一种适用于雾计算的终端节点切换认证协议

胡荣磊[①] 陈 雷*[①②] 段晓毅[①] 于秉琪[①]

[①](北京电子科技学院 北京 100070)

[②](北京邮电大学网络空间安全学院 北京 100876)

摘 要：针对当前雾计算环境下终端节点的切换认证协议在存储量、计算量和安全性等方面还存在缺陷，该文提出一种高效的终端节点切换认证协议。在该协议中，采用双因子组合公钥(TF-CPK)和认证Ticket相结合的方式，实现雾节点和终端节点的相互认证和会话密钥协商。安全性和性能分析结果表明，该协议支持不可跟踪性，可以抵抗众多已知攻击和安全威胁，且具有较小的系统开销。

关键词：雾计算；终端节点；切换认证；双因子组合公钥；认证票据

中图分类号：TN918; TP309 **文献标识码**：A **文章编号**：1009-5896(2020)10-2350-07

DOI: 10.11999/JEIT200005

A Switching Authentication Protocol of Terminal Node for Fog Computing

HU Ronglei[①] CHEN Lei[①②] DUAN Xiaoyi[①] YU Bingqi[①]

[①](*Beijing Electronic Science and Technology Institute*, *Beijing* 100070, *China*)

[②](*School of Cyberspace Security*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: In order to solve the problem that the switching authentication protocol of terminal nodes in fog computing has some defects in storage, compute and security, an efficient terminal node switching authentication protocol is proposed. In this protocol, mutual authentication and session key agreement between the fog nodes and terminal nodes are realized by the combination of Two Factors Combined Public Key(TF-CPK) and authentication ticket. The security and performance analysis results show that the protocol supports untraceability, which can resist numerous known attacks and security threats, and has a smaller system overhead.

Key words: Fog computing; Terminal node; Switching authentication; Two Factors Combined Public Key (TF-CPK); Authentication ticket

1 引言

随着物联网和云计算技术的飞速发展，原有的把终端产生的数据全部交由云中心来存储和处理的方式，不仅会造成云中心数据的堆积、无效数据占用大量存储空间，还存在服务时延过长的问题。雾计算不同于云计算的集中式，它是一种分布式计算模式[1]。雾计算层位于云层和终端层之间[2]。雾计算层由多个雾计算节点(Fog Node, FN)组成，雾计算节点可以是路由器、基站、机顶盒，甚至智能终端等拥有一定计算和存储能力的设备[3]。当终端节点(Terminal Node, TN)首次向雾节点申请服务时，需要完成初始化认证以确保只有合法节点接入雾节点，从而获得服务。随着物联网的进一步发展，人们对终端移动性的要求也越来越高[4]。终端节点移动到新雾节点时，需要新雾节点对终端节点重新认证，即切换重认证。如果切换重认证效率低，认证过程复杂，则必然造成较大的计算开销、时延开销[5,6]。

又由于雾节点和终端节点都是能量和计算能力有限的功能单元，传统的数字签名安全认证方法往往要求高计算能力和高能量消耗，所以传统的数字签名方法已经不再适用于雾计算中的安全节点认证[7]。文献[8,9]提出了一种“认证Ticket”的概念，文章

收稿日期：2020-01-02；改回日期：2020-08-07；网络出版：2020-08-12

*通信作者： 陈雷 1051274640@qq.com

基金项目：国家自然科学基金(61772047)，中央高校基本科研业务费项目(328201914)，北京电子科技学院中央高校研究基金(2017LG01)

Foundation Items: The National Natural Science Foundation of China (61772047), The Fundamental Research Funds for The Central Universities of China (328201914), The Central University Research Foundation of Beijing Electronic Science and Technology Institute (2017LG01)

采用对称密码体制，安全性不够强，初始化认证需要基站(Base Station, BS)的参与。文献[10]采用代理节点的认证模型，同样需要基站的参与。文献[11]无需基站的参与，提出了一种认证服务中心思想，但分布式的认证结构需要多个雾节点同时参与才能完成终端节点的认证。文献[12]提出了一种基于组合公钥(Combined Public Key, CPK)和切换预认证的认证方案。然而，在此方案中终端节点每次都需要预测移动位置，切换重认证过程没有为下次认证做准备，再次认证都需要进行切换预认证。

基于以上不足，本文提出了一种轻量且高效的适用于雾计算环境的终端节点切换认证协议，分别讨论了终端节点首次接入雾节点的初始化认证和终端节点移动后的切换重认证。协议通过基于双因子组合公钥(Two Factor Combined Public Key, TF-CPK)的机制，利用极小的存储空间即可实现大规模密钥的分发与管理，实现大量终端节点的初始化认证，与其他同类方法相比节省了存储空间；与采用第1代CPK技术相比，设置更新密钥和2阶复合机制为密钥更换提供了便利，同时在数字签名中保护了用户隐私。初始化认证结束后，雾节点向终端节点发放认证Ticket，在切换重认证时只需利用认证Ticket即可快速完成认证，这大大减少了认证所需时间。同现有物联网和雾计算中的终端节点认证协议相比，该协议具有更小的存储空间和更低的认证时延。

