面向移动边缘计算的高效条件隐私保护认证密钥协商方案

李森森[1]，黄一才[1]，黄美根[2]，刘燕江[1]，郁滨[1]
（1.信息工程大学密码工程学院，河南 郑州 450001；2.国防科技大学系统工程学院，湖南 长沙 410073）

摘　要：针对现有面向移动边缘计算（MEC）的匿名认证密钥协商（AKA）方案存在无法追踪恶意匿名设备、无法防范物理攻击等问题，提出一种抗物理攻击的高效条件隐私保护 AKA 方案。通过将物理不可克隆函数（PUF）与椭圆曲线上的无证书密码体制相结合，并引入变色龙哈希函数，所提方案在实现对终端设备匿名性保护的前提下，提供对恶意匿名设备的追踪机制，通过聚合验证实现了批量认证功能。分析表明，方案无须使用高计算复杂度的双线性对运算且无密钥托管问题，能够在保持较低资源开销的同时，实现更高的安全性，满足 MEC 环境下设备的安全通信需求。

关键词：移动边缘计算；认证密钥协商；物理不可克隆函数；条件隐私保护；批量认证
中图分类号：TP309
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025161

Efficient conditional privacy-preserving authenticated key agreement scheme for mobile edge computing

LI Sensen[1], HUANG Yicai[1], HUANG Meigen[2], LIU Yanjiang[1], YU Bin[1]
1. College of Cryptography Engineering, Information Engineering University, Zhengzhou 450001, China
2. College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

Abstract: Aiming at the issues in existing anonymous authenticated key agreement (AKA) schemes for mobile edge computing (MEC), such as the inability to trace malicious anonymous devices and the lack of defense against physical attacks, an efficient conditional privacy-preserving AKA scheme that was robust against physical attacks was proposed. By combining physical unclonable function (PUF) with the certificateless public key cryptography on elliptic curves and introducing chameleon hash function, anonymity of terminal devices was preserved while a mechanism for tracing malicious anonymous devices were provided by the proposed scheme. Besides, batch authentication was further supported through aggregated verification. The analysis demonstrates that the proposed scheme eliminates the need for computationally intensive bilinear pairing operations and effectively circumvents the key escrow issue. It achieves enhanced security while maintaining low resource overhead, thereby better fulfilling the secure communication requirements of devices in MEC environment.

Keywords: mobile edge computing, authenticated key agreement, physical unclonable function, conditional privacy-preserving, batch authentication

收稿日期：2025–08–07；**修回日期：**2025–09–11
通信作者：李森森，lss589@163.com
基金项目：国家自然科学基金资助项目（No.62302519）；湖南省自然科学基金资助项目（No.2023JJ40677）
Foundation Items: The National Natural Science Foundation of China (No.62302519), The Natural Science Foundation of Hunan Province (No.2023JJ40677)

0 引言

随着物联网和移动通信技术的广泛应用，海量的移动终端接入网络，网络边缘的数据呈现爆炸式增长的趋势[1]。移动云计算（MCC, mobile cloud computing）能够依托云服务器为资源受限的移动终端提供计算卸载和数据外包存储服务。然而，在MCC网络架构下，终端设备采集到的数据需上传至远程云服务器，由于通信链路过长，该方式不可避免地存在数据处理时延长的问题[2]，无法满足无人驾驶、远程手术等时延敏感型应用的需求。针对该问题，欧洲电信标准协会提出了移动边缘计算（MEC, mobile edge computing），将云中心的计算能力下沉至网络边缘，通过边缘服务器在靠近终端设备的位置为其提供服务，能够有效缓解上述问题，在智慧医疗、智能驾驶、环境监测等领域有着广阔的应用前景[3]。然而，由于终端设备在开放的网络环境中与边缘服务器进行通信，易遭受窃听、篡改等恶意攻击，使MEC在数据安全、隐私保护等方面仍面临挑战[4]。

认证密钥协商（AKA, authenticated key agreement）能够实现通信实体之间的身份认证，并建立共享的会话密钥，为数据在公开信道上的安全传输提供了一种有效的解决文献[5]。在此基础上，匿名AKA进一步增强了安全性，通过隐藏通信实体的真实身份来保护其隐私信息[6]。近年来，已有研究学者提出了面向MEC的匿名AKA方案。文献[7]、文献[8]、文献[9]分别利用基于身份的密码体制设计匿名AKA方案，可在不泄露终端设备身份信息的前提下，实现终端设备与边缘服务器间的双向认证和会话密钥协商，但该类方案完全依赖于可信第三方为实体分配安全参数，存在密钥托管的问题。针对该问题，文献[10]基于无证书密码体制提出面向MEC的匿名AKA方案，该方案不存在密钥托管问题，但不能满足前向安全性的要求[11]。文献[12]设计了不需要可信第三方实时参与的认证方案，提高了系统的实现效率，但该方案使用了高计算复杂度的双线性对运算，不能满足资源受限设备的应用需求。文献[13]指出大多面向MEC的匿名AKA方案均采用长期静态密钥，无法满足不可链接性要求，并提出采用一次性身份标识和一次性公/私钥对的认证机制，但该方案仍需要使用高计算复杂度的双线性对运算。在此基础上，文献[14]通过抗篡改设备生成一次性身份标识，无须使用双线性对运算，降低了方案的资源开销，但该方案中终端设备的部分公钥参数始终保持恒定，边缘服务器能够通过该参数关联特定的终端设备。

尽管上述方案在一定程度上满足MEC环境下的基础安全通信需求，但在万物互联时代背景下，面对海量终端设备接入、高并发数据传输的挑战，以及金融、医疗等关键领域对通信安全的严苛要求，现有方案仍存在以下3个方面的局限性。

1) 在隐私保护方面，部分方案仅提供了弱匿名性保护机制，虽然敌手无法得到终端设备的身份信息，但边缘服务器仍可获取与其通信的终端设备的真实身份标识。而能够提供强匿名性保护的方案，却无法对发送虚假消息的恶意匿名设备进行有效追踪。因此，如何在保障终端设备身份隐私的同时，实现对恶意设备的身份追踪，仍是一个亟待解决的问题。

2) 上述匿名AKA方案仅提供一对一的认证机制，难以满足MEC网络中高移动性和高并发性场景的应用需求。一方面，为保证移动终端在多服务器间迁移过程的服务连贯性，并降低因频繁认证而产生的服务时延，需设计一对多认证机制，支持单个终端设备同时与多个边缘服务器进行身份合法性认证；另一方面，为缓解在终端密集部署场景下边缘服务器面临的认证压力，需提供多对一认证机制，允许多个终端设备同时被单个服务器并行认证。虽然文献[15]提出了一种支持广播通信的一对多身份认证机制，但该方案未考虑对终端设备的匿名性保护问题。因此，如何在实现设备隐私保护的前提下，构建高效的批量认证机制，仍有待进一步研究。

3) 上述方案均基于传统的密码体制设计，需要将敏感的密钥参数保存于设备存储器中。在MEC环境下，终端设备和边缘服务器常部署于无人值守的开放环境中，存在被敌手物理捕获的风险。若敌手通过实施物理攻击[16]获取设备存储器中的敏感密钥参数，将导致传统的认证密钥协商方案失效。因此，在设计匿名AKA方案时，需要充分考虑对物理攻击的防范，以确保系统的整体安全性。

物理不可克隆函数（PUF, physical unclonable function）[17]构建于集成电路制造过程中的随机性工艺偏差基础上，能够建立起输入挑战与输出响应

之间不可克隆的映射关系，可为资源受限设备抵抗物理攻击提供可行的解决文献[18-20]。

综上所述，本文将硬件安全原语PUF与椭圆曲线上的无证书密码体制相结合，并引入变色龙哈希函数，设计出抗物理攻击的高效条件隐私保护AKA方案。本文方案通过聚合验证机制实现了批量认证功能，能够以较低的资源开销完成终端设备与边缘服务器之间的双向认证及密钥协商，满足高移动性、高并发性MEC应用场景的安全需求。本文方案在实现对终端设备匿名性保护的同时，提供对恶意匿名设备的追踪机制，兼顾了隐私性和可监管性。

本文的主要贡献如下。

1) 为进一步提升MEC环境下认证与密钥协商的安全性和实现效率，提出基于PUF的高效条件隐私保护AKA方案，既可以高效地实现终端设备与边缘服务器间的一对一认证和密钥协商，又能够满足高移动性、高并发性场景下的批量认证需求。

2) 在方案设计过程中，引入变色龙哈希函数构建条件隐私保护机制，能够在系统检测到虚假消息时，通过协同验证实现对恶意匿名终端设备身份的追溯。

3) 在定义安全模型的基础上，对方案进行形式化安全证明和安全属性分析，表明本文方案在抵抗窃听、篡改、重放等传统安全威胁的同时，提供了对物理攻击的防范。

4) 与同类方案进行了性能对比和分析，结果表明，在高移动性、高并发性MEC应用场景下，本文方案能够以较低的资源开销实现更高的安全性。

1　网络结构与系统模型

1.1　网络结构

采用与文献[10]、文献[12]相似的MEC网络结构，如图1所示。网络中的实体主要包括管理服务器、边缘服务器和终端设备。其中，管理服务器为可信实体，主要负责系统初始化和参数配置，在设备注册过程中作为密钥生成中心（KGC, key generation center）为入网设备分配安全参数；边缘服务器在靠近终端设备的位置为其提供数据外包处理、计算卸载等服务；终端设备通常资源受限，并且通过公开信道与边缘服务器进行数据交互。此外，在部分应用场景中，边缘服务器和终端设备部署于无人值守的开放环境中，存在遭受物理攻击的风险。为此，本文引入硬件安全原语PUF，并作如下假设。

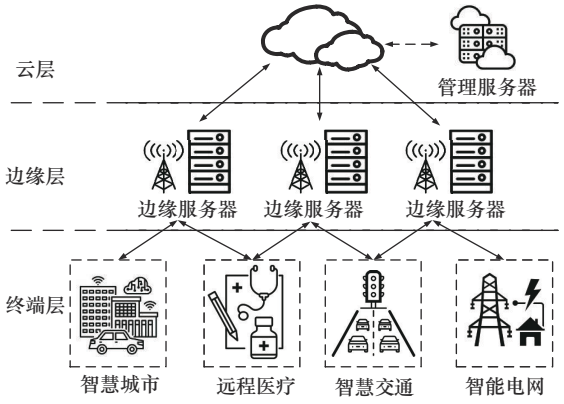


图1　MEC网络结构

1) 边缘服务器和终端设备均内置与其唯一绑定的PUF模块。

2) 设备控制器与其PUF模块间的通信不能被截获[20-21]。

3) 任何针对PUF的侵入式攻击和非法访问将改变其映射关系[20-21]。

1.2　威胁模型

在MEC网络中，终端设备与边缘服务器基于公开的网络信道进行通信，易遭受敌手的恶意攻击。Dolev-Yao模型[22]被广泛应用于定义密码方案中敌手的攻击能力，其指出敌手可以完全控制公开的网络信道，并通过窃听、篡改、重放等方式对密码方案发起攻击。考虑到MEC网络中终端设备和边缘服务器面临物理攻击威胁的现实风险，本文在经典Dolev-Yao模型的基础上进行了扩展，将物理攻击能力纳入敌手模型中。具体而言，本文考虑的威胁模型中敌手具备如下攻击能力。

1) 被动攻击能力。敌手可通过窃听方式获取公开信道中传递的消息，并对其进行分析。

2) 主动攻击能力。敌手可通过篡改、重放等方式干扰公开信道上的通信或注入恶意消息。

3) 物理攻击能力。敌手可物理捕获终端设备和边缘服务器，并得到其存储的全部参数。

1.3　安全模型

依据上述威胁模型，本文在学者Bellare、Pointcheval和Rogaway提出的安全模型[23]基础上进

行扩展，增加对敌手物理攻击能力的刻画。所构建的扩展安全模型的主要定义如下。

定义 1 参与者。方案$\mathcal{S}$的认证过程参与者包含终端设备和边缘服务器两类。用$\prod_{U}^{k}$表示参与者U的第k个会话实例，则终端设备TD_i和边缘服务器MS_j的第m和n个会话实例表示为$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$。

定义 2 接受状态。若会话实例$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$之间发送的所有消息均按序接收，则在最后一条有效消息接收后，$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$进入接受状态。

定义 3 伙伴关系。若会话实例$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$处于同一个会话并且均处于接受状态，则称$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$具备伙伴关系。

定义 4 新鲜性。若会话实例$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{m}$之间建立的会话密钥未被敌手$\mathcal{A}$获取，则称$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{n}$具有新鲜性。

定义 5 敌手。敌手$\mathcal{A}$可在多项式时间内执行如下预言机询问。

1) Hash(M)询问。通过执行该询问，敌手$\mathcal{A}$能够得到消息M对应的哈希值。

2) PUF(U,C)询问。通过执行该询问，敌手$\mathcal{A}$能够得到PUF_U（方案参与者U的PUF）对于挑战值C的响应R。为便于表述，在进行安全性分析时，将PUF和模糊提取器合并考虑，即在模糊提取器的辅助下，特定PUF对于同一挑战值始终产生一致的响应。

3) Execute($\prod_{\mathrm{TD}_i}^{m}$,$\prod_{\mathrm{MS}_j}^{n}$)询问。通过执行该询问，敌手$\mathcal{A}$能够得到会话实例$\prod_{\mathrm{TD}_i}^{m}$和$\prod_{\mathrm{MS}_j}^{m}$之间传递的全部消息。该询问用于刻画敌手的被动攻击能力。

