山西大学学报(自然科学版)48(3):505—515,2025
Journal of Shanxi University(Nat. Sci. Ed. )

**DOI**:10.13451/j.sxu.ns.2024007

# 边缘环境下基于动态变色龙认证树的完整性审计

段敬,段婕,万雪枫,刘海涛
(国网山西省电力公司信息通信分公司,山西 太原 030021)

**摘　要**:边缘计算结构比传统云计算更具复杂性,因此数据完整性问题变得尤为重要。针对目前完整性审计方案中仍然存在的隐私保护能力差、计算存储开销大、系统复杂度高、不支持动态操作等问题,本文提出基于动态变色龙认证树的完整性审计方案。该方案在数据加密阶段,引入无证书公钥密码体制,在用户端生成数据加解密钥,保证了数据在系统传输过程中的完整性、机密性;在数据上传阶段,采用动态变色龙认证树存储结构,保证数据存取过程全动态操作的同时,减少通信计算开销;并利用边缘节点的计算能力进行完整性审计,避免了因第三方审计机构而产生的单点失效问题。在随机预言模型下,基于计算性Diffie-Hellman困难问题(Computationa Diffie-Hellman Problem,CDH)和离散对数困难问题证明了本方案的机密性。实验证明,相较于其他完整性审计方案,本方案的计算效率更优,能节省约50%的计算存储开销。

**关键词**:数据完整性;变色哈希函数;无证书公钥密码;默克尔树;动态审计
**中图分类号**:TP309.7　　**文献标志码**:A　　**文章编号**:0253-2395(2025)03-0505-11

# Integrity Audit Based on Dynamic Chameleon Authentication Tree in Edge Environments

DUAN Jing, DUAN Jie, WAN Xuefeng, LIU Haitao
(*Information and Communication Branch of State Grid Shanxi Electric Power Company, Taiyuan* 030021*, China*)

**Abstract:** Edge computing structures are more complex than traditional cloud computing, so data integrity issues become even more important. In order to solve the problems of poor privacy protection ability, high computing and storage overhead, high system complexity, and lack of support for dynamic operation in the current integrity audit schemes, this paper proposes an integrity audit scheme based on dynamic chameleon authentication tree. In the data encryption stage, the scheme introduces a certificateless public key cryptography system to generate a data encryption and decoding key at the user end to ensure the integrity and confidentiality of the data in the process of system transmission; in the data upload stage, the dynamic chameleon authentication tree storage structure is adopted to ensure the full dynamic operation of the data access process and reduce the communication computing overhead; and the computing power of the edge node is used to carry out integrity audit, which avoids the problem of single point failure caused by the third-party audit agency. Under the stochastic oracle model, the confidentiality of the scheme is proved based on the computational DH difficulty problem and the discrete logarithmic difficulty problem. The experiments show that compared with other integrity audit schemes, the proposed scheme has better computing efficiency and can save about 50% of computing and storage overhead.

**Key words:** data integrity; chameleon hash function; certificateless public key cryptography; merkle tree; dynamic auditing

**收稿日期**:2023-11-03;**接受日期**:2024-01-26
**基金项目**:国网山西省电力公司科技项目(52051C220001)
**作者简介**:段敬(1983—),男,山西太原人,正高级工程师,主要研究方向为数字化业务运营管理。E-mail:catduanjing@163.com
**引文格式**:段敬,段婕,万雪枫,等.边缘环境下基于动态变色龙认证树的完整性审计[J].山西大学学报(自然科学版),2025,48(3):505-515. DOI:10.13451/j.sxu.ns.2024007.

## 0 引言

随着智能电表和各种智能设备的发展与普及[1],电力数据规模爆发式增长,传统的电力信息管理系统难以满足大量数据的实时处理需求,指数级增长的海量数据给电力信息管理系统带来了新的机遇与挑战[2-3]。如何保证数据的完整与机密性一直以来都是一个亟待解决的难题。而传统云计算模式逐渐被云边协同的工作模式所取代[4]。边缘节点的加入,利用边缘设备处理海量数据的特性,能很好弥补传统云计算中存在的低带宽、高延迟的问题,但因其结构更具复杂性,因此研究边缘环境下数据的完整性是非常重要的[5]。

传统的云存储完整性审计方案采用第三方审计的方式,但由于第三方审计服务不可信,则有可能出现数据泄露或单点失效的问题[6]。因此在边缘环境下,如何消除不可信第三方带来的影响是个亟待解决的问题。此外,传统审计方案中云服务的数据存储架构不支持动态操作,无法适应海量数据的增长,因此如何动态存取云服务中数据的同时还可以节省此过程中产生的时间通信开销是关注的重点。另外,由于边缘节点的计算存储能力有限,仍存在如何降低系统复杂度的同时也要保证数据机密性和完整性的问题。

为了解决上述问题,本文假定边缘节点以及云存储服务器都是半可信的情况下,提出一个适用于智能电网的基于动态变色龙认证树的完整性审计方案。本文的主要贡献如下。

(1)本文假定边缘节点具有一定的计算、存储能力,但其能力不强远小于云存储服务器。在此情况下,利用边缘节点的计算能力,将数据完整性审计工作交由边缘节点来完成,以此消除不可信第三方带来的影响,减少用户在审计过程中产生的计算存储开销的同时降低系统的复杂度。

(2)通过将Merkle树与变色龙哈希函数结合,在静态变色龙认证树的基础上,构建动态变色龙哈希认证树(Dynamic Chameleon Authentication Tree,DCAT),保证了数据存取过程中的全动态操作。

(3)将传统加密与无证书公钥密码相结合[7],设计了一种适用于边缘环境的完整性审计方案,保证了数据传输过程的机密性、完整性。

(4)在随机预言模型下,基于计算性DH困难问题和离散对数困难问题证明了本方案的机密性。并与其他审计方案相比,本文方案效率更高。

## 1 相关工作

数据持有性证明(Provable Data Possession,PDP)是Ateniese等于2007年首先提出的[8],这是第一个不分块且支持公开验证的数据完整性审计方案。可恢复性证明(Proof of Retrievability,POR)是由Shacham等[9-10]在2008年提出的,是对PDP的扩展,不仅能够验证远程数据是否被篡改,还可以保证数据的可恢复性。后续专家学者在此基础上,针对不同的方向如:数据机密性、批量审计、动态操作等对此问题进行扩展研究,提出了诸多方案[11-13]。