2　预备知识

2.1　TF-CPK基本原理

CPK是组合公钥的简称，是在椭圆曲线密码(Elliptic Curve Cryptosystems, ECC)上构建的基于标识的密码体制。它依据离散对数难题构建公钥矩阵和私钥矩阵，采用散列函数与密码变换将实体的身份标识映射为相应矩阵的行坐标和列坐标，并根据该坐标选取矩阵中的元素进行组合以生成大量的公私钥对，从而实现大规模密钥的分发与管理[13]。组合公钥2.0版本采用标识密钥和随机密钥相复合的公钥体制，因此也称为双因子组合公钥体制。标识密钥由实体的标识通过组合矩阵生成，标记为(IPK, isk)。其中随机密钥又分为系统密钥和更新密钥。系统密钥是系统定义的随机序列，标记为(SPK, ssk)。更新密钥由个人定义，标记为(UPK, usk)。

以实体A为例，其标识密钥$(\mathrm{IPK}_A, \mathrm{isk}_A)$的产生过程如下：在有限域$F_p$上，椭圆曲线$E: y^2 \equiv \left(x^3+ax+b\right) \bmod p$由参数$(a,b,G,n,p)$定义，其中$a,b$是系数，$a,b,x,y \in F_p$，$G$为加法群的基点，$n$是以$G$为基点的群的阶。令任意小于$n$的整数$r$为私钥，则$rG=R$为对应的公钥。在给定椭圆曲线参数$(a,b,G,n,p)$的基础上，可以构建出相应的私钥矩阵**skm**和公钥矩阵**PKM**。令**skm**为$m \times h$大小的矩阵，矩阵的元素记为$r_{i,j}\,(1 \le i \le m, 1 \le j \le h)$。公钥矩阵**PKM**的元素记为$R_{i,j}$，由$rG=R$得到公钥矩阵**PKM**。任意多对公私钥之间，其私钥之和与公钥之和构成新的公私钥对。标识到组合矩阵坐标的映射是通过对标识的HASH变换实现的。以$m=h=32$为例，将HASH输出调整成长度为165 bit的映射序列YS，构成$w_0, w_1, \cdots, w_{32}$的字符串，决定列坐标和行坐标。w_0的内容u指示列的起始坐标，以后的列坐标在前列坐标基础上加1实现。$w_1 \sim w_{32}$依次指示行坐标。标识私钥isk的计算在密钥管理中心(Key Manage Center, KMC)进行。设第i次行坐标用w_i表示，列坐标用$(u+i) \bmod 32$表示，那么标识私钥的计算以有限域F_p上的倍数加法实现，实体A的标识私钥为$\mathrm{isk}_A = \sum_{i=1}^{32} r\left[w_i, (u+i)_{32}\right] \bmod n$。标识公钥的计算以椭圆曲线$E: y^2 \equiv \left(x^3+ax+b\right) \bmod p$上的倍点加法实现，即$\mathrm{IPK}_A = \sum_{i=1}^{32} R\left[w_i, (u+i)_{32}\right]$。至此，实体$A$的标识密钥$(\mathrm{IPK}_A, \mathrm{isk}_A)$产生完成。TF-CPK除设置标识密钥、随机密钥和更新密钥外，还采用1阶复合、2阶复合的密钥机制。

密钥的1阶复合过程如下：由KMC为实体A生成一对系统密钥$(\mathrm{SPK}_A, \mathrm{ssk}_A)$。1阶组合私钥$\mathrm{csk}'_A$是标识私钥$\mathrm{isk}_A$和系统私钥$\mathrm{ssk}_A$的复合，由KMC计算得到。KMC将1阶组合私钥$\mathrm{csk}'_A$和系统公钥$\mathrm{SPK}_A$分发给实体$A$，同时删除系统私钥$\mathrm{ssk}_A$。

密钥的2阶复合过程如下：设置更新密钥为密钥更换提供了便利，同时在数字签名中保护了用户隐私。实体A随机定义一对更新密钥$(\mathrm{UPK}_A, \mathrm{usk}_A)$。更新密钥由实体$A$保管，保留到下次变更为止。2阶组合私钥$\mathrm{csk}''_A$为1阶组合私钥$\mathrm{csk}'_A$和更新私钥$\mathrm{usk}_A$的复合，由签名方实体$A$计算得到。伴随公钥$\mathrm{APK}_A$是系统公钥$\mathrm{SPK}_A$和更新公钥$\mathrm{UPK}_A$的复合，由签名方实体$A$计算得到。2阶组合公钥$\mathrm{CPK}''_A$是标识公钥$\mathrm{IPK}_A$和伴随公钥$\mathrm{APK}_A$的复合，由验证方计算得到。至此，实体$A$的2阶组合密钥$(\mathrm{CPK}''_A, \mathrm{csk}''_A)$产生完成。

2.2　认证Ticket

认证Ticket是一种认证凭证，用于终端节点的切换重认证。终端节点初始化认证阶段，初始雾节点对终端节点认证后，向其颁发认证Ticket，内容包括终端节点ID，随机数R，重认证会话密钥$K_{T,F}$和有效期限时间戳TS，并用自身认证密钥AK加密。认证密钥AK在雾节点相互认证阶段共享

给自身的邻居雾节点。终端节点重认证阶段，终端节点向新雾节点发送认证Ticket，新雾节点通过认证密钥AK解密得到认证Ticket，通过验证认证Ticket的有效性即可完成终端节点的重认证。图1为FN_1共享认证密钥示意图。

3 协议框架模型

如图2所示，雾计算终端节点认证模型主要由可信中心(Trust Center, TC)、雾节点(FN)和终端节点(TN)组成。协议分为4个阶段：(1)系统初始化；(2)雾节点邻居节点发现及认证密钥交换；(3)终端节点初始化认证；(4)终端节点切换重认证。为简化协议，本文给出以下假设：(1) TC是可信实体，诚实地响应每一个注册请求；(2) 同传统网络一样，FN与TC, TN与TC之间存在安全信道。

