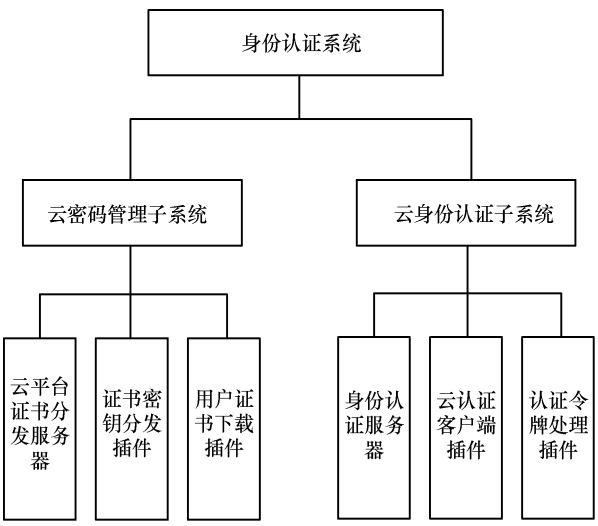

图 2 认证系统组成

本系统将数字证书藏于 UKey（USB Key）中，利用 UKey 提供的安全机制使其作为用户私钥与证书的安全载体[13]。在部署方面增加密码认证服务器和证书分发服务器，并对传统的证书认证及认证网关所采用的 socket 通信接口进行改造，统一采用 Rest-full API 进行服务调用，提高标准化及快速扩展能力。使用 Barbican 组件作为密钥容器代替 keystone 使用的文件方式密钥存储，统一将客户端与服务端的公私钥对作为 secret 经 Barbican 加密密钥加密后进行统一管理存储，使用密码认证服务器内置密码卡代替 HSM 存储 Barbican 加密密钥。考虑到系统性能要求，将密码认证服务器部署成认证服务器集群，统一为云平台中各组件提供证书或数字签名的验证服务。在数据库模块中增加新的用户证书信息表，用来存储 keystone 用户与证书之间的关系。认证系统总体架构如图 3 所示。

3.2　认证系统功能

认证系统主要完成云平台用户的证书管理、签发以及身份认证工作。

1) 云平台用户的证书管理、签发

在 keystone 后端部署云平台证书分发服务器并与 keystone 组件结合，同时另一端与公钥密码基础设施连接，提供用户和设备证书管理服务。管理员根据需要构建用户证书管理请求连接到云平台证书签发服务器，获得证书密码管理服务权限，并通过证书密钥分发插件向用户直接颁发数字证书，从而实现云平台用户证书的在线签发、更新与撤销。

2) 云平台用户身份认证

在 keystone 后端部署密码认证服务器，主要处理云用户基于 UKey 数字证书的身份认证和令牌签发工作，通过内置密码卡来提供密码技术支持，与 keystone 组件协同完成云平台证书用户的身份认证。通过认证令牌处理插件与 keystone 组件结合，使 keystone 能够接受新的认证令牌，解析协议数据，并发送至密码认证服务器进行验证，而后从本地身份管理模块中获取用户对应的证书信息，查看该用户的合法性，从而实现基于证书密码的全局身份认证。

4　基于数字证书的 openstack 认证协议

本文通过一种基于数字证书的 openstack 认证协议保障云用户身份认证的安全。

表 1 给出协议描述相关的符号记法。协议涉及五方参与者：客户端 C、密码认证服务器 AS（authentication server）、管理员终端 MT（management terminal）、keystone 服务器 KS（key stone server）、证书分发服务器 CS（certificate authorization server ），参与者能够通过密钥管理组件 Barbican 获得对方公钥 UK（public key）。协议分为云用户身份标识阶段与云用户身份鉴别阶段。

由于对称密钥加密具有实时性且加密效率高等优点，因此统一使用对称密钥 SK（symmetric key）对数据进行加密传输，同时考虑到密钥安全性，利用非对称加密公钥 UK 将对称密钥封装到数字信封中发送至服务端，服务端利用私钥 RK（private key）对其进行解密后提取 SK。

