EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Blockchain- and Zero-Knowledge–Proof-Based framework for manufacturing Data-Asset qualification verification

Yujuan Xie*

School of Accounting of Guangzhou College of Commerce, Guangzhou 511363, China

Abstract

In cross-enterprise manufacturing data spaces involving device-generated batches under dynamic authorization, trusted data sharing requires consistent verification of device identity, batch state, and current access qualification. Existing approaches either verify batch evidence and qualification states separately, which may weaken cross-context consistency, or combine them into a monolithic proof that requires unnecessary recomputation when authorization, revocation, or rule states change. To address this problem, this study proposes a state-aware dual-proof framework that separates a relatively stable data sub-proof from an epoch-specific qualification sub-proof while binding both through a shared device–asset–batch–qualification context. The framework further incorporates finalized-state synchronization, a context-bound nullifier for replay prevention, and auxiliary risk screening applied only after deterministic cryptographic qualification. Experiments on two public manufacturing datasets, with authorization, revocation, epoch, and attack states constructed under a unified protocol, show that the proposed method achieves Mean CARR values of 98.2% on CONTEXT and 97.1% on IoT-Enriched, with valid-request acceptance rates of 98.3% and 98.0%, respectively. Separated updating reduces qualification-update latency from 83.1 ms to 20.3 ms, corresponding to a 75.6% update redundancy reduction. Auxiliary risk admission achieves a Macro-F1 of 90.7% with a 4.9% false rejection rate. Scalability evaluation further shows that the main bottleneck shifts toward blockchain queuing and confirmation under high concurrency. These results indicate that context binding, separated state updates, and layered admission provide an effective verification strategy for cross-enterprise manufacturing data sharing with frequently changing qualifications.

Keywords: dynamic qualification, context binding, separated state updates, layered data admission, industrial data spaces

Received on 03 August 2026, accepted on 28 August 2026, published on 24 September 2026

doi: 10.4108/eetsis.14301

1. Introduction

With the development of the industrial Internet, intelligent manufacturing, and cross-enterprise data sharing, equipment-operation, quality-inspection, and predictive-maintenance data have become important manufacturing resources [1]. Because such data often contains sensitive production information [2], trusted sharing requires verification of data integrity, source reliability, and the requester's current authorization and usage conditions [3]. This study focuses specifically on cross-enterprise manufacturing data spaces involving device-generated batches and dynamically changing access qualifications. In this setting, industrial IoT provides the data-generation and communication environment, while smart manufacturing represents the broader deployment context.

*Corresponding author. Email: 20111035@gcc.edu.cn

Existing studies use blockchain, hash commitments, and smart contracts to record data summaries and access processes [4], while digital identities and verifiable credentials support entity authentication and credential-state management [5,6]. Zero-knowledge proofs further enable qualification conditions to be verified without revealing raw data or complete credential attributes, and context information can reduce proof reuse across verification environments [7]. Merkle trees, vector commitments, and proof aggregation can also reduce the cost of batch integrity verification [8,9]. However, these mechanisms generally address relatively stable batch integrity or dynamically changing qualification states as separate problems. The central challenge is to maintain context-consistent joint verification of the device, data batch, and current qualification without repeatedly recomputing unchanged proof components when authorization states change.

Two straightforward designs expose this trade-off. A monolithic proof can bind batch information and qualification states within one verification relation, but authorization adjustments, revocations, or rule changes may require the complete proof to be regenerated. Conversely, independently maintained proofs reduce update coupling but may lack a shared context that ensures that the device, batch, and qualification refer to the same current state. This motivates a context-bound but separately updatable dual-proof design.

Cryptographic qualification alone cannot determine whether data submitted by an authenticated entity are consistent with actual operating conditions. Legitimate entities may still submit abnormal data caused by missing records, temporal offsets, or inconsistent operating states [10]. Therefore, behavioral risk screening is treated only as an auxiliary admission mechanism rather than as a substitute for deterministic credential, authorization, or ownership verification.

To address these issues, this paper proposes a state-aware dual-proof framework for dynamic authorization in cross-enterprise manufacturing data spaces. Its contributions are threefold. First, a context-bound dual-proof mechanism combines a relatively stable data proof and a dynamic qualification proof through a shared device–asset–batch–qualification context, enabling both components to be verified consistently. Second, a separated state-update mechanism retains the stable data proof when the batch context is unchanged and regenerates only the qualification component after authorization, revocation, or rule-state changes, thereby reducing state-change-induced recomputation. Third, a layered admission process separates deterministic cryptographic qualification from auxiliary risk screening, allowing abnormal requests to be reviewed or rejected without changing the validity judgment of credentials and authorization. Together, these mechanisms target manufacturing data-sharing scenarios in which batch evidence remains relatively stable while access qualifications change frequently.

2. Related work

2.1. Identity authentication and access control

Data-sharing systems commonly employ digital certificates, digital signatures, and access-control mechanisms to verify user identities, data integrity, and access permissions [11,12]. Credential-management approaches further support revocation, decentralized identity (DID), verifiable credentials (VCs), and authority delegation [13,14]. Recent decentralized digital-product architectures also integrate identity management, traceability, and access permissions to support cross-organizational asset exchange [15]. Industrial IoT solutions extend these mechanisms to blockchain-based access management and interoperable digital environments [16].

However, identity or credential validity alone does not establish that a requester's current qualification is consistently bound to a particular device, asset, and data batch. When device relationships, batch states, or authorization conditions change, existing identity-centered mechanisms still require additional context association. The present study therefore focuses on device–asset–batch–qualification consistency rather than introducing a new identity-authentication mechanism.

2.2. Trusted Blockchain-Based sharing of manufacturing data

Blockchain has been applied to industrial product traceability and manufacturing-data management by recording data summaries, production states, and access histories [17]. Digital product passport studies further combine blockchain with decentralized identity and lifecycle information management [18,19]. Cross-enterprise manufacturing-data sharing and distributed authorization have also been investigated in collaborative manufacturing and federated IoT environments [20,21]. More recent manufacturing data-space research extends this direction toward interoperable data exchange among multiple industrial participants [22].

These approaches provide traceability, auditability, and trusted state recording, but blockchain records do not themselves constitute privacy-preserving qualification proofs. In particular, recording a batch root or an authorization state does not ensure that the device, data batch, and requester qualification are jointly verified under the same current context. This distinction separates blockchain-based state recording from the context-bound qualification verification addressed in this study.

2.3. Privacy-Preserving qualification verification based on Zero-Knowledge proofs

Zero-knowledge proofs (ZKPs) enable data or qualification conditions to be verified without disclosing raw information and have been applied to regulated data trading and selective credential disclosure [23,24]. Credential-revocation mechanisms can further support privacy-preserving validity checks when credential states change [25]. Surveys of blockchain-based identity sharing have identified state binding, privacy, and replay protection as important ZKP functions [26]. ZKP-based access control has also been introduced into Industrial Internet of Things environments [27].

Recent studies have further explored anonymous ZKP authentication [28] and efficiency-oriented zk-SNARK constructions [29], demonstrating continued progress in privacy and proof efficiency. Nevertheless, these approaches primarily optimize identity, credential, transaction, or access-control verification. They do not specifically address the manufacturing case in which relatively stable device-batch evidence must remain context-consistent with authorization, revocation, and rule states that may change more frequently. The proposed framework addresses this distinction through context-bound but separately updatable data and qualification proofs.

2.4. Batch commitments and integrity verification concerning industrial data streams

Merkle-tree-based commitments and related batch-verification mechanisms can reduce the storage, communication, and verification costs associated with large IoT data streams [30]. Sparse Merkle structures further support efficient integrity auditing in cloud–edge environments [31].

These mechanisms primarily establish whether records or batches remain complete and untampered. They do not independently determine whether the same batch remains accessible under a newly changed authorization, revocation, or rule state. Consequently, the key issue considered here is not batch commitment itself, but how stable batch evidence can remain reusable while the associated qualification state is updated without full proof regeneration.

2.5. Behavioral anomalies and trusted data admission

Manufacturing data security also involves abnormal collection, data loss, malicious modification, and deviations from expected operating conditions. Existing studies combine IoT or blockchain mechanisms with data analysis to improve manufacturing-data trust [32], while learning-based approaches can identify malicious or abnormal industrial data behavior [33]. Dynamic trust-management methods provide another means of evaluating risk in industrial IoT environments [34].

Such behavioral analysis is complementary to, rather than interchangeable with, cryptographic qualification. Anomaly detection can identify statistical or behavioral deviations but cannot prove signatures, credentials, authorization, or revocation states. Conversely, cryptographic verification cannot independently determine whether data submitted by an otherwise qualified entity are operationally abnormal. Therefore, this study uses behavioral risk screening only as an auxiliary admission signal after deterministic qualification verification.

2.6. Comparison with Existing Verification Paradigms

Table 1 provides a direct comparison among the main verification paradigms discussed above. Existing studies already provide important capabilities individually: DID/VC systems support identity and credential-state management, blockchain systems provide traceability and auditable state records, ZKP schemes enable privacy-preserving verification, and commitment structures support batch integrity. The proposed framework does not claim these individual functions as new. Its distinction lies in coupling stable manufacturing-batch evidence with dynamic qualification states through a shared context while limiting recomputation when authorization-related states change.

Table 1. Comparison of Existing Verification Paradigms

Approach	Primary Function	Batch Integrity	Dynamic Qualification / Revocation	Device–Batch–Qualification Binding	Localized State Update	Manufacturing Focus
DID/VC and access control [11–16]	Identity and credential management	Limited	Supported	Partial	Credential-state level	Partial
Blockchain manufacturing sharing [17–22]	Traceability and trusted state	Supported	Partial	Partial	State-record level	Strong

	recording					
ZKP qualification/access control [23–29]	Privacy-preserving verification	Partial	Supported in some schemes	Partial	Scheme dependent	Partial
Batch commitments [30,31]	Data/batch integrity	Supported	Not primary	Limited	Batch-state level	Applicable
Proposed framework	Joint qualification verification	Supported	Supported	Explicit shared context	Separated stable/dynamic update	Strong

The comparison therefore locates the contribution at the intersection of these capabilities rather than in any single primitive. In particular, the framework maintains a shared device–asset–batch–qualification relation while separating relatively stable data evidence from more frequently changing authorization, revocation, and rule states. This design targets cross-enterprise manufacturing data spaces in which qualification changes should not automatically require complete regeneration of otherwise unchanged batch proofs.

3. Methodology

3.1. Problem Formulation and Threat Model

This study considers cross-enterprise manufacturing data sharing involving device-generated batches under dynamic authorization. Cross-enterprise manufacturing data spaces constitute the primary application scope, while industrial IoT provides the device and communication environment and smart manufacturing represents the broader deployment context. The participating entities include devices, service providers, data requesters, credential-management entities, and blockchain nodes. Devices establish their identities through registration information, and generated data are signed by the devices or trusted gateways. Their cryptographic credibility is maintained through key management and finalized on-chain states.

The threat model covers cross-device, cross-asset, and cross-batch proof reuse, stale-epoch reuse, nullifier replay, and requests based on outdated qualification states. These threats attempt to reuse otherwise valid proofs outside the device, asset, batch, epoch, or authorization context for which they were generated. The framework assumes uncompromised device private keys, valid credential issuance, trusted gateways, and blockchain operation within its consensus fault tolerance. Device-key compromise, malicious credential issuance, node collusion beyond the consensus tolerance, physical sensor tampering, and compromised trusted gateways therefore remain outside the direct protection scope. These conditions require complementary mechanisms such as secure key management, trusted issuance, and device-side protection. The framework verifies cryptographic and state consistency among signatures, batch commitments, and qualification states; it does not establish physical measurement authenticity or legal ownership.

Device D_k, where kindexes devices, generates

$$x_{k,t} = (d_{k,t}, t, \sigma_{k,t}, v_D), \quad (1)$$

where $d_{k,t}$, $\sigma_{k,t}$, and v_D denote sensor data, device signature, and data version. Records in window w and epoch eform

$$B^e_{k,w} = \{x_{k,t} \mid t \in [t_w, t_w + \Delta_w), e(x_{k,t}) = e\}, \quad (2)$$

where t_w and Δ_w are the window start and length. Qualification changes split batches at epoch boundaries.

The stable data and dynamic qualification states are

$$S^{k,w}_D = (AID, DID_k, C_D, R^{k,w}_B, v_D), S^e_Q = (R^e_A, R^e_{Rev}, v^e_P, t^e_{exp}), \quad (3)$$

where AID, DID_k, C_D, and $R^{k,w}_B$ are the asset identifier, device identifier, credential commitment, and batch root. R^e_A, R^e_{Rev}, v^e_P, and t^e_{exp} are the authorization root, revocation root, rule version, and expiration time. $S^{k,w}_D$ remains stable when the batch and device association are unchanged, whereas S^e_Q changes with qualification updates.

