面向泛在电力物联网云端数据的轻型动态完整性审计方案

田秀霞[1)] 刘天顺[1)] 牛晓宇[1)] 周傲英[2)]

[1)](上海电力大学计算机科学与技术学院 上海 200090)

[2)](华东师范大学数据科学与工程学院 上海 200062)

摘 要 在云端数据存储模式下,由于服务器提供者是不可信的(Honest-But-Curious),因此为了保证外包数据的完整性,数据拥有者通常基于加密数据进行静态或动态的完整性审计.然而,当前提出的绝大多数方法都不适用于泛在电力物联网(UPIoT)中云端数据的完整性审计——在数据采集频率高、更新速度快的情况下,数据检索效率和动态更新方面存在一定的局限性.本文就解决此问题,首次提出一种轻型分层 MHT 的认证数据结构,并融合位置敏感哈希(LSH)技术和现有的安全加密方法,实现了面向泛在电力物联网云端数据的轻型动态完整性审计方案.一方面,在分层 MHT 中节点存储数据相关信息并且引入局部根节点的概念,以此来提高节点利用率和缩短认证路径长度;另一方面,通过 LSH 技术可以提高数据的更新效率,而且在完整性审计时能够快速查找到所需数据.性能分析以及基于真实智能电表数据集的测试实验,进一步验证了本方案能有效降低计算及通信开销,并且支持高效的数据动态更新.

关键词 泛在电力物联网;数据完整性;Merkle 哈希树;位置敏感哈希;安全加密

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2020.02298

Lightweight Dynamic Integrity Auditing Scheme for Cloud Data of Ubiquitous Power Internet of Things

TIAN Xiu-Xia[1)] LIU Tian-Shun[1)] NIU Xiao-Yu[1)] ZHOU Ao-Ying[2)]

[1)](*College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai* 200090)

[2)](*College of Data Science and Engineering, East China Normal University, Shanghai* 200062)

Abstract Service provider is Honest-But-Curious in a cloud data storage mode. Therefore, data owners usually perform static or dynamic integrity auditing on encrypted data for guaranteeing the integrity of the outsourced data. However, most of the proposed methods are inapplicable to the integrity auditing of cloud data in ubiquitous power internet of things (UPIoT). In the case of high data acquisition frequency and fast update speed, there are certain limitations in data retrieval efficiency and dynamic update. In this paper, to solve above problems, we present an authentication data structure based on lightweight hierarchical MHT, which combines an existing secure encryption method and locality sensitive hash (LSH) technology. First of all, the hierarchical MHT is composed of 16 smart meters and divided into four MHT subtrees. Their root nodes as leaf nodes constitute a new MHT tree. And each node in the tree is a data information storage

收稿日期:2019-08-28;在线发布日期:2020-05-16.本课题得到国家自然科学基金(面上项目;重点项目)(61772327,61532021)、国网甘肃省电力公司电力科学研究院横向项目(H2019-275)资助.**田秀霞**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为数据库安全、隐私保护(大数据和云计算)、安全机器学习、面向电力用户的安全计算、数字图像篡改检测.E-mail:xxtian@shiep.edu.cn.**刘天顺**,硕士研究生,主要研究方向为数据库安全、云计算中的安全和隐私保护.**牛晓宇**,学士,主要研究方向为云计算安全.**周傲英**,博士,教授,博士生导师,长江学者特聘教授,国家杰出青年科学基金入选者,中国计算机学会(CCF)会员,主要研究领域为数据库、数据管理、数字化转型、教育科技(EduTech)和物流科技(LogTech)等数据驱动的应用等.

container, which stores the relevant nodes information. Besides, nodes in the tree are given six definitions, namely Definition 1 (node information), Definition 2 (smart meter identifier), Definition 3 (local root node identifier), Definition 4 (dynamic operation identifier), Definition 5 (node hash) and Definition 6 (authentication certificate). These innovations on nodes improve node utilization and shorten the authentication path length. Secondly, the data is encrypted before uploaded to the cloud. Because the smart meter in UPIoT collects electricity consumption data every 15 minutes, a time stamp is added into the smart meter data. This encryption method ensures the privacy of the smart meter data in the cloud. Then, the data is stored to the cloud by LSH technology. The cloud constructs I hash buckets, and $I=i$ (where i is the number of MHT subtrees). According to the smart meter identifier SM_{ij}, LSH maps the data that belongs to the same region into the same hash bucket. LSH technology improves data update efficiently and retrieves data quickly. Finally, the cloud data is given to three algorithms for dynamic operation, namely Algorithm 1 (data modification algorithm), Algorithm 2 (data deletion algorithm) and Algorithm 3 (data insertion algorithm). In the experimental part, we compared our scheme with MHT scheme (2010), SL scheme (2015), and PMHT scheme (2019). And we performed a series of experiments based on 126M real smart meter data sets. Under the condition that the fixed data size was unchanged and the number of smart meters was changed, we conducted comparisons of communication overhead, storage overhead, calculation overhead, and update efficiency. In addition, under the condition that the number of fixed smart meters was unchanged and the data size was changed, we conducted comparisons of auditing efficiency and calculation overhead. The experimental results further demonstrate that our work in this paper can effectively lessen the calculation and communication overhead, and support efficient dynamic data update. Our scheme realizes a lightweight dynamic integrity auditing of cloud data in UPIoT.

Keywords　ubiquitous power Internet of Things; data integrity; Merkle Hash Tree; locality sensitive hash; secure encryption

1　引　言

随着网络数据的日益规模化和集成化，云端数据库逐渐成为大数据时代数据的主要存储模式. 云端数据库是一个以数据存储和管理为核心的云计算系统，能为数据提供动态可伸缩的存储服务，其最大特点是存储即服务(Storage as a Service，SaaS). 企业越来越倾向将本地数据的维护和管理工作外包给服务器提供商，从而降低设备升级、更新、维护等各个方面的运营成本. 2019 年 1 月，国家电网公司在“两会”报告中首次提出“泛在电力物联网”的概念[1]. 2019 年 6 月，杭州供电公司推出了“电力云计算独居老人服务”模块，这是泛在电力物联网在社区共治中的首次应用①. 通过独居老人家中的智能电表(Smart Meter，SM)，用电数据以每 15min 为频段进行一次采集，把采集到的数据保存到供电公司，供电公司再把每天(24h)的数据以表格的形式上传到云端数据库进行保存. 然后，把老人现在的用电数据与之前存储在电力云端数据库中的数据用人工智能算法进行匹配计算，反馈与正常用电的偏离率，帮助其子女和社区志愿者在第一时间判断危险、联系老人并实施救助. 供电公司通过服务的方式，按需计量随时随地地使用电力云端数据库中丰富的数据资源，这给泛在电力物联网中云端数据的处理带来了希望[2-4].

然而，云端数据库是不可信的，云端数据面临着被攻击篡改或丢失的风险[5-7]. 2016 年 10 月，美国加利福尼亚太平洋电气公司数据库被发现未受任何保护，随时面临被黑客攻击的风险，致使超过 68 万

① 智能电表福利. 微信公众号：电网头条，2019-07-06

客户和员工的个人数据处于危险之中.因此,如何确保云端数据的完整性是国内外学者需要考虑的首要问题[8-10].

目前,研究者已经提出了多种面向云端数据完整性的审计方案,代表性工作主要有文献[11-12].Sebé 等人[11]在文献中提出了基于 RSA 签名的数据完整性审计机制,在审计过程中无需取回数据拥有者的数据即可完成审计.同时,每次审计后数据拥有者不需要重新计算元数据,所以该方案也能支持无限次审计.但对于泛在电力物联网中的海量数据,在计算 RSA 签名值时,巨大的计算开销使得供电公司难以承受.为了能够处理数据拥有者的动态数据,Wang 等人[12]在文献中提出了基于 Merkle 哈希树(Merkle Hash Tree,MHT)的动态数据完整性审计机制.该方案虽然支持数据拥有者动态数据的完整性审计,但在每次执行更新或第三方审计时,需要读取所有数据来重构 MHT,这将造成较大的计算开销.由于智能电表数据每 15 min 进行一次更新,因此该方案并不能满足供电公司的需求.

智能电表每天产生 TB 级别的海量数据,若每次更新都进行完整性审计,则审计效率会大幅下降,因此本文结合泛在电力物联网中云端存储数据具有采集频率高、更新速度快等特点,针对现有完整性审计工作在数据检索效率和动态更新方面存在的不足,提出了一种适用于泛在电力物联网云端数据的轻型动态完整性审计方案.本方案融合位置敏感哈希(Locality Sensitive Hash,LSH)技术和安全加密方法,在节点上存储相关数据信息,并且引入局部根节点缩短认证路径,同时利用 LSH 技术来提高数据检索和更新效率,最后基于真实智能电表数据集对安全性、计算开销、通信开销及更新效率等方面进行较全面地分析和验证.

本文其他章节内容安排,如表 1 所示.

表 1 章节内容安排

章节	内容
2	进行本研究领域相关工作总结
3	简单介绍本文用到的一些相关知识
4	描述系统模型和安全威胁模型
5	详细介绍分层 MHT 结构
6	基于分层 MHT 的轻型动态完整性审计方案
7	实验与理论相结合对本方案进行性能分析
8	总结全文并展望未来工作

2 相关工作

数据完整性审计是云计算领域的研究热点之一,因此引起了国内外众多研究者的普遍关注[13-16].在 2007 年,有国际学者提出两大数据完整性审计模型:Juels 等人[17]提出的数据可恢复性证明(Proofs of Retrievability,PoR)机制和 Ateniese 等人[18]提出的数据持有性证明(Provable Data Possession,PDP)机制[19].本小节将从 PoR 机制和 PDP 机制对现有研究工作进行分析和总结.

2.1 数据可恢复性证明机制

PoR 机制不仅可以确定出数据是否被篡改,而且还能够对被篡改的数据进行恢复.PoR 机制主要用于重要数据的完整性审计,如泛在电力物联网中云端数据不完整或被篡改,会对分析老人是否在家或用电异常等情况带来严重错误.

2009 年 Wang 等人[20]提出了一种支持动态操作的 PoR 机制,该方案事先利用 Reed-Solomon 验证元数据,并将其在本地保存.然后,审计者通过证据的生成与保存在本地的数据进行对照.若数据对比不正确,则将输出错误的数据.但是,该方案只能进行有限次的审计.2013 年 Shacham 等人[21]针对公开验证提出了新的 PoR 机制.该方案不需要保存审计过程中的验证数据,并且还可以发起任意次验证,但该方案不支持动态操作.