随着边缘计算的概念越来越广为人知,一些应用于边缘环境的数据完整性审计方案也随之被提出。Li等[14]针对移动终端设备中计算能力的不足,提出了两种轻量级的隐私保护完整性审计协议,此方案支持批量审计和动态操作,但在完整性验证过程中需要访问所有数据块,造成了过高的计算复杂性和存储空间消耗。Lin等[15]在移动云计算环境下提出了两种移动数据可持有性证明方案,通过构造基于散列树的数据结构来支持动态数据操作,同时结合BLS(Boneh-Lynn-Shacham)短签名方法实现高效率、低复杂度的完整性审计。Zhou等[16]基于区块链去中心化的特性提出了在边缘环境下基于区块链的数据完整性审计证明,并证明了与其他方案相比更具低延迟的特性。李桐等[17]提出了基于变色龙认证树的流式数据完整性验证模型,数据存储模型适合“流式数据”的需要,但是仍存在用户绕过边缘节点直接访问云存储中的数据,用户端计算量大以及存储结构只能进行部分动态操作等问题。

综上所述,目前也有研究将区块链技术应用于智能电网中进行完整性审计[18-19],但审计方案无法解决隐私保护能力差、计算存储开销

大、系统复杂度高、不支持动态操作等问题。智能电网中每天需要更新海量数据,现有方案并不能同时满足这些要求。因此本文提出的完整性审计方案能在一定程度上解决上述问题,在保证数据机密性的同时,降低用户终端一侧的计算通信开销,与此同时,满足数据在存储过程中的全动态操作。

## 2 基本知识介绍

### 2.1 困难问题

(1)离散对数问题(Discrete Logarithm,DL):令 $p$ 和 $q$ 是满足条件 $p|q-1$ 的两个大素数,设 $g$ 是群 $Z_p^*$ 上阶为 $q$ 的任意生成元,给定元组 $g, g^b \in Z_p^*$(其中 $b \in Z_q^*$ 且未知),DL 问题的目的是计算 $b$。

算法 $A$ 概率多项式时间内成功解决 DL 问题的概率为 $\in_{dl}(A)=\Pr[A(g,g^b)=b]$,其中概率来源于算法 $A$ 的随机选择和 $b$ 在 $Z_q^*$ 上的随机选取。

DL 假设。对于任意的概率多项式时间算法 $A$,优势 $\in_{dl}(A)$ 是可忽略的。

(2)计算性 Diffie-Hellman 困难问题(Computationa Diffie-Hellman Problem,CDH):令 $p$ 和 $q$ 是满足条件 $p|q-1$ 的两个大素数,设 $g$ 是群 $Z_p^*$ 上阶为 $q$ 的任意生成元,给定元组 $g, g^a, g^b \in Z_p^*$(其中 $a,b \in Z_q^*$ 且未知),CDH 的目的是证明 $g^{ab} \in Z_p^*$。

算法 $A$ 概率多项式时间内成功解决 CDH 问题的概率为 $\in_{cdh}(A)=\Pr[A(g,g^a,g^b)=g^{ab}]$,其中概率来源于算法 $A$ 的随机选择和 $a,b$ 在 $Z_q^*$ 上的随机选取。

CDH 假设。对于任意的概率多项式时间算法 $A$,优势 $\in_{cd}(A)$ 是可忽略的。

### 2.2 变色龙哈希函数

传统哈希函数的难以找到碰撞,变色龙哈希[20](Chameleon Hash,CH)可以人为的设一个“弱点”或者“后门”,掌握了这个后门就可以轻松地找到哈希碰撞。任何人可以通过给定的公钥 $p_k$ 进行变色龙哈希后,拥有私钥 $s_k$ 的用户可以广义地找到哈希碰撞,即使得 $a_{hash}(m')=a_{hash}(m)$。

变色龙哈希函数主要由四部分组成,分别为变色龙哈希函数密钥生成算法、变色龙哈希生成算法、变色龙哈希验证算法以及变色龙哈希碰撞算法,具体如下所示:

(1)变色龙哈希函数密钥生成算法:给定一个安全常数 $\lambda$,输出变色龙哈希函数的公钥 $p_k$ 和私钥 $s_k$(陷门)。表示为:$a_{gan}(1^\lambda)=(p_k,s_k)$。

(2)变色龙哈希生成算法:输入公钥 $p_k$、随机数 $r$ 和任意消息 $m$,生成哈希值 $h$ 和随机数 $P$。表示为:$a_{hash}(p_k,m,r)=(h,p)$。

(3)变色龙哈希验证算法:输入公钥 $p_k$、任意消息 $m$ 和哈希值 $h$、随机数 $P$,若 $(h,p)$ 是正确的哈希值,则输出 1,否则输出 0。表示为:$a_{ver}(p_k,m,(h,p))$。

(4)变色龙哈希碰撞算法:输入私钥 $s_k$(陷门)、消息 $m$、新消息 $m'$ 和哈希值 $m'$、随机数 $P$,输出新随机数 $r'$,使得

$$a_{ver}(p_k,m,(h,p),r)=a_{ver}(p_k,m',(h,p),r')$$

表示为:$a_{cld}(s_k,m,m',(h,p))$。

变色龙哈希函数形式如下:

$$\text{cham\_hash}=(a_{gen},a_{hash},a_{ver},a_{cld})$$

### 2.3 静态变色龙认证树

传统的云存储服务数据存储采用 Merkle 树架构在云上存储,此方案优点在于可以获取 Merkle 根节点数据值直接进行数据完整性判断,但构建 Merkle 树时需要一次性获得所有数据,不适合数据量飞速增长的现实环境,因此将变色龙哈希函数与 Merkle 树相结合,利用其“碰撞”特性,扩大其应用场景。

静态变色龙认证树[21]形式化定义:变色龙认证树是一个由多项式时间概率算法(Probabilistic Polynomial-Time Algorithm,PPT)构成的元组,一般包含四个部分:结构初始化,数据添加,数据查询,数据完整性验证。形式如下:$\text{CAT}=(d_{Gen},d_{Append},d_{Query},d_{Verify})$。

(1)结构初始化:$d_{Gen}(1^\lambda,D)$。静态变色龙认证树初始化算法,输入安全参数 $\lambda$ 以及树的深度 $D$,返回一对公私钥 $(p_k,s_k)$。静态变色龙认证树是一种二叉树,类似于默克尔树只有叶子节点存储数据,非叶子节点由其子节点运算而成,因此深度 $D$ 决定整棵树能存储的数据量大小,即假设一棵变色龙认证树的深度为 $d$,则其能存储的最大数据量为 $N_{max}=2^{d-1}$。

(2)数据添加:$d_{\mathrm{Append}}(s_{\mathrm{k}},d)$。静态变色龙认证树添加数据算法,使用私钥进行“碰撞”,更改其随机数,保证数据添加前后变色龙哈希函数值不变,并返回更新后的认证树信息。