Requester U_i 's qualification is determined without revealing raw data or complete credential attributes:

$$q_i = Q(U_i, S^{k,w}_D, S^e_Q) \in \{0,1\}, \quad (4)$$

where $Q(\cdot)$ is the qualification predicate. $q_i = 1$ requires valid credentials, device–asset–batch consistency, authorization, non-revocation, and valid version, expiration, and epoch constraints.

Accordingly, state updates, delayed propagation, cross-context reuse, replay, and late records are handled within explicit state and context boundaries while recomputation is limited to components affected by qualification changes.

3.2. Framework overview

As shown in Figure 1, the framework follows three functional stages: state construction, context-bound dual-proof verification, and separated update with layered admission. In the first stage, device-generated records and qualification information are organized into stable and dynamic states. In the second stage, a data sub-proof π_D verifies the stable data state and a qualification sub-proof π^e_Q verifies the dynamic qualification state. A shared context C^{ctx}_D links the two proofs so that the device, batch, and current qualification are evaluated under the same verification context. Sections 3.3 and 3.4 describe state

construction and context binding, Sections 3.5 and 3.6 specify separated updating and layered admission, and Sections 3.7 and 3.8 present the security properties and end-to-end execution procedure. Figures 2 and 3 further illustrate batch-state construction and the device–batch qualification relation.

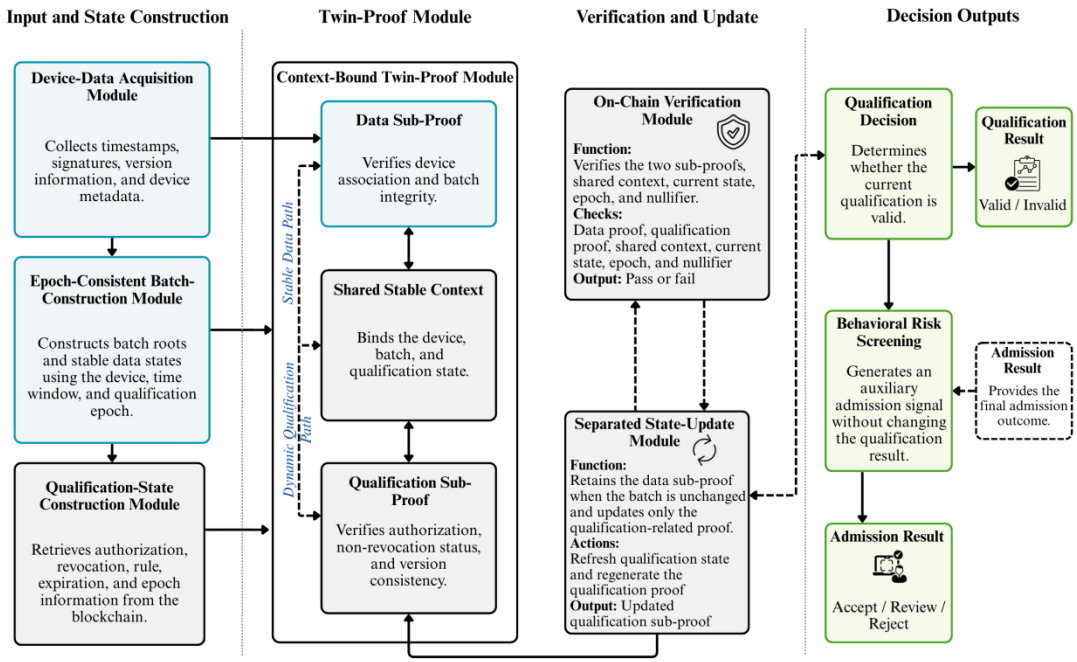

Figure 1. Framework for verifying the qualification of manufacturing data assets

The state-construction stage collects timestamps, signatures, versions, device metadata, and manufacturing records and organizes them by device and time range. Qualification states, including authorization, revocation, rule version, and validity information, are obtained from the blockchain. These two sources are not merged into a single repeatedly regenerated proof; instead, they are connected through a shared context while retaining separate stable and dynamic proof components.

The data sub-proof π_D verifies device–batch association and batch integrity, whereas the qualification sub-proof π_Q^e verifies current authorization, revocation, rule version, expiration, and epoch conditions. When the stable batch state remains unchanged, π_D is retained and only the affected qualification component is regenerated. Verification then checks both proofs, their shared context, the finalized current state, and nullifier N. Requests that satisfy deterministic qualification verification proceed to auxiliary risk screening for the final Accept/Review/Reject admission decision.

3.3. Epoch-Consistent manufacturing data-state construction

This module divides continuously generated device data streams into verifiable batches using device-related time windows and timestamp-consistency constraints. Figure 2 illustrates record verification, window partitioning, timestamp checking, batch confirmation, and record tracking. The time window Δ_w defines the batch-partitioning granularity, whereas epoch e defines the version interval of the qualification state. They are therefore not required to be one-to-one aligned: one epoch may contain multiple time windows. If the qualification state changes within a window, the batch is split at the state-transition boundary so that each resulting batch is associated with only one epoch.

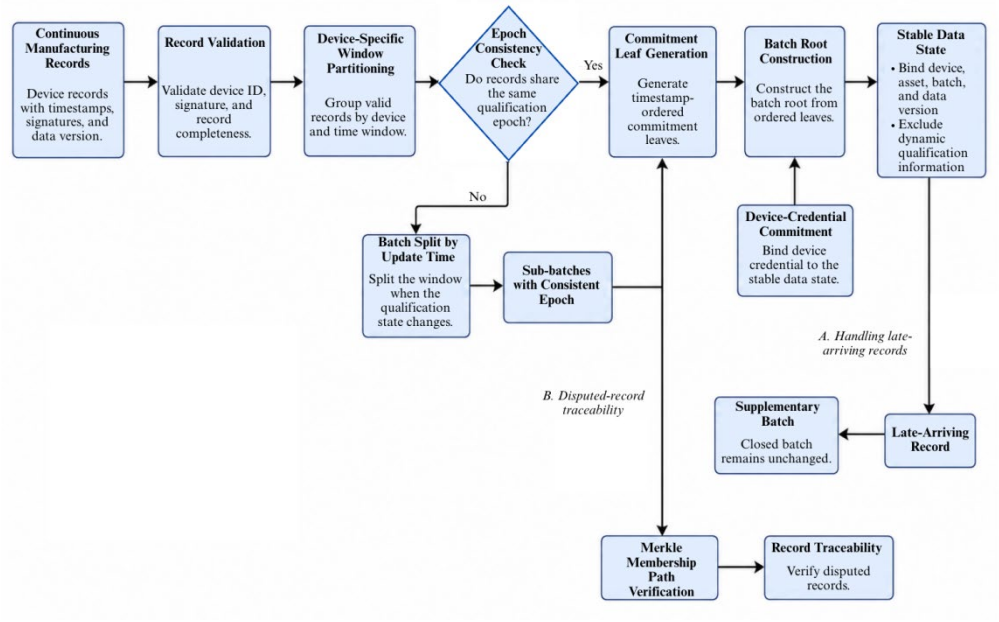

Figure 2. Sequentially consistent data state construction module

The data records within the same batch need to maintain consistent device identifiers, data versions and time information. The commitment leaf $x_{k,t}$ corresponding to the record x_(k,t) is defined as follows:

$$l_{k,t} = H\left(DID_k \parallel AID \parallel t \parallel H(d_{k,t}) \parallel v_D \parallel \sigma_{k,t}\right), \quad (5)$$

Among them, $H(\cdot)$ represents the encryption hash function, and $\parallel$ represents the concatenation operation. Timestamp-ordered leaves generate the batch root: $R_B^{k,w} = \mathrm{MerkleRoot}\left(\{l_{k,t}\}_{x_{k,t} \in B_{k,w}^e}\right)$, where $\mathrm{MerkleRoot}(\cdot)$ denotes Merkle-tree root generation. The device-credential commitment is

$$C_D = H(cred_D), \quad (6)$$

Among them, $cred_D$ represents the device credential.

The stable data state binds the device, asset, batch, and data version while excluding dynamically changing qualification information. A batch root is finalized when its corresponding batch or epoch segment is closed and is not modified by subsequent qualification updates. If authorization, revocation, or rule state changes within an open window, the current segment is closed and a new epoch-consistent segment is created. Late records arriving after closure are assigned to a supplementary batch rather than inserted into the finalized root. Historical roots therefore remain unchanged, while disputed records can still be checked through their Merkle paths. This allows the data sub-proof to remain reusable when only qualification information changes, reducing redundant proof generation.

3.4. State-Aware Device–Batch qualification relation

This module establishes a unified zero-knowledge relation among devices, data batches, and current access qualifications without disclosing raw data or complete credential information. Figure 3 illustrates the overall process of this module, including public input, private witness, joint constraints, nullifier generation, and verification process.

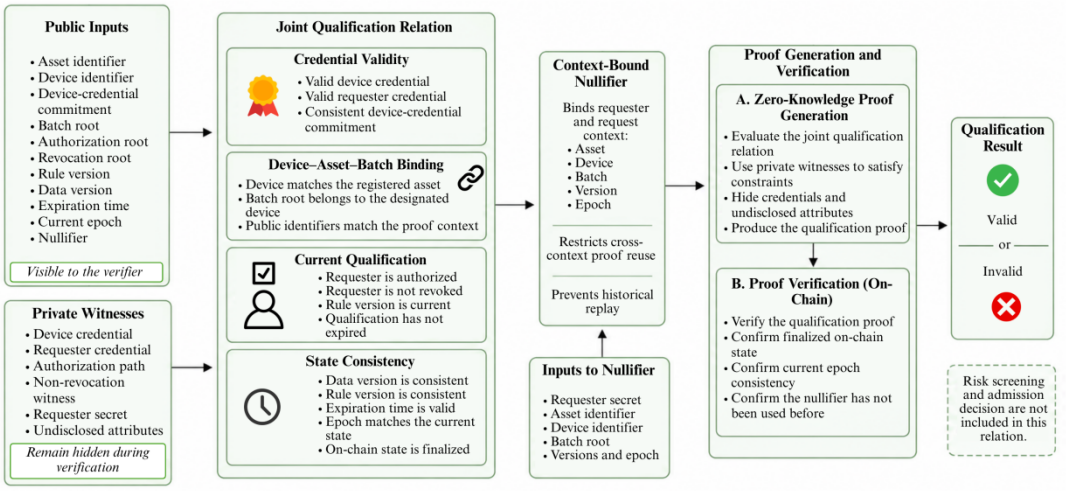

Figure 3. State-based device-batch qualification verification relationship

Public inputs include the asset identifier AID, device identifier DID_k, device-credential commitment C_D, batch root $R_B^{k,w}$, qualification state S_Q^e, version information, and nullifier. Private witnesses include the device credential, requester credential, authorization path, revocation proof, requester key, and undisclosed attributes. The shared verification context binds AID, DID_k, $R_B^{k,w}$, epoch e, rule version v_P^e, authorization root R_A^e, and revocation root R_{Rev}^e, ensuring that the data and qualification sub-proofs refer to the same finalized state.

To prevent proof reuse across different devices, assets, or batches, the context-bound nullifier is defined as

$$N = H\left(sk_{U_i} \| AID \| DID_k \| R_B^{k,w} \| v_P^e \| e\right), \quad (7)$$

where N is the nullifier and sk_{U_i} is requester U_i's secret value.

The joint relation requires valid device and requester credentials, consistency with C_D, valid device–asset–batch binding, authorization, and non-revocation. It also enforces data-version, rule-version, expiration-time, and epoch consistency. Before proof submission, the qualification component reads the latest finalized on-chain authorization, revocation, rule-version, and epoch states and synchronizes them with the shared context. If e, R_A^e, R_{Rev}^e, or v_P^e changes, the previous qualification context becomes stale and cannot be used for a new request; the stable batch context remains reusable only when AID, DID_k, and $R_B^{k,w}$ are unchanged. The verifier finally confirms that the referenced state is finalized and that N has not previously been used.

3.5. Separated Data-State and Qualification-State Update

This module separates stable data from dynamic qualification states to reduce proof regeneration after authorization, revocation, rule, or credential updates. It generates the data sub-proof π_D and epoch-specific qualification sub-proof π_Q^e separately.

