基于区块链的云边协同大规模 IoT 数据完整性验证

赵雄伟[1]，滕　威[2]，楚鸿伟[1]

（1. 佳缘科技股份有限公司，四川　成都　610097；2. 杭州电子科技大学，浙江　杭州　310018）

摘　要：针对大规模物联网（IoT）数据完整性验证中存在的计算效率低和存储开销大的问题，提出了一种基于区块链的云边协同大规模 IoT 数据完整性验证方案。该方案通过以下两个创新点实现优化：第一，引入边缘节点协助区块链智能合约完成验证任务，既降低了区块链的计算负担，又通过智能合约执行仲裁算法保障了验证公正性；第二，采用 ZSS 短签名技术构建批量验证协议，有效减少了数据块标签计算和区块链存储的开销。最后，从安全性、复杂度理论分析和性能仿真验证两个角度证明所提方案的有效性。实验结果表明，所提方案在数据块标签计算效率上较传统 BLS 签名方案提升 50.7%，验证计算效率较同类型签名方案提升 29.3%。

关键词：数据完整性验证；物联网；区块链；云边协同

中图分类号：TP309　　**文献标识码：**A　　**DOI：**10.19358/j.issn.2097-1788.2025.10.011

引用格式：赵雄伟，滕威，楚鸿伟. 基于区块链的云边协同大规模 IoT 数据完整性验证［J］. 网络安全与数据治理，2025，44(10)：66-74.

Blockchain-based cloud-edge collaborative integrity verification scheme for large-scale IoT data

Zhao Xiongwei[1]，Teng Wei[2]，Chu Hongwei[1]

（1. Jiayuan Science and Technology Co.，Ltd.，Chengdu 610097，China；
2. Hangzhou Dianzi University，Hangzhou 310018，China）

Abstract：In response to the problems of low computational efficiency and high storage overhead in large-scale Internet of Things（IoT）data integrity verification，this study proposes a blockchain-based cloud-edge collaborative large-scale IoT data integrity verification scheme. This solution achieves optimization through two innovative points：first，edge nodes are introduced to assist blockchain smart contracts in completing verification tasks，which not only reduces the computational burden of the blockchain but also ensures verification fairness through the execution of arbitration algorithms by smart contracts；second，ZSS short signature technology is used to construct batch verification protocols，which effectively reduces the overhead of data block label computation and blockchain storage. Finally，this study proves the effectiveness of the proposed scheme from two perspectives：security and complexity theory analysis and performance simulation verification. The experimental results show that the proposed scheme improves the efficiency of data block label calculation by 50.7% compared with traditional BLS signature schemes，and the verification calculation efficiency improves by 29.3% compared with similar signature schemes.

Key words：data integrity verification；Internet of Things；blockchain；cloud-edge collaborative

0　引言

随着无线网络和 5G 技术的快速发展，物联网数据在工业、农业、智慧城市和医疗健康等领域发挥着重要作用[1]。数据的爆炸性增长，对本地存储空间提出更高要求，同时也增加了采购和维护成本。为应对这一挑战，云存储服务应运而生，它提供了高效便捷的解决方案。云存储服务以其经济性成为物联网数据存储的主要选择。然而，云存储也带来数据完整性的隐患，如数据丢失、恶意攻击或服务器故意删除不常用数据等问题[2-3]。因此，对存储在云存储提供商（Cloud Storage Provider，CSP）上的数据进行完整性验证十分必要。

传统的数据完整性验证方案中，通常由数据拥有者（DO）或第三方验证机构（TPA）对数据的完整性进行验证。由 DO 进行数据完整性验证的方案称为私有验证，

这种验证方案会给 DO 带来较大的计算和通信开销，同时存在验证结果不可信问题[4]。为此，Wang 等人[5]提出了一种基于 Merkle 哈希树的完整性验证协议，借助独立 TPA 完成验证操作。Shen 等人[6]提出一种可公开验证的方案，避免用户因上传错误的元数据而导致审计失败。符庆晓等人[7]提出了一种基于邻接表的动态数据完整性验证方案。王睿垚等人[8]提出基于代数运算的安全 PDP 协议，显著降低标签生成和验证的时间开销。张晓均等人[9]面向工业云存储数据，基于双线性对构建数据完整性审计协议，引入可信 TPA 以减轻用户计算压力并实现高效验证。杨帆等人[10]针对单一审计者可能导致审计失败的问题，设计了支持审计者更换的云数据完整性审计方案，提升了审计服务的可用性与稳定性。

然而，TPA 可能在未验证 CSP 返回的证明的情况下欺骗 DO，甚至在数据被篡改的情况下，与 CSP 勾结而虚假声明数据完整性。此外，作为集中式机构，TPA 还面临单点故障风险。

区块链技术具有去中心化、不可篡改、数据可追溯的特点，可以在无需所有节点均诚实可信的情况下实现数据的去中心化验证，同时避免单点故障的风险[11]。Liu 等人[12]提出了一个面向物联网的数据完整性审计服务框架，但该方案不适合大规模物联网数据场景。Xu 等人[13]基于模幂运算构建了一个支持仲裁的数据完整性审计方案，并通过智能合约保证验证和仲裁过程的公平性。陈建伟等人[14]面向电网多维数据的双端验证与共享需求，提出基于区块链的可验证聚合与分享方案，有效降低计算与通信开销。Lin 等人[15]设计了一个支持公共审计的物联网数据审计方案，通过部署在区块链上的智能合约检查 TPA 的验证结果。杨小东等人[16]提出了一种基于区块链和雾计算协同的物联网数据完整性审计方案，降低了区块链上的计算开销。王子园等人[17]提出了一种边缘环境下的无证书的物联网数据完整性验证方案，使用边缘节点进行验证工作，有效转移 DO 的计算负担。