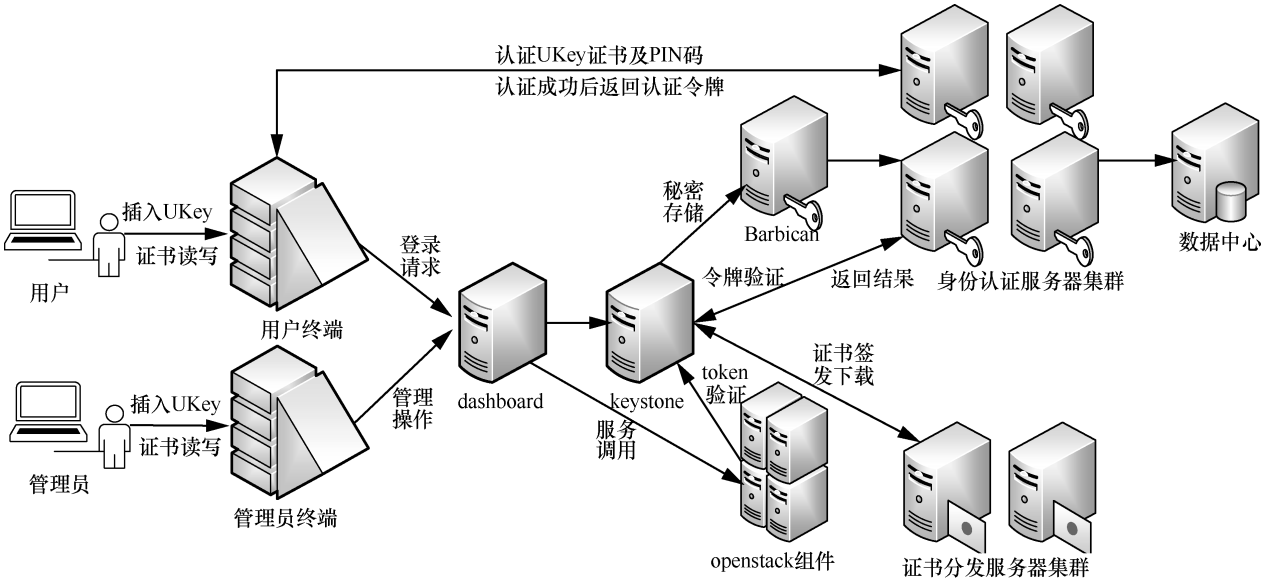

图 3　认证系统总体架构

表 1 认证协议符号记法

符号	描述
C	客户端
MT	管理员终端
AS	密码认证服务器
KS	keystone 服务器
CS	证书分发服务器
ID	用户身份标识
P	用户 PIN 码
Dm	域名
‖	字符串连接符
A→B：M	节点 A 向节点 B 发送消息*
$E(\cdot)$	加密操作
$H(\cdot)$	散列操作
Cer	数字证书
K	密钥
UK	公钥
RK	私钥
SK	共享密钥
SC	签名消息 SC：包含随机数 N_R、时间戳 T_0 和接受者信息 M_R
T	令牌

注：* A 和 B 指认证过程中交互的不同节点。

4.1 云用户身份标识阶段

1) 管理员通过管理员系统发出新建用户指令，调用 UKey 设备产生一个经 MD5（message-digest algorithm 5）Hash 算法加密后的随机数，通过云密码管理子系统向 CS 提交证书签发请求及随机数，CS 端通过对校验随机数有效性，返回 verify request。在握手过程中使用文献[14]所提出的 cookie 的交换和验证来抵御 DoS 攻击。

$$\mathrm{MT}\rightarrow\mathrm{CS}:\mathrm{client\ hello}\|H(N_R)\|\mathrm{UK_{MT}}$$

$$\mathrm{CS}\rightarrow\mathrm{MT}:\mathrm{verify\ request}\|\mathrm{Cer_{CS}}\|\mathrm{UK_{CS}}$$