The stable context digest and dynamic qualification context are defined as

$$C_D^{ctx} = H\left(AID \| DID_k \| C_D \| R_B^{k,w} \| v_D\right), \quad (8)$$

$$C_Q^e = H\left(C_D^{ctx} \| R_A^e \| R_{Rev}^e \| v_P^e \| t_{exp}^e \| e\right), \quad (9)$$

where C_D^{ctx} is the stable context digest and C_Q^e is the qualification context for epoch e. The data sub-proof π_D verifies the stable context, whereas π_Q^e takes C_D^{ctx} as a public input to verify authorization, non-revocation, rule version, expiration time, and epoch.

Proof regeneration follows an explicit state-transition procedure. When an authorization, revocation, rule, or credential change is detected, the corresponding on-chain qualification state and epoch are updated and finalized. The system then checks whether AID, DID_k, $R_B^{k,w}$, or v_D has changed. If the stable context is unchanged, (\pi_D) is retained and only the qualification witness, (C_Q^{e+1}), and π_Q^{e+1} are regenerated. The new qualification proof is accepted only against the finalized $e+1$ state. If the stable context has also changed, the affected batch context and both proof components are reconstructed. Thus, qualification-only updates do not require recalculation of unchanged batch evidence, while proofs referring to outdated qualification states become invalid.

3.6. On-Chain Verification and Layered Admission

This module integrates on-chain verification with a layered admission mechanism. The blockchain maintains the authorization root, revocation root, rule version, current epoch, and used-nullifier set.

The deterministic result is

$$z_{det} = b_D \wedge b_Q \wedge b_{ctx} \wedge b_N \wedge b_{state}, \quad (10)$$

where $z_{\mathrm{det}} \in 0{,}1$ is the deterministic verification result; b_D and b_Q indicate the validity of π_D and π_Q^e; b_{ctx} indicates shared-context consistency; b_N indicates that the nullifier is unused; and b_{state} indicates consistency with the finalized on-chain state. The qualification result is $q_i = z_{\mathrm{det}}$. If $z_{\mathrm{det}} = 0$, the request is rejected before risk screening. Only cryptographically qualified requests proceed to the auxiliary admission stage.

For a qualified batch, the system calculates an auxiliary risk score r_i from upload frequency, batch size, missing-data ratio, temporal offset, and data–operating-condition deviation. The admission decision is

$$\mathrm{a_i} = \begin{cases} \text{accept}, & \mathrm{z_{det}} = 1,\ \mathrm{r_i} < \tau_1, \\ \text{review}, & \mathrm{z_{det}} = 1,\ \tau_1 \le \mathrm{r_i} < \tau_2, \\ \text{reject}, & \mathrm{z_{det}} = 0 \text{ or } \mathrm{r_i} \ge \tau_2, \end{cases} \quad (11)$$

Here, τ_1 is the low-risk/review boundary and τ_2 is the review/reject boundary; their validation-set selection is specified in Section 4.1.1. After verification, the contract locks the nullifier, records accepted requests, and forwards review cases for inspection. Risk affects only the admission result a_i; it does not alter q_i or determine credential, authorization, or ownership validity.

3.7. Security Properties, Threat Boundaries, and Bounded Update Analysis

Under proof soundness, hash collision resistance, finalized blockchain states, and the trust assumptions in Section 3.1, the framework provides four security properties. First, context-binding correctness requires both sub-proofs to reference the same device, asset, batch, and qualification context; a proof transferred to a different context fails the shared-context check. Second, replay resistance is enforced by the context-bound nullifier and the on-chain used-nullifier set, so an already consumed N cannot authorize a second request. Third, stale-state rejection requires the qualification proof to match the finalized authorization, revocation, rule, and epoch state. Fourth, update locality limits recomputation to state components affected by a qualification change when the stable batch context remains unchanged.

The qualification update in epoch e is

$$\Delta_{\mathrm{Q}}^{\mathrm{e}} = (\Delta \mathrm{A}^{\mathrm{e}}, \Delta \mathrm{R}^{\mathrm{e}}, \Delta \mathrm{P}^{\mathrm{e}}), \quad (12)$$

where the three terms denote authorization-, revocation-, and rule-state changes. The affected components satisfy

$$\mathcal{J}_{\mathrm{e}} = \left\{ \mathrm{j} \mid \mathrm{S}_{\mathrm{Q,j}}^{\mathrm{e+1}} \neq \mathrm{S}_{\mathrm{Q,j}}^{\mathrm{e}} \right\}, \mid \mathcal{J}_{\mathrm{e}} \mid \le \mathrm{M_Q}, \quad (13)$$

where j indexes qualification-state components and M_Q is the maximum number affected by one update. If $S_D^{k,w}$ is unchanged, π_D remains valid and only the affected qualification witness, C_Q^{e+1}, and π_Q^{e+1} are regenerated.

For stale-state rejection, assuming verification occurs after final state confirmation time T_f,

$$\Pr\left[\mathrm{Verify}\left(\pi_{\mathrm{Q}}^{\mathrm{e}}, \mathrm{S}_{\mathrm{Q}}^{\mathrm{e+1}}\right) = 1 \mid \mathrm{t} \ge \mathrm{T_f}\right] \le \mathrm{negl}(\lambda), \quad (14)$$

$\mathrm{Verify}(\cdot)$ represents the verification function, λ represents the security parameter, and $\mathrm{negl}(\lambda)$ represents the probability that can be ignored under the security parameter λ.

Let $\mathrm{T_B}$, $\mathrm{T_P}$, $\mathrm{T_U}$, and $\mathrm{T_V}$ represent the time required for batch construction, proof generation, qualification update and on-chain verification respectively. Then, the end-to-end processing time of the system is:

$$\mathrm{T_{end}} = \mathrm{T_B} + \mathrm{T_P} + \mathrm{T_U} + \mathrm{T_V}, \quad (15)$$

when the qualification status remains unchanged, $\mathrm{T_U} = 0$. The security guarantees above do not independently cover leaked private keys, malicious credential issuers, node collusion beyond blockchain consensus tolerance, or physical sensor manipulation. Deployment against these threats requires complementary controls such as HSM/TPM-based key protection, secure boot or remote attestation, key rotation, and governed credential issuance.

3.8. End-to-End Qualification Verification and State Update

Algorithm 1 summarizes system initialization, epoch-consistent batch construction, dual-proof generation, finalized-state verification, layered admission, and state updating. Public parameters and proof keys are generated offline, while proof verification and state updates are executed online. The procedure explicitly distinguishes stable-context changes from qualification-only changes so that proof regeneration follows the affected state.

Algorithm 1. End-to-End Qualification Verification and State-Update Procedure

Input: Device stream $\mathrm{X} = \{\mathrm{x_{k,t}}\}$; device credential $\mathrm{cred_D}$; requester credential $\mathrm{cred_{U_i}}$; authorization state $\mathcal{A}^{\mathrm{e}}$; revocation state $\mathscr{R}^{\mathrm{e}}$; rule state $\mathcal{P}^{\mathrm{e}}$; window length Δ_{w}

Output: Qualification result $\mathrm{q_i}$; admission result $\mathrm{a_i}$; proofs π_{D} and $\pi_{\mathrm{Q}}^{\mathrm{e}}$; batch root $\mathrm{R_B^{k,w}}$

1: Initialize public parameters, proof keys, and the on-chain state.
2: Partition X by device, time window, and epoch.
3: Split an open batch at any qualification-state transition.
4: Verify each record's signature, timestamp, and data version.
5: Finalize the closed batch root and construct $S_D^{k,w}$ and C_D^{ctx}.
6: Construct $\mathrm{C_D^{ctx}}$ and generate π_{D}.
7: Read the latest finalized S_Q^e and synchronize the shared context.
8: Construct $\mathrm{C_Q^e}$, the authorization path, non-revocation witness, and N.
9: Generate $\pi_{\mathrm{Q}}^{\mathrm{e}}$ and submit both proofs, contexts, and N.
10: Verify both proofs, shared context, finalized state, and unused nullifier.
11: If verification fails, the epoch expires, or N is reused, set $\mathrm{q_i} = 0$and $\mathrm{a_i} = \mathrm{reject}$.
12: Otherwise, set $\mathrm{q_i} = 1$, lock N, and calculate $\mathrm{r_i}$.
13: On a qualification update, finalize S_Q^{e+1} and determine whether the stable context changed.
14: If unchanged, retain π_D and regenerate only C_Q^{e+1}, its witness, and π_Q^{e+1}.
15: If the stable context changed, rebuild the affected batch context and proofs; preserve previously closed roots.
16: Assign late records to a supplementary batch and return $q_i, a_i, \pi_D, \pi_Q^e, R_B^{k,w}$.

4. Experiments and results

4.1. Experimental settings

4.1.1. Datasets and preprocessing

CONTEXT is the primary dataset, and the IoT-Enriched Event Log with Data Quality Issues is the auxiliary dataset. CONTEXT was collected from a physical smart factory and contains OPC-UA and device logs with 33,089 variables. The auxiliary dataset contains 22,895 events, 3,489 cases, 52 actuators, 131 sensors, and 136,208,108 data points, including controlled missing values, complete missingness, and temporal offsets.

Neither dataset was designed for data-asset qualification verification. Therefore, only manufacturing records, device contexts, and data-quality conditions are used. Device and requester credentials, authorization and revocation states, rule versions, epochs, nullifiers, and attack requests are constructed under a unified protocol.

Timestamps are standardized, and asset–device mappings are derived from node paths, resource identifiers, and production-station numbers. After duplicate removal, records are sorted by device and time and chronologically split into training, validation, and test sets at 60:20:20. The auxiliary risk model is trained only on the training set. Model hyperparameters and the two admission thresholds τ_1 and τ_2 are determined exclusively on the validation set. Candidate threshold pairs are evaluated by Macro-F1, subject to a validation false-rejection rate not exceeding 5%; when multiple pairs satisfy the criterion, the pair producing the lower review rate is selected. The trained model and both thresholds are then frozen before test evaluation, and no test result is used for parameter or threshold adjustment. Data batches are divided into non-overlapping 60 s windows. A qualification-state transition splits an open window at the corresponding epoch boundary, while late records are assigned to supplementary batches.

Testing covers cross-device, cross-asset, cross-batch, cross-epoch, and replay attacks. Each type contains 2000 samples, resulting in a total of 10,000 attack requests. The same number of legitimate requests were set as the control group. Missing values and temporal offsets were used only for risk-admission evaluation and were not treated as contextual attack types.

4.1.2. Baselines

Table 2 summarizes 9 representative baseline methods, covering typical technical routes such as authorization management, data integrity verification, manufacturing data sharing, zero-knowledge access control, replay protection, credential revocation, and risk screening. SecuFL-IoT is only used for risk screening, and the other schemes serve as comparison baselines for different functional modules.

Table 2. Baseline methods and their main characteristics

Method	Main Characteristic
AB-DAC[35]	Attribute-based distributed access control
Fabric-IoT[36]	Consortium-blockchain access control
Consortium Auditing[37]	Blockchain integrity auditing
AIoT Integrity[38]	Cloud–edge integrity verification
BIC Manufacturing[39]	Manufacturing data sharing
DPS-IIoT[27]	Non-interactive zero-knowledge access control
ZKP Anonymous AC[7]	Anonymous blockchain access control
VC Revocation[6]	Privacy-preserving credential revocation
SecuFL-IoT[40]	Industrial anomaly-risk screening
Proposed Method	Context-bound dual proofs and separated updates

To verify the effectiveness of different design schemes, the experiment set up Monolithic Joint ZKP and two independent proofs as the control group. Among them, the monolithic joint proof needs to re-generate all the proofs after the qualification status changes, while the independent proofs separately maintain the data verification and qualification verification processes. All experimental schemes were tested based on the same data partition, attack samples, state change process and blockchain environment. For schemes that do not support joint verification, the experiment only utilized their original functions without adding additional context associations or update mechanisms. Finally, this paper's method was compared with three zero-knowledge proof schemes and two structured control schemes.

4.1.3. Implementation details

Experiments are conducted on Ubuntu 22.04 with an Intel Core i7-12700K processor, 32 GB RAM, and an NVIDIA RTX 3060 GPU. Proof generation and verification run on the CPU, while the GPU is used only for SecuFL-IoT. All methods are repeated 10 times under the same configuration; random seeds control request sampling, request order, and risk-model initialization. Results are reported as mean ± standard deviation with 95% confidence intervals.

For the scalability evaluation, the permissioned blockchain testbed uses four validation nodes and one ordering node under the default configuration, with 50 transactions per block, a 1 s block interval, eight client workers, and a 1 Gbps local network. Device scale, concurrent request load, and transaction-batching configuration are varied only in Section 4.8; other protocol and hardware settings remain fixed.

