E-Commerce Letters 电子商务评论, 2026, 15(2), 992-999
Published Online February 2026 in Hans. https://www.hanspub.org/journal/ecl
https://doi.org/10.12677/ecl.2026.152240

Hans汉斯

面向电子商务的秘密共享与可验证透明日志融合机制研究

蒲增洲，陈玉玲*

贵州大学公共大数据国家重点实验室，计算机科学与技术学院，贵州 贵阳

收稿日期：2026年1月16日；录用日期：2026年1月28日；发布日期：2026年2月28日

摘 要

电子商务平台在支付签名、订单隐私保护、履约存证、售后仲裁与监管取证等环节中长期处理高敏感数据与高价值密钥。首先，针对电子商务中平台集中持密带来的密钥滥用、内部越权与单点失效等问题，本文提出一种将秘密共享与可验证透明日志融合的关键数据保护机制，使高价值密钥材料以碎片形式分布式存储并以阈值方式协同使用，同时以不可篡改的审计证据链提升事后可追责性。其次，构建覆盖订单支付、退款仲裁、商家结算与合规取证的统一体系架构，给出密钥生成、碎片协同用密、证据追加写入与可验证审计的一体化协议，并以承诺一致性机制约束恶意节点或不一致分发。最后，在统一威胁模型下，从机密性、完整性、可用性与可审计性四类目标出发，补充安全性分析与性能评估方法，量化阈值、节点可用性与日志证明长度对业务时延与故障模式的影响，为电子商务构建“分布式信任与可验证治理”的安全基础设施提供有效路径。

关键词

秘密共享，阈值密码，透明日志，可验证数据结构，可审计协议

A Study on an Integrated Mechanism of Secret Sharing and Verifiable Transparency Logs for E-Commerce

Zengzhou Pu, Yuling Chen*

State Key Laboratory of Public Big Data, School of Computer Science and Technology, Guizhou University, Guiyang Guizhou

Received: January 16, 2026; accepted: January 28, 2026; published: February 28, 2026

*通讯作者。

文章引用: 蒲增洲, 陈玉玲. 面向电子商务的秘密共享与可验证透明日志融合机制研究[J]. 电子商务评论, 2026, 15(2): 992-999. DOI: 10.12677/ecl.2026.152240

Abstract

E-commerce platforms have long handled highly sensitive data and high-value cryptographic keys in processes such as payment signing, order privacy protection, fulfillment evidence preservation, after-sales arbitration, and regulatory forensics. First, to address risks arising from centralized key custody in e-commerce—such as key misuse, insider privilege abuse, and single points of failure—this paper proposes a critical data protection mechanism that integrates secret sharing with verifiable transparency logs. High-value key material is stored in a distributed manner as shares and used collaboratively under a threshold scheme, while a tamper-evident audit evidence chain enhances ex post accountability. Second, we build a unified architecture covering order payments, refund arbitration, merchant settlement, and compliance forensics, and present an integrated protocol suite for key generation, cooperative threshold key usage, append-only evidence logging, and verifiable auditing, together with a commitment-consistency mechanism to constrain malicious participants or inconsistent distribution. Finally, under a unified threat model, we supplement security and performance evaluation methods from four objectives—confidentiality, integrity, availability, and auditability—and quantify how the threshold, node availability, and log proof length affect service latency and failure modes, providing an effective path toward security infrastructure for "distributed trust and verifiable governance" in e-commerce.

Keywords

Secret Sharing, Threshold Cryptography, Transparency Logs, Verifiable Data Structures, Auditable Protocols