4) Send($\prod_{U}^{k}$,M)询问。在该询问中，敌手$\mathcal{A}$选择消息M发送给会话实例$\prod_{U}^{k}$，并得到相应的响应。该询问用于刻画敌手的主动攻击能力。

5) Capture(U)询问。通过执行该询问，敌手$\mathcal{A}$能够得到方案参与者U（终端设备或边缘服务器）的设备存储器中保存的全部参数。该询问用于刻画敌手的物理攻击能力。

6) Corrupt(U)询问。通过执行该询问，敌手$\mathcal{A}$能够得到方案参与者U（终端设备或边缘服务器）的长期密钥。该询问不对应于敌手$\mathcal{A}$的特定攻击能力，而是用于分析方案的完美前向安全性。

7) Reveal($\prod_{U}^{k}$)询问。通过执行该询问，敌手$\mathcal{A}$能够得到会话实例$\prod_{U}^{k}$建立的会话密钥。该询问用于分析方案抵抗已知密钥攻击的能力。

8) Test($\prod_{U}^{k}$)询问。在该询问中，通过掷币游戏随机选取1比特数据coin。若coin = 1，则敌手$\mathcal{A}$得到具有新鲜性的会话实例$\prod_{U}^{k}$建立的真实会话密钥；否则，敌手$\mathcal{A}$得到与会话密钥等长的随机数。该询问用于刻画敌手$\mathcal{A}$获取到目标会话密钥的优势。

定义 6 会话密钥的语义安全性。在执行Test($\prod_{U}^{k}$)询问后，敌手$\mathcal{A}$输出对coin的猜测值coin′。若coin′ = coin，则$\mathcal{A}$在游戏中获胜，记作Succ($\mathcal{A}$)。敌手$\mathcal{A}$破坏方案$\mathcal{S}$的会话密钥语义安全性的优势可定义为$\mathrm{Adv}_{\mathcal{S}}(\mathcal{A}) = |2\Pr[\mathrm{Succ}(\mathcal{A})] - 1|$。若对于任意概率多项式时间敌手$\mathcal{A}$，均有$\mathrm{Adv}_{\mathcal{S}}(\mathcal{A}) \leqslant \varepsilon$，其中$\varepsilon$为可忽略的值，则称方案$\mathcal{S}$满足会话密钥的语义安全性。

1.4 设计目标

为满足MEC网络的应用需求，认证密钥协商方案应具备如下安全特性。

1) 双向认证。由于敌手可能伪造合法设备的身份来发送虚假信息，方案应实现通信双方（终端设备与边缘服务器）之间的身份合法性认证。

2) 安全会话密钥建立。为保证消息在公开信道上的安全传输，通信双方应建立起共享的会话密钥，并确保仅有方案参与方能够计算出该会话

密钥。

3) 无第三方实时参与。为适应MEC网络低时延、高效率的需求，认证密钥协商过程应由终端设备和边缘服务器直接交互实现，不需要在线可信第三方的实时参与。

4) 无密钥托管。终端设备和边缘服务器的密钥参数不应完全依赖于KGC生成，以防范可能发生的密钥泄露或密钥滥用风险。

5) 匿名性。为保证MEC网络中终端设备的隐私，任何实体，包括敌手、边缘服务器和KGC，均无法独立通过分析公开信道上传输的终端设备请求得到其真实身份信息。

6) 条件隐私。当检测到恶意终端设备发送的虚假消息时，通过边缘服务器与KGC协同，可以揭示该恶意终端设备的真实身份信息。

7) 不可链接性。敌手不能通过公开信道上传递的消息对特定终端设备的身份或请求进行链接，以防止其行为被敌手关联和分析。

8) 完美前向安全性。即使敌手获得了终端设备和边缘服务器的长期密钥，方案仍应保证前期建立的会话密钥的安全性。

9) 抵抗已知攻击。方案应提供对窃听、篡改、重放、物理攻击等已知攻击的防范。

2　方案设计

本文方案用到的基础知识包括椭圆曲线离散对数（ECDL, elliptic curve discrete logarithm）问题[14]、椭圆曲线计算性Diffie-Hellman（ECCDH, elliptic curve computational Diffie-Hellman）问题[14]、变色龙哈希函数[24]、PUF以及模糊提取器[25]。其中，PUF可表示为$\omega \leftarrow \mathrm{PUF}_i(c)$，其具备唯一性、稳定性、不可预测性、不可区分性等性质[2]；模糊提取器用于消除因环境扰动导致的PUF响应比特差异，其包含生成算法$(d,\mathrm{hd}) \leftarrow \mathrm{Gen}(\omega)$与恢复算法$d \leftarrow \mathrm{Rep}(\bar{\omega},\mathrm{hd})$。本文的认证密钥协商方案主要包括系统初始化、终端设备注册、边缘服务器注册、双向认证与密钥协商、批量认证以及恶意终端设备追踪等阶段。其中，系统初始化过程在系统建立时由KGC执行，注册过程在安全环境下进行，而认证与密钥协商过程则基于公开的网络信道实现。为便于表述，定义文中主要符号如表1所示。

表1　主要符号定义

符号	含义
$E_{a,b}$	椭圆曲线
k_M	系统主密钥
P_{Pub}	系统公钥
tid_i	终端设备TD_i的身份标识
sid_j	边缘服务器MS_j的身份标识
$<\mathrm{PK}_i = X_i,\mathrm{sk}_i = x_i>$	终端设备TD_i的公/私钥
$<\mathrm{PK}_j = (X_j,R_j),\mathrm{sk}_j = (x_j,y_j)>$	边缘服务器MS_j的公/私钥
pid_i	终端设备TD_i的临时身份标识
X_i^*	终端设备TD_i的临时公钥
t_i,t_j,t_j'	时间戳
$[T]$	集合$\{1,2,\cdots,T\}$
$\mathrm{PUF}_i,\mathrm{PUF}_j$	TD_i和MS_j的PUF
$H_i(i=1,2,3)$	哈希函数
$\mathrm{CH}_X(\alpha,\beta)$	变色龙哈希函数

2.1　系统初始化

该阶段，KGC选取系统初始化参数，具体过程如下。

KGC选取安全的椭圆曲线$E_{a,b}$以及阶为大素数p、生成元$P \in E_{a,b}$的循环群G。在此基础上，随机选取系统主密钥$k_M \in Z_p^*$，计算相应公钥$P_{\mathrm{Pub}} = k_M P$，并选择安全的哈希函数$H_0:\{0,1\}^* \rightarrow Z_p^*$、$H_1:\{0,1\}^* \times G \rightarrow Z_p^*$、$H_2:\{0,1\}^* \times G \times G \rightarrow Z_p^*$、$H_3:\{0,1\}^* \times G \times G \times G \times G \rightarrow Z_p^*$。最后，KGC秘密保存系统主密钥$\mathrm{msk} = k_M$，并公开参数$\{E_{a,b},p,P,G,P_{\mathrm{Pub}},H_0,H_1,H_2,H_3\}$。

2.2　终端设备注册

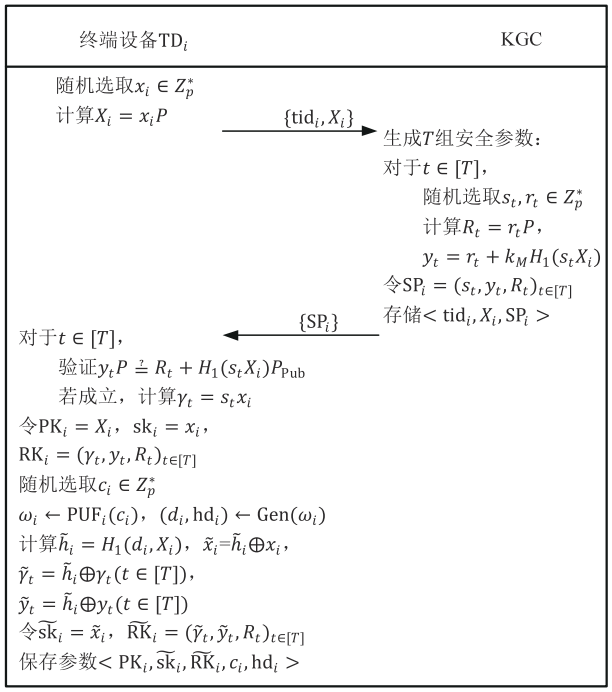
终端设备在入网前通过注册过程获得安全参数，该过程由终端设备与KGC交互实现，具体过程如图2所示。

1) 身份标识为tid_i的终端设备TD_i随机选取$x_i \in Z_p^*$，计算$X_i = x_i P$，并向KGC提交$<\mathrm{tid}_i,X_i>$。

2) KGC收到终端设备TD_i的注册请求后，为其生成T组安全参数。对于$t \in [T]$，KGC随机选取$s_t,r_t \in Z_p^*$，并计算$R_t = r_t P$和$y_t = r_t + k_M H_1(s_t X_i)$。然后，KGC令$\mathrm{SP}_i = (s_t,y_t,R_t)_{t\in[T]}$，在其参数列表中存储$<\mathrm{tid}_i,X_i,\mathrm{SP}_i>$，并向$\mathrm{TD}_i$回复$<\mathrm{SP}_i>$。

3) TD_i收到回复后，首先验证安全参数$\mathrm{SP}_i =$

$(s_t,y_t,R_t)_{t\in[T]}$的有效性，具体过程如下。对于$t\in[T]$，TD_i验证$y_tP=R_t+H_1(s_tX_i)P_{\mathrm{Pub}}$是否成立。若成立，则计算$\gamma_t=s_tx_i$，并继续执行；否则，验证失败，$\mathrm{TD}_i$终止操作。该验证过程的正确性由等式$y_tP=[r_t+k_MH_1(s_tX_i)]P=R_t+H_1(s_tX_i)P_{\mathrm{Pub}}$可知。至此，$\mathrm{TD}_i$得到其公钥$\mathrm{PK}_i=X_i$和私钥$\mathrm{sk}_i=x_i$，并将$\mathrm{RK}_i=(\gamma_t,y_t,R_t)_{t\in[T]}$作为其随机密钥。



在此基础上，TD_i随机选取$c_i\in Z_p^*$，利用其PUF和模糊提取器得到$\omega_i\leftarrow\mathrm{PUF}_i(c_i)$，$(d_i,\mathrm{hd}_i)\leftarrow\mathrm{Gen}(\omega_i)$。然后，$\mathrm{TD}_i$计算$\tilde{h}_i=H_1(d_i,X_i)$，$\tilde{x}_i=\tilde{h}_i\oplus x_i$，$\tilde{\gamma}_t=\tilde{h}_i\oplus\gamma_t(t\in[T])$以及$\tilde{y}_t=\tilde{h}_i\oplus y_t(t\in[T])$，令$\widetilde{\mathrm{sk}}_i=\tilde{x}_i$，$\widetilde{\mathrm{RK}}_i=\left(\tilde{\gamma}_t,\tilde{y}_t,R_t\right)_{t\in[T]}$，并保存参数$<\mathrm{PK}_i,\widetilde{\mathrm{sk}}_i,\widetilde{\mathrm{RK}}_i,c_i,\mathrm{hd}_i>$。

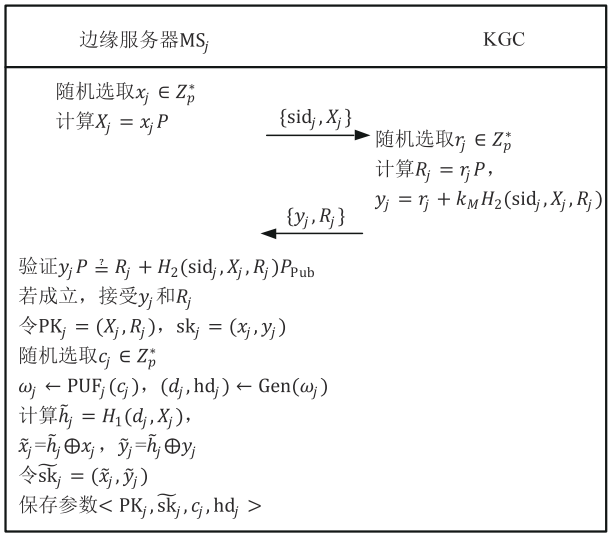
2.3　边缘服务器注册

与终端设备相同，边缘服务器在入网前也要通过注册过程获得安全参数，具体过程如图3所示。

1) 身份标识为sid_j的边缘服务器MS_j随机选取$x_j\in Z_p^*$，计算$X_j=x_jP$，并向KGC提交$<\mathrm{sid}_j,X_j>$。

2) KGC收到边缘服务器MS_j的注册请求后，随机选取$r_j\in Z_p^*$，分别计算$R_j=r_jP$，$y_j=r_j+k_MH_2(\mathrm{sid}_j,X_j,R_j)$，并向$\mathrm{MS}_j$回复$<y_j,R_j>$。

3) MS_j收到后，验证$y_jP=R_j+H_2(\mathrm{sid}_j,X_j,R_j)P_{\mathrm{Pub}}$是否成立。若成立，则继续执行；否则，验证失败，终止操作。该过程的正确性由$y_jP=[r_j+k_MH_2(\mathrm{sid}_j,X_j,R_j)]P=R_j+H_2(\mathrm{sid}_j,X_j,R_j)P_{\mathrm{Pub}}$可知。至此，边缘服务器$\mathrm{MS}_j$得到其公钥$\mathrm{PK}_j=(X_j,R_j)$和私钥$\mathrm{sk}_j=(x_j,y_j)$。