4.1.4. Evaluation metrics

The evaluation indicators include Contextual Attack Rejection Rate (CARR), Valid Request Acceptance Rate (VAR), and Update Redundancy Reduction Rate (URR).

$$CARR = \frac{N_{rej}^{attack}}{N_{attack}} \times 100\%, \quad (16)$$

$$VAR = \frac{N_{acc}^{valid}}{N_{valid}} \times 100\%, \quad (17)$$

$$URR = \left(1 - \frac{T_{sep}}{T_{full}}\right) \times 100\%. \quad (18)$$

Here, N_{rej}^{attack} and N_{attack} denote correctly rejected and total attack requests, while N_{acc}^{valid} and N_{valid} denote correctly accepted and total legitimate requests. T_{sep} and T_{full} are the separated-update and full-reproof times. Mean CARR averages the five attack types. CARR measures end-to-end attack rejection rather than cryptographic soundness. Higher values indicate better performance.

The auxiliary metrics include End-to-End Qualification-Verification Latency (EQL) and Verification Overhead (VO):

$$EQL = T_B + T_D + T_Q + T_V, \quad (19)$$

$$VO = |\pi_D| + |\pi_Q^e| + |C_D^{ctx}| + |N|. \quad (20)$$

where T_B, T_D, T_Q, and T_V denote batch construction, data-sub-proof generation, qualification-sub-proof generation, and on-chain verification times. The operator $|\cdot|$ denotes communication size. Recall, Macro-F1, and False Rejection Rate are reported only for IoT-Enriched risk screening and do not affect cryptographic qualification. The scalability evaluation additionally reports request throughput (requests·s^{-1}), P95 end-to-end latency, and per-update communication overhead.

4.2. Main experimental results

To separate cryptographic correctness from operational performance, 2,000 invalid proofs, 2,000 outdated-epoch proofs, and 2,000 reused-nullifier requests were tested. None was accepted after on-chain state finalization. Because CARR also reflects state propagation, batch splitting, and request scheduling, Tables 3 and 4 report end-to-end attack rejection rather than cryptographic soundness.

Table 3 shows that the proposed method achieves a Mean CARR of 98.2% on CONTEXT, compared with 96.1% for DPS-IIoT, while maintaining a VAR of 98.3%. The improvement is most evident in cross-batch and cross-epoch conditions because the shared context binds each proof to its device, batch root, epoch, and qualification state, reducing the possibility that a valid proof can be transferred to another context. The slightly lower VAR than conventional access-control methods reflects the stricter joint-context checks rather than weaker verification. In cross-enterprise data exchange, this trade-off is useful when preventing incorrect batch or state reuse is more important than maximizing unconditional acceptance of nominally valid requests.

Table 3. Main qualification verification results on CONTEXT

Method	Cross-Device CARR ↑	Cross-Asset CARR ↑	Cross-Batch CARR ↑	Cross-Epoch CARR ↑	Replay CARR ↑	Mean CARR ↑	VAR ↑
AB-DAC	88.7±0.9	86.9±1.0	79.8±1.3	74.6±1.5	82.4±1.2	82.5±0.8	99.1±0.4
Fabric-IoT	91.3±0.7	89.5±0.8	83.6±1.1	80.7±1.2	86.2±0.9	86.3±0.7	99.0±0.5
Consortium Auditing	78.4±1.2	75.1±1.4	97.8±0.4	69.5±1.6	79.2±1.1	80.0±0.9	97.8±0.8
AIoT Integrity	80.2±1.1	77.4±1.2	98.6±0.3	72.1±1.4	82.0±1.0	82.1±0.8	97.5±0.7
BIC Manufacturing	93.1±0.6	91.6±0.7	89.4±0.9	86.2±1.0	90.5±0.8	90.2±0.6	98.7±0.4
DPS-IIoT	97.3±0.4	96.4±0.5	95.1±0.7	94.8±0.8	97.0±0.4	96.1±0.4	98.5±0.3
ZKP Anonymous AC	96.7±0.5	95.8±0.6	93.9±0.8	93.5±0.9	99.1±0.3	95.8±0.5	98.2±0.4
VC Revocation	94.8±0.7	93.6±0.8	91.7±0.9	98.4±0.4	96.0±0.6	94.9±0.5	98.4±0.4
SecuFL-IoT	—	—	—	—	—	—	—
Proposed Method	98.1±0.4	97.6±0.5	98.0±0.6	98.2±0.4	99.0±0.3	98.2±0.3	98.3±0.4

Table 4 confirms a similar pattern on the IoT-Enriched dataset, where the proposed method reaches a Mean CARR of 97.1%. VC Revocation remains slightly stronger for epoch-specific rejection, while ZKP Anonymous AC performs slightly better for replay rejection. For risk admission, SecuFL-IoT obtains higher recall, whereas the proposed method achieves a higher Macro-F1 and a lower false rejection rate. This occurs because the proposed framework does not optimize a single attack or anomaly type in isolation; deterministic qualification and auxiliary risk screening are separated and applied sequentially. The result is a more balanced admission behavior when heterogeneous manufacturing records contain both authorization changes and data-quality anomalies.

Table 4. Qualification and Risk-Admission Results on the IoT-Enriched Dataset

Method	Mean CARR ↑	Epoch CARR ↑	Replay CARR ↑	VAR ↑	Risk Recall ↑	Macro-F1 ↑	False Rejection Rate ↓
AB-DAC	82.0±1.0	73.9±1.6	81.7±1.2	98.9±0.5	—	—	—
Fabric-IoT	85.4±0.8	79.8±1.3	85.9±1.0	99.0±0.4	—	—	—
Consortium Auditing	79.4±1.1	68.1±1.8	78.5±1.4	97.5±0.8	—	—	—
AIoT Integrity	81.3±1.0	71.0±1.6	81.0±1.3	97.3±0.7	—	—	—
BIC Manufacturing	89.7±0.7	85.3±1.1	89.8±0.8	98.6±0.5	—	—	—
DPS-IIoT	95.4±0.5	94.1±0.8	96.6±0.5	98.3±0.4	—	—	—
ZKP Anonymous AC	95.0±0.6	93.4±0.9	98.8±0.4	98.1±0.4	—	—	—
VC Revocation	94.6±0.6	98.1±0.5	95.2±0.7	98.3±0.5	—	—	—
SecuFL-IoT	—	—	—	—	93.0±0.8	90.2±0.9	6.3±0.6
Proposed Method	97.1±0.4	97.9±0.5	98.6±0.4	98.0±0.5	91.6±0.9	90.7±0.8	4.9±0.5

Note: Mean CARR represents the arithmetic mean of the CARR results across devices, assets, batches, periods, and replay attack scenarios.

Table 5 evaluates verification efficiency and resource overhead. The proposed method reduces update time from a full-reproof time of 83.1 ms to 20.3 ms, corresponding to a URR of 75.6%. The reduction results directly from update locality: when the stable device–batch context is unchanged, the data sub-proof is retained and only the qualification component is regenerated. The method therefore reduces repeated proof computation under frequent authorization changes, although its EQL and VO remain higher than those of non-zero-knowledge methods. This indicates that the main deployment benefit lies in dynamic qualification environments rather than in minimizing every verification-cost metric.

Table 5. Verification efficiency and resource overhead

Method	Proof Generation Time (ms) ↓	Verification Time (ms) ↓	Update Time (ms) ↓	Full Reproof Time (ms) ↓	URR (%) ↑	EQL (ms) ↓	VO (KB) ↓
AB-DAC	—	5.1±0.4	30.8±1.4	—	—	45.2±1.6	1.5±0.1
Fabric-IoT	—	7.1±0.3	35.0±1.2	—	—	58.3±1.9	2.2±0.1
Consortium Auditing	—	8.6±0.6	29.2±1.3	—	—	77.1±2.1	2.7±0.2
AIoT Integrity	—	9.6±0.4	33.1±1.1	—	—	82.1±2.5	3.2±0.2
BIC Manufacturing	—	8.8±0.5	40.2±1.6	—	—	92.0±2.4	3.5±0.2
DPS-IIoT	96.4±2.6	13.1±0.8	53.2±1.7	96.4±2.6	44.8±1.5	133.0±3.3	5.3±0.3
ZKP Anonymous AC	88.9±2.1	9.5±0.5	48.7±1.9	88.9±2.1	45.2±1.4	124.1±3.0	3.8±0.2
VC Revocation	91.2±2.5	12.0±0.7	42.1±1.4	91.2±2.5	53.8±1.6	127.4±2.8	4.5±0.2
SecuFL-IoT	—	—	—	—	—	—	—
Proposed Method	83.1±2.2	11.4±0.7	20.3±1.1	83.1±2.2	75.6±1.3	114.2±2.9	4.9±0.3

Figure 4 compares rejection performance across attack types. AIoT Integrity is strongest in cross-batch rejection, VC Revocation in cross-epoch rejection, and ZKP Anonymous AC in replay rejection, whereas the proposed method remains consistently high across all five categories. The flatter performance profile results from using the shared context to jointly constrain device, asset, batch, epoch, and qualification information rather than optimizing only one security function. This makes the design more suitable for cross-enterprise scenarios in which several forms of context misuse may occur concurrently.

Figure 5 shows that proof-generation time, EQL, and P95 latency increase as batch size grows from 128 to 1024. The proposed method's EQL increases from 64.8 ms to 199.1 ms. The increase is expected because larger batches require more commitment construction and proof-related computation. However, its latency remains below DPS-IIoT over the tested range, indicating that separated proof construction does not remove scale-dependent computation but limits additional overhead associated with qualification updates.

Figure 6 summarizes the security–efficiency trade-off. Conventional methods provide lower latency but weaker Mean CARR, while zero-knowledge approaches introduce additional proof costs in exchange for stronger qualification and attack rejection. The proposed method occupies an intermediate operating region: it does not minimize EQL or VO or maximize VAR, but combines high Mean CARR with substantially lower update cost. For deployment, this favors environments where authorization and revocation states change more frequently than the underlying manufacturing batch evidence.

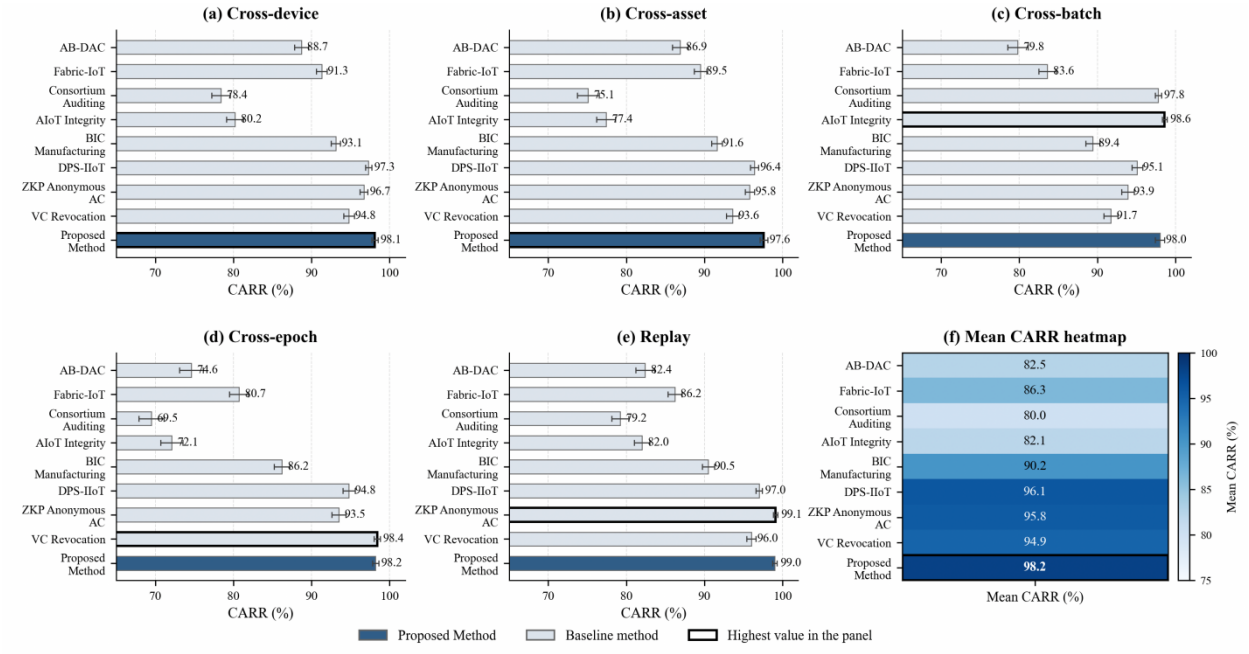

Figure 4. Context-Binding Attack Rejection under Different Attack Types.
(a) Cross-device; (b) cross-asset; (c) cross-batch; (d) cross-epoch; (e) replay; (f) Mean CARR heatmap