2) 通过本地 UKey 设备产生临时公私钥对 $\mathrm{UK_{UKey}}$ 和 $\mathrm{RK_{UKey}}$，并将用户信息提交至 keystone 端。

$$\mathrm{MT}\rightarrow\mathrm{KS}:E_{\mathrm{SK}}(\mathrm{Userinfo}\|\mathrm{UK_{UKey}}\|\mathrm{RK_{UKey}}),E_{\mathrm{UK_{KS}}}(\mathrm{SK})$$

3) keystone 在完成正常用户创建过程后，向云密码管理子系统提交证书申请。云密码管理子系统收到请求后，向 CS 提交证书签发请求。

$$\mathrm{KS}\rightarrow\mathrm{CS}:E_{\mathrm{SK}}(\mathrm{Userinfo}\|\mathrm{UK_{UKey}}\|\mathrm{RK_{UKey}}),E_{\mathrm{UK_{CS}}}(\mathrm{SK})$$

4) 申请成功后，CS 对用户信息、公钥进行签名，获得数字证书 $\mathrm{Cer_{UKey}}$ 与被加密私钥 $\mathrm{RK_E}$，返回至 KS。

$$\mathrm{CS}\rightarrow\mathrm{KS}:\mathrm{E_{SK}}(\mathrm{Cer_{UKey}}\|\mathrm{RK_E}),E_{\mathrm{UK_{KS}}}(\mathrm{SK})$$

5) KS 向服务后台 dashboard 返回证书和加密私钥，并将证书与私钥写入用户 UKey。

$$\mathrm{KS}\rightarrow\mathrm{MT}:E_{\mathrm{SK}}(\mathrm{Cer_{UKey}}\|\mathrm{RK_E}),E_{\mathrm{UK_{MT}}}(\mathrm{SK})$$

6) 云用户身份标识流程如图 4 所示。

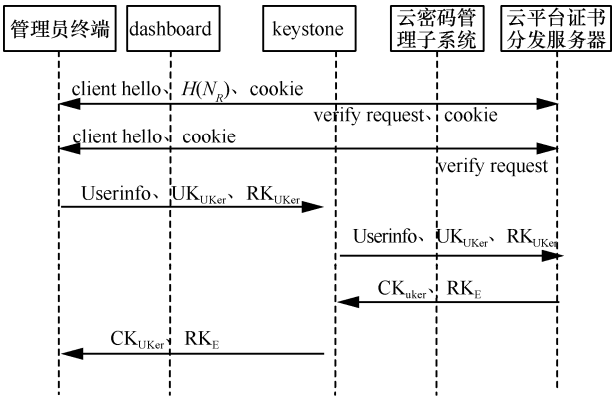

图 4 云用户身份标识流程

4.2 云用户身份鉴别阶段

1) 客户端与认证服务端通过 Rest-full API 进行握手并发送 HTTP 请求进行连接。传入 PIN（personal identification number）码激活认证客户端插件。调用本地 UKey 提取数字证书 Cer，携带 Cer 和一份签名消息 $\mathrm{SC}=H(N_R\|T_0\|M_R)$ 向密码认证服务器 AS 发送登录请求，验证成功后返回认证服务器令牌，其中签名消息中包含一个时间戳 T_0 和随机产生的 random nonce。采用成熟的对称加密算法（AES, advanced encryption standard）对传输信息进行加密/解密。

$$\mathrm{C}\rightarrow\mathrm{AS}:\mathrm{client\ hello}\|\mathrm{UK_C}$$

$$\mathrm{AS}\rightarrow\mathrm{C}:\mathrm{verify\ request}\|\mathrm{Cer_{AS}}\|UK_{\mathrm{AS}}$$

$$\mathrm{C}\rightarrow\mathrm{AS}:E_{\mathrm{SK}}(\mathrm{Cer_{UKey}}\|\mathrm{SC}),E_{\mathrm{UK_{AS}}}(\mathrm{SK})$$

$$\mathrm{AS}\rightarrow\mathrm{C}:E_{\mathrm{SK}}(T_{\mathrm{auth}}),E_{\mathrm{UK_C}}(\mathrm{SK})$$

