基于协同地址碰撞的隐蔽认证方法

曹 旭 祝跃飞 费金龙
(信息工程大学数学工程与先进计算国家重点实验室 郑州 450002)

摘 要 云计算的兴起不可避免地带来了一些安全问题,服务资源的非授权访问就是其中的一个重要威胁。对此,基于IPv6地址的新特性,提出一种协同地址碰撞技术,即待认证节点通过多个协同节点的配合,向关键主机所在网络发送在IP地址接口标识部分隐含认证秘密的数据包,以实现对节点的隐蔽认证。理论和实验分析结果表明,该方案可有效提高网络的安全性。

关键词 云计算,IPv6,地址碰撞,通信认证
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.034

Cooperative Address Knocking Based Covert Authentication

CAO Xu ZHU Yue-fei FEI Jin-long
(State Key Laboratory of Mathematic Engineering and Advanced Computing, Information Engineering University, Zhengzhou 450002, China)

Abstract With the development of cloud computing, it is inevitable that many security problems arise. Unauthorized service access is one of the most important threats. Based on the new features of IPv6 address, we proposed a new network security technique called cooperative address knocking, which can be seen as an undetectable authentication. It is a form of host-to-host communication which relies on deliberate communication attempts from some cooperative nodes. These connection attempts are monitored by a daemon which interprets the interface identifier of destination IP addresses as information. The theoretical and empirical analysis demonstrate that CAKCA scheme can effectively conduct undetectable authentication and prevent the exposure of existence of the important host. The theoretical analysis and simulation results show that the proposed scheme can effectively improve the level of network security.

Keywords Cloud computing, IPv6, Address knocking, Communication authentication

1 引言

云计算因其宽带互联、资源池共享、弹性配置、按需服务和按服务收费等独特优势,在各行业应用中快速兴起[1]。在公用云中,大量用户都可以在云中租赁资源,并且可以租赁基础设施向其他用户提供服务。如何为合法用户提供可靠服务而拒绝非法用户的非授权访问,是当前基于云计算的服务系统需要解决的重要问题。非法用户没有经过授权或有意避开系统接入控制机制就对网络信息进行访问,从而获取敏感信息或对系统进行破坏等[2]。如何保证网络服务在对授权用户访问正常开放的同时拒绝非授权用户访问,是当前相关研究面临的重要问题。

传统的用户认证技术的基本思想是要求用户在获取准入权限之前先证明自己的身份。大多情况下,这一认证往往是由具体的网络服务自行负责,例如在登录某网络服务系统之前要求在网页输入用户名和密码。有的网络服务系统在设计之初未考虑安全需要,未加入身份认证功能;有的网络服务系统设计复杂,攻击者利用其存在的某安全缺陷能够进行非授权访问等。对此,人们广泛部署了防火墙、IDS/IPS等网络防御系统以帮助网络服务发现并抵御网络攻击[3]。这些网络防御系统往往需要依靠数据包中的IP地址信息对用户进行甄别,然而IP地址欺骗、会话劫持等攻击手段层出不穷,合法的源地址并不能证明用户身份的合法性,而且当前主机IP大多采用动态方式分配,网络防护系统规则配置不易。因此,亟需不同于防火墙、IDS/IPS等防护系统的新的认证手段,以加强网络安全防护水平。

本文提出一种基于协同地址碰撞的隐蔽认证方法(Cooperative Address Knocking based Covert Authentication, CAKCA),即通过多个节点协同进行地址碰撞,实现重要服务器的"按需"开放,从而降低其被发现、受攻击的可能性。

2 相关工作

在隐蔽认证方面,Barham等首次提出了隐蔽认证服务(Silent Authentication Service, SAS)[4],其是一种利用防火墙来对客户端进行认证的方案。其基本原理是:某重要应用服务默认不对任何主机开放,只有主机发送特定数据包后才对其开放,这些数据包的TCP和IP首部中通过一定的编码方式隐含了用于认证的秘密信息。Ali等提出利用端口碰撞