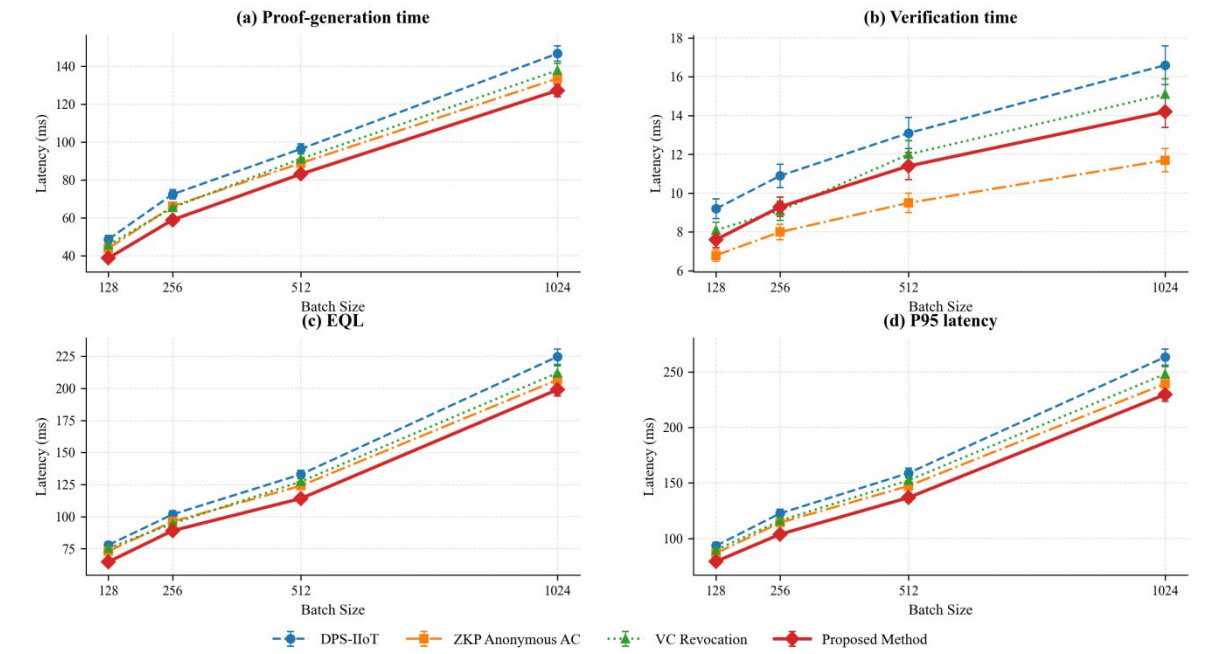

Figure 5. Calculation and verification delays under different batch sizes
(a) Proof generation time; (b) Verification time; (c) EQL; (d) P95 delay

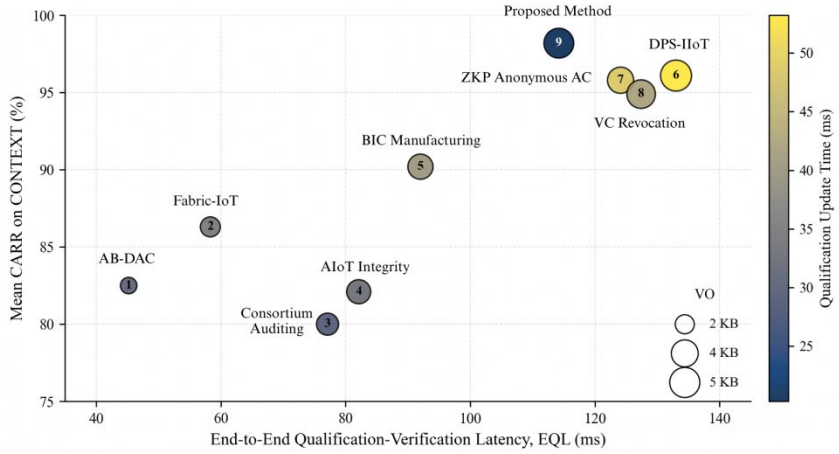

Figure 6. Security–Efficiency Trade-Off of different methods

4.3. Statistical significance analysis

Using the same data splits, attack samples, and 10 random seeds, paired two-sided t-tests were performed between the proposed method and DPS-IIoT, ZKP Anonymous AC, and VC Revocation, with Holm correction applied separately to each dataset. As summarized in Table 6, all 95% confidence intervals for Mean CARR differences exclude zero, the Holm-adjusted p-values remain below 0.05, and Cohen's d ranges from 0.99 to 1.44. These results indicate that the observed CARR gains are consistent across repeated runs rather than being driven by isolated random-seed effects.

Table 6. Statistical significance against the three strongest baselines

Dataset	Baseline	Mean Difference (pp)	95% CI	t-value	p-value	Holm-adjusted p	Cohen's d
CONTEXT	DPS-IIoT	2.1	[0.80, 3.40]	3.65	0.0053	0.0054	1.15
CONTEXT	ZKP Anonymous AC	2.4	[1.08, 3.72]	4.10	0.0027	0.0054	1.30
CONTEXT	VC Revocation	3.3	[1.66, 4.94]	4.55	0.0014	0.0042	1.44
IoT-Enriched	DPS-IIoT	1.7	[0.47, 2.93]	3.12	0.0123	0.0123	0.99
IoT-Enriched	ZKP Anonymous AC	2.1	[0.77, 3.43]	3.58	0.0059	0.0119	1.13
IoT-Enriched	VC Revocation	2.5	[1.09, 3.91]	4.02	0.0030	0.0091	1.27

Figure 7 further shows concentrated CARR distributions on both datasets, with medians above 98% on CONTEXT and similarly narrow repeated-run variation on IoT-Enriched. The proposed method also achieves a mean URR of 75.6%, compared with 44.8%–53.8% for the three zero-knowledge baselines, while its EQL remains below those baselines. Taken together, the distributions support the same mechanism observed in Table 5: separated updating provides a repeatable reduction in qualification-update overhead without relying on a single favorable run. The results therefore support the stability of the security–efficiency trade-off rather than implying that the proposed method is optimal on every individual metric.

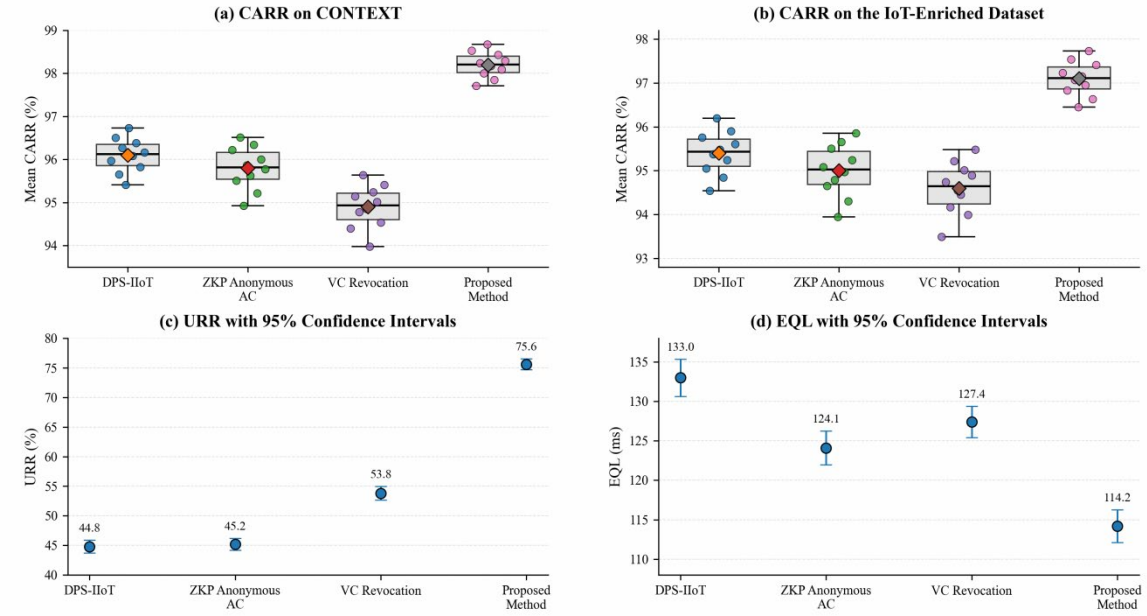

Figure 7. Repeated-Run distributions and confidence intervals.
(a) CARR on CONTEXT; (b) CARR on the IoT-Enriched dataset; (c) URR with 95% confidence intervals; (d) EQL with 95% confidence intervals

4.4. Risk-Feature and Representation-Space analysis

This section examines only auxiliary risk screening and does not interpret zero-knowledge proofs as learned representations. Upload frequency, batch size, missing-data ratio, temporal offset, and data–operating-condition deviation are processed through temporal synchronization, device mapping, and training-set standardization. The resulting five-dimensional vectors form the Preprocessed Features and are subsequently mapped into the Proposed Risk Space. Labels include normal, missing-data, temporal-offset, and compound-anomaly samples.

Table 7 shows that preprocessing improves feature-space separability: the Silhouette Score increases from 0.31 to 0.46, while the Davies–Bouldin Index and class-overlap rate decrease from 1.84 to 1.29 and from 18.7% to 11.3%, respectively. The Proposed Risk Space further reduces overlap to 8.9%, although SecuFL-IoT produces slightly greater intra-class compactness. These changes indicate that preprocessing primarily improves the distinction between normal and abnormal operating patterns rather than uniformly compressing every class, which is more relevant to auxiliary admission decisions.

Table 7. Quantitative evaluation of the Risk-Feature space

Representation	Silhouette Score ↑	Davies–Bouldin ↓	Inter-Class Distance ↑	Intra-Class Distance ↓	Class Overlap Rate ↓
Raw Features	0.31±0.02	1.84±0.06	1.72±0.08	1.21±0.05	18.7±1.0
Preprocessed Features	0.46±0.02	1.29±0.05	2.18±0.09	0.94±0.04	11.3±0.8
SecuFL-IoT Embedding	0.53±0.02	1.08±0.04	2.41±0.08	0.82±0.03	9.8±0.7
Proposed Risk Space	0.50±0.02	1.14±0.04	2.47±0.08	0.85±0.03	8.9±0.6

Figure 8 confirms this trend. Preprocessing reduces the batch-size/missing-data-ratio correlation from 0.67 to 0.32 and produces clearer separation between normal and anomalous samples. However, high-missingness and compound-anomaly samples still overlap, indicating that severe data loss can generate risk patterns similar to compound anomalies. This residual overlap explains why auxiliary risk screening retains a review region rather than assigning every abnormal-looking batch directly to rejection.

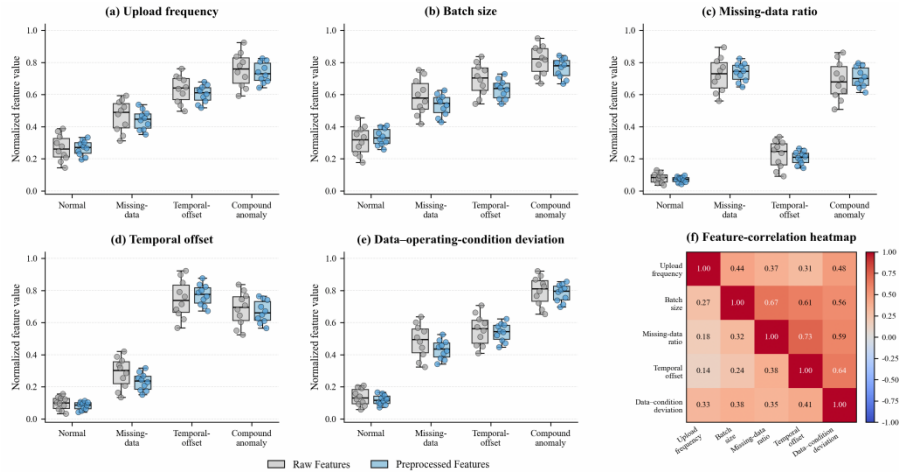

Figure 8. Risk-Feature distributions before and after preprocessing.
(a) Upload frequency; (b) batch size; (c) missing-data ratio; (d) temporal offset; (e) data–operating-condition deviation; (f) feature-correlation heatmap

Figure 9 provides consistent evidence in the lower-dimensional projections. The first two principal components retain 71.6% of the variance, while the preprocessed t-SNE representation shows a more concentrated normal cluster and greater separation from abnormal samples. Missing-data and compound-anomaly samples nevertheless retain partial overlap. These observations are consistent with the Macro-F1, Recall, and False Rejection Rate results reported for auxiliary risk screening.