PoR 机制虽然可以对被篡改的数据进行恢复,但数据拥有者想要验证时需要从云端把数据下载到本地,再进行对比验证,这将增大数据拥有者的存储开销和计算开销[22].然而,PDP 机制只需要验证云端返回给数据拥有者的证据是否正确即可.因此,PDP 机制更加适用于泛在电力物联网中云端大数据的完整性审计.

2.2 数据持有性证明机制

PDP 机制能够快速审计出云端数据库中的数据是否被篡改,主要用途是进行大数据文件完整性的审计.在 PDP 机制中,国际学者已经给出了多种审计方案.

2008 年 Shah 等人[23]提出了一种基于 MAC 值的完整性审计机制.数据拥有者事先随机选取消息认证码(Message Authentication Code,MAC)密钥,再对数据集调用哈希函数求取 MAC 值,并将其存储在本地.然后,云端返回给数据拥有者计算的 MAC 值,与本地 MAC 值对照,判断云端数据是否完整.但是,该方案在完整性审计时每次都要取回所有数据,造成了很大的通信和计算开销.2011 年 Ateniese 等人[24]提出了一种基于形式化建模的 PDP 机制,并且采用抽样的方法进行完整性审计.同时,提出

了同态认证标签(Homomorphic Verifiable Tags, HVTs)的概念，并且能够实现对分块数据进行计算. 该方案虽然通信和计算开销较小，但不能支持动态操作.

对于云端数据库中的数据，数据拥有者可能有时需要进行动态更新操作，而传统 PDP 机制对于数据拥有者提出的新要求已不能满足. 假如数据拥有者想要对元数据进行修改，那么该数据之后所有数据的证据都需要重新计算，这无疑将增大动态数据完整性审计的难度. 为解决此问题，Wang 等人[12]、Erway 等人[25]和 Hariharasitaraman 等人[26]考虑采用动态认证数据结构来验证云端数据的完整性.

2010 年 Wang 等人[12]提出了一种基于 MHT 的方案，虽然该方案可以支持数据的动态操作，但树中节点利用率不高，对于大数据动态更新会造成审计效率较低. 2015 年 Erway 等人[25]提出了一种基于跳表(Skip List, SL)的方案，虽然该方案可以缩小数据认证范围，但进行完整性审计时所需的认证路径过长，这将导致通信和计算开销较大. 2019 年 Hariharasitaraman 等人[26]提出了一种基于位置感知MHT(Position-aware Merkle Hash Tree, PMHT)的方案，该方案由于能够确定每个节点与其父节点之间的相对位置信息，因此不需要查找整棵树来计算根节点，但当云端数据量较大时，PMHT 结构会变得复杂，这将造成审计和更新效率降低.

现有所提出的方案仅仅只是理论模型，而且只能应用于特定的具体问题，并不适用于泛在电力物联网中云端数据的动态完整性审计.

3　预备知识

3.1　双线性映射

设 G 和 G_T 是两个乘法循环群，阶数是素数 p，群 G 的生成元为 g. 双线性映射 $e: G\times G\rightarrow G_T$ 为可计算的前提条件是 e 应该满足以下条件：

(1) 可计算性. 对所有 $y_1, y_2\in G$，存在一个高效的算法计算 $e(y_1, y_2)$；

(2) 双线性. 任意 $y_1, y_2\in G$ 和 $\zeta, \eta\in Z_p$，有 $e(y_1^{\zeta}, y_2^{\eta})=e(y_1, y_2)^{\zeta\eta}$；

(3) 非退化性. $e(g, g)\neq 1$.

3.2　Merkle 哈希树(MHT)

MHT 相当于二叉树，是用来存储哈希值的一棵树. 树中叶子节点是数据块的哈希值，而非叶子节点是由其对应的叶子节点进行哈希串联运算得到的，再从下向上逐级进行哈希串联运算. 最后，得到根节点 Root 的根哈希 h_{Root}，用其判断数据的完整性[27].

MHT 中叶子节点 $h(m_i)$ 为数据 m_i 的哈希值，认证结构如图 1 所示. 为了验证数据拥有者数据的完整性，需要把数据的认证路径和辅助认证信息 Ω_i 一起构成证据. 例如，随机选取数据{2,7}，想要验证它们的完整性. 首先，数据 m_2 的认证路径{$h(m_2)$, h_C, h_A}、辅助认证信息 $\Omega_2=\{h(m_1), h_D\}$ 和数据 7 的认证路径{$h(m_7)$, h_F, h_B}、辅助认证路径 $\Omega_7=\{h(m_7), h_E\}$. 最后，根据 $h_C=h(h(m_1)|h(m_2))$, $h_A=h(h_C|h_D)$, $h_F=h(h(m_7)|h(m_8))$, $h_B=h(h_E|h_F)$ (其中"|"表示字符串连接)，计算出根节点哈希值 $h'_{\text{Root}}=h(h_A|h_B)$，对比存储在本地的根哈希值 h_{Root}，来判断数据是否完整.

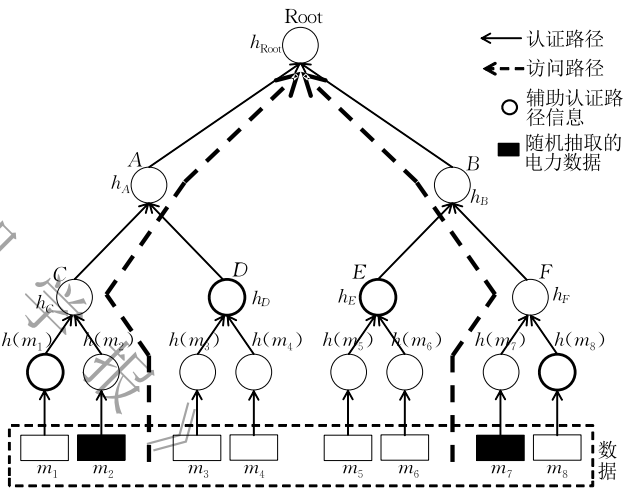


图 1　MHT 认证数据结构

3.3　位置敏感哈希

哈希桶定义：设哈希函数的值域为 Y，对 Y 等分处理，同时把 Y 中每一小段的宽度设为 L，则称这一小段距离长度 L 为一个哈希桶(Bucket)[28].

LSH 定义：对于任意 $w_1, w_2\in S$，若从欧式空间 S 到哈希编码空间 U 的函数族对 $H=\{h_1, h_2, \cdots, h_\beta\}$ 的距离函数 $d(\cdot)$ 满足以下条件，则称函数族 H 对距离 d 是 (r_1, r_2, q_1, q_2) 敏感的[29]：

(1) 如果 $d(w_1, w_2)\leq r_1$，那么 $\Pr[h(w_1)=h(w_2)]\geq q_1$；

(2) 如果 $d(w_1, w_2)\geq r_2$，那么 $\Pr[h(w_1)=h(w_2)]\leq q_2$.

其中，距离 $r_1, r_2>0$；$0<q_1, q_2<1$ 是 w_1, w_2 映射到同一个哈希桶的概率.

如图 2 所示，LSH 对区域数据进行哈希映射，

若区域空间中两点之间距离很近，那么两点哈希值经过哈希映射之后，在很大概率上是相同的. 同理，若两点之间距离很远，那么两点哈希值相同的概率就很小.

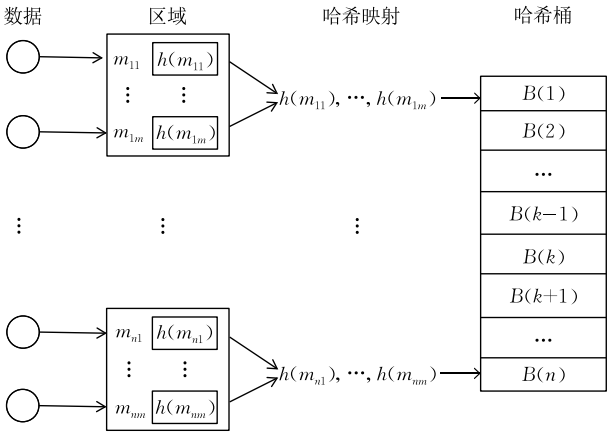


图 2 哈希映射示意图

3.4 符号含义描述

本小节列举出文中相对常用的一些符号，并对其进行含义描述，如表 2 所示.

表 2 符号含义描述

符号	含义
g	循环群的生成元
G,G_T	阶为素数 p 的乘法循环群
Z_p	mod p 所组成的整数空间，即取值范围 $[0,p-1]$
Ω_i	数据 m_i 的辅助认证信息元组
m_{ij}	加密后智能电表数据
F	m_{ij} 的集合，PSC 上传到 PCD 中存储
T	同态签名集合
$Sig(f(v_i))$	局部根节点签名
Γ	局部根节点签名集合
$Sig(f(v_R))$	PSC 根节点签名
$(f(v_{ij}),Para_{ij},\psi_{ij})$	分别表示智能电表数据的哈希值、认证证书、属性信息
(SM_{ij},R,DO)	分别表示智能电表标识符、局部根节点标识符、数据动态操作标识符

4 系统模型和安全威胁模型

4.1 系统模型

4.1.1 数据审计模型

以泛在电力物联网中云计算服务独居老人模块的应用为例，在杭州市拱墅区小河街道 326 户独居老人家中，通过老人家中的智能电表，以每 15 min 为频段进行一次采集，把采集到的数据保存在供电公司，供电公司把每天的数据以表格形式上传到云端数据库进行保存. 泛在电力物联网中云端数据审计模型由供电公司(Power Supply Company，PSC)、电力云端数据库(Power Cloud Database，PCD)及可信第三方审计(Trusted Party Auditor，TPA)组成，模型如图 3 所示.

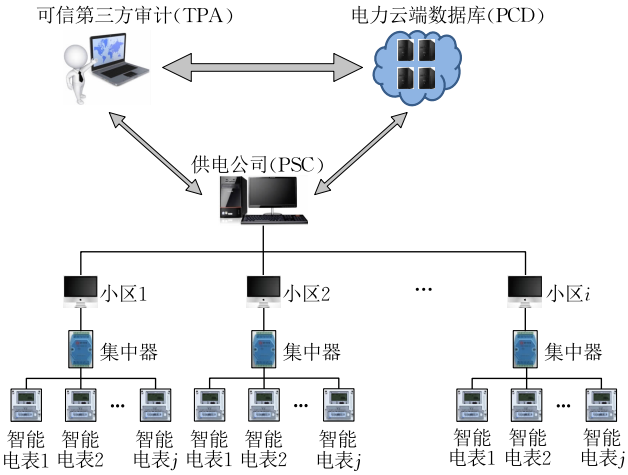


图 3 云端数据完整性审计模型

(1) 供电公司. PSC 由 326 户独居老人家中的智能电表组成. 通过智能电表可以采集到老人家中的用电数据，然后由集中器把这些数据汇聚到他们各自的小区，所有小区再把数据集中到 PSC，最后由 PSC 发送到 PCD. 由于智能电表是泛在电力物联网的智能终端，因此将智能电表和用电户看成一个整体. 集中器是中心连接点设备，它将智能电表数据集中定时(以每 15 min 为频段)传送给 PSC.