(3)数据查询:$d_{\mathrm{Query}}(i)$。查询认证树中的第$i$个数据,若查询成功,则返回证明路径$a$(包含第$i$个数据到root节点路径上的所有节点数据,以及路径中节点的兄弟节点),若查询成功则输出1,若查询不成功则输出0。

(4)数据完整性验证:$d_{\mathrm{Verify}}(p_{\mathrm{k}},i,d,a)$。使用私钥$p_{\mathrm{k}}$,认证路径$a$,验证根节点数据是否一致,若一致,则数据完整性没有被破坏,若不一致则数据已经被篡改。

## 3 方案设计

### 3.1 整体框架

本文采用云边端三者协同的方式,涉及云、边缘、终端三层,云存储服务器(Cloud Storage on Private,CSP)、边缘节点(Edge Node,EN)、用户终端(Data Owner,DO)以及密钥生成中心(Key Generating Centre,KGC)四类实体:

(1)云存储服务器(CSP):用于存储由边缘节点上传的认证树结构,以及响应由边缘节点发出的数据完整性挑战。

(2)边缘节点:本方案中根据边缘节点存储和计算的能力,将其分为边缘_存储节点(E_S)以及边缘_审计节点(E_A)。

①边缘存储节点:用于接收由用户终端输入的加密数据,构建动态变色龙认证树,以及响应由边缘_审计节点发出的数据完整性挑战。

②边缘_审计节点:用于接收用户发来的数据标识进行完整性挑战,以及接收由边缘_存储节点和云存储服务器返回的证明,计算后并将结果返回给用户。

(3)用户终端(DO):用于与密钥生成中心协商数据加密密钥,并将加密后的数据发送给边缘节点,以及在挑战-相应阶段向边缘节点发起完整性挑战请求。

(4)密钥生成中心(KGC):用于在系统初始化阶段生成变色龙认证树公私钥以及数据加解密公私钥。

### 3.2 形式化定义

本文方案形式化定义为IADCAT=($C_{\mathrm{Gen}}$,$D_{\mathrm{Gen}}$,$k_{\mathrm{Part}}$,$k_{\mathrm{Complete}}$,$d_{\mathrm{Encry}}$,$D_{\mathrm{Append}}$,$D_{\mathrm{Query}}$,$D_{\mathrm{Verify}}$)。

$C_{\mathrm{Gen}}(1^{\lambda})$:无证书公钥密码体制初始化算法。输入安全参数$\lambda$,输出系统公共参数$P$。

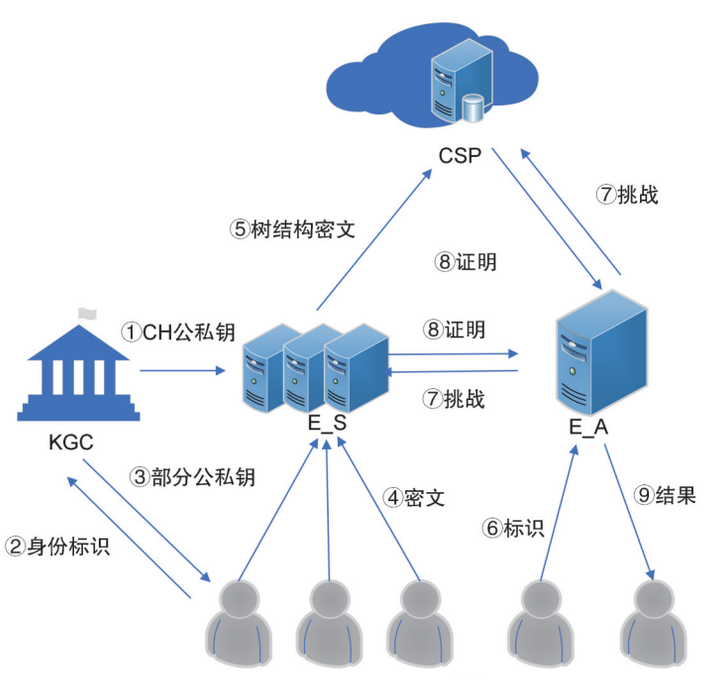


**图1** 系统流程图

**Figure 1** Flow chart of system

$D_{\mathrm{Gen}}(1^{\lambda})$:动态变色龙认证树的初始化算法,输入安全参数$\lambda$,生成一对变色龙hash函数的公私钥$(c_{\mathrm{pk}},c_{\mathrm{sk}})\leftarrow C_{\mathrm{Gen}}(1^{\lambda})$,生成的用户公私钥$(s_{\mathrm{pk}},s_{\mathrm{sk}})$,初始化数据库$D_{\mathrm{B}}$为空和树的结构$C_{\mathrm{Struct}}$,返回$(c_{\mathrm{sk}},s_{\mathrm{sk}},C_{\mathrm{Struct}})$作为私钥$s_{\mathrm{k}}$,$(c_{\mathrm{pk}},s_{\mathrm{pk}})$作为公钥$p_k$。

$R_{\mathrm{Part}}(I_{\mathrm{d}},X)$:KGC生成部分数据加密密钥算法,输入用户身份$I_{\mathrm{d}}$和公开参数$X$,输出用户部分公私钥$(Y,y)$。

$k_{\mathrm{Complete}}(Y,y)$:用户生成完整数据加密密钥算法,输入KGC返回的部分公钥$Y$和部分私钥$y$,输出完整的数据加解密密钥$\langle(X,x),(Y,y)\rangle$。

$d_{\mathrm{Encry}}(m,s_{\mathrm{pk}})$:数据加密算法,输入明文数据$m$,和用户公钥$s_{\mathrm{pk}}$,输出加密后的密文$C$。

$D_{\mathrm{Append}}(s_{\mathrm{k}},a)$:添加数据算法,使用动态变色龙私钥$s_{\mathrm{k}}$将数据$d$添加到动态变色龙认证树中。用更新后的树结构信息$d_{\mathrm{Struct}}$更新私钥$s_{\mathrm{k}}$。

$D_{\mathrm{Query}}(i)$:查询动态变色龙认证树中的第$i$个元素$d[i]$,若查询成功,返回相应的认证路径$a_i$输出1,查询失败则输出0。

$D_{\mathrm{Verify}}(p_{\mathrm{k}},i,d,a)$:使用$p_{\mathrm{k}}$、$a$验证$d$是不是动态变色龙认证树中的第$i$个元素。若验证成功则输出1,验证不成功则输出0。

### 3.3 具体设计

系统首先进行初始化,初始化系统参数并生成变色龙公私钥,当用户接入系统后,KGC会根据用户发来的身份标识生成部分公钥返回用户,在用户端生成完整公私钥,当数据上传时,先将数据分块加密然后上传至边缘节点构建DCAT[22],再将认证树结构以及加密数据一起上传至云服务器。挑战-相应阶段时,用户向系统发送数据完整性挑战,边缘节点会根据用户发来的标识,在边缘_审计节点处完成计算生成结果返回给用户。系统符号说明见表1。

