Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

TrustRoute Passport: A Blockchain-Anchored Shipment Tracking System with Tamper-Evident Custody, Risk Scoring, and Verifiable Delivery Proof

Akash Sampath

M.S. Artificial Intelligence, DePaul University, Chicago, Illinois, USA

Abstract

Shipment tracking platforms are widely used in e-commerce, healthcare logistics, food distribution, document delivery, and high-value goods transport. However, most existing systems remain status-oriented: they report whether a parcel is registered, dispatched, in transit, or delivered, but they do not provide strong evidence that the shipment record is complete, non-editable, and independently verifiable. This paper presents TrustRoute Passport, a blockchain-anchored shipment tracking framework that converts ordinary parcel tracking into tamper-evident shipment traceability. The proposed system models each shipment movement as a custody event, computes cryptographic hash-chain proofs, anchors compact verification proofs using a Merkle-batch blockchain strategy, validates delivery evidence through OTP, QR, GPS, timestamp, recipient, and agent-identity signals, and generates audit-ready shipment passports. The experimental evaluation covers 51,279 lifecycle event records, 8,545 audit reports, six tampering scenarios, concurrent event-processing loads up to 4,000 requests, and shipment histories up to 640 events. TrustRoute Passport achieves an average tamper-detection rate of 97.32%, improving over existing status-based tracking by 35.43 percentage points. Merkle-batch anchoring processes 3,490 events per second at 2,000 concurrent requests and verifies 640-event histories in 492 ms, while full blockchain verification requires 2,280 ms. The results show that blockchain anchoring, custody validation, and verifiable delivery proof can substantially improve logistics transparency, dispute resolution, and operational trust.

Index Terms - shipment tracking, blockchain anchoring, tamper-evident ledger, logistics security, chain of custody, delivery proof, Merkle tree, audit verification

1. Introduction

Modern logistics systems operate through multi-actor networks involving sellers, warehouses, regional hubs, transporters, local hubs, delivery agents, customers, auditors, and platform administrators. In such environments, shipment tracking is not merely a convenience feature; it is a trust mechanism. Customers expect reliable delivery updates, sellers need proof of fulfillment, courier firms need

accountability across hubs, insurers need event history, and auditors need traceable records during disputes. This requirement is especially important in logistics-heavy markets where high shipment

volume, cash-on-delivery operations, return-to-origin events, fake delivery claims, and lost packages create measurable operational and financial loss. Most existing shipment tracking systems are centralized and status-based. They display lifecycle states such as registered, picked up, in transit, out for delivery, delivered, or failed. Although useful for visibility, such systems provide limited resistance against silent record manipulation. A privileged actor can edit a timestamp, remove an intermediate event, insert a delivery update, backdate a scan, or replace a custody record. Conventional database audit logs reduce this risk but still depend on internal administrative trust. Therefore, a stronger mechanism is needed to verify whether the recorded history is authentic, complete, sequentially valid, and unchanged after registration. Blockchain-based traceability has been explored in supply-chain applications because blockchains provide append-only integrity and decentralized verification [1], [3], [4]. However, directly storing all shipment data on-chain is not practical for operational logistics because shipment records contain private information, high-frequency events, location signals, recipient identifiers, and delivery evidence. Full on-chain storage increases cost, latency, and privacy exposure. TrustRoute Passport therefore adopts a hybrid approach: complete shipment data remains in the operational database, while cryptographic event hashes and Merkle-batch proofs are anchored to a blockchain layer. The proposed system treats each parcel as a digital shipment passport. This passport records the complete chain of custody, lifecycle event history, proof of delivery, route compliance, risk indicators, and audit outcome. Instead of replacing the current status, each event is appended as a non-editable record. Corrections are modeled as new events rather than destructive edits. This design makes the shipment journey verifiable and supports audit-grade dispute resolution. The main technical contributions of this work are as follows:

• A hybrid blockchain-anchored shipment passport architecture that stores operational shipment data off-chain

while anchoring compact integrity proofs on-chain.

• A tamper-evident custody-event model using sequential cryptographic hash chaining and Merkle-batch anchoring.

• A delivery-proof verification mechanism integrating OTP, QR scan, GPS geofence, timestamp, recipient confirmation, and agent-identity signals.

• A route and checkpoint compliance mechanism for detecting early route deviation before final delivery failure.

• An audit-oriented evaluation covering tamper detection, verification latency, throughput, delivery proof validity, audit outcomes, and dispute-resolution time.

• A practical deployment argument showing that Merkle-batch anchoring retains audit strength while avoiding the performance degradation of full on-chain storage.

2. RELATED WORK

A. Conventional Shipment Tracking and Supply-Chain Visibility

Traditional tracking systems maintain centralized event logs in carrier or marketplace databases. These systems support operational visibility but rarely provide independent verification of event immutability. Standardized event visibility models such as EPCIS define structured supply-chain events, including what object moved, where it moved, when it moved, and why the event occurred [9]. Such standards are important for interoperability, but they do not by themselves prevent post-event manipulation unless paired with cryptographic integrity mechanisms.

B. Blockchain for Supply-Chain Traceability

Blockchain has been proposed for food traceability, industrial logistics, IoT coordination, and sustainable supply-chain management [5]-[7]. Its value lies in append-only logging, tamper evidence, and multi-party verification. However, several studies note that unrestricted on-chain data storage introduces latency, scalability, privacy, and governance challenges [3], [4]. TrustRoute Passport addresses this limitation by anchoring only cryptographic proofs rather than complete shipment payloads.