2) 将客户端与 keystone 服务端扩展为双向认证，C 发送认证服务请求，KS 收到请求之后将自己的证书及相关信息发送至 C。客户端验证 keystone 服务端证书的有效性，验证成功后携带 ID、Dm、T_{auth} 通过 dashboard 提交至 KS 身份鉴别模块。

$$\mathrm{C}\rightarrow\mathrm{KS}:\mathrm{client\ hello}\|\mathrm{UK_C}$$

$$\mathrm{KS}\rightarrow\mathrm{C}:\mathrm{Cer_{KS}}\|\mathrm{UK_{KS}}$$

$$\mathrm{C}\rightarrow\mathrm{KS}:E_{\mathrm{SK}}(\mathrm{ID}\|\mathrm{Dm}\|T_{\mathrm{auth}}),E_{\mathrm{UK_{KS}}}(\mathrm{SK})$$

3) KS 通过身份鉴别模块将认证令牌提交至认证服务器进行认证，成功后获取用户及证书信息。同时将安全域名、用户名、转化为本地 userID 提交至本地身份管理模块。

$$\mathrm{KS}\rightarrow\mathrm{AS}: E_{\mathrm{SK}}(T_{\mathrm{auth}}), E_{\mathrm{UK}_{\mathrm{AS}}}(\mathrm{SK})$$

$$\mathrm{AS}\rightarrow\mathrm{KS}: E_{\mathrm{SK}}(\mathrm{Cer\ info}\|\mathrm{Userinfo}), E_{\mathrm{UK}_{\mathrm{KS}}}(\mathrm{SK})$$

4) 将获取的证书信息同身份认证子系统返回的用户证书信息进行比较映射；成功后通过 KS 向客户端返回 keystone 令牌。

$$\mathrm{KS}\rightarrow\mathrm{C}: E_{\mathrm{SK}}(T_{\mathrm{keystone}}), E_{\mathrm{UK}_{\mathrm{C}}}(\mathrm{SK})$$

5) 当用户访问其他组件或应用服务时，携带 keystone token 和一份签名消息 SC 向 keystone 发送认证请求，认证成功后处理请求并返回认证结果。

$$\mathrm{C}\rightarrow\mathrm{KS}: E_{\mathrm{SK}}(T_{\mathrm{keystone}}\|\mathrm{SC}), E_{\mathrm{UK}_{\mathrm{KS}}}(\mathrm{SK})$$