(2) 电力云端数据库. PCD 是指一个专业提供云端数据库服务的提供商. 它可以存储供电公司的海量数据，并能正确地对数据进行插入、删除和修改等操作. 但是，PCD 是不可信的，不能保证云端存储数据的完整性[30].

(3) 可信第三方审计. TPA 是当 PSC 和 PCD 因智能电表数据完整性问题产生争议时，对存储在云端的数据执行完整性审计验证. TPA 具有一定的计算和存储能力，如 TPA 可以是计算机，也可以是手机或无线 PDA 等[31].

4.1.2 完整性审计系统架构

数据完整性审计机制本质上是一种“挑战-应答”协议. 在初始化阶段，PSC 对智能电表数据进行预处理，并将数据存储在 PCD. 为了降低存储开销，PSC 删除了本地数据. 在验证过程中，TPA 首先向 PCD 发送挑战信息 $chal$，PCD 接收到 $chal$ 后，根据存储的数据计算并生成完整性审计证据 P，并返回给 TPA. 最后，TPA 对返回的 P 进行验证并得出结论，再将结果发送给 PSC.

当泛在电力物联网中云端数据进行完整性审计时，需要 5 个多项式时间内算法，具体如下：

(1) 密钥对生成算法. $KeyGen(1^{\theta})\rightarrow(pk,sk)$ 由 PSC 执行. 输入为安全参数 θ，输出为公开密钥 pk 和私有密钥 sk；

(2) 数据标签生成算法. $TagGen(F,sk)\rightarrow T$ 由 PSC 执行，为每一个智能电表数据生成标签，并组成标签集合 T. 输入为数据 F 和私有密钥 sk，输出为标签集合 T；

(3) 挑战信息生成算法. $ChalGen(\gamma)\rightarrow chal$ 由 TPA 执行. 输入为数据参数 γ，输出为相应的挑战信息 $chal$；

(4) 证据生成算法. $ProofGen(F,pk,T,chal)\rightarrow P$ 由 PCD 执行. 输入为 PCD 保存的数据 F、公开密钥 pk、标签集合 T 和挑战信息 $chal$，输出为请求的完整性证据 P；

(5) 验证证据生成算法. $VerifyGen(pk,P,chal)\rightarrow$('true','fulse') 由 TPA 执行. 输入为公开密钥 pk、完整性证据 P 和挑战信息 $chal$，如验证通过，则输出为 true，否则输出为 fulse.

当泛在电力物联网中云端数据需要动态更新时，还需要 2 个多项式时间内算法，具体如下：

(6) 更新生成算法. $UpdateGen(F,T,Update)\rightarrow(F',T',P_{Update})$ 由 PCD 执行. 输入为 PCD 保存的数据 F、标签集合 T 和更新操作命令 $Update$，输出为更新的数据 F'、标签集合 T' 和更新证据 P_{Update}；

(7) 验证更新生成算法. $V\text{-}UpdateGen(pk,P_{Update},Update)\rightarrow$('true','fulse') 由 TPA 执行. 输入为公开密钥 pk、更新证据 P_{Update} 和更新操作命令 $Update$. 如验证通过，则输出为 true，否则输出为 fulse.

4.2　安全威胁模型

在电力云端数据库中，不仅需要确信云端数据的完整性，而且还要满足对大量数据更新的请求；另外，PCD 是不可信的，数据更新操作存在不安全问题. 数据所受攻击类型可分为 3 种：替换攻击、伪造攻击和重放攻击. 具体分析如下：

攻击模型：在 PSC 发送验证请求后，TPA 接收请求，并向 PCD 发起挑战 $chal$；在挑战阶段，攻击模型的攻击目标是获得云端存储的数据信息，进而与 PSC 授权的 TPA 进行通信，试图通过 TPA 对数据的证据进行验证，并且不被发现.

在智能电表数据完整性审计过程中，攻击模型与挑战者审计过程由以下 5 个阶段组成：

(1) Setup 阶段. 即初始化阶段. PSC 执行 $KeyGen(\cdot)$ 算法，得到验证所需的公开密钥 pk 和私有密钥 sk，PSC 将公开密钥 pk 发给攻击模型；

(2) Query 阶段. 即交互阶段. 攻击模型与挑战者之间相互通信，攻击模型随机选取一些智能电表数据，向挑战者请求这些数据的标签. 挑战者对于攻击模型的请求，运行 $TagGen(\cdot)$ 算法，计算相应的数据标签并返回给攻击模型；

(3) Challenge 阶段. 即挑战阶段. 根据攻击模型请求的数据，挑战者向攻击模型发起挑战，运行 $ChalGen(\cdot)$ 算法，并将挑战信息 $chal$ 发送给攻击模型；

(4) Proof 阶段. 即证据生成阶段. 攻击模型接受到挑战信息 $chal$ 后，运行 $ProofGen(\cdot)$ 算法，把生成的 P 发送给挑战者；

(5) Verify 阶段. 即验证阶段. 挑战者运行 $VerifyGen(\cdot)$ 算法，对收到的 P 进行验证，并得出验证结果. 如果挑战者输出'true'，表明攻击模型攻击成功.

5　分层 MHT 认证数据结构设计

认证数据结构是将查询数据结构与密码学技术相结合，使其数据可认证化的一种结构[32]. 由于泛在电力物联网中智能电表以每 15 min 为频段进行数据采集，导致每天将产生大量数据记录. 因此，为了对泛在电力物联网中云端数据进行轻型动态完整性审计，文本设计了一种分层 MHT 认证数据结构.

5.1　分层 MHT 数据结构

随着智能电表的普及，泛在电力物联网中每天将产生海量的数据，而 MHT 需要分配的初始化参数越来越多，树的构造也越来越大. 因此，传统的 MHT 结构已经无法支持大规模智能电表数据完整性审计. 为了解决上述问题，本文提出分层 MHT 方案. 为了更加合理、高效地进行数据完整性审计，一个拥有海量数据的泛在电力物联网应该将整个电网分成多个子区域，处于同一个区域的数据共同构成一棵 MHT 子树，再将所有 MHT 子树的根节点构成一棵新 MHT 树.

本方案根据智能电表所处小区地理位置不同，将传统 MHT 拆分成多棵 MHT 子树，然后再把子树的根节点作为叶子节点构成新 MHT 树. 以杭州

市拱墅区小河街道 326 户独居老人家中的智能电表为例，从拱墅区小河街道中随机选取 4 个小区，再从每个小区中随机选取 4 个独居老人家中的智能电表，如图 4 所示. 其中，分层 MHT 由 16 块智能电表组成，被划分成 4 棵 MHT 子树，4 个子树的根节点作为叶子节点，再构成一棵新 MHT 树.

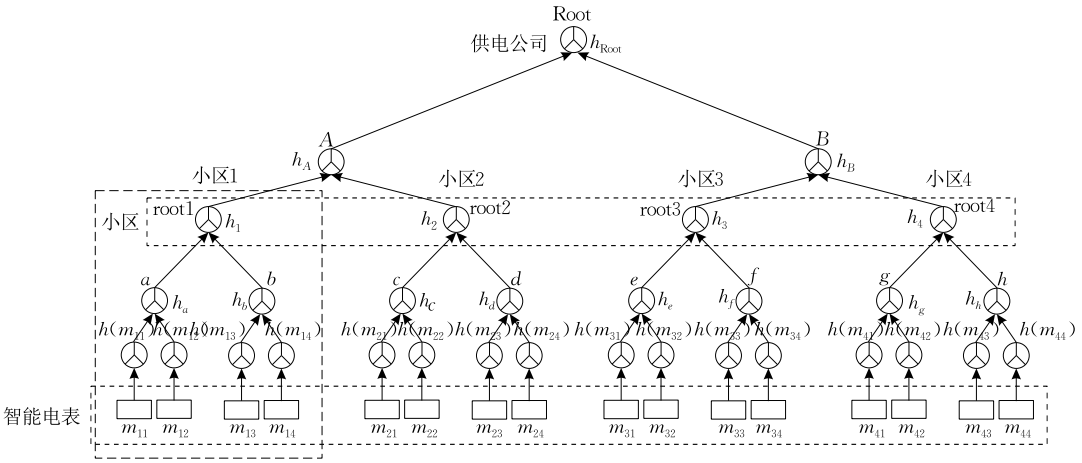


图 4 分层 MHT 认证数据结构

5.2 分层 MHT 具体构建

出于对用户数据安全等因素的考虑，本实验无法获取杭州市供电公司的智能电表数据，因此采用本课题组项目中某市供电公司智能电表真实数据. 为了更加直观地显示智能电表数据参数，并且不泄露用户用电数据信息，故随机挑选 4 个小区的 4 组智能电表数据，如图 5 所示. 其中，TG_ID 表示小区编号，ID 表示智能电表编号，DATA_DATE 表示用电日期，PAP_R 表示总用电量，PAP_R1 和 PAP_R2 分别表示已用电量和未用电量.

TG_ID	ID	DATA_DA	PAP_R	PAP_R1	PAP_R2
10001745531	4035893	2019/5/1	14449.29	11273.09	3176.2
10001745531	4035894	2019/5/1	21261.22	15935.38	5325.84
10001745531	4035895	2019/5/1	25512.5	17309.92	8202.58
10001745531	4035896	2019/5/1	10650.14	7909.64	2740.5
10001749671	3725265	2019/5/1	16198.97	10647.66	5551.3
10001749671	3725266	2019/5/1	2919.84	2034.11	885.73
10001749671	3725267	2019/5/1	28974.02	20119.22	8854.79
10001749671	3725268	2019/5/1	7224.2	4578.26	2645.94
10001750095	4643428	2019/5/1	16744.77	12739.63	4005.14
10001750095	4643429	2019/5/1	7094.31	5141.64	1952.66
10001750095	4643430	2019/5/1	21314.13	14934.21	6379.92
10001750095	4643431	2019/5/1	9292.15	6537.9	2754.25
10001724631	4149074	2019/5/1	6365.07	4116.69	2248.37
10001724631	4149075	2019/5/1	25169.79	14564.84	10604.94
10001724631	4149076	2019/5/1	4161.88	2674.59	1487.29
10001724631	4149077	2019/5/1	31737.83	20928.43	10809.4

图 5 小区智能电表数据

用 4 个小区的 4 组数据分别构建 4 棵 MHT 子树，再用 4 棵 MHT 子树的根节点作为叶子节点，构建 1 棵新 MHT 树. 具体构建如下：

(1) 构建 4 棵 MHT 子树.