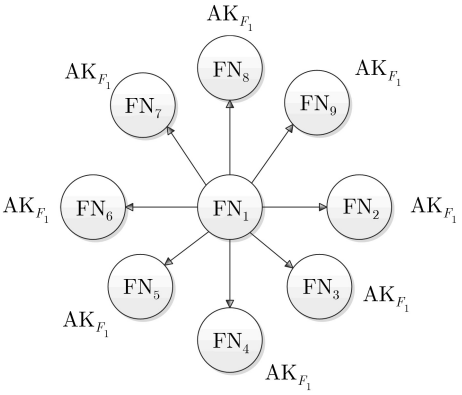

图 1 FN_1共享认证密钥AK_{F_1}

4 协议设计

本文使用的相关符号定义如下：R代表选取的随机数，ID_A代表A的身份标识，$\langle \mathrm{PK}_A, \mathrm{SK}_A \rangle$代表$A$的一对公私钥，TS为时间戳。本文使用的相关运算定义如下：$\mathrm{E}(K, m)$和$\mathrm{D}(K, c)$分别代表使用对称密钥进行加密/解密；$\mathrm{ENC}(\mathrm{SK}, m)$和$\mathrm{DNC}(\mathrm{PK}, c)$分别代表使用非对称密钥进行加密/解密；$\mathrm{SIG}_A(m)$代表$A$对消息$M$进行数字签名；$\mathrm{H}(\cdot)$代表标准散列算法[14]。

4.1 系统初始化

雾节点FN和终端节点TN向可信中心TC提出注册申请，TC验证完节点合法性后，通过安全信道将ID, 公钥矩阵**PKM**, 1阶组合私钥csk′, 系统公钥SPK, 标准散列函数$\mathrm{H}(\cdot)$, 网络共享密钥K_N和椭圆曲线参数(a, b, G, n, p)发送给雾节点和终端节点。

4.2 雾节点邻居节点发现及认证密钥交换

雾节点通过周期性的发送广播包发起与相邻雾节点间的相互认证，并在认证过程中建立邻居雾节点列表NFL。以FN_1和FN_2为例，参见步骤(1)到步骤(5)。

(1) 雾节点FN_1发送广播数据包，如图3所示。

$\mathrm{FN}_1 \to \mathrm{Broadcast}: \mathrm{E}(K_N, u_1 \| v_1)$，其中$u_1 = \mathrm{ID}_{F_1} \| \mathrm{APK}_{F_1} \| \mathrm{TS}_1, v_1 = \mathrm{H}(u_1)$.

(2) 雾节点FN_2收到FN_1的广播数据包后，用K_N解密数据包，获得ID_{F_1}，检查自己的邻居雾节点列表NFL_{F_2}，如果已经包含FN_1，则忽略该广

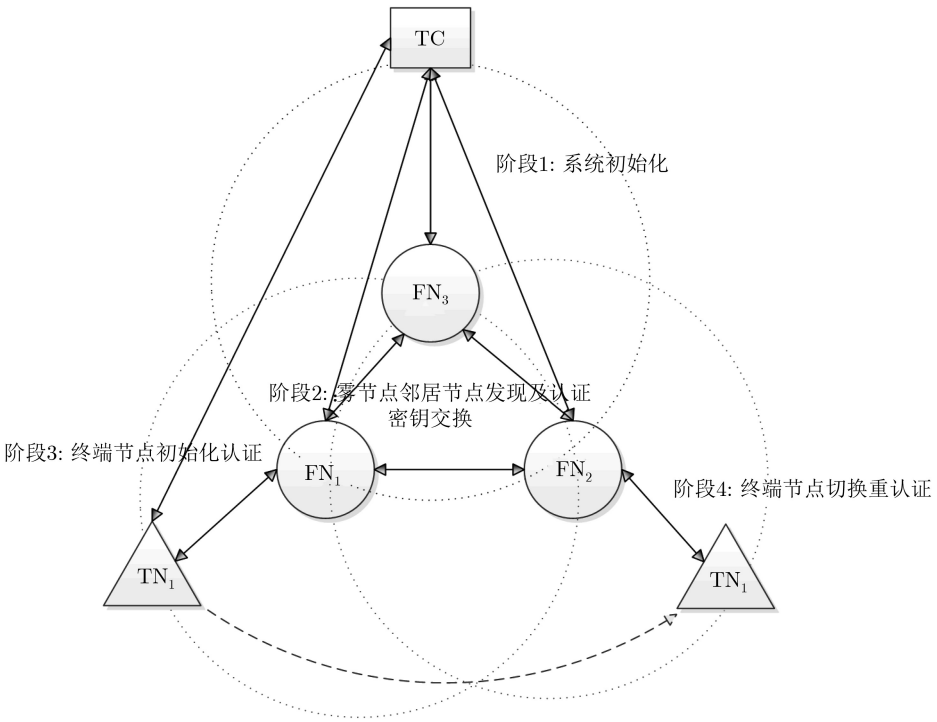

图 2 雾计算终端节点认证模型

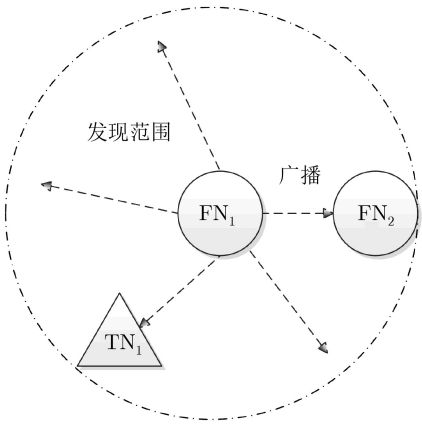

图 3 FN_1发送广播数据包

播。否则，利用ID_{F_1}和APK_{F_1}计算FN_1的2阶组合公钥CPK''_{F_1}，生成会话密钥K_{F_1,F_2}，认证密钥AK_{F_2}和时间戳TS_2。FN_2向FN_1发送消息。