云用户身份鉴别流程如图 5 所示。

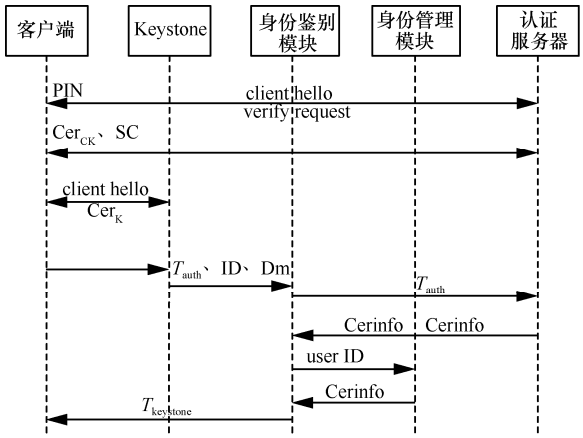

图 5　云用户身份鉴别流程

5　原型系统及安全性分析

5.1　原型系统

针对不同云用户的需求，在保留原有认证方式的基础上，为高安全需求的用户提供基于数字证书的认证方式。

系统主要功能是为管理员用户及普通用户提供身份的标识和鉴别服务。系统工作流程如图 6 所示。在系统安装时至少需要设置一名管理员用户并绑定一个 UKey。管理员使用 UKey 及 PIN 码通过图 7 所示云用户登录界面发送认证请求，通过密码认证服务器及 keystone 组件按照本文所设计的系统认证流程进行认证，成功后返回认证结果并跳转至图 8 所示 openstack 管理员系统界面。通过管理员系统中的添加用户按钮跳转至图 9 所示云用户创建界面，根据不同用户对安全等级的需求来填写用户信息和证书序列号，对于安全等级需求较高的用户需向证书分发服务器提交证书签发请求，通过证书分发及下载插件将数字证书写入 UKey 设备并颁发给普通云用户。云用户使用 UKey 及 PIN 码通过登录界面发送认证登录请求，成功后即可使用 openstack 云平台中的各项服务。

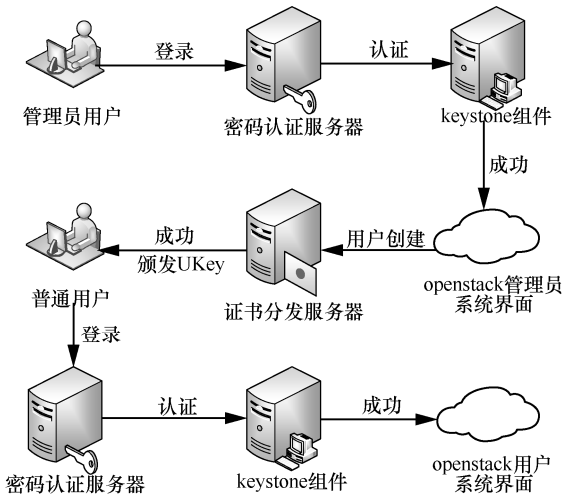

图 6　系统工作流程

图 7　云用户登录界面

认证系统主要分为云用户的标识和鉴别这两部分。在云用户身份标识部分，通过与云平台证书密钥分发插件进行集成，通过 Rest-full API 与云平台证书分发服务器进行握手并建立连接，从而支持证书的签发下载。对数据库表结构进行修改，增加 PKI 证书属性。在 openstack_auth 组件代码中对 view.py 文件中的 login 函数进行修改，将不同的 URL 请求将转至 UKey 认证的 form 页面。界面部分在 openstack 原有的用户标识页面的基础上增加了是否使用 PKI 证书的选项以及对应的证书序列号输入框。