到稿日期:2015-08-01 返修日期:2015-12-11

曹 旭(1983-),男,博士生,主要研究方向为云计算与网络信息安全,E-mail:xu20101002@126.com;祝跃飞(1964-),男,博士,教授,博士生导师,主要研究方向为密码学与网络信息安全;费金龙(1981-),男,博士,讲师,主要研究方向为云计算与网络信息安全。

(Port Knocking)的技术来保护主机，抵御 TCP 重放攻击以及端口扫描探测[5]，其基本原理是防火墙默认阻止外部节点对特定服务的访问，直到访问节点能够按一定的顺序向特定的端口发送数据包。该特定的端口序列实质上可看作是访问该特定服务的密钥。此外，研究者们针对端口碰撞技术提出了一系列的改进方案。例如，Mehran 等提出一种增强型的端口认证机制[6]——SPKT(Secure Port Knock-Tunneling)；Srivastava 等基于 QRC 和 AES 提出一种改进的端口碰撞方案[7]；此外，基于碰撞的思想，Sahu 等实现了一种改进的基于端口碰撞的混合认证方法[8]；Hadi 等提出一种混合的端口碰撞技术，其实现了对主机身份的有效认证[9]；Liew 等提出一种基于 SPA(Single Packet Authentication)和 IPsec 隧道进行隐蔽认证的新方案[10]；Singh 等人还给出了碰撞技术在移动设备认证方面的应用方案[11]。

在利用数据包地址部分传递信息方面，Dunlop 等设计并实现了 MT6D(Moving Target IPv6 Defense)系统[12]，其通过在 IPv6 地址部分引入双方共享的密钥和时间片实现通信过程中数据包地址的动态变化，以此干扰攻击者的探测，达到提高通信安全的目的。

本文提出的 CAKCA，是一种基于碰撞思想、由多个节点协同利用 IPv6 数据包中的地址来传递认证数据的新型隐蔽认证方法。与已有的碰撞式认证相比，CAKCA 具有以下特点：1)与上层报文无关。CAKCA 利用 IP 地址来传递认证数据，因此 IP 数据包内封装的上层数据既可以是 TCP，UDP 等常见的数据报文，也可以是 ICMP 等无端口的控制类报文。2)与报文顺序无关。已有的碰撞式认证大多采用了端口等序列来承载秘密信息，而 CAKCA 以多个节点协同碰撞实现隐蔽的认证，碰撞报文接收的先后不会影响秘密的验证。

3 基于协同地址碰撞的隐蔽认证方法

基于协同地址碰撞的隐蔽认证的基本思想是：服务器默认不对任意主机开放。某主机需要访问服务器时，不是直接向服务器发送连接请求，而是将包含了访问请求的秘密数据发送到其他一些主机，由其他主机根据收到的秘密数据向特殊的 IP 地址发送数据包。服务器所在网络的守护程序根据这些数据包解密认证请求，然后在防火墙上增添允许该主机访问服务器的规则，之后主机即能正常访问服务器。

在具体描述 CAKCA 前，首先介绍 CAKCA 方案涉及的几个基本概念。

定义 1(碰撞秘密，Knocking Secret) 碰撞秘密是隐蔽认证的具体信息，包括一次认证中目标服务器 IP 地址的哈希值、客户端 IP 地址的哈希值和具体的操作指令。

定义 2(碰撞秘密碎片，Knocking Secret) 碰撞秘密碎片由碰撞秘密分割得到。

定义 3(CAKCA 守护程序，CAKCA Daemon，CD) 网络内用以控制 CAKCA 的守护程序，主要完成两个功能：1)负责对碰撞报文进行处理；2)对防火墙中的规则进行编辑以允许/关闭某主机对服务器的访问。CAKCA Daemon 保存有子网内部重要服务器 IP 地址的哈希值，用于根据碰撞秘密确定待处理的服务器地址。