选取小区 1 的 TG_ID：10001745531；m_{11} 的 ID：4035893、m_{12} 的 ID：4035894、m_{13} 的 ID：4035895、m_{14} 的 ID：4035896. 首先，用这 4 组数据分别用 MD5 算法进行哈希运算，分别得到 $h(m_{11})$、$h(m_{12})$、$h(m_{13})$ 和 $h(m_{14})$. 然后，把这 4 个叶子节点哈希值经过串联哈希运算，得到 $h_a=h(h(m_{11})|h(m_{12}))$ 和 $h_b=h(h(m_{13})|h(m_{14}))$. 最后，把这 2 个非叶子节点哈希值（$h_a$ 和 h_b）经过串联哈希运算，得到 $h_1=h(h_a|h_b)$. 同理，其余小区的 MHT 子树构建与小区 1 相同.

(2) 构建 1 棵新 MHT 树.

首先，把 4 个小区根节点哈希值 h_1、h_2、h_3 和 h_4 当作叶子节点，经过串联哈希运算，得到 $h_A=h(h_1|h_2)$ 和 $h_B=h(h_3|h_4)$. 然后，把这 2 个非叶子节点哈希值（h_A 和 h_B）经过串联哈希运算，得到 $h_{Root}=h(h_A|h_B)$.

5.3 节点定义

树中每个节点都是一个数据信息存储容器，用来存储节点的相关信息.

定义 1. $\Phi=(f(v_{ij}),Para_{ij},\psi_{ij})$ 是每个节点独立存储的数据哈希值 $f(v_{ij})$、认证证书 $Para_{ij}$ 以及属性信息 ψ_{ij}，存储格式如图 6 所示.

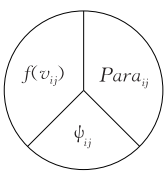


图 6 智能电表节点信息

首先，属性信息 ψ_{ij} 是三元组 $\psi_{ij}=(SM_{ij},R,DO)$. 其中，SM_{ij} 是智能电表标识符，其具体含义见定义 2；R 是局部根节点标识符，其具体含义见定义 3；DO 是数据动态操作标识符，其具体含义见定义 4.

然后，$f(v_{ij})$是节点哈希值，其具体含义见定义 5. 最后，$Para_{ij}$是认证证书，其具体含义见定义 6.

定义 2. 根据每个智能电表 ID 的不同，可以确定智能电表位置，智能电表标识符用 SM_{ij} 表示. 其中，i 表示小区（$i=1,2,3,4$），j 表示智能电表（$j=1,2,3,4$）. 例如，SM_{24}表示小区 2 的智能电表 4；SM_{31}表示小区 3 的智能电表 1.

定义 3. 在 MHT 中选取某个节点，用来描述该节点及其下方所有节点的完整性，则称该节点为局部根节点，记为 root. 例如，节点 $root_1$ 是小区 1 中所有节点的局部根节点（如图 4 中虚线所示）. 设 R 为局部根节点标识符，其具体表示如下：

$$R=\begin{cases}1, & \text{小区根节点}\\ 0, & \text{非小区根节点}\end{cases}.$$

定义 4. 动态操作标识符用 DO 表示，其具体表示如下：

$$DO=\begin{cases}00, & \text{未操作}\\ 01, & \text{插入操作}\\ 10, & \text{删除操作}\\ 11, & \text{修改操作}\end{cases}.$$

定义 5. 节点哈希值用 $f(v_{ij})$来表示，由自身哈希值 $f(v_{ij})$与左右孩子节点哈希值经过串联哈希运算得到的，其运算公式如下：

$$f(v_{ij})=\begin{cases}h(m_{left})\,|\,h(m_{right}), & \text{非叶子节点}\\ h(m_{ij}), & \text{叶子节点}\end{cases}.$$

例如，智能电表SM_{24}的哈希值 $f(v_{24})=h(m_{24})$，小区 2 的局部根节点哈希值 $f(v_2)=h_c\,|\,h_d$，供电公司的根节点哈希值 $f(v_R)=h_A\,|\,h_B$.

定义 6. 认证证书用 $Para_{ij}$来表示，$Para_{ij}$表示叶子节点到小区根节点路径上所有兄弟节点的哈希值，或者小区根节点到供电公司根节点路径上所有兄弟节点的哈希值. 例如，$Para_{13}=\{h(m_{14}),h_a\}$，$Para_{42}=\{h(m_{41}),h_h\}$，$Para_2=\{h_1,h_B\}$，$Para_3=\{h_4,h_A\}$.

6　轻型动态完整性审计方案

为了解决泛在电力物联网中云端数据的完整性审计问题，本文提出了一种基于分层 MHT 的轻型动态完整性审计方案. 方案主要包括以下三个阶段：初始化阶段、挑战-应答-审计阶段和动态更新-更新审计阶段.

6.1　初始化阶段

6.1.1　初始化

（1）数据采集

用电数据由智能终端——智能电表采集并处理. 智能电表采集独居老人家中用电数据 d_{ij}，并由集中器发送到各自小区，小区为所有采集到的数据生成智能电表标识符 SM_{ij}，并将数据及 SM_{ij} 发送到 PSC.

（2）数据加密

为了保证智能电表数据在云端的安全性，在将本地数据上传到 PCD 之前，PSC 先将数据用传统方法加密处理. 智能电表以每 15 min 为频段进行一次数据采集，因此在数据中加入时间戳. PSC 的用电数据 d_{ij} 加密后为：$m_{ij}=E(d_{ij},t)$，则智能电表数据集合为 $F=\{m_{ij}\}_{1\le i\le 4,1\le j\le 4}$.

（3）构建分层 MHT 与密钥生成

用加密后的智能电表数据 m_{ij} 构建图 4 中的分层 MHT，同时运行 $KeyGen(\cdot)$算法，随机选取 α 作为私钥 sk，并计算公钥 $pk=g^{\alpha}$，形成密钥对 (sk,pk).

（4）数据签名

运行 $TagGen(\cdot)$算法，随机选取 $u\in G$，对每个智能电表 m_{ij} 进行签名，得到 $\delta_{ij}=(h(m_{ij})\cdot u^{m_{ij}})^{\alpha}$，进而得到小区签名集合 $\sigma_i=\{\delta_{ij}\}_{1\le i\le 4,1\le j\le 4}$，继而得到 PSC 签名集合 $T=\{\sigma_i\}_{1\le i\le 4}$.

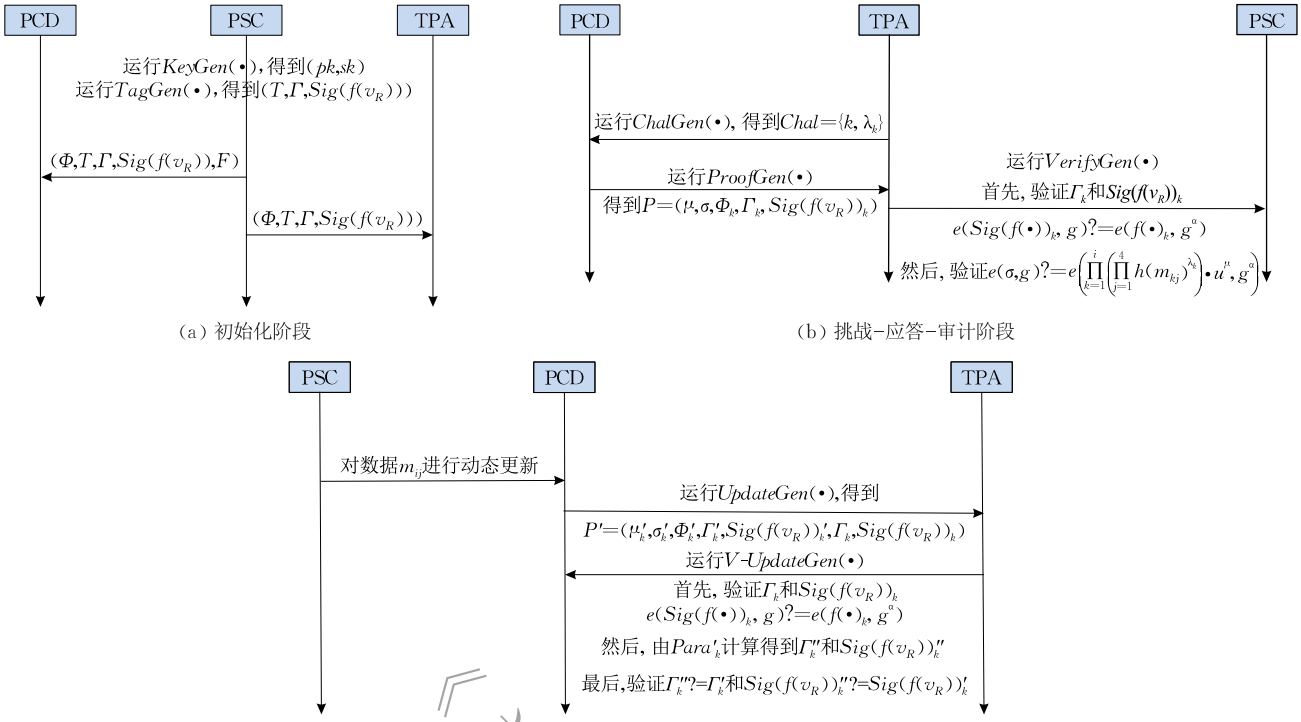
同时，对局部根节点进行签名 $Sig(f(v_i))=(f(v_i))^{\alpha}$，得到签名集合 $\Gamma=\{Sig(f(v_i))\}_{1\le i\le 4}$；然后，对根节点进行签名 $Sig(f(v_R))=(f(v_R))^{\alpha}$.

（5）生成节点信息

根据分层 MHT，写出 m_{ij} 节点的 $Para_{ij}$和局部根节点的 $Para_i$，同时写出 m_{ij} 的属性信息 $\psi_{ij}=(SM_{ij},0,00)$和局部根节点的属性信息 $\psi_i=(SM_i,1,00)$. 最后，m_{ij} 节点信息集合为 $\Phi_{ij}=(f(v_{ij}),Para_{ij},\psi_{ij})$，局部根节点信息集合为 $\Phi_i=(f(v_i),Para_i,\psi_i)$，节点信息集合为$\Phi=\{\Phi_{ij},\Phi_i\}_{1\le i\le 4,1\le j\le 4}$.

（6）上传数据

把$(\Phi,T,\Gamma,Sig(f(v_R)),F)$上传到 PCD，并且把$(\Phi,T,\Gamma,Sig(f(v_R)))$上传到 TPA，删除本地数据节省存储空间. 初始化阶段具体流程，如图 7(a)所示.