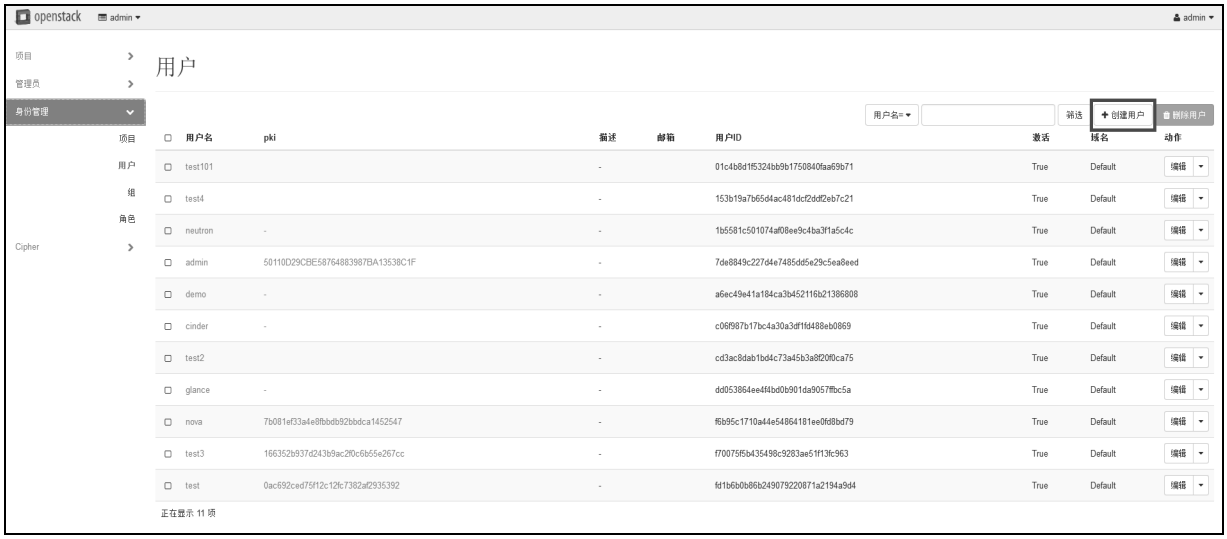

图 8　openstack 管理员系统界面

图 9　云用户创建界面

在云用户身份鉴别部分，通过与云认证客户端插件进行集成，负责读取云用户 UKey 设备中的证书。对 keystone 组件进行扩展，增加新的 PKI 认证方式及该认证方式所需要解析的数据格式和属性。在 openstack_auth/plugin 目录下新建 usbkeytoken.py 文件，定义一个新的认证插件，支持 UKey token 的认证方式，在 identities/v3/目录下实现封装这种认证方式所需要传递的参数和协议，并支持对这种协议的引用。界面部分将云用户口令输入框替换为 UKey 设备口令输入框，为云用户提供基于数字证书的身份鉴别服务。

在系统应用方面，本文基于 openstack 云平台搭建了密码云（cryptographic cloud）系统。密码云系统是对分布式密码资源进行整合，并通过虚拟化技术生成密码资源池，构建出具有较高动态扩展性和灵活性的虚拟计算平台，按需向用户提供密码服务，解决了传统密码技术受限于固定的载体和密码资源不可扩展等问题。密码云系统对身份认证技术有着极高的安全性需求，若攻击者成功获得系统的授权，可能会对密码信息资源进行窃取和篡改。目前本文系统已全面应用于基于密码云服务系统的应用场景，管理员使用 UKey 设备通过本文所设计的认证协议登录基于 openstack 的密码云系统来对密码虚机进行统一管理，提高了云用户在登录云平台系统时的安全性。

政务云（government cloud）系统指通过云计算技术为政府部分行业提供基础设施、应用系统及信息安全等综合服务的云计算平台。但是涉及政府机密数据信息的泄露是具有破坏性的。在 2017 年国家电子政务外网管理中心电子政务云集成与应用国家工程实验室联合发布的《政务云安全要求》（GW0013-2017）中明确指出，安全接入平台是政务用户通过互联网或移动专线网络访问政务云的部门业务和公共区业务的唯一接入通道，接入平台应具备数字认证、授权管理等功能。本文通过数字证书来实现系统的身份认证和访问控制，符合政务云对安全性方面的要求，可以很好地适用于政务云系统。

5.2　安全性分析

在协议安全性方面，文献[10]在 openstack 原有

认证协议的基础上提出了一种基于安全度量的方法对 openstack 认证系统进行定量化安全评估，对 keystone 组件中存在的漏洞进行分析与修补。文献[16]提出了一种基于改进 OpenID 框架的 openstack 认证协议。文献[16]提出了一种基于 keystone 的安全认证协议，通过加密技术来提高云平台身份认证的安全性。本文从加密技术、密钥管理、协议效率以及 openstack 现阶段易遭受的主要网络攻击抗性等角度将本文协议与 openstack 原有的基于 keystone 令牌的认证协议及文献[10]、文献[15-16]协议进行比较。对比分析结果如表 2 所示。