然而，在大规模 IoT 数据完整性验证场景下，上述的数据完整性验证方案仍然存在链上计算和存储开销大、面临 CSP 重放攻击、数据块标签计算效率低的问题。为解决上述问题，本文提出一种基于区块链的大规模数据完整性验证方案。一方面，通过引入边缘节点协助部署在区块链上的智能合约完成数据完整性验证，降低链上计算开销。另一方面，通过 ZSS 短签名构建数据完整性验证协议，有效减小数据块标签计算开销与链上存储开销，并支持批量验证，提升验证效率。表 1 对国内外主流的 IoT 数据完整性验证方案进行了技术层面的系统对比，同时突出了本文方案在各关键维度上的优势。

1 系统模型及设计目标

1.1 系统模型

本方案系统模型如图 1 所示，该模型包含 5 个实体：数据拥有者（DO）、传感器节点（SN）、边缘节点（EN）、云存储提供商（CSP）以及区块链（BC），对上述 5 个实体的详细描述如下：

SN：部署在现实世界中的物联网终端设备，用于收集物联网数据，将收集到的数据上传给 DO。

表 1 相关工作比较与本方案优势分析

文献	算法基础	去中心化验证	支持批量验证	支持云边协同	缺点分析	本方案优势
文献［8］	代数签名	否	否	否	验证效率低，中心化程度高	标签计算效率高，链上存储开销低，云边协同减轻链上计算压力提升系统可扩展性，支持仲裁、去中心化验证
文献［9］	同态数字签名	否	是	否	标签计算开销大，中心化程度高	
文献［10］	BLS 签名	否	否	是	标签计算开销大，验证效率低，中心化程度高	
文献［13］	模幂运算	是	否	否	标签计算与链上存储开销大	
文献［14］	RSA 签名	是	否	否	标签计算开销大，验证效率低	
文献［15］	BLS 签名	是	否	否	标签计算和链上计算开销大	
文献［16］	BLS 签名	是	否	是	标签计算和链上计算开销大	
文献［18］	BLS 签名	否	是	否	标签计算和链上计算开销大，中心化程度高	
文献［19］	ZSS 签名	是	否	否	链上计算开销大，不支持批量验证，面临 CSP 重放攻击	
本文方案	RSA 同态哈希	是	是	是	—	

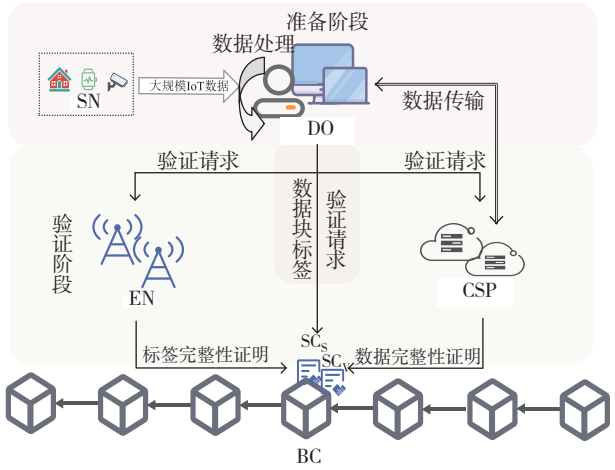

图 1　系统模型

DO：具有有限的计算和存储能力，会对 SN 上传的数据进行分块并计算对应的数据块标签，将数据块上传给远程的 CSP，数据块标签上传至区块链。

EN：部署在 DO 一侧的边缘节点，它主要接收 DO 发来的验证请求，在本地计算标签完整性证明，协助智能合约完成数据完整性验证。

CSP：具有强大的计算能力和海量的存储空间；但 CSP 可能会丢失 DO 存储的数据，因此它需要响应 DO 的验证请求并生成数据完整性证明。

BC：它是一个去中心化的存储网络，可以存储网络中所有实体的交易信息，运行智能合约 SC_S 和 SC_V。

1.2　困难问题与安全模型

离散对数问题：在一个有限群中，给定生成元 g 和群中的某个元素 h，求解一个整数 x 使得 $g^x = h$ 被认为是困难的。

计算 Diffie-Hellman 问题：一个循环群中，给定 g^a 和 g^b，计算出 g^{ab} 的问题。虽然 g、g^a 和 g^b 是公开的，但计算 g^{ab} 被认为是困难的。

1.3　安全威胁

本文方案考虑来自两方面的威胁：

（1）CSP 可能为节省存储成本在本地删除 DO 访问频率低的数据或因其他因素导致数据损坏。为保持其声誉，CSP 隐瞒数据损坏的事实。具体来说，CSP 可以通过伪造攻击、替换攻击和重放攻击来欺骗 DO。

（2）EN 可能在收到 DO 验证请求后，为节省计算成本或因计算错误导致生成错误的标签完整性证明，进而导致错误的验证结果。

1.4　设计目标

本文数据完整性验证方案要达到以下目标：

（1）验证正确性：所提方案应该确保验证结果是正确的，即能正确判断数据是否被完整保存。

（2）公开验证：验证过程和结果对 DO 和 CSP 公开，他们可以查阅验证结果。为实现此目的，由智能合约执行验证任务。

（3）批量验证：支持对大规模数据的同时验证，降低整体计算和通信成本。

2　验证方案

2.1　验证框架与算法

本文基于 ZSS 短签名构建了云边协同的物联网数据完整性验证方案，相较于 RSA、BLS 签名，ZSS 短签名具备更高的计算效率和更低的存储开销，并且支持签名聚合，非常适合大规模的 IoT 数据完整性验证。如图 2 所示，方案包含准备阶段和验证阶段共 6 个算法。