1. 引言

电子商务平台在订单、支付、退款、争议仲裁与清结算等关键链路上，持续依赖少数中心化服务持有并调用高价值密钥或敏感数据。该模式在工程上易于部署，但也带来结构性风险：一旦出现内部越权、运维误操作或供应链失守，将可能导致大规模数据泄露、伪造退款、对账篡改或合规证据缺失。更重要的是，电商场景常涉及平台、商家、支付机构、物流与仲裁监管等多角色协作，仅靠中心化托管难以自然表达“多方共同授权、按需披露、过程可证明”的治理要求。设系统中待保护的高价值秘密为 $secret$，阈值系统的参与节点集合为 $P_1,\cdots,P_n$。对任意节点 P_i，其持有的秘密碎片记为 $share_i$。阈值参数记为 t，其中任意少于 t 个碎片无法恢复 $secret$，而任意不少于 t 个碎片可协同恢复 $secret$。该类门限安全性以经典秘密共享为基础[1]。为避免不一致分发导致“关键时刻解不开”，可验证秘密共享(VSS)通过公开承诺与一致性校验，使每个参与者能够验证自身 $share_i$ 的正确性[2] [3]。

另一方面，仅有阈值持密仍不足以满足电商合规：关键操作应当被记录为不可回滚且可被第三方复核的证据链。设业务证据为 $evidence_j$，其摘要为 $h_j = H\left(evidence_j\right)$，$H$ 为哈希函数。将 h_j 追加写入透明日志可为审计提供时间锚定与可验证性支撑，这类“追加写 + 证明”的结构可由可验证数据结构实现[4]，并在证书透明体系中形成工程化范式[5] [6]。近年透明日志与门限签名在标准与工业实践上持续演进，例如证书透明协议更新为 RFC 9162 [6]，阈值签名协议 FROST 标准化为 RFC 9591 [7]，软件供应链透明日志 Sigstore 形成公开可审计实践并发表系统性研究[8]，并提供可查询镜像数据集以支撑审计与研

究[9]。

然而，电商业务对高并发与低延迟敏感，不能将全量明文证据写入日志；多主体协作还引入碎片节点作恶、重放、拆分披露、日志分叉展示等问题，需要在架构、协议与评估层进行一体化设计。本文从“体系架构－核心协议－安全性能评估”三层出发，提出秘密共享与透明日志融合机制，以实现分布式持密、阈值用密、操作可审计、证据可验证。

本文贡献点为：(1) 提出面向电子商务链路的“秘密共享 + 透明日志”融合体系，将高价值密钥从中心化持有改为阈值分布式使用，并以可验证日志固化关键操作证据，形成可追责闭环。(2) 给出覆盖密钥生成、碎片协同用密、证据追加写入与一致性审计的核心协议与数据结构，明确 $share_i$ 、承诺与证明在多方协作中的作用边界。(3) 补充统一威胁模型下的安全性与性能评估框架，量化 t 、 n 、节点可用性与证明长度对时延、成功率与审计可信度的影响，并给出电商争议仲裁场景的可落地实现路线。

2. 相关工作

2.1. 秘密共享与可验证分发

秘密共享通常在有限域上构造。设有限域为 $\mathbb{F}_p$ ，其中 p 为素数。Shamir 提出的 (t,n) 门限秘密共享以多项式插值为核心，实现信息论安全的碎片分发与恢复[1]。为抵抗分发者恶意向不同节点分配不一致碎片，Feldman 提出非交互可验证秘密共享，利用公开承诺使参与者可验证自身 $share_i$ 一致性[2]；Pedersen 进一步给出信息论隐藏的承诺式 VSS，增强对抗能力[3]。

2.2. 透明日志与可验证数据结构

透明日志依赖 Merkle 树等可验证数据结构实现“追加写、可证明包含、可证明一致性”[4]。证书透明度将证书提交日志并向客户端提供包含证明与一致性证明，从而降低误签发的不可见性风险[5] [6]。在可验证日志工程领域，透明日志也被用于软件供应链签名事件记录与审计[8] [9]。

2.3. 工业规范与治理框架

设安全开发过程控制框架为 $\mathcal{C}$ ，工业规范与控制目录为系统化落地提供支撑，例如 NIST SSDF [10]、NIST SP 800-53 控制目录[11]与 ISO/IEC 27001:2022 管理体系要求[12]。在供应链透明方面，SLSA 规范强调通过可验证溯源与分级目标提高可信度[13]；SBOM 最小要素与格式标准(如 NTIA 报告与 CycloneDX)强调可读字段、自动化支持与透明性[8]。这些规范可作为本文方案的过程控制与证据格式参考。