(a) 初始化阶段　(b) 挑战-应答-审计阶段

(c) 动态更新-更新审计阶段

图 7 审计各阶段具体流程

6.1.2 数据存储

为了使云端数据在完整性审计时,能够快速检索到所需数据,用 LSH 技术把 PSC 的智能电表数据上传到 PCD 存储. 首先,由 PCD 构造 I 个哈希桶,并且 $I=i$(其中,i 为小区个数). 然后,PCD 收到 PSC 发送过来的$(\Phi,T,\Gamma,Sig(f(v_R)),F)$后,根据智能电表标识符 SM_{ij},利用 LSH 技术将数据 m_{ij} 分别映射到各自小区的哈希桶中存储,数据存储具体步骤如下:

(1) 对于每一个智能电表的数据 m_{ij},由哈希映射函数

$$f_{\mathrm{LSH}}=\frac{a\cdot v+b}{L} \tag{1}$$

映射到对应的哈希桶中. 其中,a 使用正态分布产生,值为 d 维随机变量;v 为数据 m_{ij} 对应的特征向量;b 使用均匀分布产生,范围为$[0,L]$内的实数;宽度 L 是最重要的参数,因为 $a\cdot v+b$ 得到的是一个实数,如果不加以处理,那么就起不到哈希桶的效果.

(2) 设 $o\langle K,V\rangle$为数据的索引项,其中,K 表示智能电表标识符 SM_{ij};V 表示数据索引信息,如 $B[i]$表示数据映射到第 I 个哈希桶. 因此,同属于一个小区的数据就会映射到同一个哈希桶中.

6.2 挑战-应答-审计阶段

6.2.1 挑战阶段

泛在电力物联网中云端数据数量巨大,若每次 PSC 都要对所有云端数据进行审计,则会造成审计效率大幅度降低. 由于大部分云端数据是完整且没有被篡改的,不需要进行完整性审计,因此 PSC 只需针对 PCD 中有可能损坏的智能电表数据,请求 TPA 提出完整性审计请求.

TPA 运行 $ChalGen(\cdot)$算法,从有可能损坏的小区集合中选取 $k\in\{1,i\}_{1\le i\le 4}$. 同时,随机选取 $\lambda_k\in Z_p$,构成挑战信息集合 $chal_k=\{\lambda_k,k\}$,然后周期性地向 PCD 发送验证请求.

6.2.2 应答阶段

PCD 接收到 TPA 发送过来的挑战请求后,根据智能电表标识符 SM_{ij},用 LSH 算法对 I 个哈希桶进行检索,找到挑战信息集合 $chal_k$对应需要验证的小区数据. 然后,PCD 运行 $ProofGen(\cdot)$算法,对 k 对应的小区数据分别计算:

$$\mu=\sum_{k=1}^{i}\sum_{j=1}^{4}\lambda_k m_{kj} \tag{2}$$

$$\sigma=\prod_{k=1}^{i}\left(\prod_{j=1}^{4}h(m_{kj})\cdot u^{m_{kj}}\right)^{\lambda_k} \tag{3}$$

并且得到节点信息 Φ_k、局部根节点 Γ_k和根节点

$Sig(f(v_R))_k$. 最后，PCD 把证据 $P_k=(\mu,\sigma,\Phi_k,\Gamma_k,Sig(f(v_R))_k)$ 发送给 TPA.

6.2.3　审计阶段

TPA 根据 PCD 发来的证据 P，运行 $VerifyGen(\cdot)$ 算法，先验证 Γ_k 和 $Sig(f(v_R))_k$ 是否正确. 如果正确则继续验证，否则说明数据不完整或被篡改. 然后，通过 PCD 发送过来的 $Para_k$，计算局部根节点哈希值 $f(v_i)_k$ 和根节点哈希值 $f(v_R)_k$，验证 $e(Sig(f(\cdot))_k,g)?=e(f(\cdot)_k,g^{\alpha})$. 当 $f(v_i)_k$ 和 $f(v_R)_k$ 都验证通过后，再继续验证

$$e(\sigma,g)?=e\left(\prod_{k=1}^{i}\left(\prod_{j=1}^{4}h(m_{kj})^{\lambda_k}\right)\cdot u^{\mu},g^{\alpha}\right) \tag{4}$$

是否成立. 若成立，说明智能电表数据完好或没有被篡改；否则，说明数据被损坏. 挑战-应答-审计阶段具体流程，如图 7(b)所示.

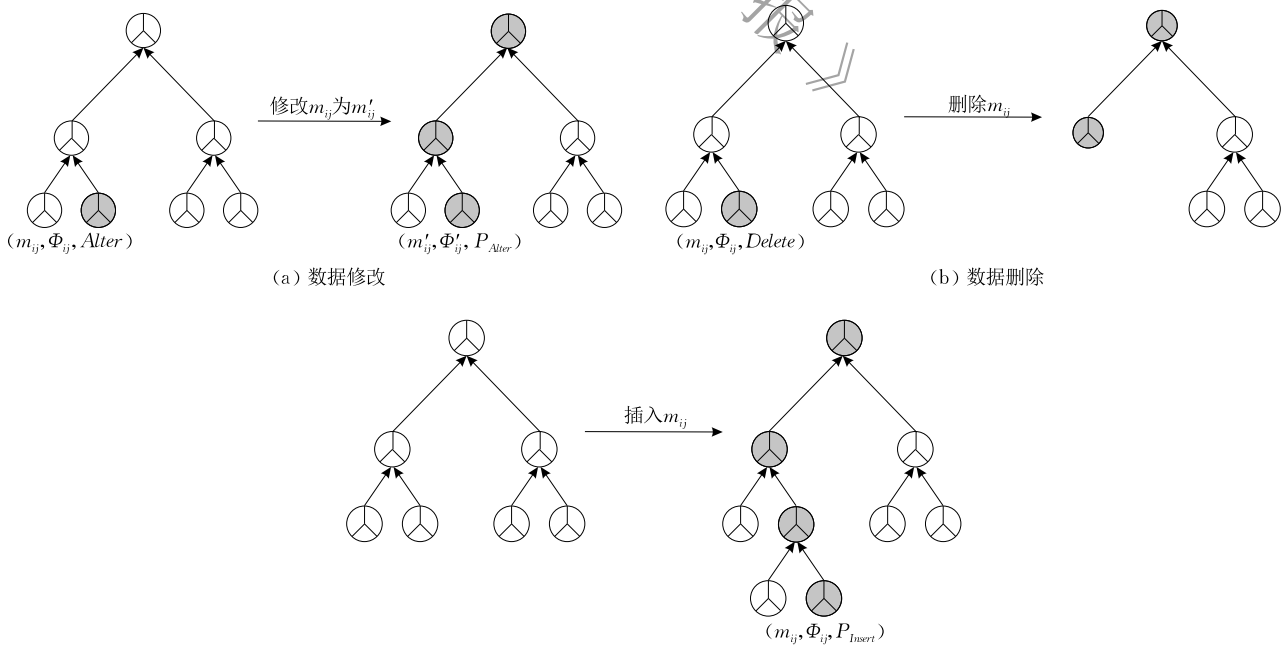
6.3　动态更新-更新审计阶段

由于泛在电力物联网中云端数据采集频率高、变化快，传统方案[33-36]已经无法满足云端数据的完整性审计要求，因此对能够实现实时处理云端数据的轻型动态完整性审计提出了更高的要求. 在 Hu 等人[37]的方案中，在每次执行数据更新时，其算法复杂度都为 $O(n)$（其中，n 为数据总数），因此造成通信和计算开销较大；而在 Liu 等人[38]的方案中，在每次执行数据更新时，随着数据量的增加，树中结构逐渐失衡，因此给审计带来很大困难.

为了解决泛在电力物联网中云端数据采集频率高和变化快的特点，本文首次引入 LSH 技术来支持云端数据的轻型动态完整性审计，用 LSH 能够迅速找到 PCD 中需要更新数据的位置，而且还可以快速找到 TPA 向 PCD 发起挑战需要验证完整性的数据信息. 因此，LSH 技术解决了泛在电力物联网中云端动态数据完整性审计的问题，并且大幅度提高了云端数据的更新效率. 云端动态数据的主要更新操作包括：修改、删除和插入，具体分析见如下小节[39].

6.3.1　修改数据

首先，当 PSC 需要修改某智能电表数据时，PSC 将需要修改的智能电表标识符 SM_{ij} 以及修改后加密数据 m'_{ij} 发送到 PCD. 然后，PCD 运行 LSH 算法，根据 $Value.Bucket[I]$ 找到所属桶的编号 I，运行 $UpdateGen(\cdot)$ 算法，即 $UpdateGen(m_{ij},\Phi_{ij},Alter)\rightarrow(m'_{ij},\Phi'_{ij},P_{Alter})$. 输入为 PCD 保存的数据 m_{ij}、节点信息 Φ_{ij} 和修改操作命令 $Alter$，输出为修改的数据 m'_{ij}、节点信息 Φ'_{ij} 和修改证据 P_{Alter}. 其中，m_{ij} 节点信息 $\Phi_{ij}=(f(v_{ij}),Para_{ij},\psi_{ij})$，属性信息 $\psi_{ij}=(SM_{ij},0,11)$，如图 8(a)所示.



(a) 数据修改　　(b) 数据删除

(c) 数据插入

图 8　数据动态操作

算法 1. 数据修改算法.

输入:PCD 保存的数据 m_{ij}、节点信息 Φ_{ij} 和修改操作命令 $Alter$

输出:修改的数据 m'_{ij}、节点信息 Φ'_{ij} 和修改证据 P_{Alter}

1. $h_0=m_{ij}.f(v_{ij})$;
2. $BID=h_0/L$;
3. FOR $i=1$ TO n DO
4. FOR $o\langle K,V\rangle\in HashTab$ DO
5. IF $B[i]==BID$ THEN
6. $UpdateGen(m_{ij},\Phi_{ij},Alter)\rightarrow(m'_{ij},\Phi'_{ij},P_{Alter})$;
7. FOR $j=1$ TO m DO
8. $UpdateGen(m_{ij},\Phi_{ij},Alter)\rightarrow(m'_{ij},\Phi'_{ij},P_{Alter})$;
9. END FOR
10. END IF
11. END FOR
12. END FOR

6.3.2 删除数据

首先,当 PSC 需要删除某智能电表数据时,PSC 将需要删除的智能电表标识符 SM_{ij} 发送到 PCD. 然后,PCD 运行 LSH 算法,根据 $Value.Bucket[I]$ 找到所属桶的编号 I,运行 $UpdateGen(\cdot)$ 算法,即 $UpdateGen(m_{ij},\Phi_{ij},Delete)\rightarrow(0,0,P_{Delete})$. 输入为 PCD 保存的数据 m_{ij}、节点信息 Φ_{ij} 和删除操作命令 $Delete$,输出为删除证据 P_{Delete},0 表示空节点,即节点信息为空. 其中,m_{ij} 节点信息 $\Phi_{ij}=(f(v_{ij}),Para_{ij},\psi_{ij})$,属性信息 $\psi_{ij}=(SM_{ij},0,10)$,如图 8(b)所示.