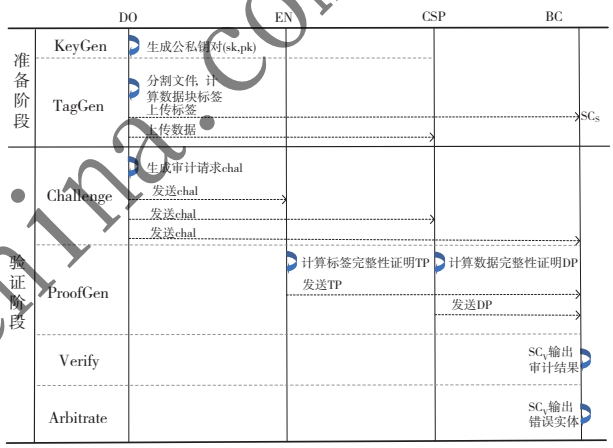

图 2　验证框架

（1）准备阶段

KeyGen（1^λ）→（sk，pk）：密钥生成算法，DO 生成用于数据块签名的私钥 sk 和公钥 pk。

TagGen（F，sk）→（Φ）：标签生成算法，由 DO 运行，输入要存储的文件 F 和 sk，对文件 F 分块，输出数据块标签集合 Φ。

（2）验证阶段

Challenge（I，V）→（chal）：挑战算法，由 DO 运行，选择要验证的 N^* 个数据块的索引集合 I 及对应的随机数集合 V，输出验证请求 chal。

ProofGen：证明生成算法，由两部分组成：

TProofGen（chal，Φ）→TP：标签证明生成算法，由 EN 运行，根据验证请求 chal 和数据块标签集合 Φ，生成标签完整性证明 TP。

DProofGen（chal，F）→DP：数据证明生成算法，由 CSP 运行，根据本地保存的数据 F 和验证请求 chal 生成数据完整性证明 DP。

Verify（DP，TP）→0/1：验证算法，由 SC_V 执行，根据 CSP 发送的 DP 和 EN 发送的 TP 检查数据的完整性，并输出验证结果。

Arbitrate（DP，TP，Φ，chal）→（CSP/EN），仲裁算法，在数据完整性验证失败时，由 SC_V 执行，获取公平的仲裁结果，输出错误实体名单。

2.2 单文件验证方案

（1）准备阶段

验证系统建立者首先生成 ZSS 签名所需要的系统参数：一个 q 阶加法循环群 G_1 以及它的一个生成元 P；一个乘法循环群 G_2；建立一对双线性映射 e：$G_1 \times G_1 \to G_2$，以及一个安全的哈希函数 H：$\{0, 1\}^* \to Z_q^*$，用于将数据映射到 Z_q^* 群上。接着，DO 随机选取一个签名私钥 $sk \in Z_q^*$，并计算对应的公钥 $pk = skP$。最后将系统公共参数 $\{pk, P, G_1, G_2, e, H\}$ 公布在区块链中。

然后，DO 将 SN 上传的文件 F 分割为 n 个小块，得到 $F = \{b_1, b_2, \cdots, b_n\}$，并计算每个数据块 b_i 对应的数据块标签：

$$\sigma_i = \frac{1}{H(b_i) + sk} P \tag{1}$$

最终得到完整的数据块标签集合 $\Phi = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$，之后 DO 将分块后的文件 F 上传至 CSP，将数据块标签集合上传至 SC_S。CSP 也会根据收到的文件 F，计算对应的数据块标签集合 Φ'，并与 SC_S 中 DO 上传的 Φ 进行对比，如果相同，则准备阶段完成，DO 可以在本地删除原始数据 F。反之，CSP 拒绝此次交易。

（2）验证阶段

DO 首先选择要挑战的 N^* 个数据块索引元素 $I = \{e_1, e_2, \cdots, e_{N^*}\}$。然后，为每一个元素生成一个对应的随机数 $V = \{v_1, v_2, \cdots, v_{N^*}\}$，$1 \leqslant i \leqslant N^*$，最后将验证请求 $chal = \{I, V\}$ 发送给 EN、CSP 以及 SC_V。CSP 在收到 DO 的验证请求 chal 后会根据本地保存的数据 F 计算：

$$\mu = \sum_{i=e_1}^{e_{N^*}} v_i H(b_i) P \tag{2}$$

$$R = \sum_{i=e_1}^{e_{N^*}} v_i pk \tag{3}$$

$$DP = e(\mu + R, P) \tag{4}$$

最后将数据完整性证明 DP 发送给 SC_V。

同时，EN 在收到 DO 的验证请求后，会根据验证请求中的索引元素 I 在 SC_S 中查询到对应的数据块标签集合，在本地计算标签完整性证明：

$$TP = e\left(P - P^2 \sum_{i=e_1}^{e_{N^*}} \frac{v_i}{\sigma_i}, P\right) \tag{5}$$

计算完成后将 TP 发送至 SC_V。

SC_V 收到 CSP 发送的 DP 以及 EN 返回的 TP 后计算等式：

$$TP \cdot DP = e(P, P) \tag{6}$$

如果式（6）成立，则代表 CSP 完整保存了 DO 的数据，反之，则表明此次完整性验证未通过，SC_V 将执行仲裁算法以找到出错的实体。具体来说，SC_V 首先根据 DO 之前发送的验证请求 chal 从 SC_S 中查询被挑选的数据块标签集合，然后再分别检查 EN 和 CSP 上传的 TP 和 DP 是否正确。

2.3 批量验证方案

在物联网环境下，DO 拥有大量的数据文件，可能需要同时验证多个文件的完整性。然而，若逐个对每个文件单独进行验证，将会导致较高的计算成本、通信开销以及验证延迟，进而影响系统的整体性能。如果能够同时满足 DO 的多个验证请求，则可以极大地提高数据完整性验证的效率。为实现此目标，本小节在原有单个文件数据完整性验证的基础上提出批量验证的方案，将 CSP 生成的多个数据完整性证明进行聚合，同时 EN 也在本地计算聚合的标签完整性证明，最终完成多个文件的数据完整性验证。下面将详细介绍批量验证方案的具体流程。