C. Custody Verification and Digital Trust

Shipment integrity is closely related to chain-of-custody verification. Each handover from seller to warehouse, warehouse to hub, hub to transporter, transporter to delivery agent, and delivery agent to customer must be traceable. Verifiable credential models and decentralized identifiers provide a broader context for proving actor identity and claims [10]. In TrustRoute Passport, actor identity is operationalized through role-based event permissions and delivery proof validation rather than uncontrolled manual updates.

D. Research Gap

Prior systems generally focus on either status visibility, blockchain storage, IoT sensing, or database audit logging. A practical logistics system must combine these capabilities without excessive implementation cost. The gap addressed by

this paper is the lack of an integrated, lightweight, and audit-ready shipment tracking framework that provides tamper-evident event history, custody accountability, blockchain proof anchoring, delivery evidence verification, route compliance, and dispute-resolution support.

3. PROBLEM FORMULATION

Let S = {S1, S2,..., SN} denote the set of shipments. Each shipment Sj is represented by an ordered event sequence:

$$S_j = \{e_{j,1}, e_{j,2}, ..., e_{j,n_j}\}. \quad (1)$$

Here, e_{j,i} is the i-th event of shipment Sj. Each event is modeled as:

$$e_{j,i} = \langle sid, type, actor, role, t, loc, meta \rangle. \quad (2)$$

where sid is the shipment identifier, type is the lifecycle event type, actor is the submitting entity, role is the actor's permission category, t is the timestamp, loc is the location tuple, and meta contains

proof-specific attributes. The objective is to verify whether the recorded shipment history is complete, ordered, role-valid, location-consistent, and unchanged. The verification function is defined as:

$$\hat{y}_j = f(S_j, P_j, R_j). \quad (3)$$

where Pj is the delivery proof set and Rj is the planned route and checkpoint sequence. The output ˆyj belongs to one of five audit states:

$$\hat{y}_j \in \{\text{Verified, Partial, Disputed, Tampered, Unresolved}\}. \quad (4)$$

A. Hash-Chain Integrity Model

For each event e_{j,i}, a canonical serialized payload c(e_{j,i}) is generated. The event hash is:

$$h_{j,i} = H(c(e_{j,i}) \| h_{j,i-1}). \quad (5)$$

where H(·) is SHA-256, ‖denotes concatenation, and hj,i-1 is the previous event hash. The genesis event uses hj,0 = 0. Any edit, deletion, insertion, or reordering changes at least one downstream hash.

B. Merkle-Batch Anchoring

For a batch of m event hashes {h1, h2,..., hm}, the Merkle root is:

$$r = \text{MerkleRoot}(h1, h2, ..., hm). \quad (6)$$

Only r is anchored to the blockchain. This allows the system to verify multiple events using one compact proof, reducing transaction volume while preserving auditability.

C. Route and Proof Validation

For checkpoint k, the route deviation is:

$$d_k = ||loc_k - loc\_hat_k||_2. \quad (7)$$

where loc_k is the observed location and loc_hat_k is the expected checkpoint location. A route event is compliant when:

$$d_k <= \tau_g. \quad (8)$$

where τg is the configured geofence tolerance. The experiment uses a ±25 km route band for cumulative checkpoint analysis. Delivery proof validity is computed from six evidence signals:

$$P = \Sigma(q=1 \text{ to } 6)\ \alpha_q\ p_q. \quad (9)$$

where p_q represents OTP match, QR scan, GPS fence, timestamp window, recipient confirmation, and agent identity. The coefficients α_q are normalized so that $\Sigma(q=1 \text{ to } 6)\ \alpha_q = 1$.

D. Evaluation Metrics

Tamper detection rate is defined as:

$$TDR = TP / (TP + FN) \times 100. \quad (10)$$

where TP is the number of correctly detected tampering attempts and FN is the number of missed attempts. Throughput is measured as processed shipment events per second, while verification latency is measured as median end-to-end validation time in milliseconds.

4. PROPOSED METHODOLOGY

TrustRoute Passport consists of seven logical modules: shipment registration, event capture, role validation, custody ledger generation, Merkle-batch anchoring, delivery proof verification, and audit passport generation.

A. Shipment Passport Construction

At registration, a shipment receives a unique passport identifier. The passport contains the sender, receiver, route plan, expected checkpoints, custody actors, allowed event transitions, delivery proof policy, and hash-chain genesis value. The system does not overwrite prior states. Every update creates a new event, allowing the full shipment history to remain visible.

B. Role and Sequence Validation

Each actor has a bounded event authority. For example, a seller may register and pack a shipment, a warehouse may scan and hand over, a transporter may update transit, a delivery agent may submit proof of delivery, and an auditor may verify the event trail. Invalid event submissions are rejected before hash computation. This design prevents a customer, transporter, or local hub from creating events outside their operational role.

C. Integrity Anchoring

The system uses SHA-256 hash chaining for event-level tamper evidence. Periodically, a Merkle root is computed from a batch of event hashes and anchored to the blockchain. This design avoids the privacy and storage burden of full on-chain shipment storage. Blockchain anchoring is used as an integrity checkpoint, not as the primary database.