算法 2. 数据删除算法.

输入:PCD 保存的数据 m_{ij}、节点信息 Φ_{ij} 和删除操作命令 $Delete$

输出:删除证据 P_{Delete},0 表示空节点,即节点信息为空

1. $h_0=m_{ij}.f(v_{ij})$;
2. $BID=h_0/L$;
3. FOR $i=1$ TO n DO
4. FOR $o\langle K,V\rangle\in HashTab$ DO
5. IF $B[i]==BID$ THEN
6. $UpdateGen(m_{ij},\Phi_{ij},Delete)\rightarrow(0,0,P_{Delete})$;
7. FOR $j=1$ TO m DO
8. $UpdateGen(m_{ij},\Phi_{ij},Alter)\rightarrow(m'_{ij},\Phi'_{ij},P_{Alter})$;
9. END FOR
10. END IF
11. END FOR
12. END FOR

6.3.3 插入数据

首先,当 PSC 需要插入某智能电表数据时,PSC 将需要插入的智能电表标识符 SM_{ij} 以及插入的加密数据 m_{ij} 发送到 PCD. 然后,PCD 运行 LSH 算法,根据 $Value.Bucket[I]$ 找到所属桶的编号 I,运行 $UpdateGen(\cdot)$ 算法,即 $UpdateGen(m_{ij},\Phi_{ij},Insert)\rightarrow(m_{ij},\Phi_{ij},P_{Insert})$. 输入为需要插入的数据 m_{ij}、节点信息 Φ_{ij} 和插入操作命令 $Insert$,输出为插入的数据 m_{ij}、节点信息 Φ_{ij} 和插入证据 P_{Alter}. 其中,m_{ij} 节点信息 $\Phi_{ij}=(f(v_{ij}),Para_{ij},\psi_{ij})$,属性信息 $\psi_{ij}=(SM_{ij},0,01)$,如图 8(c)所示.

算法 3. 数据插入算法.

输入:需要插入的数据 m_{ij}、节点信息 Φ_{ij} 和插入操作命令 $Insert$

输出:插入的数据 m_{ij}、节点信息 Φ_{ij} 和插入证据 P_{Insert}

1. $h_0=m_{ij}.f(v_{ij})$;
2. $BID=h_0/L$;
3. FOR $i=1$ TO n DO
4. FOR $o\langle K,V\rangle\in HashTab$ DO
5. IF $B[i]==BID$ THEN
6. $UpdateGen(m_{ij},\Phi_{ij},Insert)\rightarrow(m_{ij},\Phi_{ij},P_{Insert})$;
7. FOR $j=1$ TO m DO
8. $UpdateGen(m_{ij},\Phi_{ij},Alter)\rightarrow(m'_{ij},\Phi'_{ij},P_{Alter})$;
9. END FOR
10. END IF
11. END FOR
12. END FOR

6.3.4 更新审计

首先,PCD 运行 $ProofGen(\cdot)$ 算法,得到证据 $P'=(\mu_k,\sigma_k,\Phi'_k,\Gamma'_k,Sig(f(v_R))'_k,\Gamma_k,Sig(f(v_R))_k)$. 然后,TPA 根据 PCD 发来的证据 P',运行 $V\text{-}UpdateGen(\cdot)$ 算法,先验证 Γ_k 和 $Sig(f(v_R))_k$. 然后,通过 PCD 发送过来的 $Para'_k$,计算得到 Γ''_k 和 $Sig(f(v_R))''_k$. 最后,验证 $\Gamma''_k?=\Gamma'_k$ 和 $Sig(f(v_R))''_k?=Sig(f(v_R))'_k$. 当所有的值都验证通过后,说明智能电表数据更新成功;否则,说明数据更新错误. 动态更新-更新审计阶段具体流程,如图 7(c)所示.

7 实验与结果分析

本文通过与 Wang 等人[12]的 MHT 方案(2010)、Erway 等人[25]的 SL 方案(2015)和 Hariharasitaraman 等人[26]的 PMHT 方案(2019)作对比分析,并进行一系列实验测试,从通信和存储开销,到审计效率和计算开销,再到更新效率,来分析本方案在轻型动态完整性审计方面的优势.

7.1 实验环境配置

本文采用 Python 语言实现基于分层 MHT 以及 LSH 算法的轻型 MHT 动态完整性审计方案. 实

验平台为 Intel(R) Core(TM) i5-6300HQ CPU@2.30GHz(4 CPUs)～2.3GHz，内存为 8GB，所用 Python 版本为 3.7.3，软件为 Geany.

7.2　实验数据集

本测试实验选用 5.2 小节使用的智能电表真实数据作为实验原始数据集，大小为 126M. 把智能电表数据集以及时间进行对称加密，将加密后数据存储在同一 TXT 文件中作为实验数据集. 由于实验部分是对本方案的轻型动态完整性审计性能进行分析，故在实验时间效率上没有加入数据加密的时间.

7.3　安全性能分析

本方案支持云端数据的动态操作，所以当数据更新时，也需要更新数据的节点信息. 然而，PCD 可能不诚实地更新智能电表数据，或者 PCD 服务器的软硬件故障而导致智能电表数据被损坏或丢失，所以 PCD 可能利用已经过期的数据和节点信息欺骗 TPA. 本小节将对本方案的安全性能，从抗替换能力、抗伪造能力、抗重放能力和隐私保护能力四个方面进行分析.

(1) 抗替换能力. 替换攻击是 PCD 选择一个有效的智能电表数据和签名代替已经损坏的数据和签名，试图通过 TPA 的审计. 为抵抗替换攻击，本方案使用分层 MHT 验证更新后的局部根节点签名 Γ_k 和根节点签名 $Sig(f(v_R))_k$. 当 Γ_k 和 $Sig(f(v_R))_k$ 都验证通过后，TPA 通过 $e(Sig(f(\cdot))_k, g)?= e(f(\cdot)_k, g^{\alpha})$ 进一步验证数据 m_{ij} 的完整性. 根据双线性映射和哈希函数的性质，被替换的数据签名不能以一个不可忽略的概率通过 TPA 的完整性审计.

(2) 抗伪造能力. 伪造攻击是 PCD 伪造已经损坏的智能电表数据和节点信息，企图通过 TPA 的审计. PCD 不能使用伪造攻击的方式通过 TPA 对云端数据的完整性审计，因为 TPA 已经通过局部根节点签名 Γ_k、根节点签名 $Sig(f(v_R))_k$ 和 $e(Sig(f(\cdot))_k, g)?= e(f(\cdot)_k, g^{\alpha})$ 对数据 m_{ij} 和节点信息 Φ_{ij} 的正确性进行了验证. 根据双线性映射和哈希函数的性质，伪造的签名和节点信息不能通过 TPA 的完整性审计.

(3) 抗重放能力. 重放攻击是 PCD 用原先的 P 去应答当前 TPA 发来的 $chal$，准备用欺骗的手段通过 TPA 的审计. 从某种程度上讲，重放攻击可以看做是多次替换攻击，因此可以使用与抗替换攻击同样的方法来分析抗重放攻击[40]. 当 Γ_k 和 $Sig(f(v_R))_k$ 都验证通过后，TPA 才能进一步验证数据 m_{ij} 的完整性. 根据双线性映射和 Hash 函数的性质可知，重放攻击不能以一个不可忽略的概率通过 TPA 的审计.

(4) 隐私保护能力. 隐私保护是对 PSC 和 PCD 之间传输以及存储在 PCD 中的智能电表数据进行保护. 本方案对 PSC 将要上传到 PCD 的智能电表数据进行时间戳加密处理为 $m_{ij}=E(d_{ij}, t)$，因此攻击模型无法得知智能电表数据的真实内容，进而保护用户用电数据的隐私性.

本方案与其他三种方案的安全性能对比结果如表 3 所示. 本方案具有隐私保护功能，同时抗替换、抗伪造和抗重放能力都很强. 而传统的 MHT、SL 和 PMHT 方案不具有隐私保护功能，并且 MHT 和 SL 方案抗替换能力很弱，SL 方案抗伪造能力也很差，MHT 和 PMHT 方案抗重放能力很弱.

表 3　安全性能对比

方案	抗替换能力	抗伪造能力	抗重放能力	隐私保护能力
本方案	强	强	强	有
MHT	弱	强	弱	无
SL	弱	弱	强	无
PMHT	强	强	弱	无

7.4　通信开销和存储开销分析

本小节将进行存储开销和通信开销对比分析，结果如表 4 所示. 其中，n 表示数据的总数，τ 表示 SL 中每个节点中所有孩子节点的总数，ξ 表示 PMHT 中节点与父节点相对位置信息的总数，i 表示在本方案中局部根节点的总数，j 表示在本方案中每个局部根节点中所有孩子节点的总数.

表 4　存储开销和通信开销对比

方案	存储开销		通信开销		
	PCD	TPA	PSC	PCD	TPA
本方案	$O(i)$	$O(i)$	$O(\log j)$	$O(\log j)$	$O(\log j)$
MHT	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
SL	$O(\tau)$	$O(\tau)$	$O(n+\tau)$	$O(n+\tau)$	$O(n+\tau)$
PMHT	$O(\xi)$	$O(\xi)$	$O(n+\xi)$	$O(n+\xi)$	$O(n+\xi)$

实验 1.　固定数据大小为 200kB 不变，改变智能电表数量，进行通信开销对比测试. 测试方案如下：

在传统的 MHT 中，MHT 是由叶子节点通过一系列哈希运算生成的，树中每个节点都是由哈希运算计算得到的哈希值，哈希运算均采用 MD5 算法. 数据拥有者发送数据集合给云端数据库，该数据集合包含 MHT 中叶子节点到根节点所有兄弟节点的哈希值；SL 在有序链表的基础上增加了多级索引

结构，在通信时需要发送所有节点信息；PMHT 发送每个节点与其父节点相对位置信息，从而不需要查找整棵树结构来计算根节点. 本方案将智能电表进行分区域，通信时只需要发送子区域的认证信息即可. 将本方案与其他三种方案进行对比分析，通信开销结果如图 9 所示.