假设 DO 需要同时验证 z 个文件 $\{F_1, F_2, \cdots, F_z\}$ 的完整性，具体的验证方案如下：

DO 为每一个文件生成对应的验证请求 $chal_k = \{I_k, V_k\}$，$k \in [1, z]$，并将验证请求分别发送给 EN、CSP、SC_V。CSP 收到请求后分别在本地计算数据完整性证明：

$$\mu_k = \sum_{i=e_{k1}}^{e_{kN^*}} v_i H(b_i) P \tag{7}$$

$$R_k = \sum_{i=e_{k1}}^{e_{kN^*}} v_i pk \tag{8}$$

然后计算聚合的数据完整性证明 ADP：

$$ADP = e\left(\sum_{k=1}^{z} (\mu_k + R_k), P\right) \tag{9}$$

最后 CSP 将聚合的证明 ADP 发送给 SC_V。

同时，EN 也会根据验证请求，在 SC_S 中查询到对应文件的数据块标签，在本地计算聚合的标签证明：

$$ATP = e\left(P - P^2 \sum_{k=1}^{z} \sum_{i=e_{k1}}^{e_{kN^*}} \frac{v_i}{\sigma_i}, P\right) \tag{10}$$

然后将聚合的证明 ATP 发送给 SC_V。

SC_V 在收到 CSP 发送的 ADP 和 EN 发送的 ATP 后，计算等式：

$$ATP \cdot ADP = e(P, P) \tag{11}$$

如果式（11）成立，则代表 CSP 完整地保存了 DO 的 z 个文件，反之，则表明此次数据完整性验证未通过，即 DO 上传的 z 个文件中至少有一个文件损坏。此时，

SC_V将会执行仲裁算法，找到出错的实体。SC_V 首先会根据 DO 之前发送的验证请求 chal 查询 SC_S中被挑选的数据块标签集合，检查 EN 上传的标签完整性证明是否正确，然后再对 CSP 上传的完整性证明进行验证，输出出错实体。

3 安全性分析

3.1 正确性分析

在本文提出的单文件验证方案中，验证结果取决于式（6）的计算结果，正确性证明如下：

$$\begin{aligned}
&TP \cdot DP = e\Big(P - P^2 \sum_{i=e_1}^{e_{N*}} \frac{v_i}{\sigma_i}, P\Big) \cdot e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e\Big(-P^2 \sum_{i=e_1}^{e_{N*}} \frac{v_i}{\sigma_i}, P\Big) \cdot e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e\Big(-P^2 \sum_{i=e_1}^{e_{N*}} \frac{v_i (H(b_i) + \mathrm{sk})}{P}, P\Big) \cdot \\
&\quad e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e(P,P)^{-\sum_{i=e_1}^{e_{N*}} (v_i (H(b_i) + \mathrm{sk}))} \cdot e(P,P)^{\sum_{i=e_1}^{e_{N*}} (v_i (H(b_i) + \mathrm{sk}))} \\
&= e(P,P)
\end{aligned}$$

在本文提出方案的批量验证方案中，验证结果取决于式（11）的计算结果，正确性证明如下：

$$\begin{aligned}
&ATP \cdot ADP = e\Big(P - P^2 \sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} \frac{v_i}{\sigma_i}, P\Big) \cdot \\
&\quad e\Big(\sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e\Big(-P^2 \sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} \frac{v_i}{\sigma_i}, P\Big) \cdot \\
&\quad e\Big(\sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e\Big(-P^2 \sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} \frac{v_i (H(b_i) + \mathrm{sk})}{P}, P\Big) \cdot \\
&\quad e\Big(\sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e(P,P) \cdot e(P,P)^{-(\sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} v_i (H(b_i) + \mathrm{sk}))} \cdot e(P,P)^{(\sum_{k=1}^{z} \sum_{i=e_1}^{e_{N*}} v_i (H(b_i) + \mathrm{sk}))} \\
&= e(P,P)
\end{aligned}$$

3.2 CSP 攻击

伪造攻击场景定义如下：假设 CSP 为敌手，DO 向 CSP 发送挑战 chal = $\{I, V\}$，CSP 生成相应的数据完整性证明 DP。然而，由于人为或其他因素被挑战的某个块丢失或损坏，CSP 为保持其声誉，伪造证明 DP′。如果 CSP 伪造的证明通过 SC_V 的验证，则 CSP 赢得这场博弈的胜利，否则，他将会失败。假设 CSP 伪造的证明可以通过 SC_V的验证，则有：

$$TP \cdot DP' = e(P, P) \tag{12}$$

然而，实际上只有正确的证明 DP 才能通过数据完整性验证，即：

$$TP \cdot DP = e(P, P) \tag{13}$$

使用式（12）除以式（13）有：

$$DP' \cdot DP^{-1} = 1 \tag{14}$$

式（14）左边部分可以表示为：

$$\begin{aligned}
&DP' \cdot DP^{-1} = e(\mu' + R', P) \cdot e(\mu + R, P)^{-1} \\
&= e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b'_i) P + \sum_{i=e_1}^{e_{N*}} v_i \mathrm{pk}, P\Big) \cdot e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b'_i) P + \sum_{i=e_1}^{e_{N*}} v_i \mathrm{pk}, P\Big)^{-1} \\
&= e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b'_i) P + \sum_{i=e_1}^{e_{N*}} v_i \mathrm{pk}, P\Big) \cdot e\Big(-\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b_i) P + \sum_{i=e_1}^{e_{N*}} v_i \mathrm{pk}\Big), P\Big) \\
&= e\Big(\sum_{i=e_1}^{e_{N*}} v_i H(b'_i) P - v_i H(b_i) P, P\Big)
\end{aligned}$$