$\mathrm{FN}_2 \to \mathrm{FN}_1$ ： $\mathrm{E}(K_N, \mathrm{ID}_{T_1} \| \mathrm{ID}_{F_1} \| \mathrm{APK}_{T_1} \| u_7 \| v_7)$， 其 中$u_6 = \mathrm{SIG}\left(\mathrm{csk}''_{T_1}, K_{T_1,F_1} \| \mathrm{TS}_4\right)$， $u_7 = \mathrm{ENC}\left(\mathrm{CPK}''_{F_1}, u_6 \| \mathrm{TS}_1\right)$, $v_3 = \mathrm{H}(\mathrm{ID}_{F_2} \| \mathrm{ID}_{F_1} \| \mathrm{APK}_{F_2} \| u_3 \| v_3)$。

(3) FN_1收到FN_2发送的消息后，用K_N解密数据包，利用csk''_{F_1}解密得到u_2，验证FN_2的签名得到K_{F_1,F_2}, AK_{F_2}和TS_2。FN_1认证了FN_2且有了FN_2的认证密钥AK_{F_2}, FN_1更新NFL_{F_1}，生成认证密钥AK_{F_1}和时间戳TS_3。FN_1向FN_2发送消息。

$\mathrm{FN}_1 \to \mathrm{FN}_2$：$\mathrm{E}(K_N, \mathrm{ID}_{F_1} \| \mathrm{ID}_{F_2} \| u_4 \| v_4)$，其中 $u_4 = \mathrm{E}(K_{F_1,F_2}, \mathrm{AK}_{F_1} \| \mathrm{TS}_2 \| \mathrm{TS}_3)$ ， $v_4 = \mathrm{H}(\mathrm{ID}_{F_1} \| \mathrm{ID}_{F_2} \| u_4)$ 。

(4) FN_2解密得到AK_{F_1}和时间戳TS_3。FN_2认证了FN_1且有了FN_1的认证密钥AK_{F_1}, FN_2更新NFL_{F_2}。FN_2向FN_1发送确认消息。

$\mathrm{FN}_2 \to \mathrm{FN}_1$: $\mathrm{E}(K_N, \mathrm{ID}_{F_2} \| \mathrm{ID}_{F_1} \| u_5 \| v_5)$, 其中$u_5 = \mathrm{E}(K_{F_1,F_2}, \mathrm{TS}_3)$, $v_5 = \mathrm{H}(\mathrm{ID}_{F_2} \| \mathrm{ID}_{F_1} \| u_5)$。

(5) FN_1解密得到TS_3，即可确认FN_2已经收到AK_{F_1}。至此，FN_1和FN_2完成相互认证。

其他雾节点周期性的进行上述操作，因此每个雾节点都能很快地建立起邻居雾节点列表NFL，并与邻居雾节点完成认证密钥交换，如图4所示。

4.3 终端节点初始化认证

终端节点初始化认证过程如图5所示。

(1) 终端节点TN_1收到来自雾节点FN_1的广播数据包后，计算FN_1的2阶组合公钥CPK''_{F_1}，生成会话密钥K_{T_1,F_1}和时间戳TS_4。TN_1向FN_1发送消息。

$\mathrm{TN}_1 \to \mathrm{FN}_1$ ： $\mathrm{E}(K_N, \mathrm{ID}_{T_1} \| \mathrm{ID}_{F_1} \| \mathrm{APK}_{T_1} \| u_7 \| v_7)$， 其 中$u_6 = \mathrm{SIG}\left(\mathrm{csk}''_{T_1}, K_{T_1,F_1} \| \mathrm{TS}_4\right)$， $u_7 = \mathrm{ENC}\left(\mathrm{CPK}''_{F_1}, u_6 \| \mathrm{TS}_1\right)$, $v_7 = \mathrm{H}(\mathrm{ID}_{T_1} \| \mathrm{ID}_{F_1} \| \mathrm{APK}_{T_1} \| u_7)$。

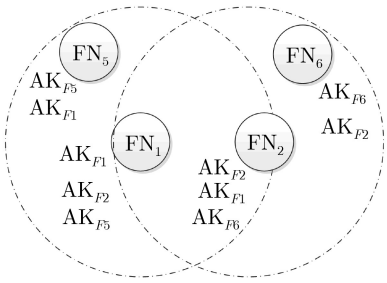

图 4 邻居雾节点密钥交换

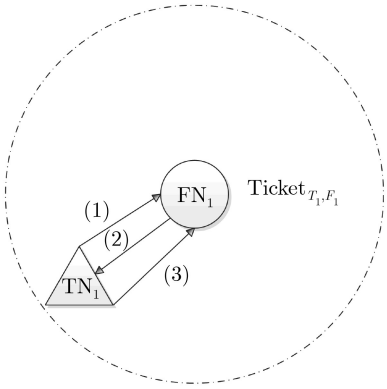

图 5 终端节点初始化认证

(2) FN_1利用ID_{T_1}和APK_{T_1}计算TN_1的2阶组合公钥，解密得到会话密钥K_{T_1,F_1}。FN_1认证了TN_1。FN_1为TN_1生成重认证会话密钥K_{T_1,F_n}, $\mathrm{Ticket}_{T_1,F_1}$, 随机数$R_1$和时间戳$\mathrm{TS}_5$。$R_1$用于$\mathrm{TN}_1$, FN_1以及重认证时的消息确认。TS_5用于指示$\mathrm{Ticket}_{T_1,F_1}$的有效期限。$\mathrm{FN}_1$向$\mathrm{TN}_1$发送消息。