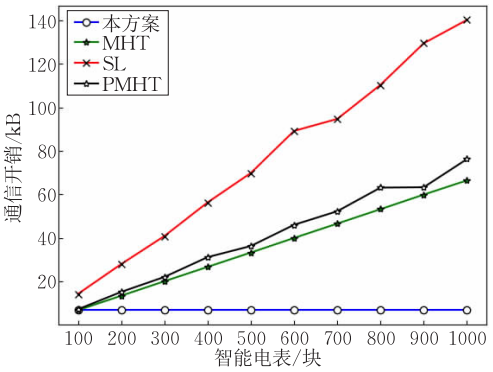


图 9 通信开销对比

由图 9 可知，在智能电表数目为 100 块时，本方案与 SL 和 PMHT 方案通信开销相等，但超过 200 块以后，本方案通信开销趋于平缓，而传统的 MHT、SL 和 PMHT 方案通信开销呈线性增加.

实验 2. 固定数据大小为 200 kB 不变，改变智能电表数量，进行存储开销对比测试. 测试方案如下：

MHT 发送从叶子节点到根节点所有节点哈希值到 TPA，SL 发送所有节点的索引信息，PMHT 发送每个节点与其父节点的相对位置信息. 本方案将智能电表进行分区域，存储时只需要发送子区域树认证信息即可. 将本方案与其他三种方案进行对比分析，存储开销结果如图 10 所示.

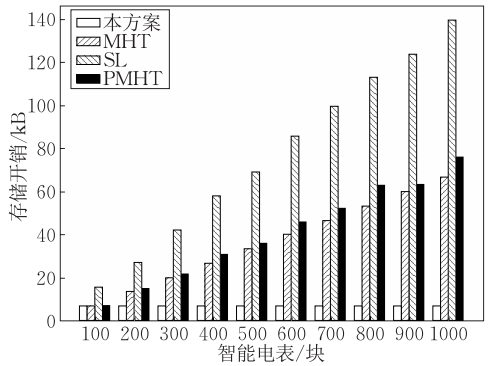


图 10 存储开销对比

由图 10 可知，本方案存储开销较小且基本保持不变，而传统的 MHT、SL 和 PMHT 方案随着智能电表块数增多，存储开销呈线性增大.

7.5 审计效率分析

实验 3. 固定智能电表数量为 16 块不变，改变数据大小，进行审计效率时间对比测试. 测试方案如下：

审计在云端数据完整性中占较大的比重，与数据标签生成算法 $TagGen(\cdot)$ 不同，审计过程需要三个多项式时间内算法 $ChalGen(\cdot)$、$ProoofGen(\cdot)$ 和 $VerifyGen(\cdot)$ 的支持. 为了验证智能电表数据大小对审计时间的影响，本小节分别对 16 块智能电表随机挑选 200 kB、400 kB、600 kB、800 kB 和 1000 kB 的数据文件，将本方案与其他三种方案进行对比分析，审计效率结果如图 11 所示.

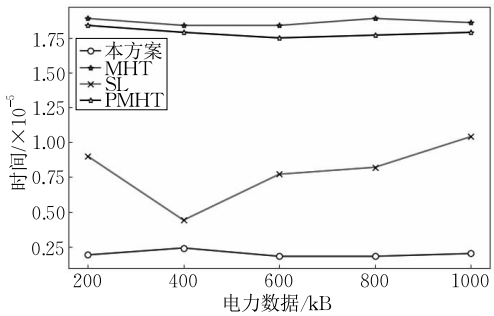


图 11 审计效率时间对比

由图 11 可知，由于本方案提高了节点利用率，缩短了认证路径长度，相比于其他三种方案所需的电力数据完整性审计时间更短，而 MHT、SL 以及 PMHT 方案审计时间相对较长.

实验 4. 固定智能电表数量为 16 块不变，改变数据大小，进行审计效率平均时间测试. 测试方案如下：

分别对 16 块智能电表随机挑选 200 kB、400 kB、600 kB、800 kB 和 1000 kB 的数据文件作为测试对象，对每组数据均进行 10 轮实验测试，并对 10 轮测试结果进行平均值计算，测试结果如图 12 所示.

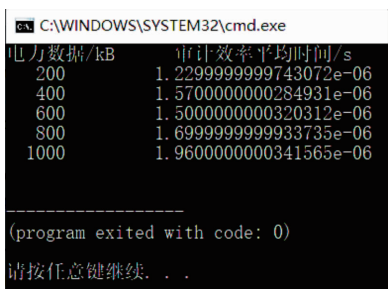


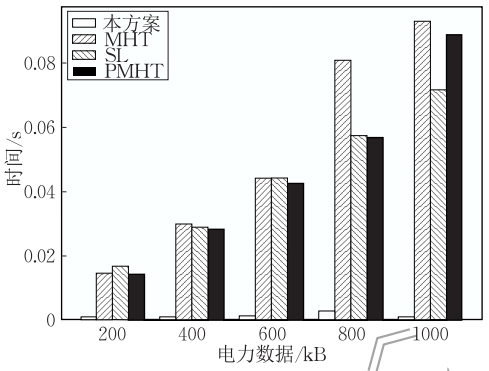
图 12 审计效率平均时间

由图 12 结果可知，本测试与实验 3 中审计效率时间保持一致，说明本方案在审计效率方面占很大优势.

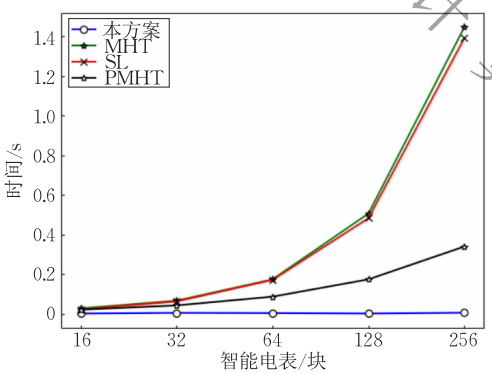
7.6　计算开销分析

实验 5.　固定智能电表数量为 16 块不变，改变数据大小，进行计算开销对比. 测试方案如下：

在 PCD 计算时，本方案只需考虑局部根节点的计算，所以相比于其他三种方案计算量减少很多. 本小节分别对 16 块智能电表随机挑选 200 kB、400 kB、600 kB、800 kB 和 1000 kB 的数据做实验测试，计算开销结果如图 13(a)所示.



(a) 基于电力数据的计算开销对比



(b) 基于智能电表的计算开销对比

图 13　计算开销时间对比

由图 13(a)可知，本方案计算开销总体较少，而其他三种方案在 600 kB 之前计算开销相差较小，但 800 kB 之后 MHT 和 PMHT 比 SL 变化明显. 因此，本方案符合预期设计目标.

实验 6.　固定数据大小为 200 kB 不变，改变智能电表数量，进行计算开销对比. 测试方案如下：

挑选大小为 200 kB 的数据文件分别以 16 块、32 块、64 块、128 块和 256 块智能电表做实验测试，将本方案与其他三种方案进行对比分析，结果如图 13(b)所示.

由图 13(b)可知，本方案计算开销较小，PMHT 方案计算开销相对较小，而 MTH 和 SL 方案随着数据块增多，计算开销呈指数增加.

7.7　更新效率分析

LSH 技术在很大概率上保证 SM_{ij} 的哈希值经过哈希运算后可以保持一致性，因此本方案以智能电表标识符 SM_{ij} 为依据，使 SM_{ij} 相同的智能电表数据映射到同一小区的哈希桶内. 进行数据更新时，通过 LSH 算法从 PCD 中找出需要更新数据信息(信息中含有 Φ, T、Γ 和 $Sig(f(v_R))$等信息)，然后运行 $UpdateGen(\cdot)$算法进行数据更新，并且运行 $ProofGen(\cdot)$算法生成更新后的 Φ'、T'、Γ' 和 $Sig(f(v_R))'$等信息.

实验 7.　固定数据大小为 200 kB 不变，改变智能电表数量，进行更新效率时间对比. 测试方案如下：

挑选大小为 200 kB 的数据文件分别以 16 块、32 块、64 块、128 块和 256 块智能电表做实验测试，将基于 LSH 更新数据的本方案与其他三种更新方案进行更新效率时间对比分析，结果如图 14 所示.

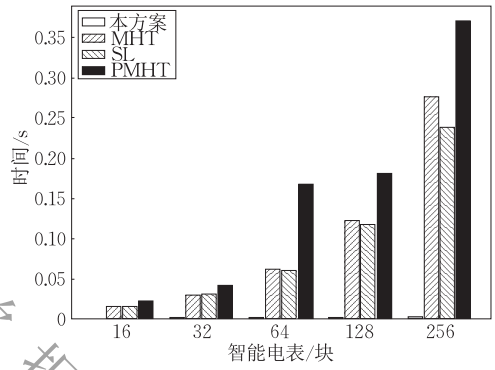


图 14　更新效率时间对比

由图 14 可知，与传统数据更新方案相比，本方案在更新效率方面具有显著优势，而 MHT、SL 以及 PMHT 更新方案随着数据块增加，更新效率时间变长. 因此，本方案实现了泛在电力物联网中云端数据的轻型动态完整性审计.

8　总结与展望

云端数据库是大数据时代主要的数据存储模式，其最重要的服务是为数据提供动态可伸缩的存储服务. 由于智能电表每天产生大量用电数据，云端数据库为泛在电力物联网大数据的存储带来了希望，然而如何确保云端存储数据的动态完整性是泛在电力物联网应该考虑的首要问题. 本文通过分析现有云端数据库中数据完整性审计方案，同时结合泛在电力物联网中数据采集频率高、更新速度快的特点，针对当前提出的方案都不适用于泛在电力物联网云端数据的轻型动态完整性审计，因此首次提

出一种基于分层 MHT 的认证数据结构，并融合 LSH 技术和现有的安全加密方法，实现了面向泛在电力物联网云端数据的轻型动态完整性审计方案. 今后工作重点是对云端中未通过审计的不完整数据进行研究，找出能够恢复不完整数据的方法，从而提高泛在电力物联网的安全性和可靠性.