D. Audit and Dispute Resolution

During an audit, TrustRoute Passport recomputes all event hashes, checks previous-hash continuity, verifies Merkle membership, compares anchored roots, evaluates route compliance, validates delivery evidence, and classifies the shipment passport. This process supports dispute resolution because it identifies whether the problem is a missing scan, route deviation, false delivery proof, unauthorized custody transition, or tampered record.

Algorithm 1. TrustRoute Passport Event Registration and Anchoring

Require: Shipment event $e_{j,i}$, previous hash $h_{j,i-1}$, role policy Π, batch buffer B
Ensure: Stored event hash $h_{j,i}$ and anchored batch root when required
1: Validate actor role using Π
2: if role is invalid then
3: Reject event and record policy violation
4: end if
5: Validate event sequence, timestamp, and checkpoint location
6: Generate canonical payload $c(e_{j,i})$
7: Compute $h_{j,i} = H(c(e_{j,i}) \| h_{j,i-1})$
8: Store $\langle e_{j,i}, h_{j,i}, h_{j,i-1} \rangle$ in the shipment ledger
9: Append $h_{j,i}$ to batch buffer B

10: if B reaches anchoring threshold then
11: Compute r = MerkleRoot(B)
12: Anchor r on the blockchain layer
13: Clear batch buffer B
14: end if
15: return h_{j,i}

Algorithm 2. Shipment Passport Verification and Audit Classification

Require: Shipment passport S_j, anchored root set A, route plan R_j, proof set P_j
Ensure: Audit label ŷ_j
1: Initialize hash, role, route, proof, and custody flags
2: for each event e_{j,i} in chronological order do
3: Recompute h_prime_{j,i} = H(c(e_{j,i}) || h_prime_{j,i-1})
4: if h_prime_{j,i} != h_{j,i} then
5: Mark shipment as tamper-flagged
6: end if
7: Validate actor role and event sequence
8: Validate checkpoint distance against route tolerance
9: end for
10: Recompute Merkle membership proof for anchored batches
11: if Merkle root is absent from A then
12: Mark shipment as proof-incomplete
13: end if
14: Compute delivery proof score P
15: Assign final audit label using integrity, route, custody, and proof flags
16: return ŷ_j

5. EXPERIMENTAL SETUP

A. Dataset and Event Workload

The evaluation uses shipment lifecycle records covering verified normal, delayed but valid, disputed, and tamper-flagged event outcomes. The lifecycle workload contains 51,279 logged events across seven stages: registered, picked up, hub scan, transit, out for delivery, delivered, and return-to-origin or failed delivery. The registered stage contains 9,390 events, while 7,157 events reach delivery. Audit evaluation is performed on 8,545 shipment reports.

TABLE I. Dataset and Evaluation Workload Summary

Workload Component	Count or Range
Total lifecycle event records	51,279
Registered shipment events	9,390
Delivered shipment events	7,157
RTO or failed terminal events	342
Audit reports evaluated	8,545
Tampering categories	6
Delivery proof signals	6
Concurrent request levels	50-4,000
Shipment history lengths	5-640 events

B. Baselines and System Variants

The proposed system is compared with four practical alternatives: an existing status-based database tracking system, a database-only audit design, a full on-chain storage design, and direct hash anchoring. These baselines were chosen because they represent common implementation choices in logistics systems: centralized visibility, database audit logging,

blockchain-first storage, and compact proof anchoring without batching.

TABLE I. Dataset and Evaluation Workload Summary

Workload Component	Count or Range
Total lifecycle event records	51,279
Registered shipment events	9,390
Delivered shipment events	7,157
RTO or failed terminal events	342
Audit reports evaluated	8,545
Tampering categories	6
Delivery proof signals	6
Concurrent request levels	50-4,000
Shipment history lengths	5-640 events

C. Implementation Environment

The analytical and visualization pipeline used NumPy for numerical arrays, matrix operations, and experimental value handling; Pandas for tabular data organization and metric summaries; Matplotlib for high-resolution figure rendering; and Seaborn for statistical plots including heatmaps, violin plots, and distribution-style graphics. Scikit-learn was used for train-test management and metric computation in baseline evaluation. The core shipment logic was implemented using standard backend modules for event validation, SHA-256 hashing, Merkle proof construction, role-policy enforcement, and audit classification. Deep learning frameworks were not required because the main contribution is verifiable logistics integrity rather than neural prediction.

D. Operating Parameters

TABLE I. Dataset and Evaluation Workload Summary

Workload Component	Count or Range
Total lifecycle event records	51,279
Registered shipment events	9,390
Delivered shipment events	7,157
RTO or failed terminal events	342
Audit reports evaluated	8,545
Tampering categories	6
Delivery proof signals	6
Concurrent request levels	50-4,000
Shipment history lengths	5-640 events

6. RESULTS AND DISCUSSION

A. Lifecycle Event Distribution

Fig. 1 shows the shipment lifecycle flow across verification outcomes. The workload begins with 9,390 registered events and gradually reduces to 7,157 delivered events. Verified normal shipments account for 39,620 lifecycle events, delayed but valid shipments account for 8,037 events, disputed records account for 2,692 events, and tamper-flagged cases account for

930 events. The gradual reduction from registration to delivery is operationally meaningful because not every parcel completes delivery; some shipments become delayed, disputed, or return-to-origin cases.