定义 4(CAKCA 客户端，CAKCA Client，CC) 采用 CAKCA 机制向服务器申请认证的通信节点。

定义 5(CAKCA 协同节点，CAKCA Cooperative Node，CCN) 收到 CC 认证请求后，配合 CC 向服务器发送碰撞报文的节点。

3.1 基本过程

基于协同地址碰撞的隐蔽认证技术的基本过程如下。

(1)首先在服务器所在子网安装 CAKCA 守护程序，对进入本地子网的数据流进行监视、判别。

(2)某主机 $Client_X$ 需要访问受保护的服务器 Server 时，直接向服务器使用过的某个地址 IP_{server} 发送数据包会被拦截，如图 1(a)所示。这时 $Client_X$ 向其同伴节点 $Client_1$，$Client_2$，$Client_3$，…，$Client_n$ 发送协同碰撞请求，其中包含了本次碰撞的秘密数据，如图 1(b)所示。

(3)同伴节点 $Client_1$，$Client_2$，$Client_3$，…，$Client_n$ 收到来自 $Client_X$ 的协同碰撞请求之后，首先根据主机信息(IP、主机名称)确认 $Client_X$ 的身份；在身份确认通过之后，根据各自所收到的秘密数据向守护程序所在主机的特定 IP 地址发送碰撞报文，如图 1(c)所示。在高安全要求的环境中，可以通过加入公钥体制进行主机身份认证，进而提高系统的安全性。

(4)CAKCA Daemon 收到达到门限数量的碰撞报文后，根据报文地址解密秘密数据内容，在网络防护设备处加入规则，在一定时间内允许主机 $Client_X$ 对服务器 Server 进行远程访问，如图 1(d)所示。

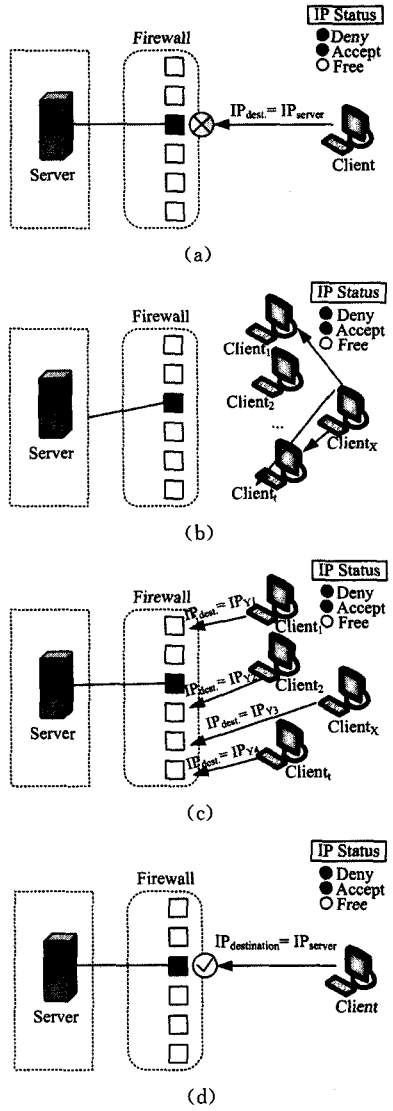

图 1 CAKCA 的基本过程

3.2 碰撞秘密数据包的处理

碰撞秘密数据包的处理过程主要包括：碰撞秘密的分割、碰撞数据包的生成和碰撞秘密数据包的验证3个核心过程。其中申请认证的节点 $Client_X$ 执行碰撞秘密的分割过程；各个客户端节点都需要执行碰撞秘密数据包的构造过程；CAKCA Daemon执行碰撞秘密数据包的验证过程。