3. 体系架构与威胁模型

3.1. 体系架构概述

本文体系由五类组件构成，分别为电子商务业务系统、阈值密钥服务、证据聚合器、透明日志服务以及审计与监控系统。

其中，业务系统负责产生支付签名、订单隐私保护、履约存证与售后仲裁等关键业务事件，并输出结构化业务证据。

证据聚合器对业务证据进行规范化处理与摘要计算，形成可审计的证据摘要并发起阈值用密请求，承担业务侧与安全侧的衔接职责。

阈值密钥服务以多节点碎片协作为基础，为高价值密钥提供分布式持有与阈值式调用能力，避免单点持密带来的越权与失效风险。

透明日志服务以追加写机制记录关键动作摘要及其最小元数据，为后续包含证明与一致性验证提供

不可回滚的时间锚定。

审计与监控系统持续获取日志树头与证明材料，对关键操作进行可验证核验，并在发现分叉、回滚或异常调用迹象时触发告警。

整体而言，本体系的核心思想是将“关键用密能力”由单点服务迁移为阈值协作能力，并将关键动作摘要写入追加写日志以支持可验证审计。其整体可视化结构如图 1 所示。

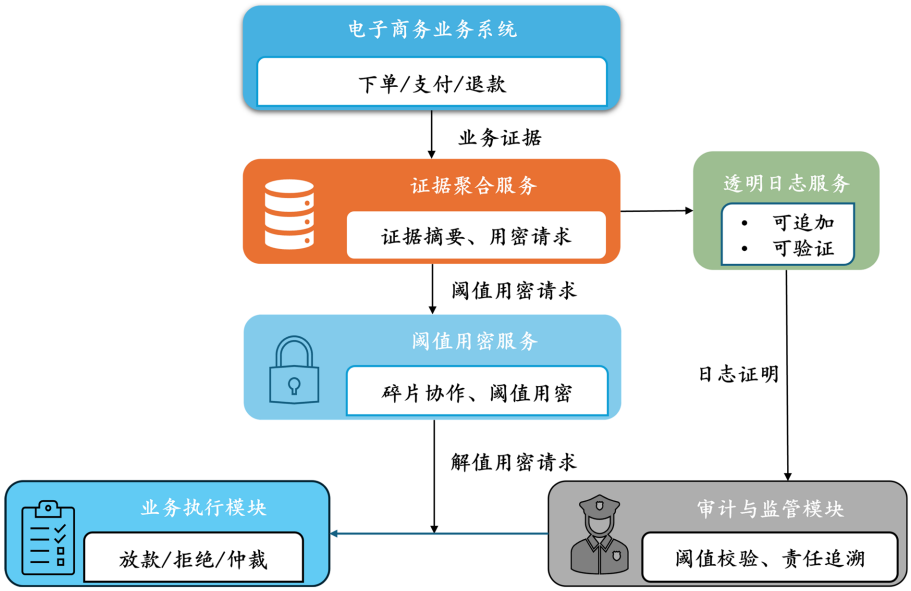

Figure 1. Schematic diagram of the architecture
图 1. 体系结构示意图

3.2. 数据对象与记号

设秘密为 $secret$，碎片为 $share_i$，阈值为 t，节点数为 n，节点索引为 $i \in 1,\cdots,n$。设证据对象为 $evidence_j$，其摘要为

$$h_j = H\left(evidence_j\right) \tag{1}$$

设日志条目为

$$entry_k = \left(h_j, meta_j, ts_j\right) \tag{2}$$

设日志树头为

$$STH_k = \left(Root_k, size_k, time_k, sig_k\right) \tag{3}$$

设密文为 ct_j，明文为 pt_j。设阈值签名为 σ_j，节点对输入产生的部分结果为 $partial_i\left(\cdot\right)$。

3.3. 威胁模型

设攻击者为 $\mathcal{A}$。$\mathcal{A}$ 的能力包括：外部攻击(入侵业务服务、重放请求、构造恶意退款)、内部越权(绕过流程进行用密)、碎片节点作恶(最多控制 $t-1$ 个碎片节点并合谋)、日志异常(对不同观察者呈现分叉视图或回滚隐藏记录)。系统目标为：机密性(少于 t 的 $share_i$ 不泄露 $secret$)、完整性、可用性以及可审计性，以体现关键操作不可被无痕篡改、节点故障下仍可完成阈值用密以及日志追加写与证明可复核，分叉可发现。