$\mathrm{FN}_1 \to \mathrm{TN}_1$：$\mathrm{E}(K_N, \mathrm{ID}_{F_1} \| \mathrm{ID}_{T_1} \| u_8 \| v_8)$，其中 $u_8 = \mathrm{E}(K_{T_1,F_1}, \mathrm{NFL}_{F_1} \| K_{T_1,F_n} \| \mathrm{Ticket}_{T_1,F_1} \| R_1 \| \mathrm{TS}_4)$ ， $\mathrm{Ticket}_{T_1,F_1} = \mathrm{E}(\mathrm{AK}_{F_1}, \mathrm{ID}_{T_1} \| K_{T_1,F_n} \| R_1 \| \mathrm{TS}_5)$ ， $v_8 = \mathrm{H}(\mathrm{ID}_{F_1} \| \mathrm{ID}_{T_1} \| u_8)$ 。

(3) TN_1认证了FN_1，并且获得了$\mathrm{Ticket}_{T_1,F_1}$。$\mathrm{TN}_1$用$K_{T_1,F_1}$加密$R_1$，向$\mathrm{FN}_1$发送确认消息。

$\mathrm{TN}_1 \to \mathrm{FN}_1$：$\mathrm{E}(K_N, \mathrm{ID}_{T_1} \| \mathrm{ID}_{F_1} \| u_9 \| v_9)$， 其中$u_8 = \mathrm{E}(K_{T_1,F_1}, R_1)$, $v_9 = \mathrm{H}(\mathrm{ID}_{T_1} \| \mathrm{ID}_{F_1} \| u_9)$。

(4) FN_1解密得到R_1，即可确认TN_1已经收到相应信息。

至此，终端节点TN_1的初始化认证完成。

4.4 终端节点切换重认证

(1) 当TN_1移动到FN_2附近时，收到FN_2发送的广播数据包，则向FN_2发送消息。

$\mathrm{TN}_1 \to \mathrm{FN}_2$：$\mathrm{E}(K_N, u_{10} \| v_{10})$，其中$u_{10} = \mathrm{ID}_{T_1} \| \mathrm{ID}_{F_2} \| \mathrm{ID}_{F_1} \| \mathrm{Ticket}_{T_1,F_1} \| \mathrm{NFL}_{F_1}$, $v_{10} = \mathrm{H}(u_{10})$。

情况1：FN_2是FN_1的邻居雾节点，如图6所示。

(2) FN_2用AK_{F_1}解密$\mathrm{Ticket}_{T_1,F_1}$，得到终端节点$\mathrm{ID}_{T_1}$，会话密钥$K_{T_1,F_n}$，随机数$R_1$，有效期限时间戳$\mathrm{TS}_5$。$\mathrm{FN}_2$认证了$\mathrm{TN}_1$。$\mathrm{FN}_2$根据$\mathrm{TS}_5$验证该

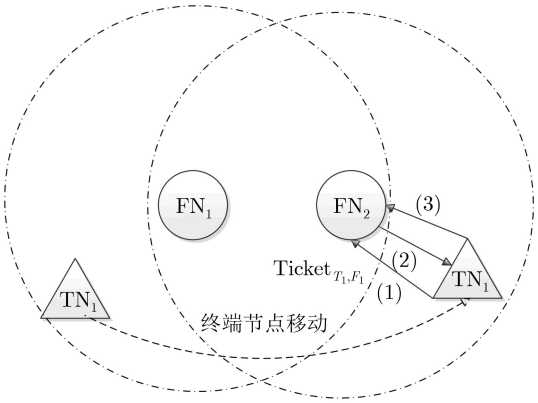

图 6 FN_2是FN_1的邻居雾节点

$Ticket_{T_1,F_1}$是否在有效期限内，如果不在，回复TN_1进行初始化认证；如果在有效期限内，FN_2为TN_1生成下次重认证会话密钥$K_{T_1,F_n'}$，随机数R_2，有效时间戳TS_6和$Ticket_{T_1,F_2}$。FN_2向TN_1发送消息。

$FN_2 \to TN_1$ ：$\mathrm{E}(K_N, \mathrm{ID}_{F_2} \| \mathrm{ID}_{T_1} \| u_{11} \| v_{11})$，其中 $u_{11} = \mathrm{E}\left(K_{T_1,F_n}, \mathrm{NFL}_{F_2} \left\| K_{T_1,F_n'} \right\| \mathrm{Ticket}_{T_1,F_2} \| R_1 \| R_2\right)$, $\mathrm{Ticket}_{T_1,F_2} = \mathrm{E}\left(\mathrm{AK}_{F_2}, \mathrm{ID}_{T_1} \left\| K_{T_1,F_n'} \right\| R_2 \| \mathrm{TS}_6\right), v_{11} = \mathrm{H}\left(\mathrm{ID}_{F_2} \| \mathrm{ID}_{T_1} \| u_{11}\right)$。