3.2.1 碰撞秘密结构

碰撞秘密是CAKCA的核心，包含了一次认证所需的信息，包括服务器IP地址哈希值、客户端IP地址哈希值、命令和时间戳4个部分，如图2所示。

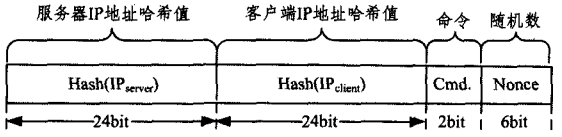

图2 碰撞秘密结构

其中，服务器IP地址哈希值 $Hash(IP_{server})$ 由服务器IP地址经过哈希函数处理后得到，长度为24bit。客户端IP地址哈希值 $Hash(IP_{client})$ 由客户端IP地址经过哈希函数处理后得到，长度为24bit。

Hash函数采用如下方式构造：$C_i=b_{i1}\oplus b_{i2}\oplus b_{i3}\oplus b_{i4}$，其中，$C_i$ 表示hash码的第 i 位，$1\leqslant i\leqslant n$，b_{i1}，b_{i2}，b_{i3} 和 b_{i4} 为128位IPv6地址的后96位按序所划分出的4个24位的输入块。

命令包括打开和关闭两种，长度为2bit。随机数用于标识一次认证，长度为6bit。

3.2.2 碰撞秘密的分割

申请认证的节点 $Client_X$ 执行碰撞秘密的分割过程。碰撞秘密的分割基于拉格朗日多项式的秘密共享来实现。首先确定构造秘密共享的门限(k,n)，即需要构造 $k-1$ 次多项式：

$$f(x)=(a_1x^{k-1}+a_2x^{k-2}+\cdots+a_{k-1}x+M) \bmod p \quad (1)$$

随后，选取 $i(k<i<n)$ 个未知元 x_i，计算相应的多项式值 $f(x_i)$。$\langle x_1,f(x_1)\rangle,\langle x_2,f(x_2)\rangle,\cdots,\langle x_i,f(x_i)\rangle$ 即成为分割后的碰撞秘密份额。完成碰撞秘密分割之后，由客户端 $Client_X$ 任选一个碰撞秘密份额作为自己生成碰撞秘密数据包所需的秘密份额，然后通过加密隧道将剩余的 $i-1$ 个碰撞秘密份额传递至 $i-1$ 个协同节点。

3.2.3 碰撞秘密数据包的生成

各个协同节点收到碰撞秘密份额后，首先根据主机信息对客户端 $Client_X$ 进行确认。在认可 $Client_X$ 之后，即表示愿意协同 $Client_X$ 完成秘密认证过程。

以每个 x_i(8bit)和对应的 $f(x_i)$ (56bit)作为数据包目的接口标识构造碰撞数据包，数据包的格式如图3所示。

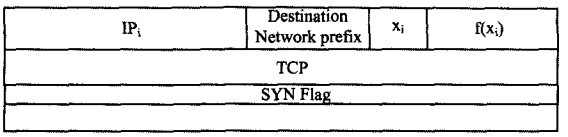

图3 碰撞数据包结构示意图

3.2.4 碰撞秘密数据包的验证

碰撞数据包的验证过程由CD(CAKCA Daemon)完成。根据所约定的秘密共享分割方案来恢复所传输的碰撞秘密 M，具体按拉格朗日插值公式 $F(x)=\sum_{i=1}^{k}y_if_i(x)$ 来恢复多项式 $f(x)=(a_1x^{k-1}+a_2x^{k-2}+\cdots+a_{k-1}x+M) \bmod p$。其中 y_i 对应着函数 $f_i(x)$ 在 x_i 位置上的取值。

$$f_i(x)=\prod_{j=1,j\neq i}^{k}\frac{x-x_j}{x_i-x_j}$$

具体验证流程可用如下伪代码表示。

```
Proc Name:CAKCAVer()
Procedure:
    监视数据流;
    若在某一时刻 t 监视发现碰撞数据报文,开始碰撞验证过程:
      在一定的时间范围内收集 k(秘密共享门限值)个碰撞数据报文;
      When (i<k)
      Extract IID_dest. of Message_i;
    //根据前述秘密共享原理还原碰撞秘密
    Calculate Knocking secret;
    //根据碰撞秘密中的相关信息设置防火墙规则
      Conduct the ACCEPT rules on firewall for IP_client
```