在此基础上，MS_j随机选取$c_j\in Z_p^*$，利用其PUF和模糊提取器得到$\omega_j\leftarrow\mathrm{PUF}_j(c_j)$，$(d_j,\mathrm{hd}_j)\leftarrow\mathrm{Gen}(\omega_j)$。然后，$\mathrm{MS}_j$计算$\tilde{h}_j=H_1(d_j,X_j)$，$\tilde{x}_j=\tilde{h}_j\oplus x_j$以及$\tilde{y}_j=\tilde{h}_j\oplus y_j$，令$\widetilde{\mathrm{sk}}_j=(\tilde{x}_j,\tilde{y}_j)$，并保存参数$<\mathrm{PK}_j,\widetilde{\mathrm{sk}}_j,c_j,\mathrm{hd}_j>$。

2.4　双向认证与密钥协商

为保证通信过程的安全性，终端设备在每次向边缘服务器请求服务之前，两者需执行双向认证与密钥协商过程，以实现身份合法性认证与会话密钥的建立。边缘服务器MS_j间歇性地通过广播信道发送其参数信息$<t_j,\mathrm{sid}_j,X_j,R_j>$，其中，$t_j$为$\mathrm{MS}_j$获取的当前时间戳。当终端设备$\mathrm{TD}_i$进入到$\mathrm{MS}_j$的覆盖区域时，两者执行双向认证与密钥协商过程，具体过程如图4所示。

1) 认证密钥协商发起

终端设备TD_i执行如下操作发起认证密钥协商。

①利用其PUF和模糊提取器得到$\bar{\omega}_i\leftarrow\mathrm{PUF}_i(c_i)$，$d_i\leftarrow\mathrm{Rep}(\bar{\omega}_i,\mathrm{hd}_i)$，并计算$\tilde{h}_i=H_1(d_i,X_i)$与$x_i=\tilde{h}_i\oplus\tilde{x}_i$。从$\widetilde{\mathrm{RK}}_i$中随机选取一组安全参数，记作$(\tilde{\gamma}_i,\tilde{y}_i,R_i)$，并计算$\gamma_i=\tilde{h}_i\oplus\tilde{\gamma}_i$与$y_i=\tilde{h}_i\oplus\tilde{y}_i$。

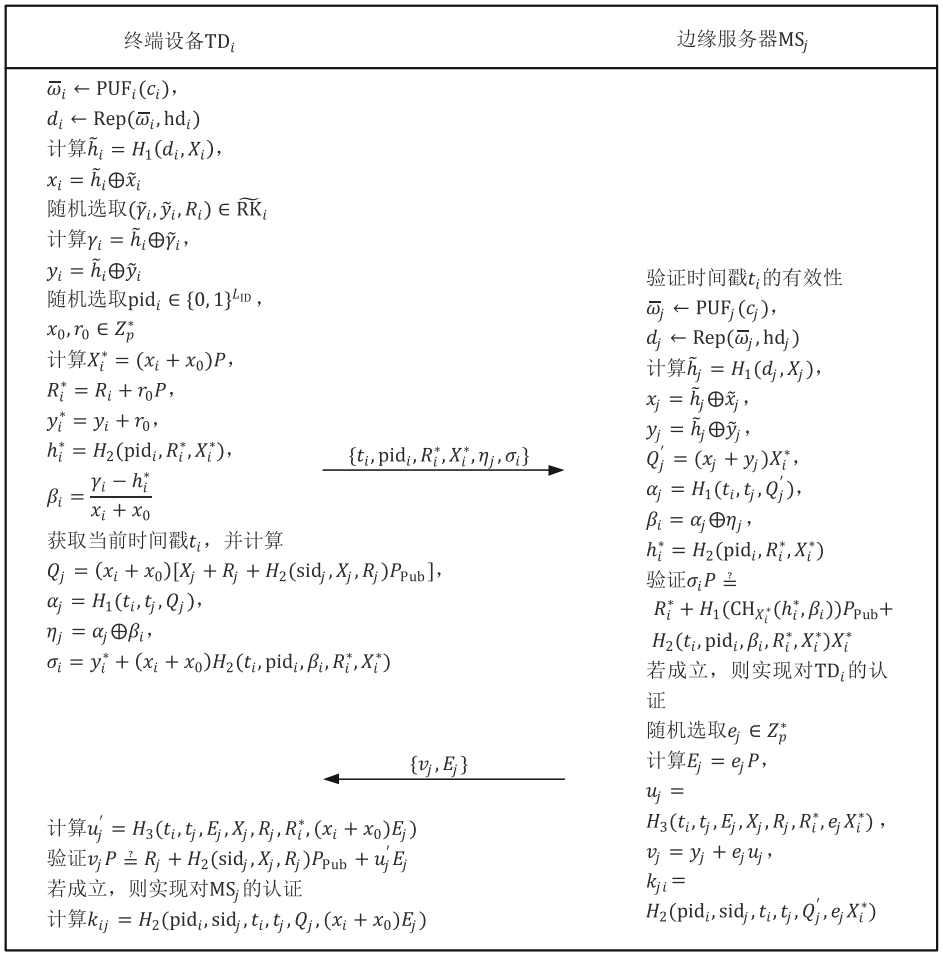


图4　双向认证与密钥协商过程

②随机选取伪身份标识$\mathrm{pid}_i \in \{0,1\}^{L_{\mathrm{ID}}}$以及随机值$x_0, r_0 \in Z_p^*$，计算临时公钥$X_i^* = (x_i + x_0)P$以及临时参数$R_i^* = R_i + r_0 P$，$y_i^* = y_i + r_0$，$h_i^* = H_2(\mathrm{pid}_i, R_i^*, X_i^*)$和$\beta_i = \dfrac{\gamma_i - h_i^*}{x_i + x_0}$，并获取当前时间戳$t_i$。

③分别计算$Q_j = (x_i + x_0)[X_j + R_j + H_2(\mathrm{sid}_j, X_j, R_j)P_{\mathrm{Pub}}]$，$\alpha_j = H_1(t_i, t_j, Q_j)$，$\eta_j = \alpha_j \oplus \beta_i$以及$\sigma_i = y_i^* + (x_i + x_0)H_2(t_i, \mathrm{pid}_i, \beta_i, R_i^*, X_i^*)$，并将$< t_i, \mathrm{pid}_i, R_i^*, X_i^*, \eta_j, \sigma_i >$发送给MS$_j$作为认证请求。

2) 边缘服务器认证终端设备

边缘服务器MS$_j$收到认证请求后，执行如下操作。

①获取当前时间戳t_j'，计算$\Delta t = |t_j' - t_i|$，并判断Δt是否在可接受的时间范围内。若可接受，则继续执行后续操作；否则，返回消息超时。

②利用其PUF和模糊提取器得到$\bar{\omega}_j \leftarrow \mathrm{PUF}_j(c_j)$，$d_j \leftarrow \mathrm{Rep}(\bar{\omega}_j, \mathrm{hd}_j)$，并分别计算$\tilde{h}_j = H_1(d_j, X_j)$，$x_j = \tilde{h}_j \oplus \tilde{x}_j$以及$y_j = \tilde{h}_j \oplus \tilde{y}_j$。然后，计算$Q_j' = (x_j + y_j)X_i^*$，$\alpha_j = H_1(t_i, t_j, Q_j')$，$\beta_i = \alpha_j \oplus \eta_j$以及$h_i^* = H_2(\mathrm{pid}_i, R_i^*, X_i^*)$。特别地，参数$Q_j'$的正确性由等式$Q_j' = (x_j + y_j)X_i^* = (x_j + y_j)(x_i + x_0)P = (x_i + x_0)[X_j + R_j + H_2(\mathrm{sid}_j, X_j, R_j)P_{\mathrm{Pub}}] = Q_j$可知。

③验证等式$\sigma_i P = R_i^* + H_1(\mathrm{CH}_{X_i^*}(h_i^*, \beta_i))P_{\mathrm{Pub}} + H_2(t_i, \mathrm{pid}_i, \beta_i, R_i^*, X_i^*)X_i^*$是否成立，其中$\mathrm{CH}_{X_i^*}$为变色龙哈希函数。若该等式成立，则MS$_j$实现对TD$_i$的合法性认证，并继续执行后续操作；否则，MS$_j$终止操作。特别地，由于$\mathrm{CH}_{X_i^*}(h_i^*, \beta_i) = h_i^* P + \beta_i X_i^* = h_i^* P + \dfrac{\gamma_i - h_i^*}{x_i + x_0}(x_i + x_0)P = \gamma_i P = s_i x_i P = s_i X_i$且$\sigma_i = y_i^* + (x_i + x_0)H_2(t_i, \mathrm{pid}_i, \beta_i, R_i^*, X_i^*)$，上述合法性认证过程的正确性如式(1)所示。

$\sigma_i P = [y_i^* + (x_i + x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)]P = (y_i + r_0)P + H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)(x_i + x_0)P = R_i + H_1(s_i X_i)P_{\mathrm{Pub}} + r_0 P + H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)X_i^* = R_i^* + H_1(\mathrm{CH}_{X_i^*}(h_i^*,\beta_i))P_{\mathrm{Pub}} + H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)X_i^*$ (1)

④ 随机选取 $e_j \in Z_p^*$，计算 $E_j = e_j P$，$u_j = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_j X_i^*)$，$v_j = y_j + e_j u_j$ 以及会话密钥 $k_{ji} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_j X_i^*)$，并向 TD_i 回复消息 $< v_j,E_j >$。

3) 终端设备认证边缘服务器

终端设备 TD_i 收到响应后，首先计算 $u_j' = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,(x_i + x_0)E_j)$，其正确性由等式 $u_j' = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,(x_i + x_0)E_j) = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,(x_i + x_0)e_j P) = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_j X_i^*) = u_j$ 可知。然后，验证等式 $v_j P = R_j + H_2\left(\mathrm{sid}_j,X_j,R_j\right)P_{\mathrm{Pub}} + u_j' E_j$ 是否成立。若成立，则 TD_i 实现对边缘服务器 MS_j 的合法性认证，并计算会话密钥 $k_{ij} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i + x_0)E_j)$；否则，$\mathrm{TD}_i$ 终止操作。该验证过程的正确性由等式 $v_j P = (y_j + e_j u_j)P = R_j + H_2(\mathrm{sid}_j,X_j,R_j)P_{\mathrm{Pub}} + u_j' E_j$ 可知。

在认证过程中，MS_j 和 TD_i 分别计算出密钥 k_{ji} 和 k_{ij}，由于 $k_{ij} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i + x_0)E_j) = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,e_j(x_i + x_0)P) = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,e_j X_i^*) = k_{ji}$，可知双方建立起了共享的会话密钥，利用该会话密钥，TD_i 和 MS_j 可进行安全通信。在安全通信过程完成后，双方均删除在双向认证与密钥协商过程中产生的全部临时参数，即双方仅需存储其通过注册过程得到的安全参数。

2.5 批量认证

为提升高移动性、高并发性场景下认证与密钥协商的效率，通过聚合验证机制实现多终端设备与多边缘服务器间的批量认证，具体过程如下。

1) 批量认证密钥协商发起

当终端设备 TD_i 收到多个边缘服务器通过广播信道发送的参数信息时，其可根据自身的移动路径选取多个边缘服务器 $\mathrm{MS}_j(j \in [J])$，并同时向它们发送认证密钥协商请求，以保证其在多服务器间迁移过程中的服务连贯性。具体而言，TD_i 执行如下操作。

① 利用其 PUF 和模糊提取器得到 $\bar{\omega}_i \leftarrow \mathrm{PUF}_i(c_i)$，$d_i \leftarrow \mathrm{Rep}(\bar{\omega}_i,\mathrm{hd}_i)$，并计算 $\tilde{h}_i = H_1(d_i,X_i)$，$x_i = \tilde{h}_i \oplus \tilde{x}_i$。从 $\widetilde{\mathrm{RK}}_i$ 中随机选取一组安全参数，记作 $(\tilde{\gamma}_i,\tilde{y}_i,R_i)$，计算 $\gamma_i = \tilde{h}_i \oplus \tilde{\gamma}_i$ 与 $y_i = \tilde{h}_i \oplus \tilde{y}_i$。

② 随机选取伪身份标识 $\mathrm{pid}_i \in \{0,1\}^{L_{\mathrm{ID}}}$ 以及随机值 $x_0,r_0 \in Z_p^*$，计算临时公钥 $X_i^* = (x_i + x_0)P$ 以及临时参数 $R_i^* = R_i + r_0 P$，$y_i^* = y_i + r_0$，$h_i^* = H_2(\mathrm{pid}_i,R_i^*,X_i^*)$ 和 $\beta_i = \dfrac{\gamma_i - h_i^*}{x_i + x_0}$，并获取当前时间戳 t_i。

③对于边缘服务器 $\mathrm{MS}_j(j \in [J])$，分别计算 $Q_j = (x_i + x_0)[X_j + R_j + H_2(\mathrm{sid}_j,X_j,R_j)P_{\mathrm{Pub}}]$ 和 $\alpha_j = H_1(t_i,t_j,Q_j)$。

④ 构造 $f(x) = \prod_{j=1}^{J}(x - \alpha_j) + \beta_i (\mathrm{mod}\, p) = x^J + a_{J-1}x^{J-1} + \cdots + a_1 x + a_0$，并令 $A_i = \{a_0,a_1,\cdots,a_{J-1}\}$。然后，计算 $\sigma_i = y_i^* + (x_i + x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)$，并将 $< t_i,\mathrm{pid}_i,R_i^*,X_i^*,A_i,\sigma_i >$ 通过广播信道发送给 $\mathrm{MS}_j(j \in [J])$ 作为认证请求。