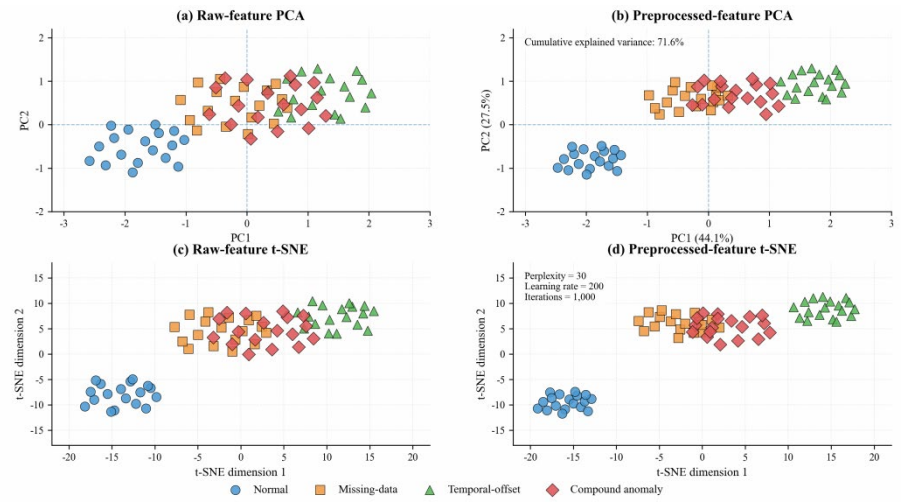

Figure 9. PCA and t-SNE Visualization of Risk-Feature Embeddings.
(a) Raw-feature PCA; (b) preprocessed-feature PCA; (c) raw-feature t-SNE; (d) preprocessed-feature t-SNE

4.5. Interpretability and Decision-Evidence analysis

To examine the basis of auxiliary admission decisions, feature masking is used to quantify the contribution of each risk feature. The IoT-Enriched labels are derived from controlled missing-data and temporal-offset conditions, whereas the CONTEXT subset follows the missing-data, temporal-offset, and compound-anomaly settings described in Section 4.1. Classification performance is reported only for IoT-Enriched because it provides complete controlled labels. For masking, each feature is replaced by its training-set mean, and its contribution is measured by the resulting decrease in Macro-F1.

Top-3 Consistency (%) is defined as the percentage of the 10 repeated runs in which a feature appears among the three highest-ranked features according to its masking-based contribution score. Thus, values of 40%, 50%, 80%, 90%, and 100% correspond to Top-3 appearances in 4, 5, 8, 9, and 10 runs, respectively.

Table 8 shows that Data–Operating-Condition Deviation and Missing-Data Ratio make the largest average contributions, whereas Upload Frequency contributes the least. Their rankings are also relatively stable across repeated runs, and the Spearman correlation between the CONTEXT and IoT-Enriched feature rankings reaches 0.90. This consistency suggests that the risk module relies mainly on data-quality and operating-state mismatch signals rather than on request frequency alone, which supports more interpretable review and rejection decisions.

Table 8. Risk-Feature Contribution and Decision Stability

Risk Feature	Mean Contribution ↑	Standard Deviation ↓	Top-3 Consistency (%) ↑	CONTEXT Rank	IoT-Enriched Rank
Upload Frequency	0.117	0.009	40	5	5
Batch Size	0.146	0.013	50	4	4
Missing-Data Ratio	0.238	0.018	90	2	3
Temporal Offset	0.214	0.015	80	3	2
Data–Operating-Condition Deviation	0.285	0.020	100	1	1

Figure 10 further shows that feature contributions change with the anomaly type. Missing-data batches are dominated by Missing-Data Ratio, whereas temporal-offset batches place greater weight on Temporal Offset and Data–Operating-Condition Deviation. Review cases generally arise from several moderate-risk signals, while rejection cases are associated with stronger individual or compound abnormalities. The evidence therefore supports the intended role of the risk module as an auxiliary admission mechanism: it explains why a qualified request is accepted, reviewed, or rejected without altering the underlying cryptographic qualification result.

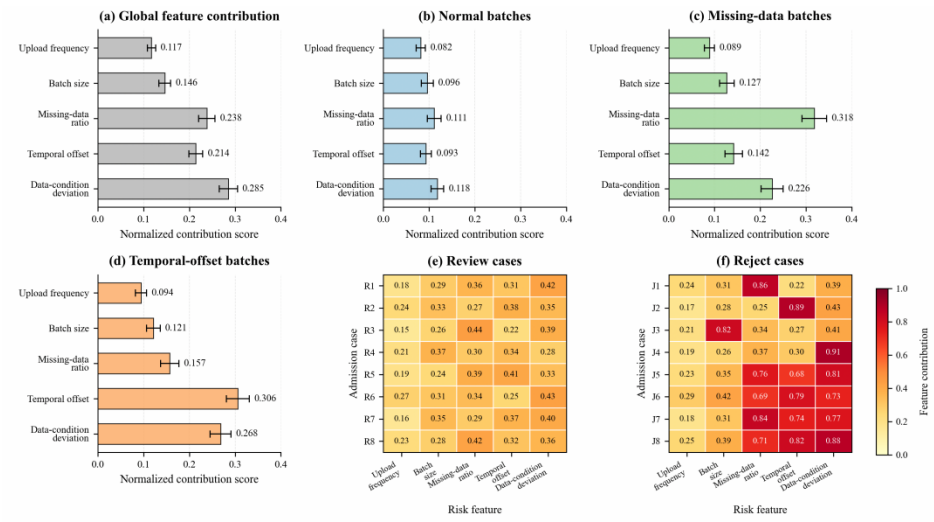

Figure 10. Risk characteristics contribution in layered admission decision (a) Global characteristic contribution; (b) Normal batch; (c) Batch with missing data; (d) Time-shifted batch; (e) Review cases; (f) Rejection cases

4.6. Industrial Scenario and Failure-Case Analysis

To clarify the practical boundary of the framework, failure cases and manufacturing-oriented protocol scenarios are analyzed using the public smart-factory records and the constructed authorization, revocation, epoch, and attack states described in Section 4.1. These experiments represent scenario-based industrial validation rather than validation from an operational enterprise deployment.

Table 9 summarizes representative failure cases. Extreme temporal offsets may send otherwise qualified batches to review, while consecutive missing records can resemble malicious deletion and cause false rejection. Records arriving after epoch closure are assigned to

supplementary batches because finalized roots are not modified. The 180 s temporal-offset threshold and 35% missing-data threshold correspond to validation-set 95th percentiles and are used only to construct representative failure conditions. These cases show that conservative state and risk controls improve protection against uncertain requests but can increase review or rejection when data quality deteriorates, motivating process-aware thresholds and bounded grace handling in deployment.

Table 9. Failure Cases, causes, and potential improvements

Failure Type	Dataset	Input Condition	Qualification Result	Admission Result	Cause	Potential Improvement
Extreme temporal offset	IoT-Enriched	Offset > 180 s	Valid	Review	Fixed temporal rule	Device-specific adaptive threshold
Consecutive missing records	IoT-Enriched	Missing-data ratio > 35%	Valid	Reject	Similarity to malicious deletion	Process-aware missingness modeling
Late record near an epoch boundary	CONTEXT	Arrival within 5 s after closure	Valid	Review	Closed batch cannot be modified	Bounded grace interval
State-propagation delay	CONTEXT	Unconfirmed state root	Temporarily unconfirmed ($q_i = 0$)	Reject	Finalized state unavailable	Cached confirmed state with expiration
Compound anomaly	IoT-Enriched	Missingness and temporal offset coexist	Valid	Review/Reject	Interacting risk signals	Rule-trace explanation

Figure 11 illustrates the three principal failure patterns: extreme temporal offset, high consecutive missingness, and late arrival near an epoch boundary. Rather than indicating cryptographic verification failure, these cases arise mainly from risk-screening boundaries or the immutability of finalized batch states. This distinction is important because their mitigation should target adaptive admission rules and late-data handling rather than weakening qualification verification.

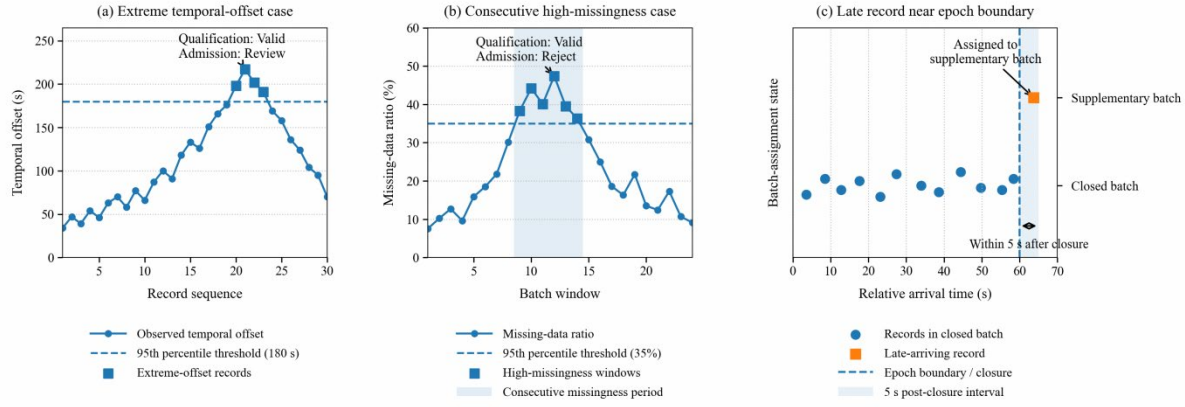

Figure 11. Representative failure cases.
(a) Extreme temporal-offset case resulting in review; (b) consecutive high-missingness case resulting in false rejection; (c) late record near an epoch boundary assigned to a supplementary batch

Figure 12 maps the existing protocol tests to four manufacturing-oriented sharing scenarios. Scenario 1 represents supplier quality-data sharing with an OEM, where a normal device batch and the supplier's current qualification must both be valid; the corresponding valid-request acceptance rates are 98.3±0.4% and 98.0±0.5% across the two datasets. Scenario 2 represents maintenance-service revocation: after the service provider's qualification is revoked, the stable batch proof is retained while the qualification component is regenerated, reducing update

time from 83.1±2.2 ms to 20.3±1.1 ms. Scenario 3 considers a qualified requester submitting data with missingness or temporal anomalies; risk recall reaches 91.6±0.9% with a false rejection rate of 4.9±0.5%, illustrating why cryptographic qualification and behavioral admission are separated. Scenario 4 represents cross-enterprise proof reuse, in which a proof associated with one device or batch is submitted under another context and is rejected through shared-context binding, with rejection rates above 98%.

These scenarios indicate that the framework is most applicable when manufacturing batches remain reusable while supplier, service-provider, or access qualifications change over time. They do not establish production-scale enterprise deployment performance, which additionally depends on organizational credential governance, network configuration, and operational integration.

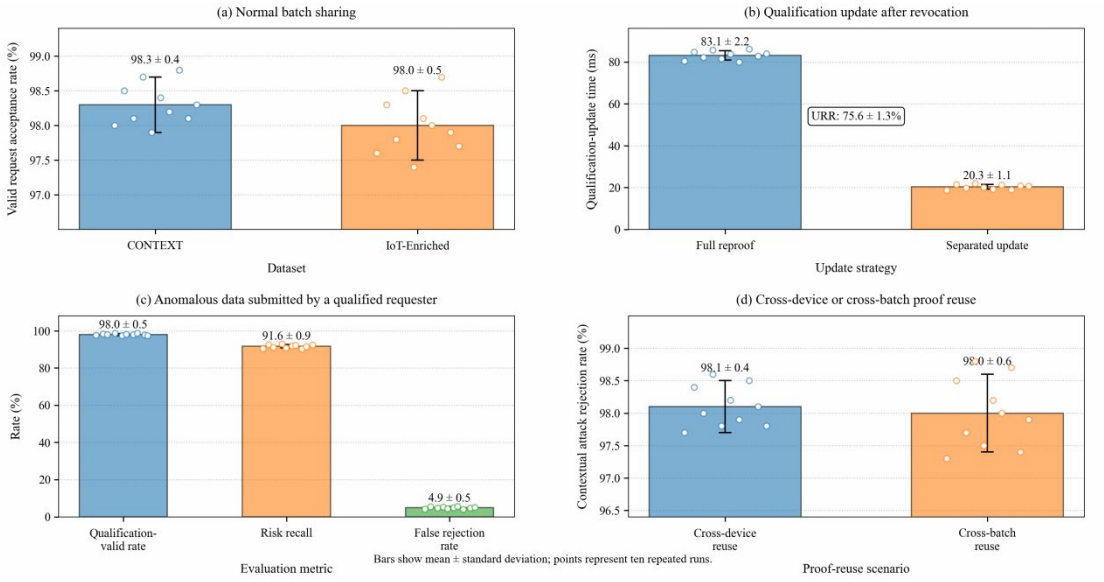

Figure 12. Qualification and Admission Outcomes under Manufacturing-Oriented Data-Sharing Scenarios. (a) Normal batch sharing; (b) qualification update after revocation; (c) auxiliary risk-screening outcomes for anomalous data submitted by qualified requesters; (d) rejection of cross-device and cross-batch proof reuse on context

4.7. Ablation study

Ablation experiments evaluate the contribution of each module and proof structure on CONTEXT and IoT-Enriched. All variants use non-overlapping 60 s windows, 512-record batches, identical attack requests, state sequences, and on-chain parameters. Results are averaged over ten fixed seeds and reported as mean ± standard deviation.