4 模型分析

4.1 模型性能分析

为考查模型性能，首先对CAKCA的碰撞认证时间开销进行分析。假设丢包率为 ε，则CAKCA客户端成功通过碰撞认证的概率 P 可通过如下公式计算得到：

$$P=(1-\varepsilon)^k \quad (2)$$

其中，k 为所需的碰撞秘密门限值。

进一步可分析得到，CAKCA客户端完成一次碰撞认证的平均时间为：

$$E(T)=\sum_{i=1}^{\infty}((i-1)\alpha+i\beta)(1-P)^{i-1}P \quad (3)$$

其中，α 是碰撞认证的超时门限时间，β 是协同碰撞节点的最大转发时延。

由上可得出如下结论：

结论1 CAKCA的系统性能与4个参数相关，即丢包率 ε、碰撞秘密的门限值 k、超时门限 α 和协同碰撞节点的最大转发时延 β。其中 k 和 α 可由系统本身设定，而 ε 和 β 则与系统部署的网络环境有较大关系。

然后，进一步对本文提出的CAKCA的平均碰撞时间进行模型仿真分析。设定碰撞秘密的门限值 $k=5$，超时门限 $\alpha=30s$，协同碰撞节点的最大转发时延 $\beta=10s$。那么在不同的丢包率下，CAKCA的平均碰撞时间如图4所示。由图4可知，即使在丢包率达到10%时，CAKCA平均碰撞时间也未超过15s。

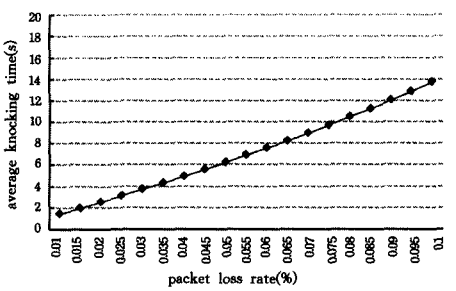

图4 平均碰撞时间分析结果

4.2 模型安全性分析

4.2.1 与端口碰撞技术的安全性对比分析

假设攻击者采用远程DDoS方式暴力猜解认证，具体通过随机发送数据包对存活节点进行探测，每个报文的目的地

址都不相同，同时实施攻击的攻击者数量为 m，其攻击报文发送频率为 r 个/秒。

首先分析已有的端口碰撞技术，端口碰撞技术要求客户端访问固定的端口序列完成认证。假设端口序列 Port_1，…，Port_v 是一个长度为 v 的序列，碰撞秘密的有效期为 t，可用端口数为 s，则攻击者成功攻破端口碰撞认证的概率为

$$P=\frac{m\cdot r\cdot t}{v}\cdot\frac{1}{A_s^v} \tag{4}$$

然后分析本文给出的协同地址碰撞方法。假设同时分发的秘密份额为 i，门限为 k，可用接口标识数为 u，即攻击者只要猜中 i 个秘密份额中的 k 个即表示破解成功，碰撞秘密的有效期为 t，那么攻击者成功攻破端口碰撞认证的概率为

$$P=\frac{m\cdot r\cdot t}{k}\cdot\frac{1}{C_u^k} \tag{5}$$

然后，进一步对本文提出的 CAKCA 和典型的 Port Knocking 的抗攻击能力进行模型仿真分析。设定攻击节点数目为 10^5，猜解报文发送频率为 $r=10^5$ 个/秒，秘密更换周期 $t=1800$s，同时分发的秘密份额为 $i=10$，门限 $k=5$，且为简化分析，端口碰撞可用的端口数 $s=2^{16}$，IPv6 中单个链路可用地址数目为 $u=2^{64}$。那么在不同攻击强度(用猜解报文发送频率表示)下，攻击者成功攻破 Port Knocking 和 CAKCA 的概率如图 5 所示。