2) 边缘服务器批量认证终端设备

边缘服务器 MS_j 收到多个终端设备 $\mathrm{TD}_i(i \in [I])$ 发来的认证请求后，执行如下批量认证操作。

①利用其 PUF 和模糊提取器得到 $\bar{\omega}_j \leftarrow \mathrm{PUF}_j(c_j)$，$d_j \leftarrow \mathrm{Rep}(\bar{\omega}_j,\mathrm{hd}_j)$，并分别计算 $\tilde{h}_j = H_1(d_j,X_j)$，$x_j = \tilde{h}_j \oplus \tilde{x}_j$ 以及 $y_j = \tilde{h}_j \oplus \tilde{y}_j$。

②对于每个请求 $< t_i,\mathrm{pid}_i,R_i^*,X_i^*,A_i,\sigma_i >$，首先验证其时间戳的有效性。若有效，则继续执行后续操作；否则，将该请求删除。然后，计算 $Q_j' = (x_j + y_j)X_i^*$，$\alpha_j = H_1(t_i,t_j,Q_j')$ 和 $h_i^* = H_2(\mathrm{pid}_i,R_i^*,X_i^*)$，并利用参数 $A_i = \{a_0,a_1,\cdots,a_{J-1}\}$ 重构 $f(x) = x^J + a_{J-1}x^{J-1} + \cdots + a_1 x + a_0$，进而计算 $\beta_i = f(\alpha_j)$。特别地，参数 Q_j' 的正确性由等式 $Q_j' = (x_j + y_j)X_i^* = (x_j + y_j)(x_i + x_0)P = (x_i + x_0)[X_j + R_j + H_2(\mathrm{sid}_j,X_j,R_j)P_{\mathrm{Pub}}] = Q_j$ 可知。

③验证等式 $(\sum_{i=1}^{I}\sigma_i)P = \sum_{i=1}^{I}R_i^* + [\sum_{i=1}^{I}H_1(\mathrm{CH}_{X_i^*}(h_i^*,\beta_i))]P_{\mathrm{Pub}} + \sum_{i=1}^{I}[H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)X_i^*]$ 是否成立。若成立，则边缘服务器 MS_j 实现对多个终端设备 $\mathrm{TD}_i(i \in [I])$ 的批量认证，并继续执行后续操作；否则，批量认证失败。由于 $\mathrm{CH}_{X_i^*}(h_i^*,\beta_i) =$

$h_i^* P + \beta_i X_i^* = h_i^* P + \frac{\gamma_i - h_i^*}{x_i + x_0}(x_i + x_0)P = \gamma_i P = s_i x_i P = s_i X_i$ 且 $\sigma_i = y_i^* + (x_i + x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)$，上述批量认证过程的正确性如式(2)所示。

$$\left(\sum_{i=1}^{I}\sigma_i\right)P = \left[\sum_{i=1}^{I}(y_i^* + (x_i + x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*))\right]\times$$

$$P = \sum_{i=1}^{I}(y_i + r_0)P + \sum_{i=1}^{I}(x_i + x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)\times$$

$$P = \sum_{i=1}^{I}[R_i + H_1(s_i X_i)P_{\mathrm{Pub}} + r_0 P] +$$

$$\sum_{i=1}^{I}[H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)X_i^*] =$$

$$\sum_{i=1}^{I}R_i^* + \left[\sum_{i=1}^{I}H_1(\mathrm{CH}_{X_i^*}(h_i^*,\beta_i))\right]P_{\mathrm{Pub}} +$$

$$\sum_{i=1}^{I}[H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)X_i^*] \quad (2)$$

④随机选取 $e_j \in Z_p^*$，计算 $E_j = e_j P$。对于 $\mathrm{TD}_i(i \in [I])$，计算 $u_i = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_j X_i^*)$。

⑤随机选取 $\xi_j \in Z_p^*$，构造 $g(x) = \prod_{i=1}^{I}(x - u_i) + \xi_j (\mathrm{mod}\, p) = x^I + b_{I-1}x^{I-1} + \cdots + b_1 x + b_0$，计算 $v_j = y_j + e_j\xi_j$，并令 $B_j = \{b_0,b_1,\cdots,b_{I-1}\}$。对于终端设备 $\mathrm{TD}_i(i \in [I])$，计算会话密钥 $k_{ji} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_j X_i^*)$，并将 $< v_j,B_j,E_j >$ 通过广播信道发送给 $\mathrm{TD}_i(i \in [I])$ 作为响应。

3) 终端设备批量认证边缘服务器

终端设备 TD_i 收到多个边缘服务器 $\mathrm{MS}_j(j \in [J])$ 的认证响应后，执行如下操作。

①对于每个认证响应 $< v_j,B_j,E_j >$，计算 $u_i' = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,(x_i + x_0)E_j)$，并利用参数 $B_j = \{b_0,b_1,\cdots,b_{I-1}\}$ 重构 $g(x) = x^I + b_{I-1}x^{I-1} + \cdots + b_1 x + b_0$，进而计算 $\xi_j = g(u_i')$。特别地，u_i' 的正确性由等式 $u_i' = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,(x_i + x_0)E_j) = H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_j X_i^*) = u_i$ 可知。

②验证等式 $\left(\sum_{j=1}^{J}v_j\right)P = \sum_{j=1}^{J}R_j + [\sum_{j=1}^{J}H_2(\mathrm{sid}_j,X_j,R_j)]P_{\mathrm{Pub}} + \sum_{j=1}^{J}\xi_j E_j$ 是否成立。若成立，则 TD_i 实现对多个边缘服务器 $\mathrm{MS}_j(j \in [J])$ 的批量认证，并继续执行；否则，批量认证失败。该验证过程的正确性由等式 $\left(\sum_{j=1}^{J}v_j\right)P = \left[\sum_{j=1}^{J}(y_j + e_j\xi_j)\right]P = \sum_{j=1}^{J}y_j P + \sum_{j=1}^{J}e_j\xi_j P = \sum_{j=1}^{J}R_j + \left[\sum_{j=1}^{J}H_2(\mathrm{sid}_j,X_j,R_j)\right]P_{\mathrm{Pub}} + \sum_{j=1}^{J}\xi_j E_j$ 可知。

③对于边缘服务器 $\mathrm{MS}_j(j \in [J])$，计算会话密钥 $k_{ij} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i + x_0)E_j)$。

在批量认证过程中，对于终端设备 TD_i，边缘服务器 MS_j 计算会话密钥 k_{ji}；对于 MS_j，TD_i 计算会话密钥 k_{ij}。由于 $k_{ij} = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i + x_0)E_j) = H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_j X_i^*) = k_{ji}$，可知边缘服务器和终端设备间建立了两两共享的会话密钥。

上述批量认证过程描述了多个终端设备和多个边缘服务器进行认证和密钥协商的一般情况。特别地，当终端设备数量 $I = 1$ 时，方案转化为一对多的认证过程，支持单个终端设备同时与多个边缘服务器进行身份合法性认证，可保证移动终端在多服务器间迁移时的服务连贯性，降低因频繁认证而产生的服务时延；当边缘服务器数量 $J = 1$ 时，方案转化为多对一的认证过程，允许多个终端设备同时被单个服务器并行认证，以缓解在终端密集部署场景下边缘服务器面临的认证压力。

2.6　恶意终端设备追踪

对恶意终端设备身份信息的追踪过程由边缘服务器与KGC联合执行，具体操作如下。

1) 边缘服务器将认证过程中得到的参数 $< X_i^*,h_i^*,\beta_i >$ 提交给KGC。

2) KGC通过遍历其存储的终端设备参数列表，查找使等式 $\mathrm{CH}_{X_i^*}(h_i^*,\beta_i) = s_i X_i$ 成立的参数 $< \mathrm{tid}_i,X_i,\mathrm{SP}_i >$ 以及 $(s_i,y_i,R_i) \in \mathrm{SP}_i$，进而得到对应的终端设备真实身份标识 tid_i。

由 $\gamma_i = s_i x_i$，$h_i^* = H_2(\mathrm{pid}_i,R_i^*,X_i^*)$ 以及 $\beta_i = \frac{\gamma_i - h_i^*}{x_i + x_0}$ 可知，$\mathrm{CH}_{X_i^*}(h_i^*,\beta_i) = h_i^* P + \beta_i X_i^* = h_i^* P + \frac{\gamma_i - h_i^*}{x_i + x_0}(x_i + x_0)P = \gamma_i P = s_i x_i P = s_i X_i$。因此，当遍历得到相应参数使等式 $\mathrm{CH}_{X_i^*}(h_i^*,\beta_i) = s_i X_i$ 成立时，KGC可以确认当前认证参数由对应的终端设备产生，从而实现对恶意设备的追踪。

由上述认证过程和恶意设备追踪过程可知，作为陷门拥有者，终端设备可借助变色龙哈希函数的陷门碰撞特性，在每次认证过程中生成新的临时身份标识和临时参数，且所生成的身份标识和参数能

够通过边缘服务器的合法性验证。当系统检测到虚假消息时，通过边缘服务器与KGC的协同，可追溯发送该虚假消息的恶意终端设备真实身份，从而实现条件隐私保护。

2.7 终端设备参数更新

依据参数使用时长、网络状态等信息，终端设备可通过与KGC交互实现其安全参数的动态更新。该过程可在线完成，具体操作如下。

1) 终端设备 TD_i 随机选取 $x_i' \in Z_p^*$，计算 $X_i' = x_i' P$，$\varsigma_i = H_2(X_i', x_i' P_{\mathrm{Pub}})$ 以及 $\mathrm{pid}_i = \mathrm{tid}_i \oplus H_0(\varsigma_i)$，并向KGC提交 $< \mathrm{pid}_i, X_i' >$。

2) KGC收到终端设备 TD_i 的参数更新请求后，计算 $\varsigma_i' = H_2(X_i', k_M X_i')$ 以及 $\mathrm{tid}_i = \mathrm{pid}_i \oplus H_0(\varsigma_i')$，从而得到该终端设备的身份标识 tid_i，并通过遍历终端设备参数列表得到相应的参数 $< \mathrm{tid}_i, X_i, \mathrm{SP}_i >$。在此基础上，KGC随机选取 $\phi_i \in Z_p^*$，计算 $\Phi_i = \phi_i P$ 以及 $\zeta_i = H_3(\mathrm{tid}_i, X_i, X_i', \Phi_i, (k_M + \phi_i) X_i', \phi_i X_i)$，并向 TD_i 回复 $< \zeta_i, \Phi_i >$。

3) 终端设备 TD_i 收到响应后，利用其PUF和模糊提取器得到 $\bar{\omega}_i' \leftarrow \mathrm{PUF}_i(c_i)$，$d_i \leftarrow \mathrm{Rep}(\bar{\omega}_i', \mathrm{hd}_i)$，并计算 $\tilde{h}_i = H_1(d_i, X_i)$ 与 $x_i = \tilde{h}_i \oplus \tilde{x}_i$。然后，验证等式 $\zeta_i = H_3(\mathrm{tid}_i, X_i, X_i', \Phi_i, x_i'(P_{\mathrm{Pub}} + \Phi_i), x_i \Phi_i)$ 是否成立。若成立，则 TD_i 实现对KGC的身份认证，并继续执行后续操作；否则，认证失败，终止操作。

在此基础上，终端设备 TD_i 计算 $\xi_i = H_2(\mathrm{tid}_i, x_i' \Phi_i, x_i P_{\mathrm{Pub}})$，并向KGC回复 $< \xi_i >$。

4) KGC收到响应后，首先验证 $\xi_i = H_2(\mathrm{tid}_i, \phi_i X_i', k_M X_i)$ 是否成立。若成立，则KGC实现对 TD_i 的身份认证，并继续执行后续操作；否则，认证失败，终止操作。

在此基础上，KGC为终端设备生成 T 组新的安全参数。对于 $t \in [T]$，KGC随机选取 $s_t', r_t' \in Z_p^*$，计算 $R_t' = r_t' P$，$y_t' = r_t' + k_M H_1(s_t' X_i')$，$\widehat{s_t'} = s_t' \oplus H_2 \times (\mathrm{tid}_i, \phi_i X_i', \phi_i X_i)$，$\widehat{y_t'} = y_t' \oplus H_2(\mathrm{tid}_i, \phi_i X_i', \phi_i X_i)$，$\widehat{R_t'} = R_t' + X_i + X_i'$，并令 $\mathrm{SP}_i' = (s_t', y_t', R_t')_{t \in [T]}$ 且 $\widehat{\mathrm{SP}_i'} = (\widehat{s_t'}, \widehat{y_t'}, \widehat{R_t'})_{t \in [T]}$。然后，KGC将其参数列表中的参数 $< \mathrm{tid}_i, X_i, \mathrm{SP}_i >$ 更新为 $< \mathrm{tid}_i, X_i', \mathrm{SP}_i' >$，并向 TD_i 回复 $< \widehat{\mathrm{SP}_i'} >$。