(3) TN_1解密得到随机数R_1，与之前存储的随机数比较，如果不同，则不信任FN_2；如果相同，TN_1可以相信FN_2是已经和FN_1完成相互认证的合法雾节点，并且得到下次重认证会话密钥$K_{T_1,F_n'}$和$Ticket_{T_1,F_2}$。TN_1向FN_2发送确认消息。

$TN_1 \to FN_2$ ：$\mathrm{E}(K_N, \mathrm{ID}_{T_1} \| \mathrm{ID}_{F_2} \| u_{12} \| v_{12})$，其中 $u_{12} = \mathrm{E}(K_{T_1,F_n}, R_2)$，$v_{12} = \mathrm{H}(\mathrm{ID}_{T_1} \| \mathrm{ID}_{F_2} \| u_{12})$。

(4) FN_2解密得到R_2，即可确认TN_1已经收到$Ticket_{T_1,F_2}$和NFL_{F_2}等信息，TN_1切换重认证完成。

情况2：FN_2不是FN_1的邻居雾节点，如图7所示。

FN_2将NFL_{F_2}与NFL_{F_1}对照，如果没有相同的雾节点，则向TN_1返回其应进行初始化认证响应；如果有相同的雾节点，例如FN_3，则执行以下步骤。

(5) FN_2向FN_3发送辅助认证请求。

$FN_2 \to FN_3$ ：$\mathrm{E}(K_N, \mathrm{ID}_{F_2} \| \mathrm{ID}_{F_3} \| u_{13} \| v_{13})$，其中 $u_{13} = \mathrm{E}(K_{F_2,F_3}, \mathrm{Ticket}_{T_1,F_1} \| \mathrm{TS}_7)$，$v_{13} = \mathrm{H}(\mathrm{ID}_{F_2} \| \mathrm{ID}_{F_3} \| u_{13})$。

(6) FN_3解密$Ticket_{T_1,F_1}$，验证TS_5是否在有效期限内，如果不在，向FN_2返回$Ticket_{T_1,F_1}$为无效信息；如果在有效期限内，FN_3生成时间戳TS_8，将解密得到的ID_{T_1}和R_1返回给FN_2。FN_3向FN_2发送辅助认证响应。

$FN_3 \to FN_2$: $\mathrm{E}(K_N, \mathrm{ID}_{F_3} \| \mathrm{ID}_{F_2} \| u_{14} \| v_{14})$，其中$u_{14} = \mathrm{E}(K_{F_2,F_3}, \mathrm{ID}_{T_1} \| R_1 \| \mathrm{TS}_5 \| \mathrm{TS}_7 \| \mathrm{TS}_8), v_{14} = \mathrm{H}(\mathrm{ID}_{F_3} \| \mathrm{ID}_{F_2} \| u_{14})$。

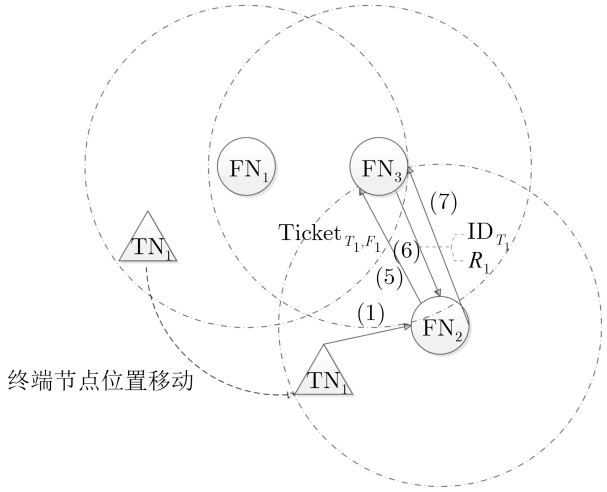

图 7 FN_2不是FN_1的邻居雾节点

(7) FN_2向FN_3发送确认响应。

$FN_2 \to FN_3$: $\mathrm{E}(K_N, \mathrm{ID}_{F_2} \| \mathrm{ID}_{F_3} \| u_{15} \| v_{15})$，其中，$u_{15} = \mathrm{E}(K_{F_2,F_3}, \mathrm{TS}_8)$，$v_{15} = \mathrm{H}(\mathrm{ID}_{F_2} \| \mathrm{ID}_{F_3} \| u_{15})$。

(8) FN_3解密得到TS_8，即可确认FN_2已经收到辅助认证响应，FN_3的辅助认证工作完成。

接下来继续执行情况1中的(2)(3)(4)，即可完成TN_1的切换重认证。

5 协议分析

5.1 安全性分析

5.1.1 切换重认证

初始雾节点对终端节点完成初始化认证后，向其发送认证Ticket，内容包括重认证会话密钥、终端节点ID、随机数R和标识认证Ticket有效期限的时间戳TS，并用初始雾节点的认证密钥加密；之后，当终端节点移动到新的雾节点附近时，发送认证Ticket给新雾节点，新雾节点只需验证该认证Ticket即可完成对终端节点的切换重认证。整个切换重认证过程没有初始雾节点的参与，且为下次认证做好了准备，生成了新的认证Ticket。

5.1.2 不可跟踪性

雾节点与终端节点完成初始化认证后，采用自身的认证密钥AK加密认证Ticket，然后发送给终端节点。终端节点发生切换重认证时，只需将认证Ticket发送给新雾节点，由新雾节点利用与原雾节点共享的认证密钥AK解密Ticket，从而实现对终端节点的重认证。假设重认证发生时终端节点已经超出了初始雾节点的认证范围，此种方式下新雾节点只知道终端节点是从哪里切换来的，却并不知道终端节点将向哪个雾节点进行切换，因此基本满足终端节点的不可跟踪性。