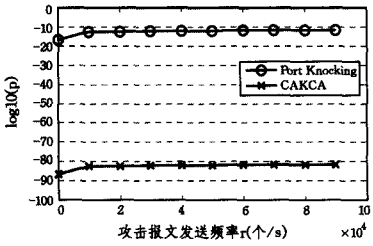

图 5　抗攻击能力对比分析结果

由结果可知，本文所提出的 CAKCA 被攻击者成功攻破的概率远远低于传统的端口碰撞(Port Knocking，PK)方案，即 CAKCA 比 PK 具有更好的抗攻击能力。

4.2.2　CAKCA 抗干扰能力分析

CAKCA 面临的主动攻击者可能将网络协议中的保留比特予以置位(如使用 0 比特进行填充)，随机化协议中的 ID 编号字段，然而这些手段都不会对 CAKCA 产生任何影响。CAKCA 进行认证信息传递的位置是地址接口标识部分，如果对所有报文的此部分进行干预，将导致所有主机无法正常通信。

主动攻击者还有可能采用包填充的手段，以统一包长。然而 CAKCA 用于存储信息的位置在数据包的首部，对数据包的统一填充不会对 CAKCA 产生任何影响。主动攻击者还可能随机延迟数据包的发送时间，破坏数据通信的时间特性。然而 CAKCA 并非使用包间隔等时间特性作为信息的载体，所以针对时间特性的处理不会对 CAKCA 产生任何影响。由上分析可得出如下结论：

结论 2　CAKCA 中存放于接口标识部分的认证数据难以被攻击者主动干扰破坏。

此外，由 CAKCA 碰撞秘密的组成结构可知，碰撞秘密中包含了随机数字段，即每个碰撞秘密都有一定的有效期。即使攻击者截获所有碰撞报文并成功将其发送至 CAKCA Daemon，也会由于 CAKCA Daemon 在解密碰撞秘密后发现重放的碰撞秘密中的时间戳重复，而使得相关命令得不到执行。

5　实验分析

为测试 CAKCA 的性能，构建如图 6 所示的实验环境。整个 CAKCA 的实验系统包括客户端 $Client_X$、协同节点 CCN_1－CCN_6、服务器 Covert Server、与服务器同网络的普通主机节点(Normal Node)以及守护程序 CAKCA Daemon。其中客户端、协同节点、普通主机节点和服务器采用 Windows7 主机(Intel Core 2 Duo E5700 (3.00GHz)，4GB RAM)，由一台 Ubuntu 主机作为网关，在其上部署 CAKCA Daemon 以及用以控制访问的防火墙 iptables(开启 ip6table 模块)。

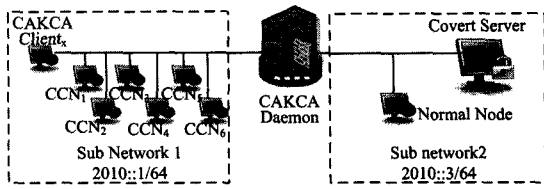

图 6　实验环境示意图

首先测试客户端生成秘密并对秘密进行分割的时间。CAKCA 系统参数设定为：碰撞报文为 TCP (带 SYN 标志)报文，碰撞秘密分割份额数为 10，碰撞门限值为 5。重复测试 10 次，结果如图 7 所示。