**表1** 系统符号化说明

**Table 1** System symbolization

| 参数 | 含义 |
|---|---|
| $H_1$, $H_2$ | 安全的哈希函数 |
| $s$ | 密钥生成中心主密钥 |
| $d_{\mathrm{r}}$ | 根节点数据 |
| $n$ | 数据总量 |
| $c$ | 变色龙认证树容量 |
| $d_{\mathrm{v}}$ | 认证树深度 |
| $d_{\mathrm{Struct}}$ | 认证树的结构 |
| $D_{\mathrm{B}}$ | 存储数据的数据库 |
| $(p_{\mathrm{k}},s_{\mathrm{k}})$ | 系统公私钥 |
| $l$ | 左子树根节点 |
| $r$ | 右子树根节点 |

系统一共包括四个阶段,接下来将对每一个阶段进行具体说明:

(1)初始化阶段。初始化阶段包括系统参数生成以及动态变色龙认证树的公私钥生成。系统初始化阶段,主要进行以下的操作:

①系统参数生成[23]:执行无证书公钥密码体制初始化算法,输入安全参数$k$,输出满足条件$p|q-1$的两个大素数$p$和$q$,$g$为$Z_p^*$中任意阶为$q$生成元,定义抗碰撞的安全哈希函数:$H_1:\{0,1\}^{L_1}\times Z_p^*\times Z_p^*\rightarrow Z_q^*$,$H_2:\{0,1\}^{L_1}\times\{0,1\}^{L_2}\times Z_p^*\rightarrow Z_q^*$,定义异或运算$\oplus$连接符$/\!/$,随机选取系统主密钥$s\in Z_q^*$,计算$P_{\mathrm{pub}}=g^s$,公开$P=\langle p,q,g,P_{\mathrm{pub}},H_1,H_2,\oplus,/\!/\rangle$,秘密保存主密钥$s$。

②变色龙认证树的公私钥的生成:输入安全参数$k$,执行变色龙哈希函数初始化算法生成一对变色龙hash函数的公私钥$(c_{\mathrm{pk}},c_{\mathrm{sk}})\leftarrow c_{\mathrm{Gen}}(1^{\lambda})$,$c_{\mathrm{sk}}=x\in Z_p^*$,$c_{\mathrm{pk}}=g^x \bmod p$,初始化数据库$D_{\mathrm{B}}$、树的结构$c_{\mathrm{Struct}}$、树的深度$d_{\mathrm{v}}$、根节点$R$、树容量$n$为空,返回$(c_{\mathrm{sk}},s_{\mathrm{sk}},c_{\mathrm{Struct}})$作为私钥$s_{\mathrm{k}}$,$(c_{\mathrm{pk}},s_{\mathrm{pk}})$作为公钥$p_{\mathrm{k}}$。

(2)用户与KGC协商密钥并对数据进行加密阶段。在初始化阶段用户公私钥并未生成,其是在用户接入系统后与KGC协商实现的,此阶段分为以下三个部分。

①密钥提取:随机选取秘密值$x$,计算$X=g^x$,发送身份标识$I_{\mathrm{d}}$和公开参数$X$给KGC,KGC随机选取秘密值$r\in Z_q^*$,分别计算$Y=g^r$和$y=r+sH_1(I_{\mathrm{d}},X,Y)$。

②生成完整密钥:用户收到KGC返回的部分公私钥对,将其与用户自身的部分公私钥对相结合,得到完整的公私钥对$\langle s_{\mathrm{pk}}=(X,x),s_{\mathrm{sk}}=(Y,y)\rangle$。

③数据加密阶段:数据加密阶段主要在用户端完成,用户先对要上传的数据进行分块,$F=m_1/\!/m_2/\!/\cdots/\!/m_n,m_i\in Z_q^*$,然后随机选取

秘密值$x$,计算$R=g^x$,计算$h_1=H_1(I_d,X,Y)$,$V=(XYP_{pub}^{h_1})^x$和$U=d(x+y)xf$,生成密文$C=(m/\!/U)\oplus H(V)$,$h=H(I_d,R,C)$和$S=x(x+y+h)^{-1}$,并将生成的密文数据$\delta=(h,S,C)$传输给边缘_存储节点。

(3)构建动态变色龙认证树阶段:动态变色龙认证树是在静态变色龙认证树基础上进行了改进,因为在初始化阶段并没有定义认证树能容纳的最大数据量,因此DCAT更适合数据量不清的情况,在构建时,首先要考虑是否要对树结构进行扩展,具体算法如算法1,算法1第1行表示,判断此时的数据量是否超过目前认证树所能容纳的最大数据量,算法1第2—8行是指需要扩展的情况,先将认证树容量扩大两倍,然后将认证树结构作为新认证树的左子树,根据左子树的结构构建右子树,最后采用深度优先搜索遍历算法,形成新的认证树结构,最后添加数据,构造"碰撞"。

**算法1** 动态变色龙认证树扩展算法

输入:加密数据$C$
输出:变色龙认证树结构$d_{Struct}$
1 $e\leftarrow(n==c)$
2 if($e$)
3 $c\leftarrow 2c$
4 $l\leftarrow r$
5 $r\leftarrow$catappend($d_{cpk},C$)
6 $d_r\leftarrow$H($l,r$)
7 $d_{Struct}\leftarrow$bfs($l_c,r_c$)
8 else
9 Col($d_{csk},C$)
10 $D_B\leftarrow C$
11 return($d_{Struct}$)

执行添加数据算法,图2为数据添加示意图,图中展现了动态变色龙认证树的树结构扩展情况,图中"key"表示根节点数据,"ch"表示变色龙哈希节点数据,"h"表示哈希节点数据,"d"表示傀儡节点数据,只为构造树结构,无实际含义。根据树的不断扩展,可以看出,直线左边左子树的结构与未扩展前树结构一致,这也是动态变色龙认证树易扩展的一个原因。根据数据上传位置选择普通哈希算法加密$H(\delta)$或变色龙哈希算法加密$C_H(\delta,r)=g^{\delta}c_{pk}^{r}\bmod p$。然后自底向上进行"碰撞",假设$m_1$为原傀儡节点数据,$r_1$为原傀儡节点随机数,则原傀儡节点的变色龙哈希值为$C_H(m_1,r_1)=g^{m_1}c_{pk}^{r_1}\bmod p$,而添加数据后,$m_2$实际节点数据,$r_2$为实际节点随机数,为了使其"碰撞",即$C_H(m_1,r_1)=C_H(m_2,r_2)$,则$r_2=\dfrac{m_1-m_2}{x}+r_1\bmod p$。