5) 终端设备 TD_i 收到响应后，首先验证安全参数 $\widehat{\mathrm{SP}_i'} = (\widehat{s_t'}, \widehat{y_t'}, \widehat{R_t'})_{t \in [T]}$ 的有效性，具体过程如下。对于 $t \in [T]$，TD_i 计算 $s_t' = \widehat{s_t'} \oplus H_2(\mathrm{tid}_i, x_i' \Phi_i, x_i \Phi_i)$，$y_t' = \widehat{y_t'} \oplus H_2(\mathrm{tid}_i, x_i' \Phi_i, x_i \Phi_i)$，$R_t' = \widehat{R_t'} - X_i - X_i'$，并验证 $y_t' P = R_t' + H_1(s_t' X_i') P_{\mathrm{Pub}}$ 是否成立。若成立，则计算 $\gamma_t' = s_t' x_i'$，并继续执行后续操作；否则，验证失败，TD_i 终止操作。至此，终端设备 TD_i 得到新的公钥 $\mathrm{PK}_i' = X_i'$ 和私钥 $\mathrm{sk}_i' = x_i'$，并将 $\mathrm{RK}_i' = (\gamma_t', y_t', R_t')_{t \in [T]}$ 作为新的随机密钥。

在此基础上，TD_i 随机选取 $c_i' \in Z_p^*$，利用其PUF和模糊提取器得到 $\omega_i' \leftarrow \mathrm{PUF}_i(c_i')$，$(d_i', \mathrm{hd}_i') \leftarrow \mathrm{Gen}(\omega_i')$。然后，$\mathrm{TD}_i$ 计算 $\tilde{h}_i' = H_1(d_i', X_i')$，$\tilde{x}_i' = \tilde{h}_i' \oplus x_i'$，$\tilde{\gamma}_t' = \tilde{h}_i' \oplus \gamma_t' (t \in [T])$ 以及 $\tilde{y}_t' = \tilde{h}_i' \oplus y_t' (t \in [T])$，令 $\widetilde{\mathrm{sk}_i'} = \tilde{x}_i'$，$\widetilde{\mathrm{RK}_i'} = (\tilde{\gamma}_t', \tilde{y}_t', R_t')_{t \in [T]}$，并将存储的旧参数 $< \mathrm{PK}_i, \widetilde{\mathrm{sk}}_i, \widetilde{\mathrm{RK}}_i, c_i, \mathrm{hd}_i >$ 替换为新参数 $< \mathrm{PK}_i', \widetilde{\mathrm{sk}_i'}, \widetilde{\mathrm{RK}_i'}, c_i', \mathrm{hd}_i' >$。

2.8 边缘服务器参数更新

与终端设备相同，边缘服务器也可通过与KGC交互实现其安全参数的动态更新。该过程可在线完成，具体操作如下。

1) 边缘服务器 MS_j 随机选取 $x_j' \in Z_p^*$，计算 $X_j' = x_j' P$，并向KGC提交 $< \mathrm{sid}_j, X_j, R_j, X_j' >$。

2) KGC收到边缘服务器 MS_j 的参数更新请求后，随机选取 $r_j' \in Z_p^*$，计算 $R_j' = r_j' P$，$\varphi_j = H_3(\mathrm{sid}_j, X_j, X_j', R_j, R_j', (k_M + r_j') X_j)$，并向 MS_j 回复 $< \varphi_j, R_j' >$。

3) MS_j 收到响应后，利用其PUF和模糊提取器得到 $\bar{\omega}_j' \leftarrow \mathrm{PUF}_j(c_j)$，$d_j \leftarrow \mathrm{Rep}(\bar{\omega}_j', \mathrm{hd}_j)$，计算 $\tilde{h}_j = H_1(d_j, X_j)$，$x_j = \tilde{h}_j \oplus \tilde{x}_j$ 以及 $y_j = \tilde{h}_j \oplus \tilde{y}_j$，并验证等式 $\varphi_j = H_3(\mathrm{sid}_j, X_j, X_j', R_j, R_j', x_j(P_{\mathrm{Pub}} + R_j'))$ 是否成立。若成立，则 MS_j 实现对KGC的身份认证，并继续执行；否则，认证失败，终止操作。

然后，MS_j 计算 $\tau_j = H_3(\mathrm{sid}_j, \varphi_j, X_j, X_j', R_j, R_j', x_j' P_{\mathrm{Pub}})$ 与 $\varrho_j = y_j + x_j' \tau_j$，并向KGC回复 $< \varrho_j >$。

4) KGC收到响应后，首先计算 $\tau_j' = H_3(\mathrm{sid}_j, \varphi_j, X_j, X_j', R_j, R_j', k_M X_j')$，并验证等式 $\varrho_j P = R_j + H_2(\mathrm{sid}_j, X_j, R_j) P_{\mathrm{Pub}} + \tau_j' X_j'$ 是否成立。若成立，则KGC实现对 MS_j 的身份认证，并继续执行后续操作；否则，认证失败，终止操作。

在此基础上，KGC计算 $y_j' = r_j' + k_M H_2(\mathrm{sid}_j, X_j, R_j)$ 与 $z_j = y_j' \oplus \tau_j'$，并向 MS_j 回复 $< z_j >$。

5) MS_j收到响应后，计算$y_j' = z_j \oplus \tau_j$，并验证等式$y_j'P = R_j' + H_2\left(\mathrm{sid}_j, X_j', R_j'\right)P_{\mathrm{Pub}}$是否成立。若成立，则$\mathrm{MS}_j$接受新参数，并继续执行后续操作；否则，参数验证失败，终止操作。至此，MS_j得到新的公钥$\mathrm{PK}_j' = (X_j', R_j')$与新的私钥$\mathrm{sk}_j' = (x_j', y_j')$。

在此基础上，MS_j随机选取$c_j' \in Z_p^*$，利用其PUF和模糊提取器得到$\omega_j' \leftarrow \mathrm{PUF}_j(c_j')$，$(d_j', \mathrm{hd}_j') \leftarrow \mathrm{Gen}(\omega_j')$。然后，$\mathrm{MS}_j$计算$\tilde{h}_j' = H_1\left(d_j', X_j'\right)$，$\tilde{x}_j' = \tilde{h}_j' \oplus x_j'$以及$\tilde{y}_j' = \tilde{h}_j' \oplus y_j'$，令$\widetilde{\mathrm{sk}}_j' = (\tilde{x}_j', \tilde{y}_j')$，并将存储的旧参数$< \mathrm{PK}_j, \widetilde{\mathrm{sk}}_j, c_j, \mathrm{hd}_j >$替换为新参数$< \mathrm{PK}_j', \widetilde{\mathrm{sk}}_j', c_j', \mathrm{hd}_j' >$。

3　安全性分析

3.1　形式化证明

在1.3节安全模型的基础上，本节通过形式化方法证明本文方案在随机预言机模型下满足会话密钥的语义安全性。

定理1　若$\mathcal{A}$为攻击本文方案的概率多项式时间敌手，其破坏方案会话密钥语义安全性的优势为

$$\mathrm{Adv}_{\mathcal{S}}(\mathcal{A}) \leqslant \frac{q_h^2 + (q_s + q_e)^2}{p} + 2\varepsilon_{\mathrm{PUF}} + 4\varepsilon_{\mathrm{ECDL}} + 2\varepsilon_{\mathrm{ECCDH}}$$

其中，p为方案中选取的安全大素数，q_h、q_s和q_e分别表示敌手执行Hash询问、Send询问和Execute询问的次数。

证明　通过构造敌手$\mathcal{A}$与挑战者$\mathcal{C}$之间的序列化游戏$\mathrm{Game}_i\ (i = 0,1,2,3,4,5)$来证明上述定理。在该系列游戏中，任意2个相邻的游戏对于敌手$\mathcal{A}$而言，是计算不可区分的。此外，在证明过程中利用$\Pr[\mathrm{Succ}_i]$来表示敌手$\mathcal{A}$在游戏Game_i中获胜的概率。

1) Game_0。在该游戏中，敌手$\mathcal{A}$对方案发起真实攻击。由定义6的会话密钥语义安全性可知

$$\mathrm{Adv}_{\mathcal{S}}(\mathcal{A}) = |2\Pr[\mathrm{Succ}_0] - 1|$$

2) Game_1。在该游戏中，挑战者$\mathcal{C}$首先随机选取$k_M \in Z_p^*$，计算$P_{\mathrm{Pub}} = k_M P$，并将系统公开参数发送给敌手$\mathcal{A}$。$\mathcal{A}$可在多项式时间内向$\mathcal{C}$发起预言机询问，$\mathcal{C}$按如下方式进行响应。

①Hash(M)询问。$\mathcal{C}$按照随机预言机方式模拟该询问，并返回相应的响应值给$\mathcal{A}$。

②PUF(U,C)询问。$\mathcal{C}$按照随机预言机方式模拟该询问，并返回相应的响应值给$\mathcal{A}$。

③Capture(U)询问。对于该询问，$\mathcal{C}$返回实体U存储的参数信息给$\mathcal{A}$。具体而言，若询问实体为终端设备TD_i，$\mathcal{C}$返回参数$< \mathrm{PK}_i, \widetilde{\mathrm{sk}}_i, \widetilde{\mathrm{RK}}_i, c_i, \mathrm{hd}_i >$；若询问实体为边缘服务器$\mathrm{MS}_j$，$\mathcal{C}$返回参数$< \mathrm{PK}_j, \widetilde{\mathrm{sk}}_j, c_j, \mathrm{hd}_j >$。

④Corrupt(U)询问。对于该询问，$\mathcal{C}$返回实体U的长期密钥参数给$\mathcal{A}$。具体而言，若询问实体为终端设备TD_i，$\mathcal{C}$返回参数$< x_i, X_i, \mathrm{RK}_i >$；若询问实体为边缘服务器$\mathrm{MS}_j$，$\mathcal{C}$返回参数$< x_j, y_j, X_j, R_j >$。

⑤Send$(\prod_U^k, M)$询问。该询问可分为以下几种情况。

Send$(\prod_{\mathrm{TD}_i}^m, \mathrm{Start})$：对于该询问，$\mathcal{C}$首先通过Corrupt$(\mathrm{TD}_i)$询问得到参数$< x_i, X_i, \mathrm{RK}_i >$，然后随机选取伪身份标识$\mathrm{pid}_i \in \{0,1\}^{L_{\mathrm{ID}}}$以及随机值$x_0, r_0 \in Z_p^*$，并获取当前时间戳$t_i$；在此基础上，依据方案计算相关参数，并返回$< t_i, \mathrm{pid}_i, R_i^*, X_i^*, \eta_j, \sigma_i >$给$\mathcal{A}$。

Send$(\prod_{\mathrm{MS}_j}^n, < t_i, \mathrm{pid}_i, R_i^*, X_i^*, \eta_j, \sigma_i >)$：对于该询问，$\mathcal{C}$首先验证时间戳$t_i$的有效性；若有效，则通过Corrupt$(\mathrm{MS}_j)$询问得到参数$< x_j, y_j, X_j, R_j >$，并依据方案计算相关参数，验证等式$\sigma_i P = R_i^* + H_1(\mathrm{CH}_{X_i^*}(h_i^*, \beta_i))P_{\mathrm{Pub}} + H_2(t_i, \mathrm{pid}_i, \beta_i, R_i^*, X_i^*)X_i^*$是否成立；若成立，则计算会话密钥$k = H_2(\mathrm{pid}_i, \mathrm{sid}_j, t_i, t_j, Q_j, e_j X_i^*)$，并返回$< v_j, E_j >$给$\mathcal{A}$。

Send$(\prod_{\mathrm{TD}_i}^m, < v_j, E_j >)$：对于该询问，$\mathcal{C}$依据方案计算相关参数，并验证等式$v_j P = R_j + H_2(\mathrm{sid}_j, X_j, R_j)P_{\mathrm{Pub}} + u_j' E_j$是否成立；若成立，计算会话密钥$k = H_2(\mathrm{pid}_i, \mathrm{sid}_j, t_i, t_j, Q_j, (x_i + x_0)E_j)$。

⑥Execute$(\prod_{\mathrm{TD}_i}^m, \prod_{\mathrm{MS}_j}^n)$询问。对于该询问，$\mathcal{C}$返回上述Send询问执行过程中产生的全部消息$<< t_i, \mathrm{pid}_i, R_i^*, X_i^*, \eta_j, \sigma_i >, < v_j, E_j >>$给$\mathcal{A}$。

⑦Reveal$(\prod_U^k)$询问。对于该询问，$\mathcal{C}$检查会话

实例$\prod_U^k$是否处于接受状态。若是，则$\mathcal{C}$返回相应会话密钥k给$\mathcal{A}$；否则，$\mathcal{C}$拒绝询问并返回⊥。

⑧Test($\prod_U^k$)询问。在游戏的最后，$\mathcal{A}$能够执行一次该询问。对于该询问，$\mathcal{C}$首先通过Reveal($\prod_U^k$)询问得到相应会话密钥k，并通过掷币游戏随机选取1比特数据coin。若coin = 1，则$\mathcal{C}$将密钥k返回给$\mathcal{A}$；否则，$\mathcal{C}$选取与会话密钥等长的随机数，并返回给$\mathcal{A}$。

由上述过程可知，相较于Game_0，Game_1按随机预言机方式模拟Hash询问和PUF询问。在随机预言机模型下，以该方式模拟Hash询问并未增加敌手攻击成功的概率，并且由PUF的不可区分性[2]可知，敌手区分PUF响应值与等长随机数的优势不超过ε_{PUF}。因此，敌手$\mathcal{A}$在该游戏中增加的优势不超过ε_{PUF}，可以得到

$$|\Pr[\text{Succ}_1]-\Pr[\text{Succ}_0]|\leqslant\varepsilon_{\text{PUF}}$$

3) Game_2。该游戏在Game_1的基础上，排除了哈希碰撞和随机数碰撞出现的可能。根据生日悖论，其概率不超过$\frac{q_h^2}{2p}+\frac{(q_s+q_e)^2}{2p}$。显然，当上述碰撞事件不发生时，$\text{Game}_2$和$\text{Game}_1$对敌手$\mathcal{A}$而言是不可区分的，可以得到

$$|\Pr[\text{Succ}_2]-\Pr[\text{Succ}_1]|\leqslant\frac{q_h^2}{2p}+\frac{(q_s+q_e)^2}{2p}$$