只有当 $\sum_{i=e_1}^{e_{N*}} v_i H(b'_i) - v_i H(b_i) = 0$ 时，$DP' \cdot DP^{-1}$ 才等于 1。也就是说只有 CSP 使用正确的数据生成证明时才能通过 SC_V的验证。因此，本文提出的方案能抵抗 CSP 的伪造攻击。

替换攻击场景如下：假设 CSP 为敌手，DO 向 CSP 发送挑战，CSP 用证明 DP′来响应 SC_V，该证明是使用未被质疑的数据块集合 b_q 来替换受质疑的数据块 b_t，$b_q \neq b_t$。如果 CSP 能使用替换证明通过 SC_V 的验证，则 CSP 赢得博弈的胜利，否则，他将会失败。

假设 CSP 可以通过伪造的证明通过 SC_V的验证，由于存储在区块链上的数据块标签是不可篡改的，因此 CSP 想要通过 SC_V验证，其伪造的证明 DP′只能和正确的证明 DP 相等，要使 DP′ = DP 则有：

$$\begin{aligned}
&e(\mu' + R', P) = e(\mu + R, P) \\
\Rightarrow &e\Big(\sum_{i \in I, i \neq t} v_i H(b_i) P + v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
&= e\Big(\sum_{i \in I, i \neq q} v_q H(b_q) P + v_i H(b_i) P + v_i \mathrm{pk}, P\Big) \\
\Rightarrow &v_t H(b_t) P = v_q H(b_q) P \\
\Rightarrow &H(b_t) = H(b_q)
\end{aligned}$$

根据哈希函数对于不同的输入 b_q 和 b_t，产生相同输出的概率极低，因此本文提出的方案可以有效抵抗 CSP 的替换攻击。

重放攻击场景如下：假设 CSP 是敌手，DO 向 CSP 发送挑战，CSP 用证明 DP′来响应 SC_V，其中 DP′是先前的数据完整性证明。如果 CSP 可以通过伪造的证明通过 SC_V的验证，则 CSP 赢得这场博弈的胜利，否则，他将会失败。

假设 CSP 取得胜利。与替换攻击的证明类似，需要满足 DP′与正确的数据完整性证明 DP 相等，同样可以得出：

$$v_t H(b_t) P = v_q H(b_q) P \tag{15}$$

每次完整性验证的数据块索引是随机选取的，并且随机数 v_i 也是不一样的，CSP 不可能通过重放之前的证明来通过 SC_V 的验证。因此本文提出的方案可以有效抵抗 CSP 的重放攻击。

3.3 EN 攻击

恶意的 EN 可能给出错误的标签完整性证明，导致 SC_V 输出错误的验证结果。与 CSP 伪造攻击类似，EN 在使用不正确的数据块标签时，会导致数据完整性验证失败，从而触发仲裁算法。

假设 EN 可以通过 SCV 的检查，它需要伪造一个数据块标签：

$$\sigma_i' = \frac{1}{H(b_i') + \mathrm{sk}} P \tag{16}$$

然而，DO 的私钥是保密的，EN 在不知道私钥的情况下无法伪造一个数据块标签满足等式（17）：

$$\frac{1}{H(b_i') + \mathrm{sk}} P = \frac{1}{H(b_i) + \mathrm{sk}} P \tag{17}$$

因此，EN 上传错误的标签证明后总会被 SC_V 发现，本文提出方案可以抵抗恶意 EN 的攻击。

4 性能分析

本文是面向 IoT 环境提出的大规模数据完整性验证方案，本节将与现有的物联网环境下的数据完整性验证方案进行比较。实验平台配置为 AMD Ryzen 7 7840H CPU、32 GB 内存和 64 位 Windows 11 操作系统，实验环境基于 Java 语言实现，采用 JPBC 密码学库完成双线性映射操作，椭圆曲线类型为 Type-A 类型曲线。实验中，使用的私钥 sk 大小为 160 bit，挑战的随机数 v_i 大小为 80 bit。为了增强实验的可对比性，将文献［11］中的扇区数目 s 设置为 1。

4.1 开销分析

在数据完整性验证方案中，计算开销主要来自标签生成（TagGen）、证明生成（ProofGen）和验证（Verify）。为直观对比本文提出的方案与现有方案的计算开销，表 2 中给出 ProofGen、TagGen、Verify 算法的计算开销分析结果。

在表 2 中 Mul 表示乘法操作，Exp 表示椭圆曲线上的标量乘法或指数运算操作，Pair 表示配对操作，Mod 表示模运算。n 表示数据块数量，s 表示扇区数目，c 表示被抽取到用于完整性验证的数据块数量。可以看出本文所提方案算法的整体开销相对较低，且本文提出的方案在验证阶段只执行一次群上的标量乘法操作，极大降低了智能合约执行算法的复杂度，减少了区块链计算开销。

4.2 单文件验证性能分析

为评估本文方案的性能，本小节首先通过实际实验对 TagGen、ProofGen、Verify 三种算法的时间消耗进行了测评，然后与文献［13］和文献［15］的方案进行对比，以验证本方案在算法时间开销上的合理性与优越性。

如图 3 所示，随着数据块大小的增加，其标签计算的时间和证明计算的时间在缓慢增加，而验证的时间基本保持不变。出现这种情况的原因在于，随着数据块文件大小的增加，其所需要的哈希运算的时间也呈线性增加，所以标签计算和证明计算的时间也在缓慢增加，并且证明计算时用到了双线性配对的操作，所以其时间消耗远大于标签计算的时间。而验证不涉及文件的哈希操作，仅进行一次椭圆曲线上的标量乘法操作，所以其计算时间基本不变，且耗时较少。因此，本文所提出的方案更适用于数据量庞大且单个数据文件体积较大的物联网数据完整性验证场景。

然后，将数据块文件大小固定为 50 KB，在数据块文件的数量 n 从 100 到 1 000，步进 100 的情况下，比较本文方案和文献［13］、文献［15］方案的 TagGen 算法时间消耗。