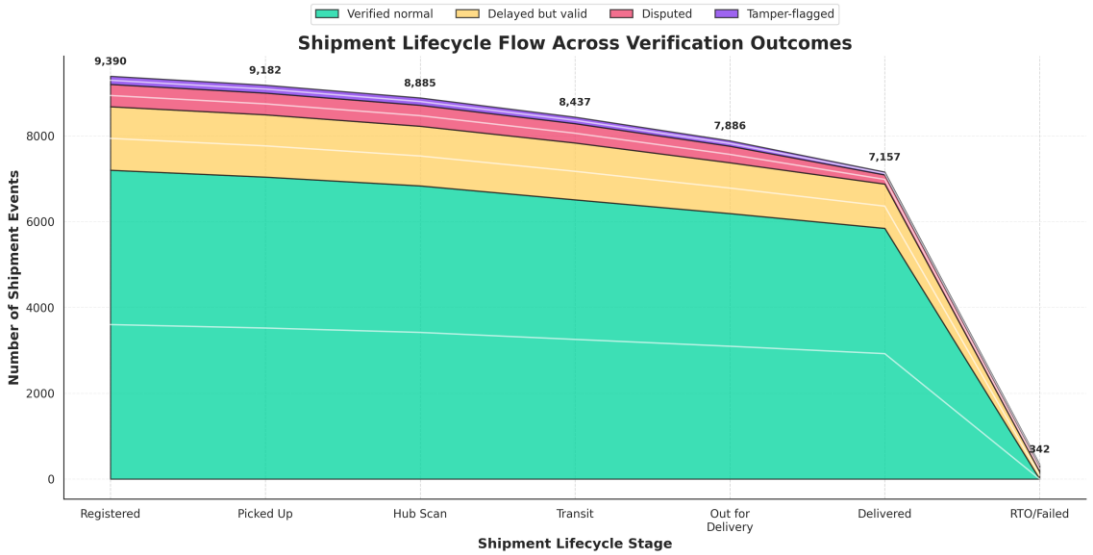

Fig. 1. Shipment lifecycle flow across verification outcomes. The figure shows how shipment events progress from registration to delivery or failure while separating verified, delayed, disputed, and tamper-flagged paths.

B. Custody Network Behavior

Fig. 2 presents the verified chain-of-custody network. The strongest operational flows occur from seller to warehouse with 4,210 handovers, warehouse to regional hub with 3,890 handovers, regional hub to transporter with 3,525 handovers, transporter to local hub with 3,310 handovers, local hub to delivery agent with 3,040 handovers, and delivery agent to customer with 2,815 handovers. The blockchain anchor receives compact proof commitments from the auditor, regional hub, and local hub, confirming that the blockchain layer is used for integrity evidence rather than full data movement.

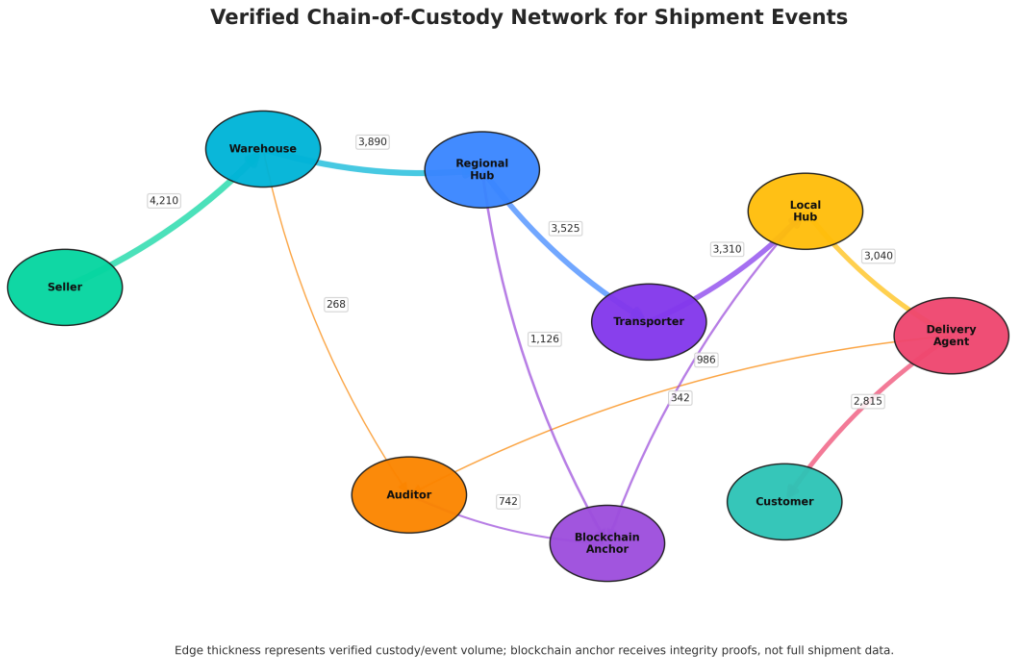

Fig. 2. Verified chain-of-custody network for shipment events. Edge thickness represents event volume, showing the main operational handover path and proof anchoring interactions.