5.1.3 抵抗单点失效

雾节点失效可以分为两种情况：第1种是雾节

点自身原因；第2种是雾节点被攻破。当雾节点自身失效时，终端节点可以连接其他雾节点或者重新进行初始化认证。因此，我们只讨论雾节点被攻破的情况。假设当某个雾节点被攻破时，该雾节点的所有密钥信息被泄露。然而，每个雾节点仅保存自身和邻居雾节点的认证密钥，即使某个雾节点被攻破，影响的也仅是自身及其邻居雾节点，这有效预防了单点失效问题。

终端节点被攻破时，由于其只拥有雾节点发送的对应于自身的认证Ticket，所以也不会影响其他节点。即使攻击者拥有终端节点的认证Ticket，但是当新雾节点发现解密认证Ticket得到的终端节点ID与攻击者身份不符，就会拒绝向其提供服务。

5.2 性能分析

5.2.1 通信跳数

令t表示雾节点个数，假设终端节点只需通过一跳即可与雾节点进行通信。在终端节点初始化认证时，认证过程的通信跳数同传统方案相比如表1所示。

如表1所示，终端节点初始化认证阶段，Han方案[8,9]、Ibriq方案[10]和Fantacci方案[11]耗费的开销都较大。房帅磊论文[12]和本文协议在初始化认证过程中，终端节点通信跳数为2，雾节点通信跳数为1，BS节点不参与，总通信跳数为3，大大减少了初始化认证过程的系统开销，因此较传统认证方案有更好的性能。

在终端节点切换重认证阶段，认证过程的通信跳数同传统方案相比如表2所示。

如表2所示，在终端节点切换重认证阶段，Ibriq方案[10]、Fantacci方案[11]与终端节点初始化认证一样，需要较大的系统开销。Han方案[8,9]和房帅磊论文[12]的切换重认证过程无需BS参与，系统开销小，但Fang硕士论文[12]的方案增加了切换预认证过程，而且终端节点每次切换前需要预测移动位置，这大大增加了系统开销。本文协议通过增加认证Ticket，使得终端节点无需预测移动位置，减小了系统开销。

表 1　初始化认证通信跳数比较表

	Han方案[8,9]	Ibriq方案[10]	Fantacci方案[11]	房帅磊方案[12]	本文协议
TN	2	2	2	2	2
FN	$2t$	$2t$	$2t$+1	1	1
BS	1	2	–	–	–
系统总跳数	$2t$+3	$2t$+4	$2t$+3	3	3

表 2　切换重认证通信跳数比较表

	Han方案[8,9]	Ibriq方案[10]	Fantacci方案[11]	房帅磊论文[12]	本文协议
TN	2	2	2	1	2
FN	1(3)	$2t$	$2t$+1	1	1(4)
BS	–	2	–	–	–
系统总跳数	3(5)	$2t$+4	$2t$+3	2	3(6)

5.2.2 计算开销

本文协议采用非对称密码体制与对称密码体制相结合的方式，其中非对称密码采用基于ECC的TF-CPK密码技术。理论上讲，单次非对称密码加解密方式所消耗的能量必然大于对称加解密方式。然而，本文协议仅仅用非对称密码进行邻居雾节点发现和终端节点初始化认证的部分过程，切换重认证均采用认证Ticket的对称密码技术，同时终端节点无需预测移动位置，因此只需很小的计算开销。文中不再测量依赖于加密和哈希运算的计算时间，而是采用TinySEC[15]和TinyHash[16]的方法来实现本方案，本文初始化认证和切换重认证的计算开销比较表如表3所示。

5.2.3 存储开销

下面比较本文与文中所提方案节点所需存储空间。基于文献[8]，我们假设随机数占8 Byte，密钥占16 Byte，源ID和目的ID各占1 Byte。相比于其他方案，本文方案中雾节点所需存储空间变化不大；然而，终端节点所需存储空间大大减小，仅为Han方案[8,9]和Ibriq方案[10]的1/3左右；同时，由于没有基站节点的参与，所以基站存储空间为零。与房帅磊论文[12]相比，本文方案所有节点所需存储空间变化不大，但由于房帅磊论文[12]有切换预认证过程，在通信开销、计算开销和节点移动性方面均有所限制。本文与各方案节点所需存储空间比较表如表4所示。

6　结束语

本文针对当前雾计算环境下雾节点和终端节点的资源受限特性以及终端节点移动引起的切换重认证效率低等问题，提出了一种适用于雾计算环境的终端节点切换认证协议。该协议采用TF-CPK和认证Ticket相结合的方式，邻居雾节点发现以及终端节点初始化认证阶段采用TF-CPK的方式，减少了

表 3　初始化认证和切换重认证计算开销比较表

运算种类	初始化认证	切换重认证
对称加密/解密	3	2
非对称加密/解密	1	0
签名或验证	1	0
哈希运算	3	2

表 4 本文与各方案节点所需存储空间比较表(Byte)

	Han 方案[8,9]	Ibriq 方案[10]	Fantacci 方案[11]	房帅磊 论文[12]	本文协议
TN	56	68	18	19	17
FN	62	76	103	64	64
BS	92	180	0	0	0
总数	210	324	121	83	81

节点的信息存储量；终端节点切换重认证采用认证Ticket的对称密码体制，这提高了认证效率。通过安全性和性能分析表明，该协议相较传统的认证方案有明显的改善，能够很好地适用于资源受限的雾计算环境。

参 考 文 献

[1] 王颖, 王懿, 陈文瑛, 等. 一种面向分布式分析的雾计算架构及其在电网安全风险评估中的应用[J]. 自动化与仪器仪表, 2016(9): 128–130, 132. doi: 10.14016/j.cnki.1001-9227.2016.09.128.
WANG Ying, WANG Yi, CHEN Wenying, *et al*. A fog computing infrastructure for distributed analytics and its application in risk assessment to power grid[J]. *Automation & Instrumentation*, 2016(9): 128–130, 132. doi: 10.14016/j.cnki.1001-9227.2016.09.128.