在加密技术方面，本文与文献[16]协议采用对称加密和非对称加密相结合的加密技术，相较于其他认证协议，具有更高的数据机密性。基于 PKI token 的认证方式虽然可以通过非对称加密技术拓展为 HTTPS 加密认证环境，但是在后续令牌的交互过程数据仍以明文传输，不能保证整个认证过程的数据安全。文献[15]协议通过 MD5 算法对云用户在登录时填写的认证码进行加密，但是无法保证数据传输过程中的安全。在认证因素方面，本文所设计的认证协议是双因素认证协议，相较于其他单因素认证协议，攻击者必须获得全部认证因素才能冒充合法用户。在密钥管理方面，本文方案相较于其他对比方案采用了 Barbican 组件提供的密钥管理和存储服务，通过密码认证服务器的内置密码卡充当硬件安全模块来存储 Barbican 密钥，能够有效防止后端数据库发生密钥泄露。另外，本文方案通过密码认证服务器与 keystone 组件协同完成认证服务，攻击者难以仅通过对 keystone 组件的漏洞进行攻击来窃取用户数据。

本文主要针对 openstack 云平台现阶段易遭受的主要网络攻击进行分析。在抵抗重放攻击方面，本文方案通过客户端向服务端发送服务请求时携带的签名消息中包含一个时间戳 T_0 和一个随机数 N_R，并通过 Hash 算法签名保证其完整性。当攻击者重放认证请求时，认证服务器端通过验证随机数以及对比时间戳判断并有效抵抗重放攻击。文献[16]方案通过添加时间戳在一定程度上也能够抵抗重放攻击，而其他认证协议中没有抵抗重放攻击的有效手段。在抵抗中间人攻击方面，本文协议与文献[16]在认证协议中采用了双向认证机制，客户端需验证服务端证书保证其身份的合法性，在一定程度上可以抵御中间人攻击。本文方案相较于其他对比方案提供了重复登录失败后限制登录的方法。若攻击者取得 UKey，但是 PIN 码错误次数超过规定限制，UKey 将自动死锁。这种机制可以很好地抵抗 UKey 丢失攻击、暴力攻击、字典攻击及用户身份鉴别阶段的 DoS 攻击，而 cookie 的交换和验证机制能够抵抗用户标识阶段可能遭受的 DoS 攻击。

对比结果显示，本文所提出的认证协议虽然在工作效率方面有所牺牲，且在一定程度上会增加认证组件的负载，但具有更高的安全性，更加完善的密钥管理方式，并且能够有效抵抗 openstack 云平台现阶段易遭受的网络攻击。在政务、军事等应用场景中，高安全性往往是第一考虑要素，所以本文方案对于安全等级需求较高的应用场景具有重要意义。

表 2　　**协议对比分析结果**

对比项	基于 UUID token 的认证协议	基于 PKI token 的认证协议	文献[10]协议	文献[15]协议	文献[16]协议	本文协议
加密技术	无	非对称加密	无	Hash 加密	对称加密/非对称加密	对称加密/非对称加密/Hash 加密
双因素认证	否	否	否	否	否	是
密钥管理	低	低	低	低	低	高
安全性						
抗重放攻击	否	否	否	否	是	是
抗中间人攻击	否	否	否	否	是	是
抗 DoS 攻击	否	否	否	否	否	是
协议效率	高	低	高	中	低	低
认证组件负载	中	中	中	中	高	高

6 结束语

本文重点对 openstack 云平台中基于 keystone 的身份认证机制进行了研究，分析了基于 keystone 组件的认证方式在安全性上的不足，提出了一种基于数字证书的身份认证协议并实现了相应的身份认证系统。根据云用户对安全等级的不同需求，提供一种具有更高安全性的身份认证方式。分析表明，基于数字证书认证方式能够更有效地防范多种可能遭受的攻击行为，从而更好地保证云平台用户隐私数据的安全。下一步工作是对系统性能进行评估并通过适当的负载均衡策略使认证服务器集群负载分布更加均衡，从而降低认证服务器负载量，提高整个系统的工作效率。

参考文献：

[1] 王斌锋, 苏金树, 陈琳.云计算数据中心网络设计综述[J]. 计算机研究与发展, 2016, 53(9):2085-2106.
WANG B F, SU J S, CHEN L. Overview of cloud computing data center network design [J]. Computer Research and Development, 2016, 53(9): 2085-2106.