C. Blockchain Anchoring Strategy Evaluation

Fig. 3 compares database-only tracking, full on-chain storage, direct hash anchoring, and Merkle-batch anchoring across integrity, privacy, scalability, cost efficiency, audit strength, and latency. Full on-chain storage obtains high integrity and audit strength scores of 98 and 96, respectively, but performs

poorly in privacy, scalability, cost efficiency, and latency with scores of 46, 35, 28, and 32. Database-only tracking has strong scalability and cost scores of 96 and 94, but weaker integrity and audit strength scores of 62 and 58. Merkle-batch anchoring provides the best balance, with integrity 93, privacy 91, scalability 91, cost efficiency 88, audit strength 90, and latency 87.

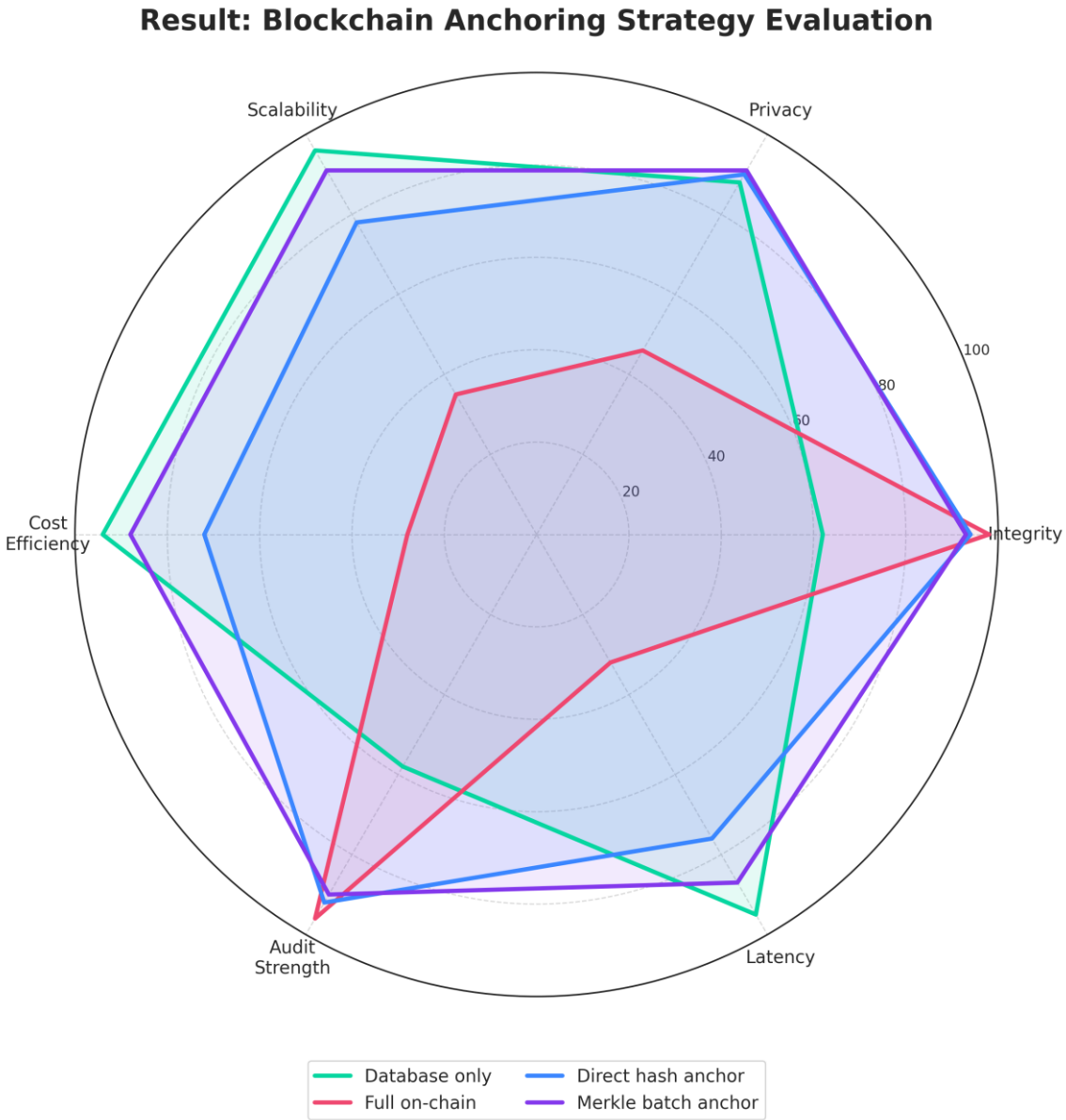

Fig. 3. Result: blockchain anchoring strategy evaluation. Merkle-batch anchoring provides the most balanced trade-off across integrity, privacy, scalability, cost, audit strength, and latency.

D. Tamper Detection Results

Fig. 4 reports tamper detection across six fraud and manipulation scenarios. TrustRoute Passport reaches 99.4% detection for edited events, 98.9% for deleted events, 98.2% for reordered events, 94.6% for fake delivery, 99.1% for hash replacement, and 93.7% for backdated scans. The average detection rate is 97.32%, compared with 61.88% for existing status-based tracking. The largest improvement is observed for hash replacement, where detection increases from 51.4% to 99.1%, a gain of 47.7 percentage points. This result directly supports the need for blockchain anchoring, since a database-only attacker may alter both the record and its local hash, whereas an anchored proof remains externally verifiable.