4) Game_3。在上述游戏中，敌手$\mathcal{A}$可通过相关询问得到终端设备TD_i和边缘服务器MS_j存储的相关参数，以及TD_i的长期密钥$<x_i,X_i,\text{RK}_i>$和MS_j的长期密钥$<x_j,y_j,X_j,R_j>$。对于会话密钥$k=H_2(\text{pid}_i,\text{sid}_j,t_i,t_j,Q_j,e_jX_i^*)=H_2(\text{pid}_i,\text{sid}_j,t_i,t_j,Q_j,e_j(x_i+x_0)P)=H_2(\text{pid}_i,\text{sid}_j,t_i,t_j,Q_j,(x_i+x_0)E_j)$，$\mathcal{A}$能够得到参数$<\text{pid}_i,\text{sid}_j,t_i,t_j>$以及参数$Q_j=(x_j+y_j)X_i^*$，但为了计算出会话密钥$k$，其仍需要得到参数$e_jX_i^*=e_j(x_i+x_0)P=(x_i+x_0)E_j$。该游戏在$\text{Game}_2$的基础上，排除了$\mathcal{A}$通过参数$E_j=e_jP$计算出$e_j$的概率。根据ECDL假设[14]，$\mathcal{A}$求解出该问题的优势不超过$\varepsilon_{\text{ECDL}}$。显然，当上述事件不发生时，$\text{Game}_3$和$\text{Game}_2$对敌手$\mathcal{A}$而言是不可区分的，可以得到

$$|\Pr[\text{Succ}_3]-\Pr[\text{Succ}_2]|\leqslant\varepsilon_{\text{ECDL}}$$

5) Game_4。该游戏在Game_3的基础上，进一步排除了$\mathcal{A}$通过参数$X_i^*=(x_i+x_0)P$计算出x_i+x_0的概率。根据ECDL假设[14]，$\mathcal{A}$求解出该问题的优势不超过$\varepsilon_{\text{ECDL}}$。显然，当上述事件不发生时，$\text{Game}_4$和$\text{Game}_3$对敌手$\mathcal{A}$而言是不可区分的，可以得到

$$|\Pr[\text{Succ}_4]-\Pr[\text{Succ}_3]|\leqslant\varepsilon_{\text{ECDL}}$$

6) Game_5。该游戏在Game_4的基础上，进一步排除了$\mathcal{A}$通过参数$E_j=e_jP$和参数$X_i^*=(x_i+x_0)P$计算出$e_j(x_i+x_0)P$的概率。根据ECCDH假设[14]，$\mathcal{A}$求解出该问题的优势不超过$\varepsilon_{\text{ECCDH}}$。显然，当上述事件不发生时，$\text{Game}_5$和$\text{Game}_4$对敌手$\mathcal{A}$而言是不可区分的，可以得到

$$|\Pr[\text{Succ}_5]-\Pr[\text{Succ}_4]|\leqslant\varepsilon_{\text{ECCDH}}$$

由于该游戏已排除了敌手$\mathcal{A}$用于计算会话密钥k的所有可能情况，$\mathcal{A}$仅能够通过猜测来赢得Game_5，可以得到

$$\Pr[\text{Succ}_5]=\frac{1}{2}$$

在上述分析过程的基础上，结合三角不等式$|\Pr[\text{Succ}_i]-\Pr[\text{Succ}_k]|\leqslant|\Pr[\text{Succ}_i]-\Pr[\text{Succ}_j]+|\Pr[\text{Succ}_j]-\Pr[\text{Succ}_k]|$，可以得到敌手$\mathcal{A}$获胜的优势

$$\text{Adv}_{\mathcal{S}}(\mathcal{A})\leqslant\frac{q_h^2+(q_s+q_e)^2}{p}+2\varepsilon_{\text{PUF}}+4\varepsilon_{\text{ECDL}}+2\varepsilon_{\text{ECCDH}}$$

其中，参数p为安全的大素数，而参数q_h、q_s和q_e均为多项式有界的值，可知$\frac{q_h^2+(q_s+q_e)^2}{p}$为可忽略的值。同时，由PUF的不可区分性可知$\varepsilon_{\text{PUF}}$为可忽略的值，由ECDL假设和ECCDH假设可知$\varepsilon_{\text{ECDL}}$和$\varepsilon_{\text{ECCDH}}$均为可忽略的值。因此，当敌手$\mathcal{A}$执行多项式有界次预言机询问时，其获胜的优势$\text{Adv}_{\mathcal{S}}(\mathcal{A})$是可忽略的。结合1.3节定义的安全模型可知，本文认证密钥协商方案在随机预言机模型下满足会话密钥的语义安全性。

3.2 安全属性分析

在上述形式化证明的基础上，本节进一步通过非形式化方法分析本文方案具备的安全属性。

1) 双向认证。本文方案中，边缘服务器MS_j通过验证等式$\sigma_iP=R_i^*+H_1(\text{CH}_{X_i^*}(h_i^*,\beta_i))P_{\text{Pub}}+H_2(t_i,\text{pid}_i,\beta_i,R_i^*,X_i^*)X_i^*$是否成立来实现对终端设备$\text{TD}_i$的合法性认证。由于$\sigma_i=y_i^*+(x_i+x_0)H_2$

$(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)$，$\mathrm{CH}_{X_i^*}(h_i^*,\beta_i)=\gamma_i P$ 且 $y_i^*=y_i+r_0$，仅拥有参数 (x_i,X_i) 和 (γ_i,y_i,R_i) 的合法设备能够计算出正确的参数 σ_i 使上式成立。同样，终端设备 TD_i 通过验证等式 $v_jP=R_j+H_2\left(\mathrm{sid}_j,X_j,R_j\right)P_{\mathrm{Pub}}+u_j'E_j$ 是否成立来实现对边缘服务器 MS_j 的合法性认证。由于 $v_j=y_j+e_ju_j$ 且 $u_j'=u_j=H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_jX_i^*)$，仅拥有参数 (x_j,y_j) 和 (X_j,R_j) 的合法设备能够计算出正确的参数 v_j 使上式成立。因此，敌手不能伪造出有效的参数来通过合法性认证，即本文方案能够满足双向认证要求。

2) 安全会话密钥建立。本文方案中，边缘服务器 MS_j 通过执行 $k_{ji}=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_jX_i^*)$ 计算会话密钥，而终端设备 TD_i 通过执行 $k_{ij}=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i+x_0)E_j)$ 计算会话密钥。由于 $e_jX_i^*=e_j(x_i+x_0)P=(x_i+x_0)E_j$，可知 $k_{ji}=k_{ij}$，即双方在认证过程中建立起了共享的会话密钥。此外，由定理 1 可知，当 ECDL 假设和 ECCDH 假设成立时，敌手无法计算出该会话密钥。

3) 无第三方实时参与。由方案执行过程可知，KGC 仅用于系统初始化和设备注册过程，本文方案的双向认证和密钥协商过程由终端设备和边缘服务器直接交互实现，不需要 KGC 的实时参与。

4) 无密钥托管。本文方案中，设备的密钥参数由 KGC 和该设备共同生成，即 KGC 仅生成设备的部分密钥。具体而言，对于边缘服务器 MS_j，在其完整密钥参数 $<\mathrm{PK}_j=(X_j,R_j),\mathrm{sk}_j=(x_j,y_j)>$ 中，KGC 仅生成 y_j 和 R_j，并且不掌握秘密值 x_j；对于终端设备 TD_i，在其完整密钥参数 $<\mathrm{PK}_i=X_i,\mathrm{sk}_i=x_i,\mathrm{RK}_i=(\gamma_t,y_t,R_t)_{t\in[T]}>$ 中，KGC 仅生成 RK_i，并且不掌握秘密值 x_i。因此，本文方案无密钥托管的问题。

5) 匿名性。在每次通信过程中，终端设备 TD_i 均随机选取新的伪身份标识 $\mathrm{pid}_i\in\{0,1\}^{L_{\mathrm{ID}}}$ 和随机值 $x_0,r_0\in Z_p^*$，并利用这些随机值计算临时公钥和临时参数。对于敌手而言，无法通过分析终端设备 TD_i 的认证请求 $<t_i,\mathrm{pid}_i,R_i^*,X_i^*,\eta_j,\sigma_i>$ 得到该设备的真实身份信息；对于边缘服务器 MS_j 而言，仅能够验证终端设备 TD_i 的合法性，但无法得到该设备的真实身份标识；对于 KGC 而言，由于不掌握 TD_i 的秘密值 x_i 和 MS_j 的秘密值 x_j，仍无法独立分析得到该设备的真实身份信息。

6) 条件隐私。由恶意终端设备追踪过程可知，通过边缘服务器和 KGC 的协同，可以揭示出恶意终端设备的身份信息。具体而言，边缘服务器根据该终端设备的请求计算出参数 β_i，并将参数 $<X_i^*,h_i^*,\beta_i>$ 提交给 KGC。在此基础上，KGC 通过遍历参数列表查找使等式 $\mathrm{CH}_{X_i^*}(h_i^*,\beta_i)=s_iX_i$ 成立的参数，进而得到该终端设备的真实身份标识。特别地，由于 KGC 无法计算参数 β_i，其不能独立得到终端设备的真实身份信息，对恶意终端设备的追踪必须通过边缘服务器和 KGC 协同才能实现。

7) 不可链接性。根据方案执行过程可知，终端设备在每次通信时均选取新的伪身份标识和临时公钥，并通过随机数计算新参数来生成认证请求。具体而言，在认证请求 $<t_i,\mathrm{pid}_i,R_i^*,X_i^*,\eta_j,\sigma_i>$ 中，t_i 为当前时间戳，pid_i 为随机选取的伪身份标识，$R_i^*=R_i+r_0P$，$X_i^*=(x_i+x_0)P$，$\eta_j=\alpha_j\oplus\beta_i$，$\sigma_i=y_i^*+(x_i+x_0)H_2(t_i,\mathrm{pid}_i,\beta_i,R_i^*,X_i^*)$ 且 $\alpha_j=H_1(t_i,t_j,Q_j)$，由于随机数 x_0、r_0 以及时间戳 t_i 的新鲜性，各个认证请求之间的参数相互独立，敌手无法通过收集多个认证请求来对特定终端设备的行为进行关联和分析。

8) 完美前向安全性。在认证和密钥协商过程中，终端设备 TD_i 和边缘服务器 MS_j 计算出的共享会话密钥 $k=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i+x_0)E_j)=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_jX_i^*)$。即使敌手得到终端设备 TD_i 的长期密钥 $<x_i,X_i,\mathrm{RK}_i>$ 和边缘服务器 MS_j 的长期密钥 $<x_j,y_j,X_j,R_j>$，由于不掌握参数 x_0 和 e_j，其仍然无法得到参数 $(x_i+x_0)E_j$ 或 $e_jX_i^*$ 来计算会话密钥。因此，设备长期密钥的泄露并不会影响前序会话密钥的安全性，即本文方案具备完美前向安全性。

9) 抗窃听攻击。通过执行窃听攻击，敌手可以得到公开信道上传递的全部消息。以终端设备 TD_i 和边缘服务器 MS_j 之间的认证和密钥协商为例，敌手通过窃听可以得到 $<<t_i,pid_i,R_i^*,X_i^*,\eta_j,\sigma_i>,<v_j,E_j>>$。根据定理 1，由于敌手无法解决 ECDL 和 ECCDH 问题，其不能从这些公开消息中得到任何有用信息。因此，本文方案能够抵抗窃听攻击。

10) 抗篡改攻击。本文方案通过带密钥的哈希函数来保证消息的抗篡改特性。例如，对于参数 $u_j=H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,e_jX_i^*)=H_3(t_i,t_j,E_j,X_j,R_j,R_i^*,$

$(x_i + x_0)E_j$)，未掌握秘密值e_j或(x_i,x_0)的敌手无法计算出有效的哈希值。同时，合法的方案参与者能够在消息验证过程中检测到敌手对于消息的恶意篡改。因此，本文方案能够抵抗篡改攻击。

11) 抗重放攻击。本文方案选取当前时间戳和随机数作为新鲜因子来保证消息的新鲜性，并通过带密钥的哈希函数来保护这些新鲜因子。若敌手截获旧消息并进行重放，则其无法计算出带有新时间戳和随机数的有效响应。因此，本文方案能够抵抗重放攻击。

12) 抗中间人攻击。由于敌手不能计算出有效的认证参数，其无法成功伪造成合法设备来参与方案执行。同时，由于本文方案采用了带密钥的哈希函数来实现消息验证机制，能够有效检测出敌手对消息的恶意篡改行为。因此，本文方案能够抵抗中间人攻击。

13) 抗物理攻击。通过执行物理攻击，敌手可获取设备存储器中保存的全部参数。对于边缘服务器MS_j而言，敌手可通过物理攻击得到参数$<\mathrm{PK}_j=(X_j,R_j),\widetilde{\mathrm{sk}}_j=(\tilde{x}_j,\tilde{y}_j),c_j,\mathrm{hd}_j>$。$\mathrm{MS}_j$的私钥$(x_j,y_j)$的计算方法为$x_j=\tilde{h}_j\oplus\tilde{x}_j$，$y_j=\tilde{h}_j\oplus\tilde{y}_j$，其中$\tilde{h}_j=H_1(d_j,X_j)$且$d_j\leftarrow\mathrm{Rep}(\bar{\omega}_j,\mathrm{hd}_j)$，$\bar{\omega}_j\leftarrow\mathrm{PUF}_j(c_j)$。由于PUF具备唯一性、不可预测性和不可区分性，并且任何针对PUF的侵入式攻击和非法访问将改变其映射关系[20-21]，即使敌手能够得到挑战值c_j，其仍无法得到对应的参数d_j来计算私钥(x_j,y_j)。若敌手采用错误的参数d_j'来执行方案，则无法通过合法性验证。同样，敌手无法通过实施物理攻击来获取终端设备的密钥参数。因此，方案能够提供对物理攻击的防范。