[2] 张玉清, 王晓菲, 刘雪峰, 等. 云计算环境安全综述[J]. 软件学报, 2016, 27(6):1328-1348.
ZHANG Y Q, WANG X F, LIU X F, et al. Survey on cloud computing security.[J] Journal of Software, 2016, 27(6):1328-1348.

[3] HARN L, REN J. Generalized digital certificate for user authentication and key establishment for secure communications[J]. IEEE Transactions on Wireless Communications, 2011, 10(7): 2372-2379.

[4] WEN X, GU G, LI Q, et al. Comparison of open-source cloud manegement platforms: openstack and OpenNebula[C]//IEEE Fuzzy Systems and Knowledge Discovery.2012:2457-2461.

[5] SEFRAOUI O, AISSAOUI M, ELEULDJ M. openstack: toward an open-source solution for cloud computing[J]. International Journal of Computer Applications, 2012, 55(3): 38-42.

[6] KHAN R H, YLITALO J, AHMED A S. Openid authentication as a service in openstack[C]//The 7th International Conference on Information Assurance and Security.2011:372-377.

[7] MARTINELLI S, NASH H, TOPOL B. Identity, authentication, and access management in openstack: implementing and deploying keystone[M]. O'Reilly Media, 2015.

[8] ABDULLA N, ERÇELEBI E. Identify cloud security weakness related to authentication and identity management (IAM) using openstack keystone model[C]//International Conference on Engineering and Technology, Computer, Basics and Applied Sciences. 2017:1-5.

[9] COOPER J D. Analysis of security in cloud platforms using openstack as case study[D]. AGDER: The University of AGDER Faculty of Engineering and Science, 2013.

[10] TORKURA K A, CHENG F, MEINEL C. Application of quantitative security metrics in cloud computing[J]. Internet Technology & Secured Transactions. 2015:256-262.

[11] WOO S W, JOH H C, ALHAZMI O H, et al. Modeling vulnerability discovery process in apache and iis http servers[J] Computers & Security, 2011, 30(1):50-62.

[12] SITARAM D, HARWALKAR S, SIMHA U, et al. standards based integration of advanced key management capabilities with openstack[C]//IEEE International Conference on Cloud Computing in Emerging Markets. 2016 :98-103.

[13] 王帅, 常朝稳, 魏彦芬. 基于云计算的 USB Key 身份认证方案[J]. 计算机应用研究, 2014, 31(7):2130-2134.
WANG S, CHANG C W, WEI Y F. USB key authentication scheme based on cloud computing [J]. Computer Application Research, 2014, 31(7): 2130-2134.

[14] 李鹏坤, 王小峰, 苏金树, 等. 基于标识密码的数据报传输层安全协议[J]. 软件学报, 2017, 28(2): 90-97.
LI P K, WANG X F, SU J S, et al. Datagram transport layer security protocol based on identity cipher[J]. Journal of Software, 2017, 28(2): 90−97.

[15] 周长春, 田晓丽, 张宁, 等. 云计算中身份认证技术研究[J]. 计算机科学, 2016, 43(6A):339-341.
ZHOU C C, TIAN X L, ZHANG N, et al. Research on identity authentication technology in cloud computing[J]. Computer Science, 2016, 43(6A):339-341.

[16] CUI B, XI T. Security analysis of openstack keystone[C]//International Conference on Innovative Mobile & Internet Services in Ubiquitous Computing. 2015: 283-288.

[作者简介]

朱智强（1961− ），男，河南汝南人，博士，解放军战略支援部队信息工程大学教授，主要研究方向为云计算与信息安全。

林韧昊（1993− ），男，河南郑州人，解放军战略支援部队信息工程大学硕士生，主要研究方向为云计算安全与云环境下的资源调度技术。

胡翠云（1985− ），女，河南辉县人，博士，解放军战略支援部队信息工程大学讲师，主要研究方向为云计算安全。