如图 4 所示，随着数据块数量的增加，三种方案标签计算的时间都线性增长，但文献［13］的增长速度最快，这是由于数据块文件越大，其对应的指数运算规模会急剧增大。本文方案相较于文献［15］的方案，在标签计算效率上提升 50.7%，这是由于文献［15］需要使用特定的哈希函数，因此其时间开销高于本文使用的 ZSS 签名方案。

表 2 计算复杂度对比

方案	TagGen				ProofGen				Verify			
	Mul	Exp	Pair	Mod	Mul	Exp	Pair	Mod	Mul	Exp	Pair	Mod
文献［13］	0	n	0	n	0	n	0	c	0	c	0	c
文献［15］	$n(s+1)$	ns	0	0	$2c+s$	$c+s$	0	0	$c+2$	$c+2s$	2	0
文献［18］	$n(s+1)$	$n(s+1)$	0	0	$c(s+1)$	c	1	0	$c+1$	$c+3$	2	0
文献［19］	n	n	0	0	c	$3c+1$	0	0	2	0	2	0
本文	n	n	0	0	c	$3c+1$	2	0	1	0	0	0

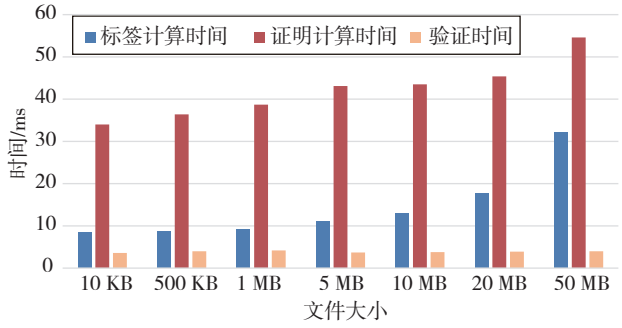

图 3　文件大小对计算开销的影响

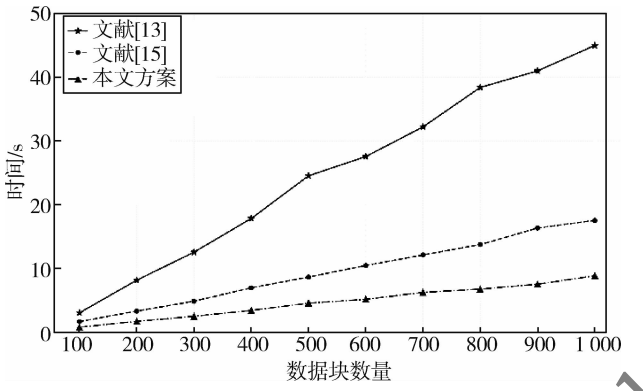

图 4　标签生成时间对比

接着，在数据块文件总数为 1 000 时，比较不同抽样数据块数量下 ProofGen 的计算开销，实验结果如图 5 所示。随着抽样数量的增加，三种方案的证明计算时间均呈上升趋势。其中，文献［13］的计算开销最高，文献［15］方案的计算时间略低于本文方案。然而，由于证明的生成由 CSP 和 EN 共同完成，而这两个实体通常具备较强的计算能力，因此整体计算开销在可接受的范围内。

图 6 展示了验证计算的时间消耗对比。可以看出文献［13］和文献［15］都随着数据块数量的增加而增长，而本文方案验证的时间开销为常数。这是由于在计算证

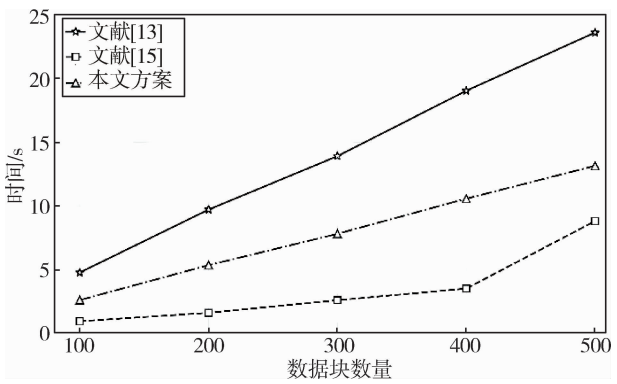

图 5　证明计算时间对比

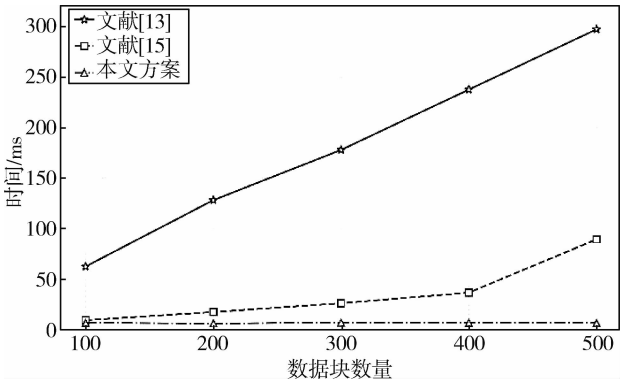

图 6　验证计算时间对比

明时，提前对生成的证明做了聚合和双线性运算，在验证时只需要做一次椭圆曲线上的标量乘法即可。文献［13］不支持签名的聚合，在验证时需要逐个比较，进行多次模幂运算，因此其计算开销较大。

最后，将文献［19］的方案与本文方案在区块链验证时间开销上进行对比。由于两者均采用 ZSS 签名，该对比能更直观地反映其计算成本的优劣。如图 7 所示，两种方案的验证计算开销在整体上保持稳定，虽存在一定波动，但变化幅度较小，相较于文献［19］，本文方案在验证计算效率上平均提升 29. 3%。这是由于本文所提方案在验证阶段只进行一次乘法操作，而文献［19］还需要进行双线性配对操作。因此本文提出的方案可以有效降低部署在区块链上智能合约的计算开销，验证方案的可扩展性更强。