14) 抗已知密钥攻击。由本文方案可知，通信双方计算出的会话密钥$k=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j,(x_i+x_0)E_j)=H_2(\mathrm{pid}_i,\mathrm{sid}_j,t_i,t_j,Q_j',e_jX_i^*)$。由于$x_0$和$e_j$为双方在通信时选取的新随机值，该会话密钥与其他会话过程中产生的密钥无任何关联。因此，即使敌手得到了前序或后序密钥协商过程中建立的会话密钥，仍不能增加其攻击方案的优势，即本文方案能够抵抗已知密钥攻击。

4 性能分析与对比

本节对本文方案的安全属性、计算开销和通信开销进行分析，并与同类方案进行对比。

4.1 安全属性

本文方案与同类方案的安全属性对比如表2所示。由表2可知，本文方案和同类方案均能实现双向认证和会话密钥协商，但文献[8]、文献[9]、文献[13]要求在线第三方的实时参与，降低了认证和密钥协商的效率。文献[7]、文献[9]采用基于身份的密码体制设计，该类方案存在密钥托管的问题。在隐私保护方面，除文献[15]外，其余方案均提供了对终端设备的匿名性保护机制，但

表2 本文方案与同类方案的安全属性对比

方案	双向认证	安全会话密钥建立	无第三方实时参与	无密钥托管	匿名性	条件隐私	不可链接性	完美前向安全性	抗窃听攻击	抗篡改攻击	抗重放攻击	抗中间人攻击	抗物理攻击	抗已知密钥攻击
文献[7]	√	√	√	×	√	×	×	×	√	√	√	×	×	√
文献[8]	√	√	×	×	√	×	×	√	√	√	×	√	×	√
文献[9]	√	√	×	×	√	×	×	√	√	√	√	√	×	√
文献[10]	√	√	√	√	√	×	×	×	√	√	√	√	×	√
文献[11]	√	√	√	√	√	×	×	√	√	√	√	√	×	√
文献[12]	√	√	√	√	√	×	×	√	√	√	√	√	×	√
文献[13]	√	√	×	√	√	×	√	√	√	√	√	√	×	√
文献[14]	√	√	√	√	√	×	×	√	√	√	√	√	×	√
文献[15]	√	√	√	√	×	×	×	√	√	√	√	√	×	√
本文方案	√	√	√	√	√	√	√	√	√	√	√	√	√	√

只有文献[13]和本文方案满足不可链接性的要求，并且仅本文方案能够实现条件隐私保护。此外，同类方案均未提供对物理攻击的防范机制。因此，相较于同类方案，本文方案具备更高的安全性。

4.2　计算开销

在实验过程中，利用 Python 语言基于 py_ecc 函数库进行测试。实验平台硬件参数如表 3 所示。椭圆曲线选用实现 128 bit 安全等级的 secp256k1 曲线，在该曲线定义下，大素数 p 为 128 bit，群 G 上的点为 256 bit。由于文献[7]、文献[12]采用了双线性对运算，为便于性能对比，在实现同等安全等级的双线性对友好的 BLS12-381 曲线上构造 Optimal ATE 对[26]：$G_2 \times G_1 \rightarrow G_T$，其中 G_1 为有限域 $F_{\bar{q}}$ 上阶为大素数 $\bar{p}$ 的循环群，G_2 和 G_T 分别为扩域 $F_{\bar{q}^2}$ 和 $F_{\bar{q}^{12}}$ 上的循环群，且 $\bar{q}$ 为 381 bit、$\bar{p}$ 为 256 bit。

表 3　实验平台硬件参数

实体	实验平台	硬件参数
边缘服务器	计算机设备	Inter Core i7-1165G7 CPU, 16 GB RAM, Ubuntu 18.04 OS
终端设备	Raspberry Pi 4B 开发板	Broadcom BCM2711 CPU, 8 GB RAM, Debian 11 OS

采用与文献[12]、文献[14]相同的评估方法，通过复杂密码运算的执行次数来衡量方案的计算开销。在上述实验平台的基础上，测试得到方案涉及的各类复杂密码运算的平均运行时间，如表 4 所示。

表 4　复杂密码运算的平均运行时间

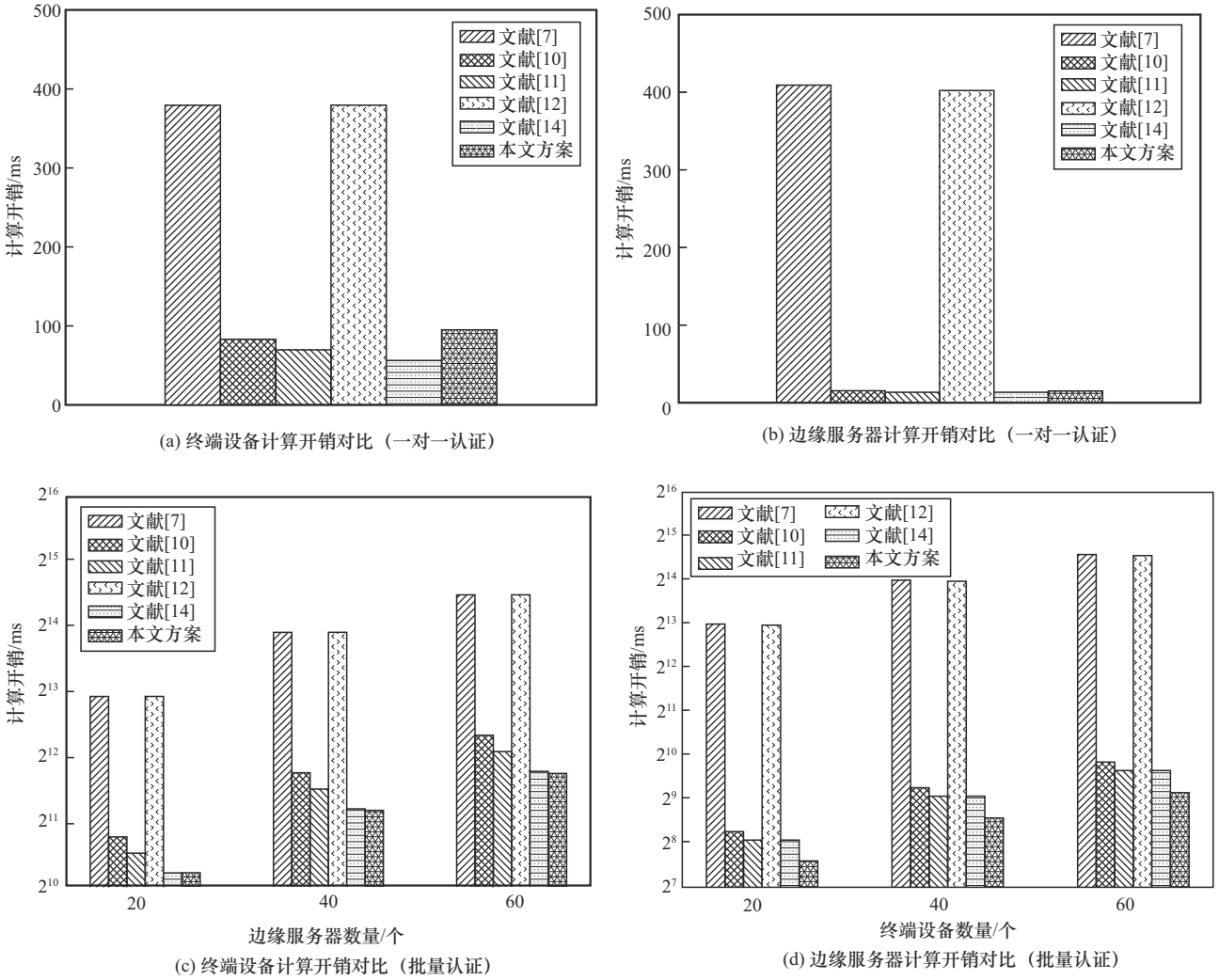
符号	密码运算	边缘服务器/ms	终端设备/ms
T_{aG}	群 G 上的点加运算	0.052	0.226
T_{smG}	群 G 上的点乘运算	1.873	13.410
T_{hpG}	群 G 上的 hash-to-point 运算	0.223	2.169
T_{aG_1}	群 G_1 上的点加运算	0.027	0.166
T_{smG_1}	群 G_1 上的点乘运算	6.908	48.914
T_{hpG_1}	群 G_1 上的 hash-to-point 运算	3.318	23.273
T_{aG_2}	群 G_2 上的点加运算	0.107	0.703
T_{smG_2}	群 G_2 上的点乘运算	31.808	219.407
T_{hpG_2}	群 G_2 上的 hash-to-point 运算	92.220	646.044
T_{eG_T}	群 G_T 上的幂运算	20.039	160.158
T_{bp}	双线性对运算	371.201	2 924.306

结合 MEC 网络的特点，分别考虑一对一认证、批量认证两类典型应用场景。本文方案与同类方案（可实现匿名性保证且不需要第三方参与的方案）的计算开销对比如表 5 所示，其中，I 为终端设备的数量，J 为边缘服务器的数量。特别地，当 $I = 1$ 且 $J \neq 1$ 时，对应数据为单个终端设备和多个边缘服务器认证（一对多认证）的计算开销；当 $I \neq 1$ 且 $J = 1$ 时，对应数据为多个终端设备和单个边缘服务器认证（多对一认证）的计算开销。

在此基础上，结合表 4 中密码运算的运行时间，可得如图 5 所示的计算开销对比。由图 5 可知，文献[7]和文献[12]的计算开销显著高于其他方案，这主要归因于其采用了计算复杂度较高的双线性对运算。相比之下，本文方案与文献[10]、文献[11]、

表 5　计算开销对比

方案	一对一认证		批量认证	
	终端设备/ms	边缘服务器/ms	终端设备/ms	边缘服务器/ms
文献[7]	$T_{aG_1} + 4T_{smG_1} + T_{hpG_1} + T_{eG_T}$	$3T_{aG_1} + 5T_{smG_1} + T_{hpG_1} + T_{bp}$	$J \times T_{aG_1} + 4J \times T_{smG_1} + J \times T_{hpG_1} + J \times T_{eG_T}$	$3I \times T_{aG_1} + 5I \times T_{smG_1} + I \times T_{hpG_1} + I \times T_{bp}$
文献[10]	$2T_{aG} + 6T_{smG} + T_{hpG}$	$4T_{aG} + 8T_{smG} + T_{hpG}$	$2J \times T_{aG} + 6J \times T_{smG} + J \times T_{hpG}$	$4I \times T_{aG} + 8I \times T_{smG} + I \times T_{hpG}$
文献[11]	$2T_{aG} + 5T_{smG} + T_{hpG}$	$3T_{aG} + 7T_{smG} + T_{hpG}$	$2J \times T_{aG} + 5J \times T_{smG} + J \times T_{hpG}$	$3I \times T_{aG} + 7I \times T_{smG} + I \times T_{hpG}$
文献[12]	$2T_{aG_1} + 4T_{smG_1} + T_{hpG_1} + T_{eG_T}$	$2T_{aG_1} + 4T_{smG_1} + T_{hpG_1} + T_{bp}$	$2J \times T_{aG_1} + 4J \times T_{smG_1} + J \times T_{hpG_1} + J \times T_{eG_T}$	$2I \times T_{aG_1} + 4I \times T_{smG_1} + I \times T_{hpG_1} + I \times T_{bp}$
文献[14]	$2T_{aG} + 4T_{smG} + T_{hpG}$	$3T_{aG} + 7T_{smG} + T_{hpG}$	$2J \times T_{aG} + 4J \times T_{smG} + J \times T_{hpG}$	$3I \times T_{aG} + 7I \times T_{smG} + I \times T_{hpG}$
本文方案	$5T_{aG} + 7T_{smG}$	$3T_{aG} + 8T_{smG}$	$(3J + 2)T_{aG} + (4J + 3)T_{smG}$	$(I + 2)T_{aG} + (5I + 3)T_{smG}$



(a) 终端设备计算开销对比（一对一认证）

(b) 边缘服务器计算开销对比（一对一认证）

(c) 终端设备计算开销对比（批量认证）

(d) 边缘服务器计算开销对比（批量认证）

图5　计算开销对比

文献[14]则具有相近的计算开销。具体而言，在一对一认证场景下，本文方案的计算开销略高于文献[10]、文献[11]、文献[14]；而在批量认证场景下，本文方案的计算开销则优于上述3个方案。因此，本文方案更适用于具有高移动性和高并发性特点的MEC应用场景。

4.3　通信开销

与文献[12]、文献[14]一致，采用实体发送的消息长度作为衡量其通信开销的关键指标。为便于清晰描述，引入以下符号表示不同元素的长度：L_T表示时间戳的长度，L_{ID}表示身份标识的长度，$L_{Z_p^*}$表示Z_p^*中元素的长度，L_G表示群G中元素的长度，$L_{Z_{\bar{p}}^*}$表示$Z_{\bar{p}}^*$中元素的长度，L_{G_1}表示群G_1中元素的长度。依据方案的执行过程，两类典型应用场景下各方案的通信开销对比如表6所示。