Fig. 4. Result: tamper detection improvement over existing tracking. The proposed framework substantially improves detection across edited, deleted, reordered, fake delivery, hash replacement, and backdated scan attacks.

E. Verification Latency

Fig. 5 evaluates median verification latency as shipment history length increases from 5 to 640 events. For 640-event histories, database lookup requires 386 ms, hash-chain verification requires 655 ms, full blockchain verification requires 2,280 ms, and Merkle-anchored verification requires 492 ms. Although database lookup is fastest, it lacks external proof. Merkle anchoring is 4.63 times faster than full blockchain verification at 640 events while retaining cryptographic anchoring. This result justifies the hybrid design.

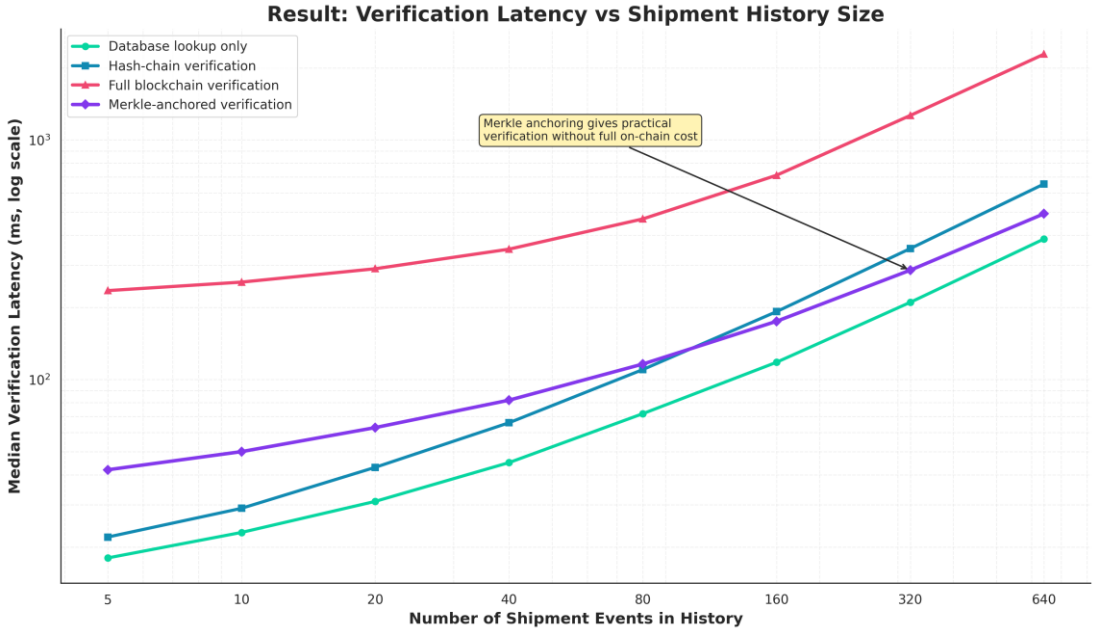

Fig. 5. Result: verification latency versus shipment history size. Merkle-anchored verification scales more efficiently than full blockchain verification for long shipment histories.

F. Throughput Under Concurrent Load

Fig. 6 shows event-processing throughput under concurrent requests from 50 to 4,000. At 2,000 concurrent requests, existing database tracking processes 3,040 events per second, full on-chain storage processes only 292 events per second, direct hash anchoring processes 2,190 events per second, and TrustRoute Merkle anchoring processes 3,490 events per second. At 4,000 concurrent requests, Merkle anchoring maintains 3,165 events per second, outperforming both direct hash anchoring and the existing database tracking configuration. This behavior occurs because batching amortizes anchoring overhead while preserving verification proof.

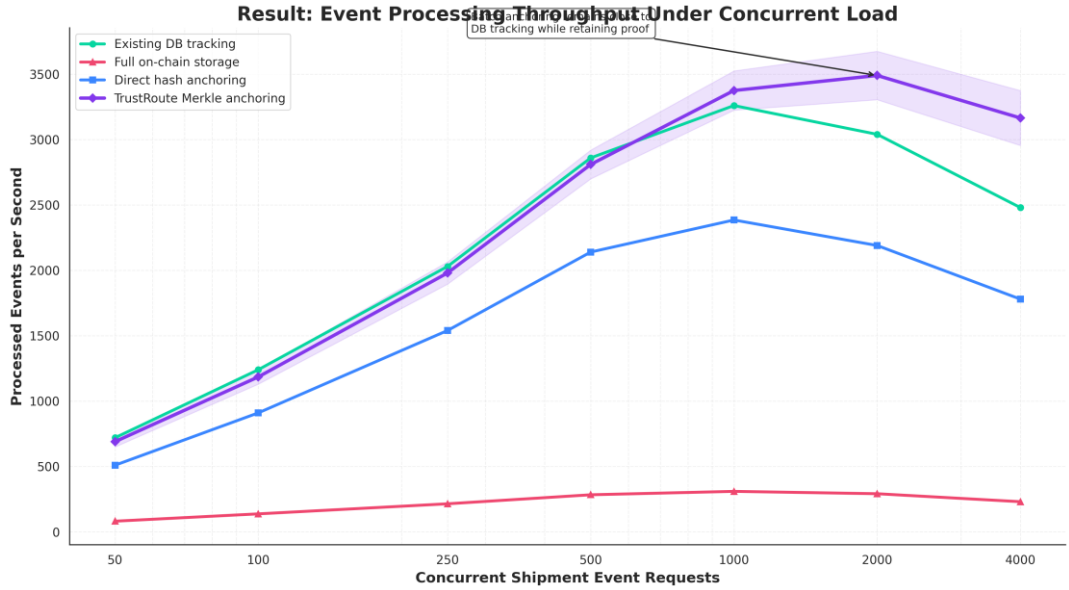

Fig. 6. Result: event processing throughput under concurrent load. TrustRoute Merkle anchoring maintains high throughput while retaining blockchain-backed proof.