First, the context-binding mechanism primarily determines cross-context consistency. Removing the shared context reduces the CONTEXT Mean CARR to 93.0% and Cross-Batch CARR to 89.1%, whereas the full method reaches 98.2% and 98.0%, respectively. This reduction occurs because independently valid data and qualification proofs no longer establish that they refer to the same device–batch context, confirming that shared binding is the main mechanism against cross-context proof reuse.

Second, separated updating mainly affects update efficiency rather than baseline qualification accuracy. Removing this mechanism reduces URR from 75.6% to 2.4% and increases update-cycle latency from 20.3 ms to 81.7 ms; Monolithic Joint ZKP similarly requires 86.4 ms because the complete proof is regenerated. The result supports the intended update-locality design: retaining unchanged batch evidence is particularly useful when authorization or revocation changes are more frequent than underlying manufacturing data changes.

Third, freshness and replay controls have more targeted effects. Removing epoch binding reduces average Cross-Epoch CARR to 82.6%, while removing the nullifier reduces average Replay CARR to 74.8%. Disabling risk screening leaves the cryptographic metrics unchanged but removes the Risk Macro-F1 output. These effects show that epoch binding, nullifier control, and auxiliary screening perform distinct functions rather than contributing interchangeable improvements to a single aggregate metric.

Table 10. Ablation and Structural-Control results

Variant	CONTEXT Mean CARR ↑	IoT Mean CARR ↑	Cross-Batch CARR (CONTEXT) ↑	Cross-Epoch CARR (Average) ↑	Replay CARR (Average) ↑	URR (%) ↑	Update-Cycle Latency (ms) ↓	VO (KB) ↓	Risk Macro-F1 (IoT) ↑
Full Method	98.2±0.3	97.1±0.4	98.0±0.6	98.1±0.4	98.8±0.4	75.6±1.3	20.3±1.1	4.9±0.3	90.7±0.8
Monolithic Joint ZKP	97.8±0.4	96.7±0.5	97.6±0.6	97.9±0.5	98.4±0.4	0.0	86.4±2.7	4.3±0.2	90.7±0.8
Two Independent Proofs w/o Shared Context	93.0±0.8	91.9±0.9	89.1±1.2	92.8±1.0	96.2±0.7	69.8±1.5	24.9±1.3	4.6±0.2	90.7±0.8
w/o Separated Update	98.0±0.4	96.9±0.5	97.8±0.6	97.9±0.5	98.5±0.4	2.4±0.5	81.7±2.5	4.9±0.3	90.7±0.8
w/o Epoch Binding	94.9±0.7	93.8±0.8	97.4±0.6	82.6±1.4	97.3±0.6	71.5±1.4	22.8±1.2	4.8±0.2	90.7±0.8
w/o Nullifier	93.6±0.8	92.3±0.9	97.5±0.6	96.9±0.6	74.8±1.6	74.9±1.3	19.1±1.0	4.9±0.3	90.7±0.8
w/o Risk Screening	98.2±0.3	97.1±0.4	98.0±0.6	98.1±0.4	98.8±0.4	75.6±1.3	20.3±1.1	4.9±0.3	—

Note: Cross-Epoch and Replay CARR are averaged across both datasets. URR is calculated against each variant's full-reproof time. Update-cycle latency and VO use 512-record batches. Risk Macro-F1 is measured on a fixed IoT-Enriched subset; cryptographic ablations do not alter the risk module.

Figure 13 provides a compact comparison of qualification performance and update efficiency across the ablation variants, while Figure 14 links individual components to their principal target functions. Together with Table 10, the figures show that context binding mainly supports cross-batch consistency, separated updating determines recomputation efficiency, and epoch/nullifier controls provide freshness and replay protection. This modular correspondence is consistent with the three principal design objectives introduced earlier in the framework.

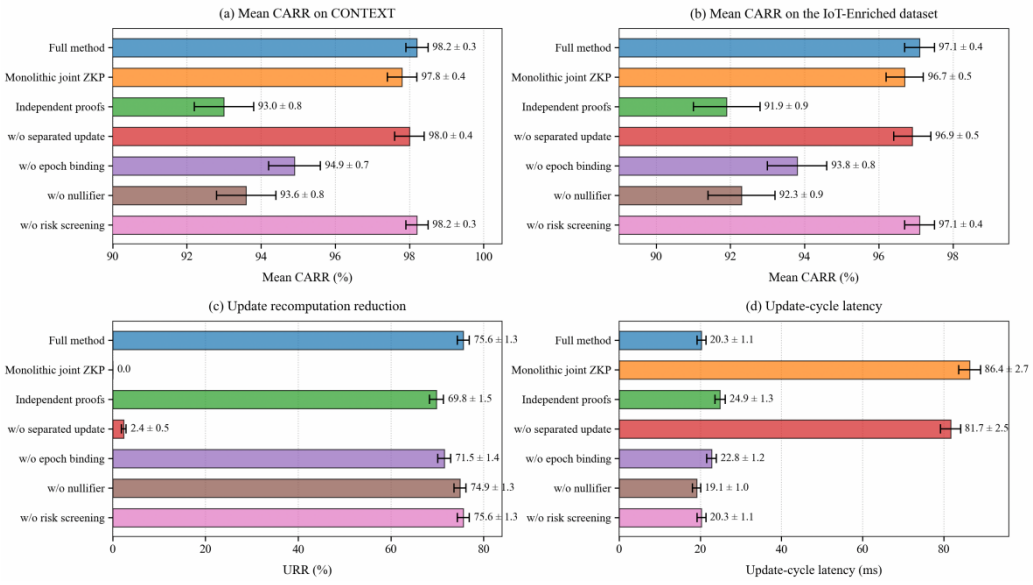

Figure 13. Performance comparison of ablation experiments and structured control schemes (a) Mean CARR on the CONTEXT dataset; (b) Mean CARR on the IoT-Enriched dataset; (c) Reduction rate of re-computation upon update; (d) Delay of update cycle

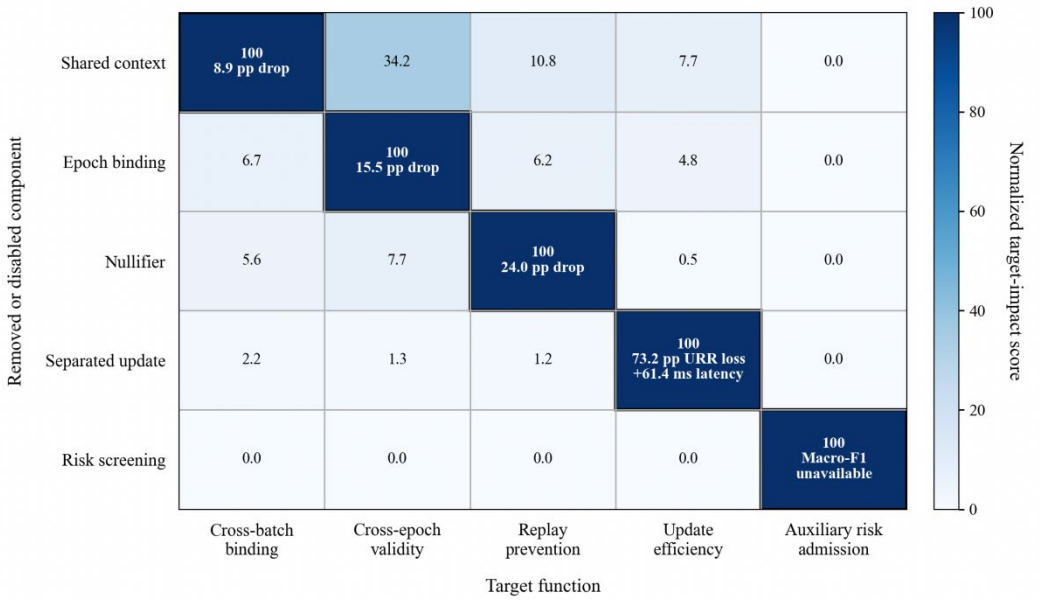

Figure 14. Correspondence between the modules and the target functions

4.8. Scalability and Concurrent-Update Evaluation

To evaluate scalability, the Proposed Method is tested under increasing device scale, concurrent qualification updates, and different blockchain configurations. Monolithic Joint ZKP and DPS-IIoT are retained as representative controls.

As shown in Table 11, increasing the number of active devices from 100 to 5,000 raises EQL from 114.2 to 158.6 ms and P95 latency from 137.6 to 244.8 ms, while throughput decreases from 47.2 to 39.1 requests·s^{-1}. Communication overhead remains approximately 4.9 KB per request because proof size is independent of the registered-device population.

Table 11. Scalability Settings and Results

Evaluation	Scale	Latency (ms)	Throughput (requests·s^{-1})	P95 (ms)	VO / Queue
Device scale	100	114.2	47.2	137.6	4.9 KB
	500	117.5	46.8	149.3	4.9 KB
	1,000	123.1	45.9	166.7	4.9 KB
	2,000	134.6	43.7	194.5	4.9 KB
	5,000	158.6	39.1	244.8	4.9 KB
Concurrent updates	1	20.3	38.4	27.4	0%
	10	21.5	44.7	34.8	0%
	50	31.6	47.1	71.2	2.1%
	100	44.3	47.6	119.6	7.8%
	200	69.7	47.8	201.3	18.6 %

Under concurrent updates, latency remains 69.7 ms at 200 requests, although queueing increases as blockchain capacity becomes saturated. At this load, Monolithic Joint ZKP and DPS-IIoT require 231.4 ms and 148.6 ms, respectively, showing that separated updating retains its advantage under concurrent qualification changes.

Three combined blockchain configurations were evaluated: 2 validators with 25 transactions per block, 4 validators with 50 transactions per block, and 6 validators with 100 transactions per block. Throughput increases from 23.4 to 87.9 requests·s^{-1}, while confirmation latency rises from 0.82 to 1.14 s. This indicates a trade-off between throughput and state-finalization delay.

Figure 15 summarizes the results. Overall, increasing device and request loads raises latency gradually, while high-concurrency conditions shift the main bottleneck from proof regeneration to blockchain queuing and confirmation.

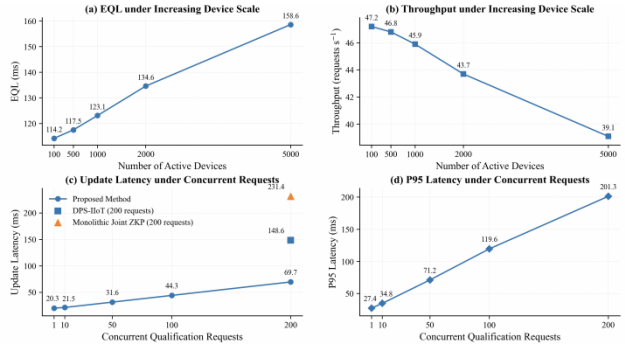

Figure 15. Scalability under Increasing Device and Request Loads.
(a) EQL versus device number; (b) throughput versus device number; (c) update latency under concurrent requests; (d) P95 latency under concurrent requests

5. Discussion and Conclusion

The results indicate that the main value of the proposed framework lies in combining context binding with separated state updates rather than in any single cryptographic primitive. Shared context binds each valid proof to the corresponding device, asset, batch, epoch, and qualification state, thereby restricting cross-context reuse. Separated updating is effective because authorization or revocation changes do not necessarily alter finalized batch evidence; complete proof regeneration would therefore introduce redundant computation. Auxiliary risk screening serves a different purpose: it identifies behavioral or data-quality abnormalities but cannot establish credential, authorization, or revocation validity. Together, these functions explain the observed balance between attack rejection, update efficiency, and admission control.