4. 核心协议设计

为减少章节碎片化，本章按功能给出三部分协议：可验证分发、阈值用密、证据透明审计。

4.1. 可验证的秘密共享生成与分发

设分发多项式为 $f(x)$，其次数为 $t-1$。用数学语言表述，$f(x)$ 由系数 $a_0, a_1, \cdots, a_{t-1}$ 构成，并满足

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \pmod{p} \tag{4}$$

其中 $a_0 = secret$。对第 i 个碎片节点，分发碎片定义为 $share_i = f(i)$。

设 $S \subseteq 1, \cdots, n$ 为参与恢复的索引集合且 $|S| \geq t$。恢复过程以拉格朗日插值为基础，可表述为：$secret = f(0)$，并且 $f(0)$ 可由 $\sum_{i \in S} share_i \cdot \lambda_i \pmod{p}$ 计算得到，其中插值系数

$$\lambda_i = \prod_{j \in S, j \neq i} \frac{-j}{i-j} \pmod{p} \tag{5}$$

为抵抗分发不一致，采用 VSS 承诺验证。设承诺群的生成元为 g，对每个系数 a_k 发布承诺 $C_k = g^{a_k}$ (其中 $k \in 0, \cdots, t-1$)。节点 i 的一致性验证可表述为：检查 g^{share_i} 是否等于 $\prod_{k=0}^{t-1} C_k^{i^k}$。若验证失败则拒绝接收并记录异常事件摘要写入审计链路，以便后续追责与恢复。

4.2. 阈值用密：协同解密与协同签名

设业务侧提交密文为 ct_j。每个节点 i 基于自身 $share_i$ 产生局部结果 $partial_i(ct_j)$。聚合器收集任意 t 个局部结果并组合得到明文 pt_j。该组合过程在语义上等价于在目标密码体制的“部分解密/部分计算”机制下，用与插值权重同构的方式合成最终结果，从而避免频繁显式重构 $secret$，降低暴露面。

设高风险操作消息为 m_j。对 m_j 生成阈值签名 σ_j，使验证端可将 σ_j 视为标准签名进行验证，同时授权从单点迁移为阈值协作。阈值签名可用于背书证据摘要 h_j 或日志条目摘要 $H(entry_k)$，使事后否认与单点伪造更难成立。相关的两轮阈值 Schnorr 协议可参考 FROST 标准[7]。

4.3. 证据提交透明日志与可验证审计

设透明日志追加写条目序列为 $entry_1, \cdots, entry_k$。定义叶哈希为

$$L_i = H(entry_i) \tag{6}$$

对任意内部节点 u，其哈希递归关系可表述为：

$$H_u = H(H_{left(u)} \| H_{right(u)}) \tag{7}$$

根记为 $Root_k$，日志发布

$$STH_k = (Root_k, size_k, time_k, sig_k) \tag{8}$$

设被查询条目为 $entry_x$，包含证明为 π_x，则包含验证谓词为

$$\text{VerifyInclusion}(H(entry_x), \pi_x, Root_k) = 1 \tag{9}$$

设两次树头为 $Root_{k_1}$ 与 $Root_{k_2}$ 且 $k_2 > k_1$，一致性证明为 π_{k_1,k_2}，则一致性验证谓词为

$$\text{VerifyConsistency}(Root_{k_1}, Root_{k_2}, \pi_{k_1,k_2}) = 1 \tag{10}$$

该类机制在证书透明体系中已形成规范化接口。

为避免泄露隐私，日志中仅追加 h_j 与最小元数据 $meta_j$，敏感明文 $evidence_j$ 保存在受控存储中。当触发合规条件时，通过阈值解密或阈值签名生成可验证结果，并在审计侧同时校验：其一，h_j 是否在时间 $time_k$ 之前已被追加写入；其二，用密或授权是否由不少于 t 方协作产生。该路径压缩“事后补录证据”与“流程外越权用密”的空间。