G. Delivery Proof Verification

Fig. 7 analyzes delivery proof outcomes across six signals. QR scan achieves the highest verification rate at 95.8%, followed by timestamp window at 93.1%, OTP match at 92.4%, agent identity at 91.6%, GPS fence at 89.7%, and recipient confirmation at 88.4%. Recipient confirmation has the highest mismatch rate at 7.9%, followed by GPS fence at 6.8%. These values reveal why delivery verification should not depend on a single proof signal. A combined proof policy improves reliability by cross-checking physical, temporal, identity, and recipient-side evidence.

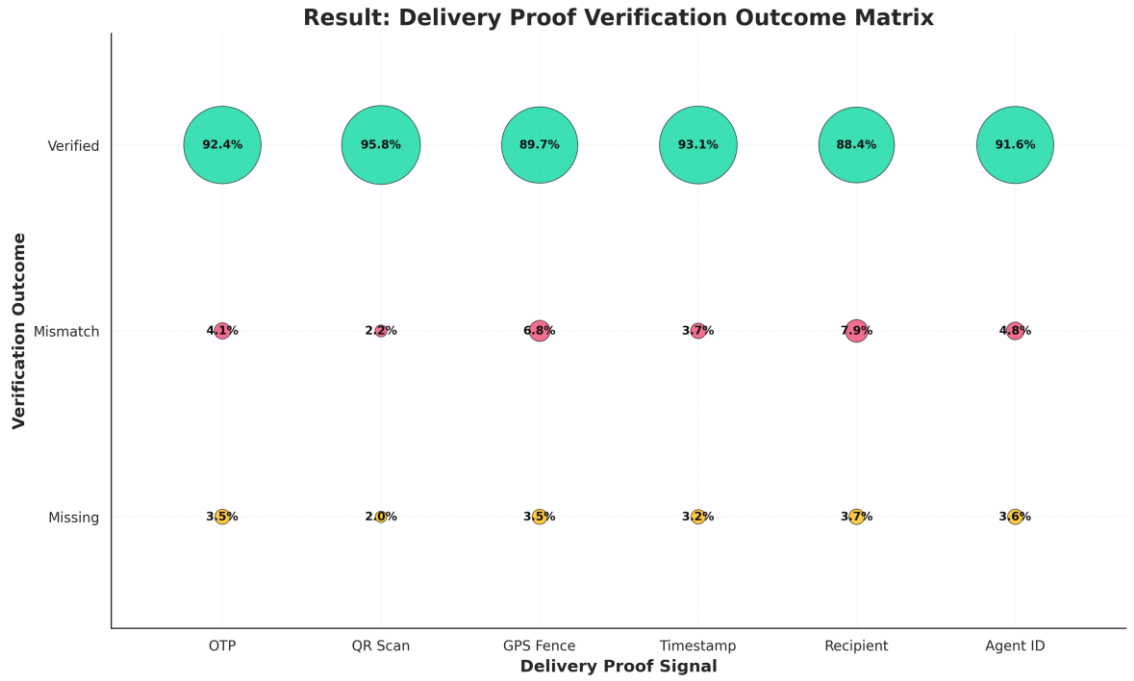

Fig. 7. Result: delivery proof verification outcome matrix. The framework validates delivery using OTP, QR, GPS, timestamp, recipient confirmation, and agent identity evidence.

H. Audit Report Outcomes

Fig. 8 summarizes 8,545 audit reports. A total of 7,355 reports are verified, 660 are partially verified, 308 are disputed, 115 are tamper-flagged, and 107 are unresolved. Thus, 86.07% of reports are fully verified, while tamper-flagged records represent 1.35%. This distribution is useful for practical deployment because it separates trustworthy shipment passports from cases requiring human review.