From an industrial perspective, the framework is particularly relevant to supplier data sharing and third-party equipment-maintenance services. In supplier–OEM exchange, quality-inspection records, machine-operation data, and production-batch information can be associated with the supplier's current qualification before access. In maintenance services, access scope may expire, be revoked, or change while historical equipment batches remain unchanged; separated qualification updating therefore avoids regenerating stable batch proofs. The practical benefit lies primarily in reducing repeated-proof overhead when qualification states change more frequently than batch evidence. The scenario-based experiments in Section 4.6 support these use cases without implying operational enterprise deployment.

The intended application scope is cross-enterprise manufacturing data spaces involving device-generated batches, dynamic authorization, and auditable blockchain states. The framework is less suitable for offline standalone devices without reliable state synchronization, environments lacking trusted credential issuance, or applications where physical sensor authenticity is the primary concern. It may also be unsuitable for extremely low-latency control loops because proof generation and blockchain confirmation introduce additional delay. The scalability results further indicate that blockchain queuing and confirmation increasingly become bottlenecks under high concurrency.

Several limitations remain. First, the public datasets were not originally designed for qualification verification; credentials, authorization and revocation states, epochs, nullifiers, and attack requests were constructed under a unified protocol. Second, compromised device keys and malicious credential issuance remain outside the direct protection scope and require secure key management and governed credential issuance. Third, protection against blockchain-node collusion is bounded by the underlying consensus assumptions. Fourth, physical sensor manipulation requires complementary secure sensing, trusted hardware, or remote attestation. Fifth, scalability remains dependent on blockchain throughput and network conditions. Finally, fixed risk thresholds and strict epoch boundaries may increase review or false rejection under severe missingness, temporal offsets, or delayed records.

Overall, context-bound verification and separated qualification updates reduce redundant proof regeneration while preserving consistency among manufacturing devices, data batches, and current qualifications. The framework is therefore most appropriate for dynamically authorized cross-enterprise data-sharing environments where stable batch evidence coexists with frequently changing qualification states, rather than as a universal solution for industrial data security.

Acknowledgement

This research was supported by China Commercial Accounting Society Data Assets Branch 2025 Research Project (2025DA034): The Assetization Paradox of Generative AI Training Data:

The Disruptive Impact of Copyright Compliance on Measurement Reliability

References

[1] Gabellini M, Civolani L, Ronchi M, Naldi LD, Regattieri A. Data spaces in manufacturing and supply chains: a review and insights from European initiatives. Appl Sci (Basel). 2025;15(11):5802. doi: 10.3390/app15115802.

[2] Mabkhot MM, Kalawsky RS, Liaqat A. Introducing the manufacturing digital passport (MDP): a new concept for realising digital thread data sharing in aerospace and complex manufacturing. Systems (Basel). 2025;13(8):700. doi: 10.3390/systems13080700.

[3] Tapia E, Sastoque-Pinilla L, Gamecho B, Lopez-Novoa U. Integration of blockchain-based digital identity management in an aeronautical manufacturing setting. Int J Adv Manuf Technol. 2025 Aug 14. doi: 10.1007/s00170-025-16260-w. [Epub ahead of print].

[4] Li C, Yu T, Li W, Liu Y, Yang H. Enhancing CAD data integrity and security in supply chain networks using blockchain. Int J Inf Syst Supply Chain Manag. 2025;18(1):1-22. doi: 10.4018/IJISSCM.389716.
[5] Cortés-Santacruz F, Carrillo-Martínez LA, García-Bañuelos L, Fortoul-Díaz JA. DLT in manufacturing: a systematic review of applications, taxonomy, enablers, maturity, and challenges. Results Eng. 2026;29:108277. doi: 10.1016/j.rineng.2025.108277.
[6] Zhang T, Wang Y, Gong B, Xu J, Wu J, Wan C. Privacy protection during the issuance and revocation of verifiable credentials in self-sovereign identity. Concurr Comput. 2025;37(9-11):e70084. doi: 10.1002/cpe.70084.
[7] Wu Y, Matsubara Y, Kasahara S. Enhancing account information anonymity in blockchain-based IoT access control using zero-knowledge proofs. Electronics (Basel). 2025;14(14):2772. doi: 10.3390/electronics14142772.
[8] Sud E, Agarwal S, Upadhyay L. The power I know: zero-knowledge proofs and their transformative role in the future of cryptography. IEEE Access. 2025;13:147317-29. doi: 10.1109/ACCESS.2025.3599555.
[9] Liu M, Yang T, Shi W, Vasilakos AV, Lu N. ESDI: an efficient and secure data integrity verification scheme for indoor navigation. Future Gener Comput Syst. 2025;168:107759. doi: 10.1016/j.future.2025.107759.
[10] Liu Z, Lei Z, Wen G, Xi Y, Su Y, Feng K, et al. Anomaly detection of machinery under time-varying operating conditions based on state-space and neural network modeling. Adv Eng Inform. 2025;65:103285. doi: 10.1016/j.aei.2025.103285.
[11] Logrippo L. Data flow security in role-based access control. J Inf Secur Appl. 2025;90:103997. doi: 10.1016/j.jisa.2025.103997.
[12] Cao Z, Wen X, Ai S, Shang W, Huan S. A decentralized authentication scheme for smart factory based on blockchain. Sci Rep. 2024;14(1):24640. doi: 10.1038/s41598-024-76065-x.
[13] Lee Y, Shin H, Choi D. A survey on credential revocation and DID deactivation in self-sovereign identity systems. IEEE Access. 2026;14:16089-115. doi: 10.1109/ACCESS.2025.3649792.
[14] Čučko Š, Turkanović M. A novel model for authority and access delegation utilizing self-sovereign identity and verifiable credentials. IEEE Access. 2025;13:115102-34. doi: 10.1109/ACCESS.2025.3582312.
[15] de Diego S, Gutiérrez-Aguero I. Decentralized digital product passport building blocks for enhancing supply chain sovereignty and circular economy practices. IEEE Access. 2025;13:137973-85. doi: 10.1109/ACCESS.2025.3594826.
[16] Ali G, Shah S, Elaffendi M, Ahmad N. Blockchain-based access management framework for interoperable digital twins in industrial IoT. Front Blockchain. 2025;8:1693926. doi: 10.3389/fbloc.2025.1693926.
[17] Xiao L, Sun W, Chang S, Lu C, Jiang R. Research on the construction of a blockchain-based industrial product full life cycle information traceability system. Appl Sci (Basel). 2024;14(11):4569. doi: 10.3390/app14114569.
[18] Hulea M, Miron R, Muresan V. Digital product passport implementation based on multi-blockchain approach with decentralized identifier provider. Appl Sci (Basel). 2024;14(11):4874. doi: 10.3390/app14114874.
[19] Voulgaridis K, Lagkas T, Angelopoulos CM, Boulogeorgos AAA, Argyriou V, Sarigiannidis P. Digital product passports as enablers of digital circular economy: a framework based on technological perspective. Telecommun Syst. 2024;85(4):699-715. doi: 10.1007/s11235-024-01104-x.
[20] Zheng K, Ding K, Hui J, Zhang F, Lv J, Chan FTS. Blockchain-based credible manufacturing data sharing for a collaborative manufacturing supply chain. Int J Prod Res. 2024;62(6):2233-54. doi: 10.1080/00207543.2023.2217292.
[21] Jarosz M, Wrona K, Zieliński Z. Distributed ledger-based authentication and authorization of IoT devices in federated environments. Electronics (Basel). 2024;13(19):3932. doi: 10.3390/electronics13193932.
[22] Nizamis A, Gkonis P, Ioannidis D, Ntafalias A, Tzovaras D, Trakadas P. Manufacturing data spaces applications in Europe—a survey. Data Brief. 2025;63:112149. doi: 10.1016/j.dib.2025.112149.
[23] Li D, Ke X, Zhang X, Zhang Y. A trusted and regulated data trading scheme based on blockchain and zero-knowledge proof. IET Blockchain. 2024;4(4):443-55. doi: 10.1049/blc2.12070.
[24] Flamini A, Sciarretta G, Scuro M, Sharif A, Tomasi A, Ranise S. On cryptographic mechanisms for the selective disclosure of verifiable credentials. J Inf Secur Appl. 2024;83:103789. doi: 10.1016/j.jisa.2024.103789.
[25] Begum N, Nakanishi T. Issuer-revocable issuer-hiding attribute-based credentials using an accumulator. In: 2023 Eleventh International Symposium on Computing and Networking (CANDAR); 2023 Nov 28-Dec 1; Matsue, Japan. Los Alamitos (CA): IEEE Computer Society; 2023. p. 93-9. doi: 10.1109/CANDAR60563.2023.00019.
[26] Zhou L, Diro A, Saini A, Kaisar S, Hiep PC. Leveraging zero knowledge proofs for blockchain-based identity sharing: a survey of advancements, challenges and opportunities. J Inf Secur Appl. 2024;80:103678. doi: 10.1016/j.jisa.2023.103678.
[27] Li D, Crespi N, Minerva R, Liang W, Li KC, Kołodziej J. DPS-IIoT: non-interactive zero-knowledge proof-inspired access control towards information-centric Industrial Internet of Things. Comput Commun. 2025;233:108065. doi: 10.1016/j.comcom.2025.108065.
[28] Madine M, Salah K, Jayaraman R, Yaqoob I. Zero-knowledge proofs for anonymous authentication of patients on public and private blockchains. Array. 2025;28:100590. doi: 10.1016/j.array.2025.100590.
[29] Zhu X, Song X, Deng Y, Yang G. Fast and designated-verifier friendly zk-SNARKs in the BPK model. Cybersecurity. 2025;8(1):113. doi: 10.1186/s42400-025-00432-y.
[30] Liang C, Zhang J, Ma S, Zhou Y, Hong Z, Fang J, et al. Study on data storage and verification methods based on improved Merkle mountain range in IoT scenarios. J King Saud Univ Comput Inf Sci. 2024;36(6):102117. doi: 10.1016/j.jksuci.2024.102117.
[31] Du R, Wang Z, Shen J. Certificateless data integrity auditing with sparse Merkle trees for the cloud-edge environment. Sci Rep. 2025;15(1):39202. doi: 10.1038/s41598-025-14041-9.
[32] Li Z, Zheng P, Tian Y. Application of IoT and blockchain technology in the integration of innovation and industrial chains in high-tech manufacturing. Alex Eng J. 2025;119:465-77. doi: 10.1016/j.aej.2025.01.020.
[33] Ali H, Cano A, Ahmed I. Machine learning-based early detection of malicious G-code manipulations in 3D printing. J Manuf Process. 2025;145:211-35. doi: 10.1016/j.jmapro.2025.04.012.
[34] Kumar R, Sharma R. AI-driven dynamic trust management and blockchain-based security in industrial IoT. Comput

Electr Eng. 2025;123:110213. doi: 10.1016/j.compeleceng.2025.110213.
[35] Wang P, Yue Y, Sun W, Liu J. An attribute-based distributed access control for blockchain-enabled IoT. In: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob); 2019 Oct 21-23; Barcelona, Spain. New York: IEEE; 2019. p. 1-6. doi: 10.1109/WiMOB.2019.8923232.
[36] Liu H, Han D, Li D. Fabric-IoT: a blockchain-based access control system in IoT. IEEE Access. 2020;8:18207-18. doi: 10.1109/ACCESS.2020.2968492.
[37] Lu N, Zhang Y, Shi W, Kumari S, Choo KKR. A secure and scalable data integrity auditing scheme based on Hyperledger Fabric. Comput Secur. 2020;92:101741. doi: 10.1016/j.cose.2020.101741.
[38] Li Y, Shen J, Ji S, Lai YH. Blockchain-based data integrity verification scheme in AIoT cloud-edge computing environment. IEEE Trans Eng Manag. 2024;71:12556-65. doi: 10.1109/TEM.2023.3262678.
[39] Zhu J, Zhang GY. Blockchain-based smart and secure manufacturing systems. Internet Technol Lett. 2025;8(4):e589. doi: 10.1002/itl2.589.
[40] Alqazzaz A. SecuFL-IoT: an adaptive privacy-preserving federated learning framework for anomaly detection in smart industrial networks. Sci Rep. 2026;16(1):4107. doi: 10.1038/s41598-025-11883-1.