5. 安全性与性能评估

5.1. 安全性分析

机密性方面，设攻击者可获得的碎片集合为 $share_{i,i\in T}$ 且 $|T|<t$，则其对 $secret$ 不获得信息论增益。采用阈值用密接口可避免频繁显式重构 $secret$，降低密钥暴露面。若结合 VSS 承诺验证，则可在分发阶段发现不一致碎片，避免“数学上安全但工程上不可用”的风险[2] [3]。

完整性与不可抵赖方面，透明日志通过包含证明与一致性证明确保追加写语义，使攻击者难以在不被发现的情况下删除、回滚或替换条目[5] [6]。将高风险授权以阈值签名 σ_j 背书，可使任何单点难以伪造授权；结合日志条目 $entry_k$ 的时间锚定，可形成“授权发生于某时刻且已记录”的可验证证据链[7] [8]。

抗重放与拆分披露方面，引入一次性标识 $nonce_j$ 与有效期 exp_j，并要求每次阈值用密对请求摘要 $H\left(m_j \| nonce_j \| exp_j\right)$ 进行签名或绑定，使重放难以通过验证。通过字段分层、分级授权与频控策略限制同一对象在窗口期内的解封次数，从而将隐私泄露控制在最小必要范围。

抗分叉展示方面，设日志可能对不同观察者呈现不同 $Root_k$。审计端通过交换 STH_k 并验证 $\mathrm{VerifyConsistency}=1$ 来检测分叉，一旦对账失败即可触发告警与调查。

边界条件方面，方案安全收益依赖碎片节点独立性与审计监控的对账参与度：若多个节点被同一控制面同时攻陷，则可获得的 $share_i$ 数量可能显著增加；若审计监控不参与对账，则分叉风险将退化为“不可证伪”。因此，需要以安全开发与管理体系作为过程保障[10] [12]。

5.2. 性能评估

阈值用密开销方面，在线阶段主要成本在通信与组合。设每次阈值操作需收集 t 份响应，则通信复杂度约为 $O(t)$，组合开销与密码体制相关但常在 $O(t)$ 到 $O(t^2)$ 量级。对电商可采用风险分级函数 $t=t\left(risk_j\right)$，将高风险操作映射为更高阈值，将低风险操作保持较低时延。

日志证明开销方面，包含证明与一致性证明长度与树规模 k 相关，典型为 $O(\log k)$。为适配高并发，可采用批量提交：将窗口 Δ 内摘要 $h_1,\cdots,h_B$ 聚合为 $H\left(h_1\|\cdots\|h_B\right)$ 后写入日志，降低写入频率，同时保留批内索引映射以支持后续细粒度取证。

可用性与容错方面，系统允许最多 $n-t$ 个碎片节点不可用仍可完成阈值操作。设单节点在窗口内可用概率为 p，则阈值可用概率为

$$P_{avail}=\sum_{k=t}^{n}\binom{n}{k}p^k(1-p)^{n-k} \tag{11}$$

该模型用于规划多机房、多组织部署下的冗余度与故障恢复策略。

6. 工程化落地

碎片节点独立性与运维隔离方面，设节点集合为 $P_1,\cdots,P_n$，应在组织、账号、网络与发布链路上隔离，禁止共享同一运维密钥与发布流水线，并将碎片存储与用密执行分权，降低直接导出 $share_i$ 的可能性。对节点异常(拒绝率升高、响应延迟异常、签名不一致)建立审计告警闭环。

透明日志监控与对账方面，仅有日志写入不足以对抗分叉展示。至少部署两个独立监控者周期拉取

STH_k 并交换比对；对关键操作抽样请求包含证明并且验证是否存在 VerifyInclusion = 1。必要时将 STH_k 镜像到独立存储以增强事后调查能力[9]。