[2] IBRAHIM M H. Octopus: An edge-fog mutual authentication scheme[J]. *International Journal of Network Security*, 2016, 18(6): 1089–1101.

[3] YI Shanhe, QIN Zhengrui, and Li Qun. Security and privacy issues of fog computing: A survey[C]. The 10th International Conference on Wireless Algorithms, Systems, and Applications, Qufu, China, 2015: 685–695. doi: 10.1007/978-3-319-21837-3_67.

[4] 张海波, 程妍, 刘开健, 等. 车联网中整合移动边缘计算与内容分发网络的移动性管理策略[J]. 电子与信息学报, 2020, 42(6): 1444–1451. doi: 10.11999/JEIT190571.
ZHANG Haibo, CHENG Yan, LIU Kaijian, *et al*. The mobility management strategies by integrating mobile edge computing and CDN in vehicular networks[J]. *Journal of Electronics & Information Technology*, 2020, 42(6): 1444–1451. doi: 10.11999/JEIT190571.

[5] MUKHERJEE M, MATAM R, SHU Lei, *et al*. Security and privacy in fog computing: Challenges[J]. *IEEE Access*, 2017, 5: 19293–19304. doi: 10.1109/ACCESS.2017.2749422.

[6] NI Jianbing, ZHANG Kuan, LIN Xiaodong, *et al*. Securing fog computing for internet of things applications: Challenges and solutions[J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(1): 601–628. doi: 10.1109/COMST.2017.2762345.

[7] 黄彬, 刘广钟, 徐明. 基于簇的无线传感器网络安全节点认证协议[J]. 计算机工程, 2016, 42(7): 117–122, 128. doi: 10.3969/j.issn.1000-3428.2016.07.020.
HUANG Bin, LIU Guangzhong, and XU Ming. Security authentication protocol for nodes in wireless sensor networks based on clusters[J]. *Computer Engineering*, 2016, 42(7): 117–122, 128. doi: 10.3969/j.issn.1000-3428.2016.07.020.

[8] HAN K, KIM K, and SHON T. Untraceable mobile node authentication in WSN[J]. *Sensors*, 2010, 10(5): 4410–4429. doi: 10.3390/s100504410.

[9] HAN K, SHON T, and KIM K. Efficient mobile sensor authentication in smart home and WPAN[J]. *IEEE Transactions on Consumer Electronics*, 2010, 56(2): 591–596. doi: 10.1109/TCE.2010.5505975.

[10] IBRIQ J and MAHGOUB I. A hierarchical key establishment scheme for wireless sensor networks[C]. The 21st International Conference on Advanced Information Networking and Applications, Niagara Falls, Canada, 2007: 210–219. doi: 10.1109/AINA.2007.14.

[11] FANTACCI R, CHITI F, and MACCARI L. Fast distributed bi-directional authentication for wireless sensor networks[J]. *Security and Communication Networks*, 2008, 1(1): 17–24. doi: 10.1002/sec.1.

[12] 房帅磊. 层次化无线传感器网络移动节点认证技术研究[D]. [硕士论文], 西安电子科技大学, 2011.
FANG Shuailei. Study on mobile nodes authentication in hierarchical wireless sensor networks[D]. [Master dissertation], Xidian University, 2011.

[13] 南湘浩. CPK密码体制与网际安全[M]. 北京: 国防工业出版社, 2008: 23–28.
NAN Xianghao. CPK-Crypotosystem and Cyber Security[M]. Beijing: National Defense Industry Press, 2008: 23–28.

[14] 张鑫, 杨晓元, 朱率率. 移动网络可信匿名认证协议[J]. 计算机应用, 2016, 36(8): 2231–2235. doi: 10.11772/j.issn.1001-9081.2016.08.2231.
ZHANG Xin, YANG Xiaoyuan, and ZHU Shuaishuai. Trusted and anonymous authentication protocol for mobile networks[J]. *Journal of Computer Applications*, 2016, 36(8): 2231–2235. doi: 10.11772/j.issn.1001-9081.2016.08.2231.

[15] KARLOF C, SASTRY N, and WAGNER D. TinySec: A link layer security architecture for wireless sensor networks[C]. The 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, USA, 2004: 3–5.

[16] LEE H R, CHOI Y J, and KIM H W. Implementation of TinyHash based on hash algorithm for sensor network[C]. The World Academy of Science, Engineering and Technology, Saint Louis, USA, 2005: 135–139.

胡荣磊：男，1977年生，博士，副研究员，研究方向为密码学与信息安全.

陈 雷：男，1992年生，博士生，研究方向为密码学与信息安全.

段晓毅：男，1979年生，博士，讲师，研究方向为密码学与信息安全.

于秉琪：男，1995年生，硕士生，研究方向为密码学与信息安全.

责任编辑：余 蓉