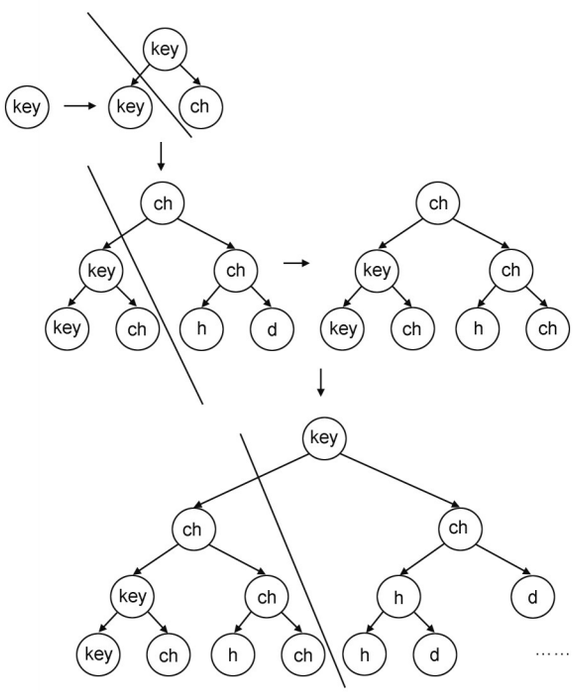


**图2** 数据添加示意图

**Fig. 2** Schematic diagram of data addition

**证明**

$$C_H(m_1,r_1)=g^{m_1}y^{r_1}\bmod p=g^{m_1}g^{xr_1}\bmod p=g^{m_1+xr_1}\bmod p,$$

$$C_H(m_2,r_2)=g^{m_2}y^{r_2}\bmod p=g^{m_2}g^{xr_2}\bmod p=g^{m_2+xr_2}\bmod p,$$

若证明:

$$C_H(m_1,r_1)=C_H(m_2,r_2),$$

即:

$$g^{m_1+xr_1}\bmod p=g^{m_2+xr_2}\bmod p,$$

则:

$$r_2=\frac{m_1+xr_1-m_2}{x}\bmod p=\frac{m_1-m_2}{x}+r_1\bmod p。$$

(4)挑战-认证阶段:$d[i]\rightarrow 1/0$,此阶段,用户作为数据完整性验证发起方,向边缘_审计节点发送需要验证信息的数据标识$d[i]$,审计系统执行数据查找算法$f(d[i])$,先向边缘_

存储节点和云存储服务器发送完整性挑战请求,查询数据是否存在,若存在,则根据返回的数据位置$ad$以及数据标识$d[i]$,返回完整性证明路径$a_p$,若不存在则返回0,返回的验证路径在边缘_审计节点处进行计算,验证返回的根节点信息与计算得到的根节点数据是否一致,若一致,则证明数据完整性没有被破坏,若不一致,则数据完整性结果已经被破坏,并将计算结果返回给用户,具体算法如算法2,证明路径示意图见图3。

**算法2** 挑战-认证阶段算法

输入:数据标识$d[i]$
输出:完整性验证结果1/0
1 $a_d \leftarrow f(d[i])$
2 if($ad$)
3 $a_p \leftarrow$ atth($d[i]$,$a_d$)
4 $r_s = c_h(a)$
5 if($r_s == r$)
6 return(1)
7 else
8 return(0)
9 else
10 return(0)

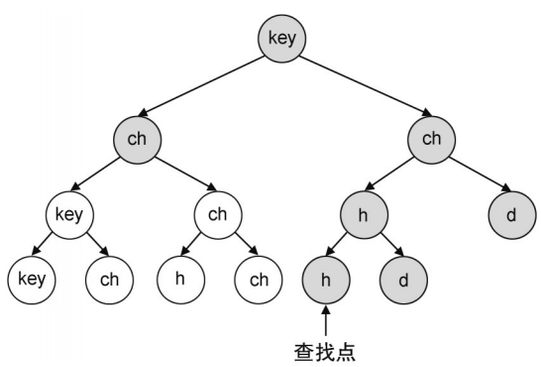


**图3** 证明路径示意图

**Fig. 3** Schematic diagram of the proof path

## 4 安全性分析

本文将从模型安全性和密码安全性两个方面对本文方案进行安全性分析。

### 4.1 模型安全性

本文利用边缘节点的计算能力代替第三方审计实体,规避了可能因第三方审计服务不可信、单点失效的问题,系统中所有数据皆是以密文形式进行存储、传输,在安全方面有保障。

### 4.2 攻击模型分析

基于无证书公钥密码机制和变色龙认证树,本文提出了两类敌手攻击:Type-Ⅰ类型敌手$A_1$,一般用户攻击,不能获取KGC主密钥但能替换任意用户私钥;Type-Ⅱ类型敌手$A_2$,恶意KGC攻击,可以获取KGC主密钥但不能替换用户私钥。

首先,挑战者$C$执行本文方案的系统初始化阶段,生成主密钥$s$及系统参数$P$,初始化变色龙公私钥对$(p_k, s_k)$。

Type-Ⅰ类型敌手$A_1$,基于CDH问题困难性假设,在随机预言模型下,若本方案对敌手是不可伪造的,则本文方案是安全的。

**证明** 对于此类攻击,敌手可以随意替换用户公钥$(X,x)$,故挑战者设秘密值$r \in Z_q^*$,则计算$Y = g^r$和$y = r + sH_1(I_d, X, Y)$,$I_d$为用户身份信息,则完整的公私钥对为$\langle s_{pk} = (X,x), s_{sk} = (Y,y)\rangle$。

假设明文数据为$m$,生成密文$h = H(I_d, R, C)$,$C = (m /\!/ U) \oplus H(V)$和$S = x(x + y + h)^{-1}$,即密文数据为$\delta = (h, S, C)$。

假设在挑战-认证阶段,挑战者利用哈希重放询问同一个挑战,从而生成两个不同的证据

$$e(C_{H_1}, g) = e(H_1(I_d(X,x)_1)R, (Y,y)_1)e(H_1(P_{pub}), \mu_1, g),$$

$$e(C_{H_2}, g) = e(H_1(I_d(X,x)_2)R, (Y,y)_2)e(H_1(P_{pub}), \mu_2, g),$$

则

$$r_1 r_2 g = ((s_2 - s_1)R)^{-1}((C_{H_1} - C_{H_2}) - H_1(P_{pub})(\mu_1 - \mu_2))。$$