根据secp256k1曲线和BLS12-381曲线的定义，可得各元素的长度：$L_{Z_p^*}$为256 bit，L_G为512 bit，$L_{Z_{\bar{p}}^*}$为256 bit，L_{G_1}为762 bit。此外，不失一般性，将时间戳的长度L_T和身份标识的长度L_{ID}均定义为32 bit。在此基础上，可得各方案的通信开销对比如图6所示。

由图6可知，在一对一认证场景下，本文方案的终端设备通信开销优于文献[7]、文献[12]、文献[14]，但相较于文献[10]、文献[11]略有增加；同时，本文方案在边缘服务器端的通信开销表现最优。在批量认证场景下，本文方案的终端设备通信开销和边缘服务器通信开销均展现出明显优势，且随着认证设备数量的增加，这一优势呈现显著提升。

表6　通信开销对比

方案	一对一认证		批量认证	
	终端设备/bit	边缘服务器/bit	终端设备/bit	边缘服务器/bit
文献[7]	$L_T + L_{\mathrm{ID}} + L_{Z_{\tilde{p}}^*} + 3L_{G_1}$	$L_T + L_{Z_{\tilde{p}}^*} + L_{G_1}$	$J \times L_T + J \times L_{\mathrm{ID}} + J \times L_{Z_{\tilde{p}}^*} + 3J \times L_{G_1}$	$I \times L_T + I \times L_{Z_{\tilde{p}}^*} + I \times L_{G_1}$
文献[10]	$L_T + L_{\mathrm{ID}} + L_{Z_p^*} + 2L_G$	$L_T + L_{Z_p^*} + L_G$	$J \times L_T + J \times L_{\mathrm{ID}} + J \times L_{Z_p^*} + 2J \times L_G$	$I \times L_T + I \times L_{Z_p^*} + I \times L_G$
文献[11]	$L_T + L_{\mathrm{ID}} + L_{Z_p^*} + 2L_G$	$L_T + L_{Z_p^*} + L_G$	$J \times L_T + J \times L_{\mathrm{ID}} + J \times L_{Z_p^*} + 2J \times L_G$	$I \times L_T + I \times L_{Z_p^*} + I \times L_G$
文献[12]	$L_T + L_{\mathrm{ID}} + 3L_{Z_{\tilde{p}}^*} + 2L_{G_1}$	$L_T + L_{Z_{\tilde{p}}^*} + L_{G_1}$	$J \times L_T + J \times L_{\mathrm{ID}} + 3J \times L_{Z_{\tilde{p}}^*} + 2J \times L_{G_1}$	$I \times L_T + I \times L_{Z_{\tilde{p}}^*} + I \times L_{G_1}$
文献[14]	$L_T + L_{\mathrm{ID}} + L_{Z_p^*} + 3L_G$	$L_T + L_{Z_p^*} + L_G$	$J \times L_T + J \times L_{\mathrm{ID}} + J \times L_{Z_p^*} + 3J \times L_G$	$I \times L_T + I \times L_{Z_p^*} + I \times L_G$
本文方案	$L_T + L_{\mathrm{ID}} + 2L_{Z_p^*} + 2L_G$	$L_{Z_p^*} + L_G$	$L_T + L_{\mathrm{ID}} + (J+1)L_{Z_p^*} + 2L_G$	$(I+1)L_{Z_p^*} + L_G$

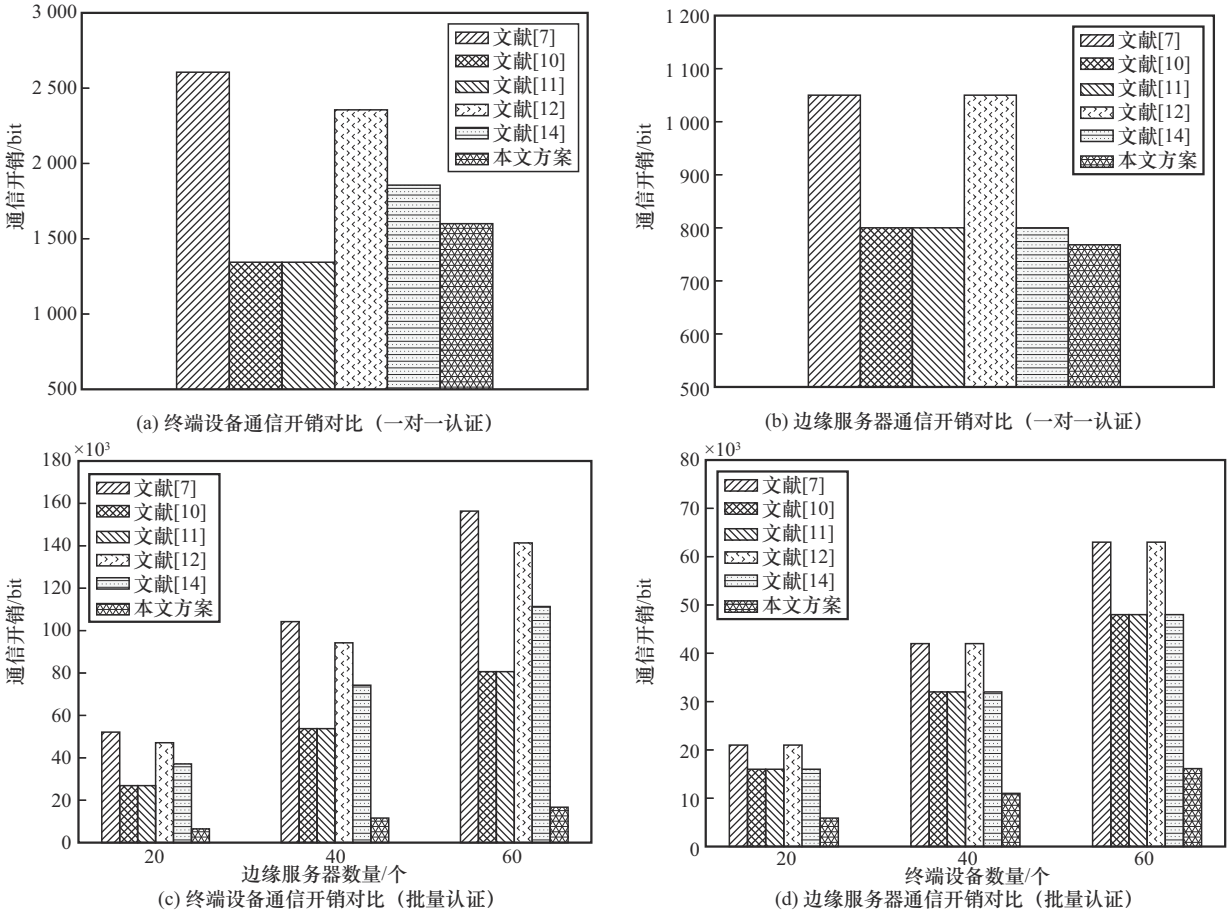


图6　通信开销对比

5　结束语

本文通过将硬件安全原语PUF与椭圆曲线上的无证书密码体制相结合，并引入变色龙哈希函数，提出一种面向MEC的高效条件隐私保护AKA方案，可以有效解决现有方案存在的无法追踪恶意匿名设备、认证模式单一、无法抵抗物理攻击等问题。安全性分析和性能分析表明，本文方案能够在保持较低资源开销的同时，实现更高的安全性，满足MEC环境下设备的安全通信需求。

参考文献：

[1] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89.
SHI W S, ZHANG X Z, WANG Y F, et al. Edge computing: state-of-the-

art and future directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.

[2] 李森森, 刘燕江, 郁滨, 等. 边缘计算环境下基于PUF的多接收者匿名签密文献[J]. 电子学报, 2024, 52(12): 4087-4100.
LI S S, LIU Y J, YU B, et al. PUF-based multi-receiver anonymous signcryption scheme in edge computing[J]. Acta Electronica Sinica, 2024, 52(12): 4087-4100.

[3] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.

[4] HE D J, CHAN S, GUIZANI M. Security in the Internet of Things supported by mobile edge computing[J]. IEEE Communications Magazine, 2018, 56(8): 56-61.

[5] 张晶辉, 张起嘉, 刘海, 等. 面向数据出域安全的鲁棒认证密钥协商协议[J]. 通信学报, 2025, 46(2): 29-43.
ZHANG J H, ZHANG Q J, LIU H, et al. Robust authentication key agreement protocol for cross-domain data security[J]. Journal on Communications, 2025, 46(2): 29-43.

[6] LIANG Y F, LUO E T, LIU Y N. Physically secure and conditional-privacy authenticated key agreement for VANETs[J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7914-7925.

[7] JIA X Y, HE D B, KUMAR N, et al. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing[J]. IEEE Systems Journal, 2020, 14(1): 560-571.

[8] LI Y T, CHENG Q F, LIU X M, et al. A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing[J]. IEEE Systems Journal, 2021, 15(1): 935-946.

[9] JIA X Y, LUO M, CHOO K R, et al. A redesigned identity-based anonymous authentication scheme for mobile-edge computing[J]. IEEE Internet of Things Journal, 2022, 9(12): 10108-10120.

[10] XU Y, ZHOU Y W, YANG B, et al. An efficient identity authentication scheme with provable security and anonymity for mobile edge computing[J]. IEEE Systems Journal, 2022, 17(1): 1012-1023.

[11] MA Y Q, CHENG Q F. An anonymous and certificateless identity authentication protocol for mobile edge computing[J]. IEEE Systems Journal, 2023, 17(4): 5604-5615.

[12] LEE H, RYU J, WON D. Secure and anonymous authentication scheme for mobile edge computing environments[J]. IEEE Internet of Things Journal, 2024, 11(4): 5798-5815.

[13] TIAN J F, WANG Y T, SHEN Y. An identity-based authentication scheme with full anonymity and unlinkability for mobile edge computing[J]. IEEE Internet of Things Journal, 2024, 11(13): 23561-23576.

[14] TIAN J F, WANG Y T, SHEN Y. A security-enhanced certificateless anonymous authentication scheme with high computational efficiency for mobile edge computing[J]. IEEE Transactions on Network and Service Management, 2025, 22(4): 3555-3572.

[15] 周彦伟, 许渊, 杨波, 等. 面向移动边缘计算的广播身份认证协议[J]. 中国科学: 信息科学, 2023, 53(9): 1734-1749.
ZHOU Y W, XU Y, YANG B, et al. Broadcast identity authentication scheme for mobile edge computing[J]. Scientia Sinica (Informationis), 2023, 53(9): 1734-1749.

[16] MARCHAND C, BOSSUET L, MUREDDU U, et al. Implementation and characterization of a physical unclonable function for IoT: a case study with the TERO-PUF[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(1): 97-109.

[17] PAPPU R, RECHT B, TAYLOR J, et al. Physical one-way functions[J]. Science, 2002, 297(5589): 2026-2030.

[18] LI S S, ZHANG T K, YU B, et al. A provably secure and practical PUF-based end-to-end mutual authentication and key exchange protocol for IoT[J]. IEEE Sensors Journal, 2021, 21(4): 5487-5501.

[19] LIU G, LI H, LIANG Y F, et al. PSRAKA: physically secure and robust authenticated key agreement for VANETs[J]. IEEE Transactions on Vehicular Technology, 2025, 74(5): 7953-7968.

[20] LI S S, HUANG Y C, YU B. A practical and flexible PUF-based end-to-end anonymous authentication protocol for IoT[J]. Computer Networks, 2024, 247: 110426.

[21] SEIFELNASR M, ALTAWY R, YOUSSEF A. SKAFS: symmetric key authentication protocol with forward secrecy for edge computing[J]. IEEE Internet of Things Journal, 2023, 11(1): 510-525.

[22] DOLEV D, YAO A. On the security of public key protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2): 198-208.

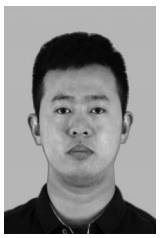
[23] BELLARE M, POINTCHEVAL D, ROGAWAY P. Authenticated key exchange secure against dictionary attacks[C]//Advances in Cryptology — EUROCRYPT 2000. Berlin: Springer, 2000: 139-155.

[24] LIU G, LI H, WANG N, et al. PECHA: privacy-preserving and efficient cross-domain handover authentication for heterogeneous networks[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(3): 2806-2822.

[25] DODIS Y, OSTROVSKY R, REYZIN L, et al. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data[J]. SIAM Journal on Computing, 2008, 38(1): 97-139.

[26] VERCAUTEREN F. Optimal pairings[J]. IEEE Transactions on Information Theory, 2010, 56(1): 455-461.

[作者简介]



李森森（1993– ），男，河南洛阳人，信息工程大学讲师、博士生，主要研究方向为物联网安全、云数据安全、可搜索加密。

黄一才（1985– ），男，湖北恩施人，博士，信息工程大学讲师，主要研究方向为安全云存储系统、物联网安全、可搜索加密。

黄美根（1990– ），男，湖南娄底人，博士，国防科技大学讲师，主要研究方向为数据安全、软件定义网络、物联网安全。

刘燕江（1990– ），男，河南南阳人，博士，信息工程大学讲师，主要研究方向为物理不可克隆函数设计及其应用、安全芯片设计。

郁滨（1964– ），男，河南郑州人，博士，信息工程大学教授、博士生导师，主要研究方向为数据安全、算法设计与分析、视觉密码、网络安全等。