合规证据格式与最小披露方面，设证据对象为 $evidence_j$，应结构化，最小元数据 $meta_j$ 与摘要 h_j 足以支持检索、复核与责任界定。参考 SBOM 最小要素对字段与自动化的强调，可为电商证据定义最小字段集合并保持机器可读[8]。对隐私字段采用分层加密与分级授权，使每次阈值用密只解封最小必要部分，并将授权与证明同时写入审计链路。

7. 结论及展望

本文提出面向电子商务的秘密共享与透明日志融合机制，将中心化持密迁移为阈值协作用密，并以追加写可验证日志固化关键操作证据链。方案在密码学层面利用秘密共享与 VSS 机制保证机密性与一致性，在系统层面通过包含证明与一致性证明提升审计可信度，在工程层面通过节点独立性、监控对账与结构化证据实现可落地治理，并可与现有风控、权限控制与合规审计流程形成闭环联动，从而在高并发业务场景下兼顾安全性与可用性。该机制在不显著扩大隐私暴露面的前提下，将“信任平台”升级为“可验证地信任系统行为”，为电商多主体协作与合规取证提供可用技术路径。未来工作可进一步面向隐私增强与跨域协同展开研究，例如引入更强的最小披露证明与细粒度访问策略、探索多日志互认证与监管接口标准化、以及在更复杂故障与对抗条件下开展大规模性能压测与可用性验证，以提升方案在跨平台、跨组织环境中的推广性与鲁棒性。

基金项目

国家自然科学基金联合基金重点项目(U24A20241)；贵州省重大专项(黔科合重大专项字[2024]014，黔科合重大专项字[2024]003)；现代商贸深度融合新零售电商数字经济平台关键技术研究(合同编号：黔科合支撑[2023]一般 231)。

参考文献

[1] Shamir, A. (1979) How to Share a Secret. *Communications of the ACM*, **22**, 612-613. https://doi.org/10.1145/359168.359176

[2] Feldman, P. (1987) A practical Scheme for Non-Interactive Verifiable Secret Sharing. *Proceedings of the* 28*th Annual Symposium on Foundations of Computer Science* (*FOCS*). Los Alamitos, 12-14 October 1987, 427-438. https://doi.org/10.1109/SFCS.1987.4

[3] Pedersen, T.P. (1991) Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J., Ed., *Lecture Notes in Computer Science*, Springer, 129-140. https://doi.org/10.1007/3-540-46766-1_9

[4] 蔡晓晴, 王勇, 陈纯. 区块链原理及其核心技术[J]. 计算机学报, 2021, 44(1): 84-131.

[5] 于戈, 王帅, 赵宏宇. 区块链系统中的分布式数据管理技术——挑战与展望[J]. 计算机学报, 2021, 44(1): 28-54.

[6] 邵奇峰, 金澈清, 张召辉, 等. 企业级区块链技术综述[J]. 软件学报, 2019, 30(9): 2603-2625.

[7] 冯琦, 何德彪, 罗敏, 李莉. 移动互联网环境下轻量级 SM2 两方协同签名[J]. 计算机研究与发展, 2020, 57(10): 2136-2146.

[8] 孙泽雨, 吴敬征, 凌祥, 等. 软件供应链 SBOM 关键技术研究[J]. 软件学报, 2025, 36(6): 2604-2642.

[9] (2024) Nowhere to Hide: Using Transparency Logs to Secure Your Supply Chain. ACM Digital Library.

[10] 中国互联网协会. 软件安全开发能力成熟度模型与评估方法[S]. 北京: 中国互联网协会, 2024. https://www.isc.org.cn/

[11] 国家市场监督管理总局, 国家标准化管理委员会. 信息安全技术网络安全等级保护基本要求: GB/T 22239-2019 [S]. 北京: 中国标准出版社, 2019.

[12] 国家市场监督管理总局, 国家标准化管理委员会. 信息技术安全技术信息安全管理体系要求: GB/T 22080-2016

[S]. 北京: 中国标准出版社, 2016.

[13] Open Source Security Foundation (2023) Supply-Chain Levels for Software Artifacts (SLSA) Specification. https://slsa.dev/spec/