如果$A_1$能够被攻破,则说明上式有解,与假设矛盾。

Type-Ⅱ类型敌手$A_2$,基于CDH问题困难性假设,在随机预言模型下,若本方案对敌手是不可伪造的,则本文方案是安全的。

**证明** 对于此类攻击,敌手可以随意更换KGC主密钥但不能替换用户私钥。则设置$P_{pub} = g^s$,其中$s$为系统主密钥。挑战者猜测$P = g^{s'}$,用户随机选取秘密值$x$,计算$X = g^x$,发送身份标识$I_d$和公开参数$X$给KGC,KGC随机选取秘密值$r \in Z_q^*$,分别计算$Y = g^r$和$y =$

$r+sH_1(I_d,X,Y)$。

假设在挑战-认证阶段,挑战者利用哈希重放询问同一个挑战,从而生成两个不同的证据

$$e(C_{H_1},g)=e(g^{s'}R,sg^{s})e(H_1(P_{pub}),\mu_1,g),$$

$$e(C_{H_2},g)=e(g^{s'}R,s'g^{s})e(H_1(P_{pub}),\mu_2,g),$$

则

$$r_1r_2g=((s_2-s_1)R)^{-1}((C_{H_1}-C_{H_2})-H_1(P_{pub})(\mu_1-\mu_2))。$$

如果$A_2$能够被攻破,则说明上式有解,与假设矛盾。

综上所述,本文方案能够抵抗两类敌手攻击,因此本文方案是安全的。

## 5 实验分析

### 5.1 实验环境配置

本文方案共涉及4种类型的实体,全部部署在本地计算机上,采用Intel(R)Core(TM)i7-8565U CPU@1.80 GHz, 1.99 GHz处理器、8 GB内存。使用Java和Python进行代码编写,采用gmpy2库实现无证书公钥密码,以及JPBC库实现变色龙哈希算法。实验结果均为多次计算后的平均值。

### 5.2 性能分析

边缘资源条件有限,终端用户的计算能力有限,不能将计算量大的工作交由终端用户来完成,终端用户进行的计算量应尽可能少。本文将本方案与其他审计方案(基于证书的可证明数据持有方案(Certificate-based Provable Data Possession,CLPDP)、基于时间戳的动态哈希密钥加权完整性审计方案(Timestamp-based Hash Key Weighted-Integrity Checking with Write Capability,tHKW-WC)、基于证书的可证明审计方案(Certificate-based Provable Audit Scheme,CL-PAS)、跨云虚拟机服务的可共享数据完整性验证方案(Cross-Cloud Integrity Verification Scheme for Shared Data, CCIVS-SD)、文献[13])中数据上传用户需进行的计算量进行对比,证明本文的计算量更低。

本文方案引用动态变色龙认证树后支持批量审计动态操作。因为DCAT的树结构在数据插入时会将其结构先扩展一倍,并且每次扩展前后整体结构不变,并且其具有基本的变色龙哈希函数的"碰撞"特性,支持动态操作。表2中将目前已有的数据完整性审计方案与本文的审计方案从批量审计、动态操作、可扩展性这三个方面进行对比,结果如表3所示。

**表2** 数据上传的时间复杂度分析

**Table 2** Time complexity analysis of data uploading

| 方案 | 用户层 | 边缘层 | 边缘节点 |
|---|---|---|---|
| CLPDP | $nH+2nM$ | — | — |
| tHKWWC | — | — | — |
| CL-PAS | $nH+2nM$ | $2nH+2nM$ | — |
| CCIVS-SD | $nH+nM+2nE$ | $nH+nM+2nE$ | 可信 |
| 文献[13] | $nM+2nE$ | — | 半可信 |
| 本文方案 | $2nE$ | — | — |

注:$H$表示群上的哈希运算,$M$表示群上的点乘运算,$E$表示群上的指数运算。"—"表示该方案在该层的时间复杂度未定义或不适用。

Note: Where $H$ represents the hash operation on the group and $M$ represents the dot multiplication operation on the group, $E$ denotes exponential operations on groups. "—" indicates that the time complexity for that layer is undefined or not applicable.

**表3** 本文方案在批量审计、动态操作、可扩展性三方面与现有完整性审计方案比较

**Table 3** This solution compares with the existing integrity audit scheme in three aspects: batch audit, dynamic operation, and scalability

| 文献 | 批量审计 | 动态操作 | 可扩展性 |
|---|---|---|---|
| 文献[15] | 支持 | 不支持 | 高 |
| 文献[16] | 不支持 | 支持 | 高 |
| 文献[17] | 不支持 | 支持 | 高,消耗大 |
| 文献[24] | 不支持 | 部分动态 | 低 |
| 文献[22] | 不支持 | 部分动态 | 低 |
| 本文方案 | 支持 | 支持 | 支持 |

### 5.3 实验对比

本文提出一种面向边缘计算环境下的基于动态变色龙认证树的数据完整性审计方案,通过引入动态变色龙认证树这一数据存储结构,解决了数据无法动态存储的问题,并节省了终端用户以及审计工作时产生的通信计算开销,为证明这一优点,本文从数据插入更新的长度、时间以及认证所需的证据长度和认证时间等角度,将本文方案与文献[16]和文献[17]中的完整性审计方案进行实验对比,文献[16]采用Merkle树进行数据存储,而文献[17]则采用静态变色龙认证树存储结构,通过比较以上两

种性能指标,衡量整个模型在数据存储与认证阶段的通信计算开销。

由于Merkle树在构建的时候需要一次性获得所有的数据节点,每次构建树会产生大量的时间通信开销,本文采用动态变色龙认证树构架,利用其“碰撞”的特性节省了计算通信开销。

从图4和图5可以看出,随着插入数据量的增长,平均更新时间与平均更新的长度相对应有所增加,最终会趋于一个定值,相较于其他方案更优。

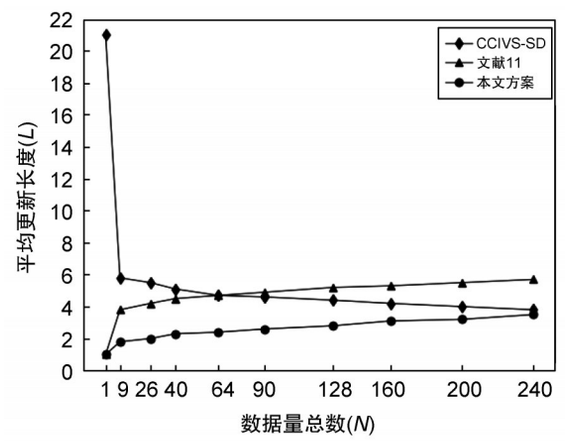


**图4** CCIVS-SD方案、文献[16]方案与本文方案中随着数据量不断增加认证树的平均更新长度的变化

**Fig. 4** The change of the average update length of the authentication tree in the CCIVS-SD scheme, the Ref. 16 scheme and the proposed scheme in this paper increases with the increase of data volume