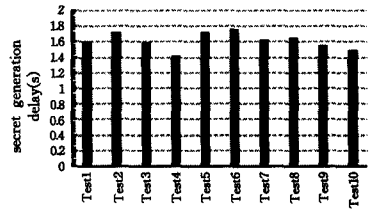

图 7　生成并分割碰撞秘密的时间测试结果

由图 7 可知，在给定的硬件条件下，时间消耗在 1.6s 左右，碰撞秘密生成并分割时间较短，且耗时稳定。

由前节分析可知，CAKCA Daemon 的运行效率将直接关系到系统整体的性能。为此，进一步测试在网关处 CAKCA Daemon 对数据包的处理能力。由 $Client_X$ 向 Server 申请访问，系统参数设定与前文相同，随机选取将 6 个秘密份额发向 CCN_1－CCN_6。记录 CAKCA 从收到第 5 个碰撞报文(达到解密门限)到设置 iptables 规则(表明 $Client_X$ 认证通过)的时间。重复实验 10 次，结果如图 8 所示。

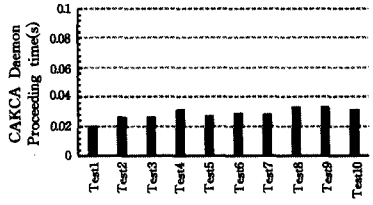

图 8　CAKCA Daemon 对数据包处理能力的测试结果

由结果可知，在给定的硬件条件下，CAKCA Daemon 对碰撞数据的处理能力较好，所测时间在 0.02～0.032s 之间。

由于 CAKCA 设计的目的是隐蔽地实现认证，因此如果 CAKCA Daemon 的处理效率太低，将使得攻击者怀疑网关处可能对数据包进行了额外的处理，进而怀疑目标服务器可能部署防御系统以使其无法访问。因此，本文进一步测试部署 CAKCA Daemon 带来的数据包处理的时间开销。首先开启 CAKCA Daemon，然后使用节点 $Client_X$ 与 Normal Node 进

行数据交互，在 $Client_X$ 处记录从发出 TCP SYN 数据包到收到 TCP SYN 确认数据包的时间，然后停止 CAKCA Daemon，记录同样的时间作为对比。重复实验 10 次，结果如图 9 所示。

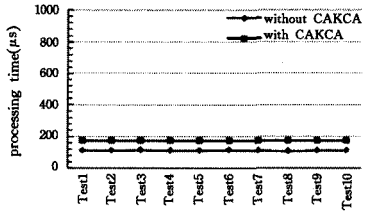

图 9　CAKCA 数据包处理时延的测试结果

由结果可知，CAKCA 的部署使得网关处的数据包处理时延有一定的增加，但增幅较小，也就是说攻击者无法通过比对数据包处理时延来确定 CAKCA 的部署。

结束语　网络用户身份认证是网络安全的核心问题之一。随着网络攻击技术的不断多样化、复杂化，阻止未授权用户对关键主机的访问是尽量保证网络安全的有效手段。本文提出一种协同地址碰撞技术，基于 IPv6 地址的新特性，结合秘密分享方法，通过多个节点的协同碰撞，实现对用户身份的隐蔽认证。本文的研究成果有望为基于 IPv6 新特性提出云计算环境下网络安全防护新技术提供一些思路。

参 考 文 献

[1] Yu Neng-hai, Hao Zhuo, Xu Jia-jia, et al. Review of Cloud Computing Security[J]. Acta Electronica Sinica, 2013, 41(2): 371-381(in Chinese)
俞能海，郝卓，徐甲甲，等. 云安全研究进展综述[J]. 电子学报，2013，41(2)：371-381

[2] Reeja S L. Role Based Access Control mechanism in cloud computing using cooperative secondary antuorization recucling methd[J]. International Journal of Emerging Technology and Advanced Engineering, 2012, 2(10): 444-450

[3] Rodas O, Morales G, Alvarez J. A reliable and scalable classification-based hybrid IPS[C]// IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Gwangiu: IEEE, 2015: 599-604