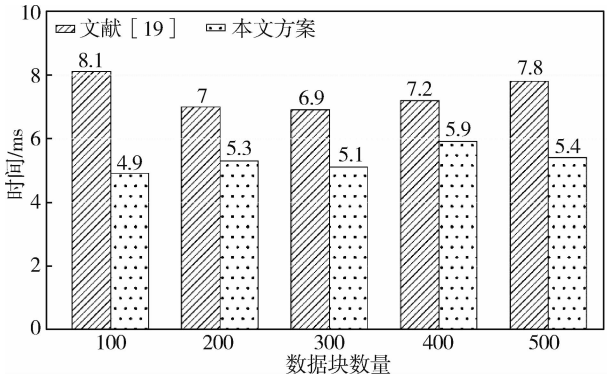

图 7　验证计算开销比较

4. 3　不同网络条件下方案的稳定性评估

为验证所提方案在真实物联网云边协同审计场景中的稳定性，本文通过模拟实际物联网部署环境中可能出现的网络延迟以及数据丢包的网络异常情况，对方案在时延、验证成功率和吞吐能力三个关键性能指标上的稳

定性进行评估。通信过程中采用 HTTP 协议，并通过 Clumsy 工具来模拟网络延迟与丢包，将 5 MB 文件划分为 1 000 块，每次随机挑战 50 个块。其余实验配置与 4.2 小节保持一致。DO、EN、CSP、SC_V 各组件运行在独立线程模拟分布式结构。实验结果取 20 次实验的平均值。结果如表 3、表 4 所示。

表 3　网络延迟对验证性能的影响

延迟/ms	成功率/%	平均验证时延/ms	标准差/ms
20	100	76.1	1.4
50	100	105.2	2.1
100	100	155.4	3.9
200	100	263.7	5.8

表 4　丢包率对验证性能的影响

丢包率/%	成功率/%	平均验证时延/ms	标准差/ms
0	100	76.1	1.4
5	100	92.5	2.6
10	98	114.8	4.7
15	93	143.3	6.9

表 3 与表 4 表明，在不同网络条件下，本文提出的云边协同数据完整性验证方案仍具备良好的鲁棒性与稳定性。在轻度丢包和中等延迟（如 5% 丢包、50 ms 延迟）下，验证成功率维持在 100%；即便在延迟和丢包率显著增加的环境（如 200 ms 延迟、15% 丢包）下，验证成功率也能保持在 93% 以上，充分验证了方案的可靠性与适应性，适合部署于实际 IoT 系统中的边缘环境和低质量网络场景。

4.4　批量验证效率

为验证本文方案在批量验证方面的优势，在抽样容量为 500，文件数量分别为 10、20、30、40 和 50 时，评估单文件验证与批量验证的平均时间消耗（包括证明计算开销与验证计算开销）。如图 8 所示，批量验证的平均时间消耗小于单文件验证。这是由于批量验证能够有效减少双线性配对运算的次数从而达到降低计算开销的效果。

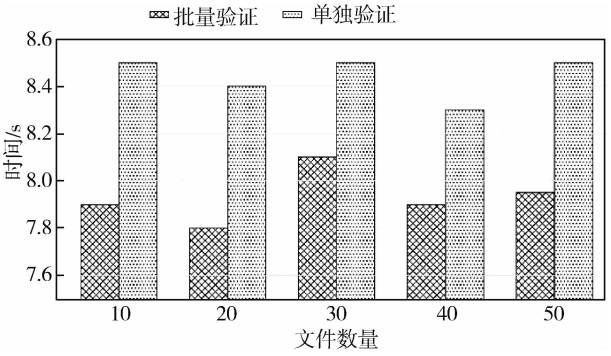

图 8　批量验证效率

5　结束语

本文提出了一种基于区块链技术的云边协同数据完整性验证方案，支持批量验证并引入 EN 以减轻区块链计算负担，防止 CSP 伪造、替换和重放攻击。该方案通过智能合约实现去中心化的数据完整性验证，并设计了有效的仲裁机制以防止不诚实行为。此外，采用 ZSS 短签名构建了一种高效的物联网数据验证协议，并在正确性和攻击防护等方面进行了理论分析。实验结果表明，所提方案在标签计算和验证时间上优于对比方案，显著降低了数据块标签计算和区块链计算、存储开销，具有大规模物联网场景下的显著优势。

参考文献

[1] 李天慈，赖贞，陈立群，等. 2020 年中国智能物联网（AIoT）白皮书 [J]. 互联网经济，2020（3）：90－97.

[2] 李晖，孙文海，李凤华，等. 公共云存储服务数据安全及隐私保护技术综述 [J]. 计算机研究与发展，2014，51（7）：1397－1409.

[3] 李艳红. 大数据背景下云存储数据安全研究 [J]. 网络安全技术与应用，2022（9）：70－71.

[4] DESWARTE Y，QUISQUATER J J，SAÏDANE A. Remote integrity checking：how to trust files stored on untrusted servers [C]// Working Conference on Integrity and Internal Control in Information Systems. Boston，MA：Springer US，2003：1－11.

[5] WANG Q，WANG C，LI J，et al. Enabling public verifiability and data dynamics for storage security in cloud computing [C]// 14th European Symposium on Research in Computer Security，2009：355－370.

[6] SHEN J，SHEN J，CHEN X，et al. An efficient public auditing protocol with novel dynamic structure for cloud data [J]. IEEE Transactions on Information Forensics and Security，2017，12（10）：2402－2415.

[7] 符庆晓，陈兰香，李继国，等. 一种高效的基于分治邻接表的动态完整性验证方案 [J]. 密码学报，2021，8（4）：601－615.