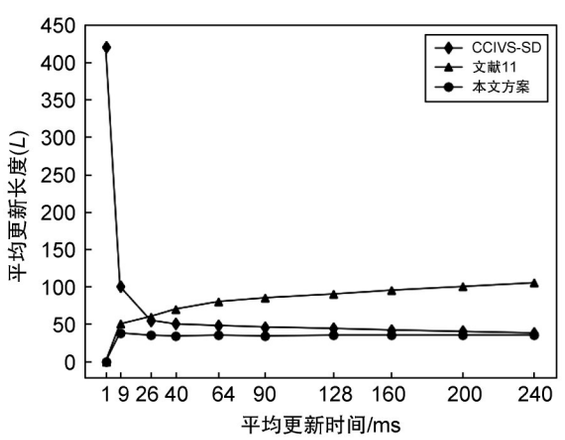


**图5** CCIVS-SD方案、文献[16]方案与本文方案中认证树的平均更新长度与更新时间的开销比较

**Fig. 5** The average update length and update time of the authentication tree in the CCIVS-SD scheme, the Ref. 16 scheme and the proposed scheme are compared with the overhead of the certificate tree

从图6和图7可以看出,随着系统中数据总数的增长,数据平均验证时间以及返回的认证路径长度相对应有所增加,但整体偏低,与其他方案相比结果更优。

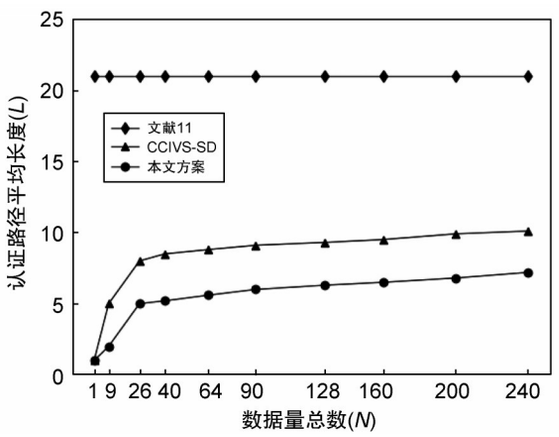


**图6** CCIVS-SD方案、文献[16]方案与本文方案中随数据量总数的增加认证路径的平均长度比较

**Fig. 6** The average length of the authentication path is compared with the CCIVS-SD scheme and the ref. 16 scheme with the increase of the total number of data in the scheme in this paper

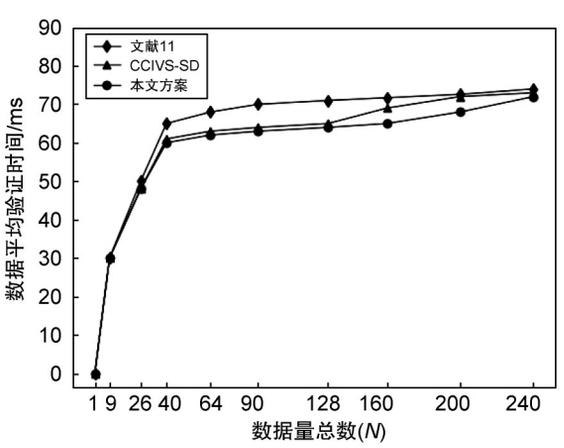


**图7** CCIVS-SD方案、文献[16]方案与本文方案中数据量对数据审计的时间开销比较

**Fig. 7** Comparison of the time cost of data audit in the CCIVS-SD scheme, the 16 scheme in ref. 16 and the scheme in this paper

## 6 结论

本文利用无证书公钥加密体制和动态变色龙认证树模型,针对电力信息管理系统中目前海量数据的机密性完整性问题,提出了智能电网中基于变色龙认证树的数据完整性审计方案,利用无证书公钥加密体制和变色龙哈希算

法保证数据的机密性,并在此基础上引入认证树的动态操作,扩大了其模型的应用场景,实验证明,本文方案时间通信开销更优。

**参考文献:**


[1] 彭小圣, 邓迪元, 程时杰, 等. 面向智能电网应用的电力大数据关键技术[J]. 中国电机工程学报, 2015, **35**(3): 503−511. DOI: 10.13334/j.0258-8013.pcsee.2015.03.001.
PENG X S, DENG D Y, CHENG S J, *et al*. Key Technologies of Electric Power Big Data and Its Application Prospects in Smart Grid[J]. *Proc CSEE*, 2015, **35**(3): 503−511. DOI: 10.13334/j. 0258-8013. pcsee.2015.03.001.

[2] 陈冬,周潭平,宋子超等.智能电网中隐私保护的数据聚合研究综述[J]. 密码学报,2023, **10**(6): 1 − 13. DOI: 10.13868/j.cnki.jcr.000653.
CHEN D, ZHOU T P, SONG Z C, *et al*. A Review of Data Aggregation on Privacy Protection in Smart Grid[J]. *J Cryptol Res*, 2023, **10**(6): 1 − 13. DOI:10.13868/j.cnki.jcr.000653.

[3] 李千叶, 郎帅. 基于物联网技术的智能电网数据安全问题分析[J]. 模具制造, 2023, **23**(11): 193−195. DOI: 10.12147/j.cnki.1671-3508.2023.11.061.
LI Q Y, LANG S. Analysis of Data Security Issues in Smart Grid Based on IoT Technology[J]. *Die Mould Manuf*, 2023, **23**(11): 193−195. DOI: 10.12147/j.cnki.1671-3508.2023.11.061.

[4] 程钎, 陈羽, 孙伶雁. 考虑服务配置的细粒度电力任务云边协同优化调度策略[J]. 电力系统保护与控制, 2023, **51**(7): 53−62. DOI: 10.19783/j.cnki.pspc.221116.
CHENG Q, CHEN Y, SUN L Y. Cloud-edge Collaborative Optimization Scheduling Strategy for Fine-grained Power Tasks Considering Service Configuration [J]. *Power Syst Prot Contr*, 2023, **51**(7): 53−62. DOI: 10.19783/j.cnki.pspc.221116.

[5] 张佳乐, 赵彦超, 陈兵, 等. 边缘计算数据安全与隐私保护研究综述[J]. 通信学报, 2018, **39**(3): 1−21. DOI: 10.11959/j.issn.1000-436x.2018037.
ZHANG J L /Y), ZHAO Y C, CHEN B, *et al*. Survey on Data Security and Privacy-preserving for the Research of Edge Computing[J]. *J Commun*, 2018, **39**(3): 1−21. DOI: 10.11959/j.issn.1000-436x.2018037.

[6] 王惠莅. 面向云计算环境的数据安全技术研究[D]. 西安: 西安电子科技大学, 2022.
WANG H L. Research on Data Security Technology for Cloud Computing Environment[D].Xi'an: Xidian University, 2022.