[4] Barham P, Hand S, Isaacs R, et al. Techniques for lightweight concealment and authentication in IP networks: Technical Report IRB-TR-02-009[R]. Berkeley: Intel Research, 2002

[5] Ali F H M, Yunos R, Alias M A M. Simple port knocking method: against TCP replay attack and port scanning[C]// International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). Kuala Lumpur: IEEE, 2012: 247-252

[6] Mehran P, Reza E A, Laleh B. SPKT: Secure port knock-tunneling, an enhanced port security authentication mechanism[C]// IEEE Symposium on Computers & Informatics (ISCI). Malaysia: IEEE, 2012: 145-149

[7] Srivastava V, Keshri A K, Roy A D, et al. Advanced port knocking authentication scheme with QRC using AES[C]// Proceeding of Trends in Networks and Communications. Chennai: Springer, 2011: 159-163

[8] Sahu P, Singh M, Kulhare D. Implementation of modified hybrid port knocking (MHPK) with strong authentication[J]. Journal of Commerce and Management Thought, 2013, 4(2): 490-504

[9] Hadi A H, Al-Bahadili H. A Hybrid Port-knocking technique for host authentication[M]// Simulation in Computer Network Design and Modeling: Use and Analysis. 2012: 336

[10] Liew J H, Lee S, Ong I, et al. One-time knocking framework using SPA and IPsec[C]// Proceeding of 2nd International Conference on Education Technology and Computer. 2010: v5-209-v5-213

[11] Singh K, Zhong J, Mirchandani V, et al. Securing data privacy on mobile devices in emergency health situations[M]// Security and Privacy in Mobile Information and Communication Systems. Springer Berlin Heidelberg, 2012: 119-130

[12] Dunlop M, Groat S, Urbanski W, et al. MT6D: a moving target IPv6 defense[C]// Proceeding of the 2011 Military Communication Conference-Track3-Cyber Security and Network Operations. Baltimore, MD: IEEE, 2011: 1321-1326

(上接第 168 页)

[5] Bethencourt J, Sahai A, Waters B. The cpabe toolkit [EB/OL]. [2015-11-13]. http://acsc.csl.sri.com/cpabe

[6] Wang G, Liu Q, Wu J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services [C]// Proceedings of the 17th ACM Conference on Computer and Communications Security. Chicago: ACM, 2010: 735-737

[7] Wan Z, Liu J E, Deng R H. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing [J]. Information Forensics and Security, 2012, 7(2): 743-754

[8] Li M, Yu S, Ren K, et al. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings [M]// Security and Privacy in Communication Networks. Berlin: Springer, 2010: 89-106

[9] Barua M, Liang X, Lu R, et al. ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing [J]. International Journal of Security and Networks, 2011, 6(2/3): 67-76

[10] Yin K Z, Wang H H. A cloud storage system with fine-grained access control and low storage space overhead [J]. Journal of Computer Applications, 2015, 35(12): 3413-3418(in Chinese)
印凯泽，汪海航. 具有细粒度访问控制和低存储空间开销的云存储系统[J]. 计算机应用，2015，35(12)：3413-3418

[11] Doan A H, Madhavan J, Domingos P, et al. Learning to map between ontologies on the semantic web [C]// Proceedings of the 11th International Conference on World Wide Web. Honolulu: ACM, 2002: 662-673

[12] Wiederhold G. An algebra for ontology composition [C]// Proceedings of 1994 Monterey Workshop on Formal Methods. Monterey, 1994, 56: 61

[13] Borthakur D. The hadoop distributed file system: architecture and design [EB/OL]. [2015-12-15]. http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[14] Noessner J, Niepert M. CODI: Combinatorial Optimization for Data Integration-Results for OAEI 2010 [C]// Proceedings of the 5th International Workshop on Ontology Matching. Shanghai, 2010: 142-149

[15] Enterprise Ontology [EB/OL]. [2016-04-25]. http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html