参 考 文 献

[1] Yang Dong-Sheng, Wang Dao-Hao, Zhou Bo-Wen, et al. Key technologies and application prospects of ubiquitous power internet of things. Power Generation Technology, 2019, 40(2): 107-114(in Chinese)
(杨东升，王道浩，周博文等. 泛在电力物联网的关键技术与应用前景. 发电技术，2019，40(2)：107-114)

[2] Yang B, Xu G, Zeng X, et al. A lightweight anonymous mobile user authentication scheme for smart grid//Proceedings of the 2018 Ubiquitous Intelligence and Computing. Guangzhou, China, 2018: 821-827

[3] Feng Chao-Sheng, Qin Zhi-Guang, Yuan Ding. Techniques of secure storage for cloud data. Chinese Journal of Computers, 2015, 38(1): 150-163(in Chinese)
(冯朝胜，秦志光，袁丁. 云数据安全存储技术. 计算机学报，2015，38(1)：150-163)

[4] Feng Deng-Guo, Zhang Min, Zhang Yan, et al. Study on cloud computing security. Journal of Software, 2011, 22(1): 71-83(in Chinese)
(冯登国，张敏，张妍等. 云计算安全研究. 软件学报，2011，22(1)：71-83)

[5] Wazid M, Das A K, Kumar N, et al. Secure three-factor user authentication scheme for renewable-energy-based smart grid environment. IEEE Transactions on Industrial Informatics, 2017, 13(6): 3144-3153

[6] Siano P. Demand response and smart grids—A survey. Renewable and Sustainable Energy Reviews, 2014, 30: 461-478

[7] Jarrah M, Jaradat M, Jararweh Y, et al. A hierarchical optimization model for energy data flow in smart grid power systems. Information Systems, 2015, 53: 190-200

[8] Hu B, Hamid G. Smart grid mesh network security using dynamic key distribution with Merkle tree 4-way handshaking. IEEE Transactions on Smart Grid, 2014, 5(2): 550-558

[9] Xu J, Wei L, Zhang Y, et al. Dynamic fully homomorphic encryption-based Merkle tree for lightweight streaming authenticated data structures. Journal of Network and Computer Applications, 2018, 107: 113-124

[10] Peng S, Zhou F, Li J, et al. Efficient, dynamic and identity-based remote data integrity checking for multiple replicas. Journal of Network and Computer Applications, 2019, 134: 72-88

[11] Sebé F, Domingo-Ferrer J, Martinez-Balleste A, et al. Efficient remote data possession checking in critical information infrastructures. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(8): 1034-1038

[12] Wang Q, Wang C, Ren K, et al. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 2010, 22(5): 847-859

[13] Koo D, Shin Y, Yun J, et al. Improving security and reliability in Merkle tree-based online data authentication with leakage resilience. Applied Sciences, 2018, 8(12): 2532

[14] Zafar F, Khan A, Malik S U R, et al. A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends. Computers & Security, 2017, 65: 29-49

[15] Wei D S L, Murugesan S, Kuo S Y, et al. Enhancing data integrity and privacy in the cloud: An agenda. Computer, 2013, 46(11): 87-90

[16] Yu Y, Au M H, Ateniese G, et al. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Transactions on Information Forensics and Security, 2016, 12(4): 767-778

[17] Juels A, Kaliski B S. PORs: Proofs of retrievability for large files//Proceedings of the 14th ACM Conference on Computer and Communications Security. Whistler, Canada, 2007: 584-597

[18] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores//Proceedings of the 14th ACM Conference on Computer and Communications Security. Alexandria, USA, 2007: 598-609

[19] Tan Shuang, Jia Yan, Han Wei-Hong. Research and development of provable data integrity in cloud storage. Chinese Journal of Computers, 2015, 38(1): 164-177(in Chinese)
(谭霜，贾焰，韩伟红. 云存储中的数据完整性证明研究及进展. 计算机学报，2015，38(1)：164-177)

[20] Wang C, Wang Q, Ren K, et al. Ensuring data storage security in cloud computing//Proceedings of the 17th International Workshop on Quality of Service. Charleston, USA, 2009: 1-9

[21] Shacham H, Waters B. Compact proofs of retrievability. Journal of Cryptology, 2013, 26(3): 442-483

[22] Su Di, Liu Zhu-Song. New type of Merkle Hash tree for integrity audit scheme in cloud storage. Computer Engineering and Applications, 2018, 54(1): 70-76(in Chinese)
(苏迪，刘竹松. 一种新型的 Merkle 哈希树云数据完整性审计方案. 计算机工程与应用，2018，54(1)：70-76)

[23] Shah M A, Swaminathan R, Baker M. Privacy-preserving audit and extraction of digital contents. IACR Cryptology ePrint Archive, 2008, 2008: 186

[24] Ateniese G, Burns R, Curtmola R, et al. Remote data checking using provable data possession. ACM Transactions on Information and System Security, 2011, 14(1): 12

[25] Erway C, Papamanthou A C, Tamassia R. Dynamic provable data possession. ACM Transactions on Information and System Security, 2015, 7(4): 1-29

[26] Hariharasitaraman S, Balakannan S P. A dynamic data security mechanism based on position aware Merkle tree for health rehabilitation services over cloud. Journal of Ambient Intelligence and Humanized Computing, 2019: 1-15

[27] Li L, Yang Y, Wu Z. FMR-PDP: Flexible multiple replica provable data possession in cloud storage//Proceedings of the 2017 International Symposium on Computers and Communications. Heraklion, Greece, 2017: 1115-1121

[28] Sun Xu, Wen Mi, Zhang Xu, et al. A scheme for dynamic data integrity verification in smart grid. Computer Engineering, 2017, 43(8): 38-43(in Chinese)
(孙旭，温蜜，张栩等. 智能电网中一种动态数据完整性验证方案. 计算机工程，2017，43(8)：38-43)

[29] Jiang Rong. Research on Issues Related to Smart Grid Security and Privacy Protection [Ph. D. dissertation]. National University of Defense Technology, Changsha, 2014 (in Chinese)
(江荣. 智能电网安全与隐私保护相关问题研究[博士学位论文]. 国防科学技术大学，长沙，2014)

[30] Li Jin-Guo, Tian Xiu-Xia, Zhou Ao-Ying. Privacy preserving fuzzy keyword search in database as a service paradigm. Chinese Journal of Computers, 2016, 39(2): 414-428(in Chinese)
(李晋国，田秀霞，周傲英. 面向 DaaS 保护隐私的模糊关键字查询. 计算机学报，2016，39(2)：414-428)

[31] Tian Xiu-Xia, Wang Xiao-Ling, Gao Ming, et al. Database as a service-security and privacy preserving. Journal of Software, 2010, 21(5): 991-1006(in Chinese)
(田秀霞，王晓玲，高明等. 数据库服务—安全与隐私保护. 软件学报，2010，21(5)：991-1006)

[32] Papamanthou C, Tamassia R, Triandopoulos N. Optimal authenticated data structures with multilinear forms//Proceedings of the 2010 International Conference on Pairing Based Cryptography. Ishikawa, Japan, 2010: 246-264

[33] Diamantoulakis P D, Kapinas V M, Karagiannidis G K. Big data analytics for dynamic energy management in smart grids. Big Data Research, 2015, 2(3): 94-101

[34] Garcia M, Giani A, Baldick R. Smart grid data integrity attacks: Observable islands//Proceedings of the 2015 Power and Energy Society General Meeting. Denver, USA, 2015: 1-5

[35] Ahmed S, Lee Y D, Seung-Ho H, et al. Unsupervised machine learning－Based detection of covert data integrity assault in smart grid networks utilizing isolation forest. IEEE Transactions on Information Forensics and Security, 2019, 14(10): 2765-2777

[36] Sharma M, Agarwal A. Survey on authentication and encryption techniques for smart grid communication//Proceedings of the 7th India International Conference on Power Electronics. Patiala, India, 2016: 1-5

[37] Hu De-Min, Yu Xing. A dynamic data integrity verification method based on homomorphic labels. Computer Application Research, 2014, 31(5): 1362-1365(in Chinese)
(胡德敏，余星. 一种基于同态标签的动态数据完整性验证方法. 计算机应用研究，2014，31(5)：1362-1365)

[38] Liu Gang. Data Integrity Verification and Recovery in Distributed Storage Networks [M. S. dissertation]. Shanghai Jiaotong University, Shanghai, 2012(in Chinese)
(刘刚. 分布式存储网络中的数据完整性校验与恢复[硕士学位论文]. 上海交通大学，上海，2012)

[39] Wei J, Zhang R, Liu J, et al. Dynamic data integrity auditing for secure outsourcing in the cloud. Concurrency and Computation: Practice and Experience, 2017, 29(12): e4096

[40] Wei Jin-Xia. Research on Integrity Verification of Data in Cloud Storage Environment [Ph. D. dissertation]. Beijing University of Posts and Telecommunications, Beijing, 2017 (in Chinese)
(魏金侠. 云存储环境下数据完整性验证研究[博士学位论文]. 北京邮电大学，北京，2017)

TIAN Xiu-Xia, Ph. D., professor. Her main research interests include database security, privacy preserving (large data and cloud computing), secure machine learning, security computing for the benefit of power users, digital image forgery detection.

LIU Tian-Shun, M. S. candidate. His main research interests include database security, security and privacy preserving for cloud computing.

NIU Xiao-Yu, B. S. candidate. His main research interest is cloud computing security.

ZHOU Ao-Ying, Ph. D., professor, Ph. D. supervisor. His main research interests include database, data management, digital transformation, data-driven applications such as Educational Technology (EduTech) and Logistics Technology (LogTech).

Background

In the field of applied cryptography, data integrity auditing is a very popular research. Data integrity auditing is a method used to detect data tampering. It is applied to many aspects, such as data transmission, data stored in clouds and so on. Some researchers have proposed multiple schemes for cloud data integrity auditing. Based on such schemes, data owners can perform integrity auditing of data stored in clouds. However, the current data stored in clouds has the characteristics of high collection frequency and fast update speed. And we analyze the characteristics of ubiquitous power internet of things (UPIoT), which updates smart meter data every 15 minutes. For the dynamic data stored in clouds, scholars have proposed some integrity auditing methods, but these methods have great shortcomings in data retrieval efficiency and dynamic update. Therefore, the existing methods are not suitable for integrity auditing of cloud dynamic data in UPIoT.

To solve this problem, we propose an authentication data structure based on hierarchical MHT. We store data information in nodes and introduce a concept of local root node. Our innovation on nodes has improved the utilization of nodes and shortened the length of authentication path. At the same time, we use time stamp to encrypt the smart meter data. This method guarantees the security of data in the cloud. What's more, we use locality sensitive hash (LSH) technology in data integrity auditing. This technology can greatly improve the efficiency of data update and quickly retrieve the data we need. We have done a lot of experiments based on the real smart meter data set. We compared security performance of our scheme with other three ones. It is concluded that the smart meter data stored in the cloud has high security. In addition, we analyzed overhead performance of our scheme by doing great amount experiments. Experimental results show that our scheme can effectively lessen computational and communication overhead and support efficient data update. Our project implements a lightweight dynamic integrity auditing for UPIoT cloud data.

This work has been supported in part by the National Natural Science Foundation of China (General Program; Key Program) (Nos. 61772327, 61532021), the Electric Power Research Institute of State Grid Gansu Electric Power Company Enterprise Project (H2019-275). The research aims of these projects include privacy preserving issues, data integrity auditing, access control mechanisms, efficient search techniques on encrypted outsourced database.