[7] 张振超, 刘亚丽, 殷新春, 等. 无证书签名方案的分析及改进[J]. 密码学报, 2020, **7**(3): 389−403. DOI: 10.13868/j.cnki.jcr.000375.
ZHANG Z C, LIU Y L, YIN X C, *et al*. Analysis and Improvement of Certificateless Signature Schemes[J]. *J Cryptologic Res*, 2020, **7**(3): 389−403. DOI: 10.13868/j.cnki.jcr.000375.

[8] ATENIESE G, BURNS R, CURTMOLA R, *et al*. Provable Data Possession at Untrusted Stores[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. Alexandria, Virginia, USA: ACM, 2007: 598−609. DOI: 10.1145/1315245.1315318.

[9] JUELS A, KALISKI B S. Pors: Proofs of Retrievability for Large Files[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. Alexandria, Virginia, USA: ACM, 2007. 584−597. DOI: 10.1145/1315245.1315317.

[10] SHACHAM H, WATERS B. Compact Proofs of Retrievability[C]//Proceedings of the Advances in Cryptology−ASIACRYPT 2008. Berlin, Heidelberg: Springer−Verlag, 2008. 90−107. DOI: 10.1007/978-3-540-89255-7_7.

[11] 李秀艳, 刘明曦, 史闻博, 等. 基于云存储的动态组共享数据完整性验证方案[J]. 计算机工程与设计, 2022, **43**(6): 1510−1519. DOI: 10.16208/j. issn1000-7024.2022.06.002.
LI X Y, LIU M X, SHI W B, *et al*. Dynamic Group Shared Data Integrity Verification Scheme Based on Cloud Storage[J]. *Comput Eng Des*, 2022, **43**(6): 1510−1519. DOI: 10.16208/j.issn1000-7024.2022.06.002.

[12] 刘峰, 赵俊峰. 基于区块链的云存储数据完整性验证方案[J]. 应用科学学报, 2021, **39**(1): 164−173. DOI: 10.3969/j.issn.0255-8297.2021.01.014.
LIU F, ZHAO J F. Cloud Storage Data Integrity Verification Scheme Based on Blockchain[J]. *J Appl Sci*, 2021, **39**(1): 164−173. DOI: 10.3969/j. issn. 0255-8297.2021.01.014.

[13] 雷莹. 云存储系统数据完整性安全审计的研究[D]. 北京: 北京交通大学, 2019.
LEI Y. Research on Data Integrity Security Audit of Cloud Storage System[D]. Beijing: Beijing Jiaotong University, 2019.

[14] LI J T, ZHANG L, LIU J K, *et al*. Privacy-preserving Public Auditing Protocol for Low-performance End Devices in Cloud[J]. *IEEE Trans Inf Forensics Secur*, 2016, **11**(11): 2572−2583. DOI: 10.1109/TIFS.2016.2587242.

[15] LIN C, SHEN Z D, CHEN Q, *et al*. A Data Integrity Verification Scheme in Mobile Cloud Computing[J]. *J Netw Comput Appl*, 2017, **77**(C): 146−151. DOI:

10.1016/j.jnca.2016.08.017.

[16] ZHOU J, JIN Y, HE H, *et al*. Dynamic Audit Model of Cloud Data Based on Nested Merkle Hash Tree Blockchain[J]. *J Netw Comput Appl*, 2019, **39**(12): 3575−3583.

[17] 李桐, 任帅, 王刚, 等. 基于变色龙认证树的云边端协同流式数据完整性验证模型[J]. 信息网络安全, 2022(1): 37−45. DOI: 10.3969/j.issn.1671-1122.2022.01.005.
LI T, REN S, WANG G, *et al*. Cloud-edge-device Collaborative Integrity Verification Scheme Based on Chameleon Authentication Tree for Streaming Data[J]. *Netinfo Secur*, 2022, **22**(1): 37−45. DOI: 10.3969/j.issn.1671-1122.2022.01.005.

[18] 张丽娟. 基于区块链技术的智能电网安全聚合方案[J]. 数字技术与应用, 2023, **41**(10): 229−231. DOI: 10.19695/j.cnki.cn12-1369.2023.10.72.
ZHANG L J. Smart Grid Security Aggregation Scheme Based on Blockchain Technology[J]. *Digit Technol Appl*, 2023, **41**(10): 229−231. DOI: 10.19695/j. cnki.cn12-1369.2023.10.72.

[19] 韦涛, 周治平. 基于区块链的用能数据完整性保护框架[J]. 电力自动化设备, 2021, **41**(12): 102−107. DOI: 10.16081/j.epae.202108010.
WEI T, ZHOU Z P. Integrity Protection Framework for Energy Consumption Data Based on Blockchain[J]. *Electr Power Autom Equip*, 2021, **41**(12): 102−107. DOI: 10.16081/j.epae.202108010.

[20] 李丽娟. 变色龙哈希函数设计及应用研究[D]. 郑州: 河南工业大学, 2014.
LI L J. Chameleon Hash Function Design and Application Research[D]. Zhengzhou: Henan University of Technology, 2014.

[21] 黄雪刚, 高天寒, 李宇溪. 面向流式数据认证的变色龙认证树算法研究[J]. 四川大学学报(工程科学版), 2016, **48**(2): 139−144. DOI: 10.15961/j.jsuese.2016.02.020.
HUANG X G, GAO T H, LI Y X. Research on Chameleon Certification Tree Algorithm for Streaming Data Authentication[J]. *J Sichuan Univ Eng Sci Ed*, 2016, **48**(2): 139−144. DOI: 10.15961/j.jsuese.2016.02.020.

[22] 陈科. 基于动态变色龙认证树的流式数据完整性验证研究与应用[D]. 沈阳: 东北大学, 2014.
CHEN K. Research and Application of Streaming Data Integrity Verification Based on Dynamic Chameleon Authentication Tree[D]. Shenyang: Northeastern University, 2014.

[23] 周彦伟, 杨波, 张文政. 不使用双线性映射的无证书签密方案的安全性分析及改进[J]. 计算机学报, 2016, **39**(6): 1257−1266. DOI: 10.11897/SP.J.1016.2016.01257.
ZHOU Y W, YANG B, ZHANG W Z. Security Analysis and Improvement of Certificateless Signcryption Scheme without Bilinear Pairing[J]. *Chin J Comput*, 2016, **39**(6): 1257−1266. DOI: 10.11897/SP.J.1016.2016.01257.

[24] PRINCE G, DU R Z. Data Integrity Audit Scheme Based on Certificateless Public Key Cryptography in Edge Environment[J]. *J Commun*, 2022, **43**(7): 62−72.