[8] 王睿垚，吴黎兵，张壮壮，等. 云存储环境下基于代数恒等式的低开销动态安全审计协议 [J]. 小型微型计算机系统，2025，46（5）：1190－1198.

[9] 张晓均，郝云溥，李磊，等. 基于工业云存储系统的数据防篡改批量审计方案 [J]. 计算机应用，2025，45（3）：891－895.

[10] 杨帆，袁艺林，张邓凡，等. 支持审计者更换和数据动态的云数据完整性审计方案 [J]. 信息安全学报，2025，10

(3)：197 – 208.

[11] 代闯闯，栾海晶，杨雪莹，等. 区块链技术研究综述 [J]. 计算机科学，2021，48 (S2)：500 – 508.

[12] LIU B，YU X L，CHEN S，et al. Blockchain based data integrity service framework for IoT data [C]//2017 IEEE International Conference on Web Services (ICWS). IEEE，2017：468 – 475.

[13] XU Y，REN J，ZHANG Y，et al. Blockchain empowered arbitrable data auditing scheme for network storage as a service [J]. IEEE Transactions on Services Computing，2019，13 (2)：289 – 300.

[14] 陈建伟，王姝妤，张美平，等. 基于区块链且可验证的智能电网多维数据聚合与分享方案 [J]. 通信学报，2024，45 (1)：167 – 179.

[15] LIN Y，LI J，KIMURA S，et al. Consortium blockchain-based public integrity verification in cloud storage for IoT [J]. IEEE Internet of Things Journal，2021，9 (5)：3978 – 3987.

[16] 杨小东，王秀秀，李茜茜，等. 基于区块链和雾计算的去中心化云端数据完整性审计方案 [J]. 电子与信息学报，2023，45 (10)：3759 – 3766.

[17] 王子园，杜瑞忠. 边缘环境下基于无证书公钥密码的数据完整性审计方案 [J]. 通信学报，2022，43 (7)：62 – 72.

[18] ZHANG Q，SUI D，CUI J，et al. Efficient integrity auditing mechanism with secure deduplication for blockchain storage [J]. IEEE Transactions on Computers，2023，72 (8)：2365 – 2376.

[19] WANG H，ZHANG J. Blockchain based data integrity verification for large-scale IoT data [J]. IEEE Access，2019，7：164996 – 165006.

(收稿日期：2025 – 06 – 19)

作者简介：

赵雄伟 (1979 –)，通信作者，男，博士，主要研究方向：信息系统、区块链和信息安全。E-mail：3909082@ qq. com。

滕威 (1999 –)，男，硕士，主要研究方向：云存储与信息安全。

楚鸿伟 (1989 –)，男，本科，主要研究方向：区块链应用。

(上接第 65 页)

参考文献

[1] 倪淑燕，陈世淼，付琦玮，等. 卫星导航欺骗干扰检测与抑制技术综述 [J]. 电讯技术，2024，64 (5)：812 – 820.

[2] HARTMANN K，GILES K. UAV exploitation：a new domain for cyber power [C]//Proceedings of 2016 8th International Conference on Cyber Conflict. Tallinn：IEEE，2016：205 – 221.

[3] GREGORY P. EASA reports GNSS jamming/spoofing amid ukraine conflict [EB/OL]. (2022 – 03 – 18) [2023 – 08 – 08]. https：//www. ainonline. com/aviation-news/air-transport/2022 – 03 – 18/easa-reports-gnss-jamming-spoofing-amid-ukraine-conflict.

[4] MOSAVI M R，NASRPOOYA Z，MOAZEDI M. Advanced anti-spoofing methods in tracking loop [J]. Journal of Navigation，2016，69 (4)：883 – 904.

[5] 马克，孙迅，聂裕平. GPS 生成式欺骗干扰关键技术 [J]. 航天电子对抗，2014，30 (6)：24 – 26，34.

[6] 王传福. 数字化混沌系统的动力学分析与伪随机序列生成算法设计 [D]. 哈尔滨：黑龙江大学，2020.

[7] 束礼宝，宋克柱，王砚方. 伪随机数发生器的 FPGA 实现与研究 [J]. 电路与系统学报，2003 (3)：121 – 124.

[8] VOSTA K P，GHOLAMI M. Pseudo random bit generator in QCA for high speed communications [J]. Heliyon，2024，10 (22)：e40238.

[9] HOU S，GUO Y，LI S. A lightweight LFSR-based strong physical unclonable function design on FPGA [J]. IEEE Access，2019，7：64778 – 64787.

[10] 中国卫星导航系统管理办公室 (CSNO). 北斗卫星导航系统空间接口控制文件 公开服务信号 [S]. 2.1 版. BDS-SIS-ICD-2.1，2016.

[11] 王迪，郝士琦，朱斌. “北斗”2 代 B1I 信号导航电文分析 [J]. 航天电子对抗，2013，29 (6)：30 – 32.

[12] 郭维. 北斗 B1I 信号的模拟与捕获算法研究 [D]. 太原：中北大学，2021.

[13] 虞玲. 北斗二号导航信号软件模拟源的研究与实现 [D]. 西安：西安电子科技大学，2014.

[14] 刘清，谢坚，王伶，等. 卫星导航欺骗式干扰源高精度直接定位方法 [J]. 电子学报，2022，50 (5)：1117 – 1122.

[15] 付晨罡，陈潇，刘婷，等. 面向北斗导航电文认证的符号级欺骗方法研究 [J]. 导航定位与授时，2024，11 (4)：16 – 25.

(收稿日期：2025 – 06 – 27)

作者简介：

王凯迪 (2001 –)，男，硕士研究生，主要研究方向：卫星导航信号应用。

曹新亮 (1970 –)，通信作者，男，博士，教授，硕士生导师，主要研究方向：综合电子系统设计与微波遥感。E-mail：caoxinliang874@ 163. com。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部