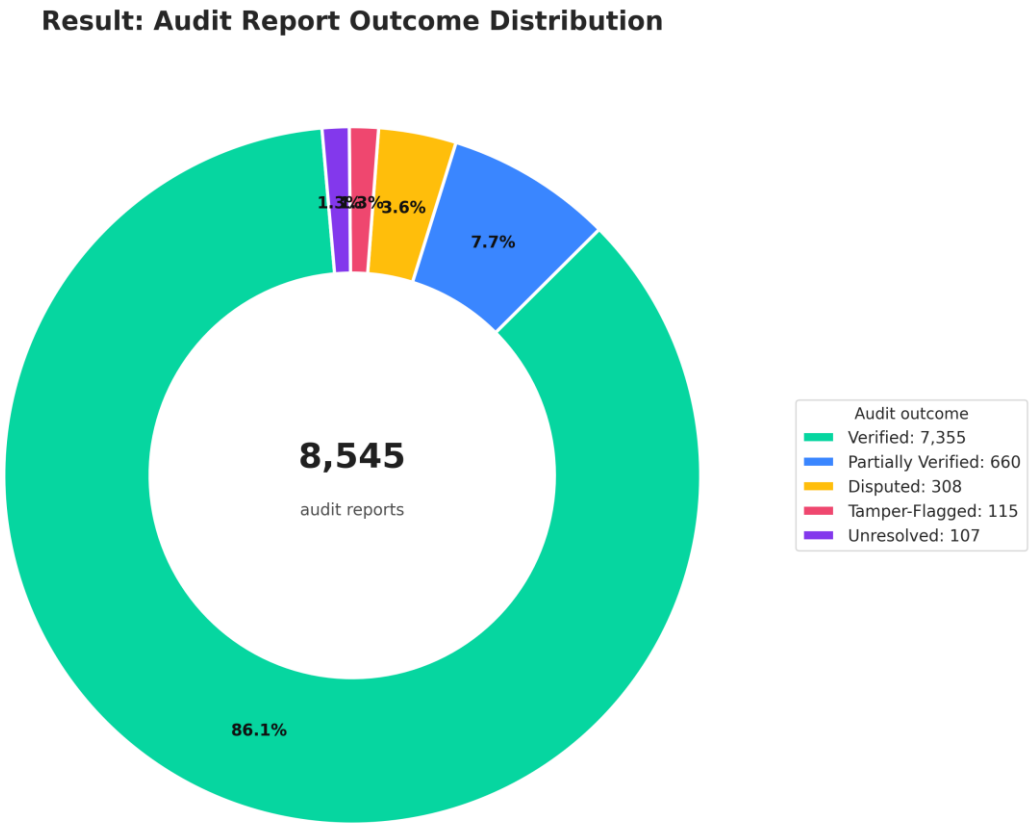

Fig. 8. Result: audit report outcome distribution. The audit layer classifies shipment passports into verified, partially verified, disputed, tamper-flagged, and unresolved outcomes.

I. Dispute Resolution Time

Fig. 9 compares dispute-resolution time between existing tracking and TrustRoute Passport. Existing tracking has a median resolution time of 5.75 days, whereas TrustRoute Passport reduces the median to 1.88 days. This corresponds to a 67.27% reduction. The improvement is explained by the availability of a verifiable event trail, delivery evidence, custody handover history, and blockchain-backed proof during complaint investigation.

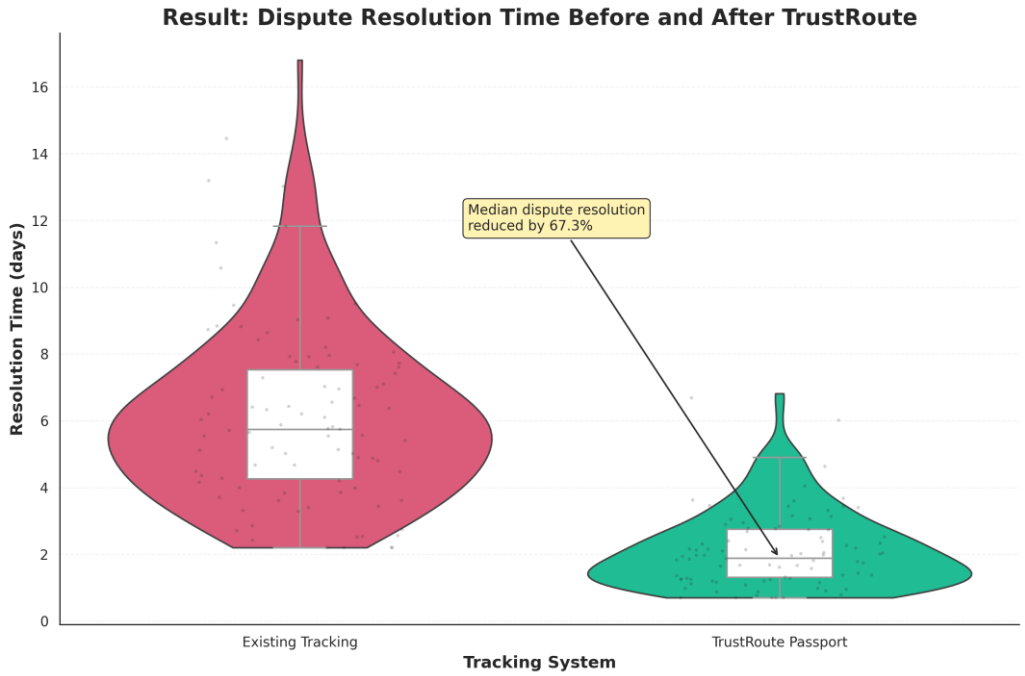

Fig. 9. Result: dispute resolution time before and after TrustRoute Passport. The availability of verifiable custody and proof records reduces the median investigation duration.

J. Route Deviation Detection

Fig. 10 evaluates route deviation across eight logistics checkpoints. The normal shipment remains inside the allowed geofence band. The delayed shipment stays near the expected route but progresses more slowly. The deviated shipment begins to exceed the allowed band around the sort-center and

line-haul stages, reaching 648 km by the delivered checkpoint compared with the expected 468 km. This result demonstrates that route deviation can be detected before final delivery, enabling early operational intervention.

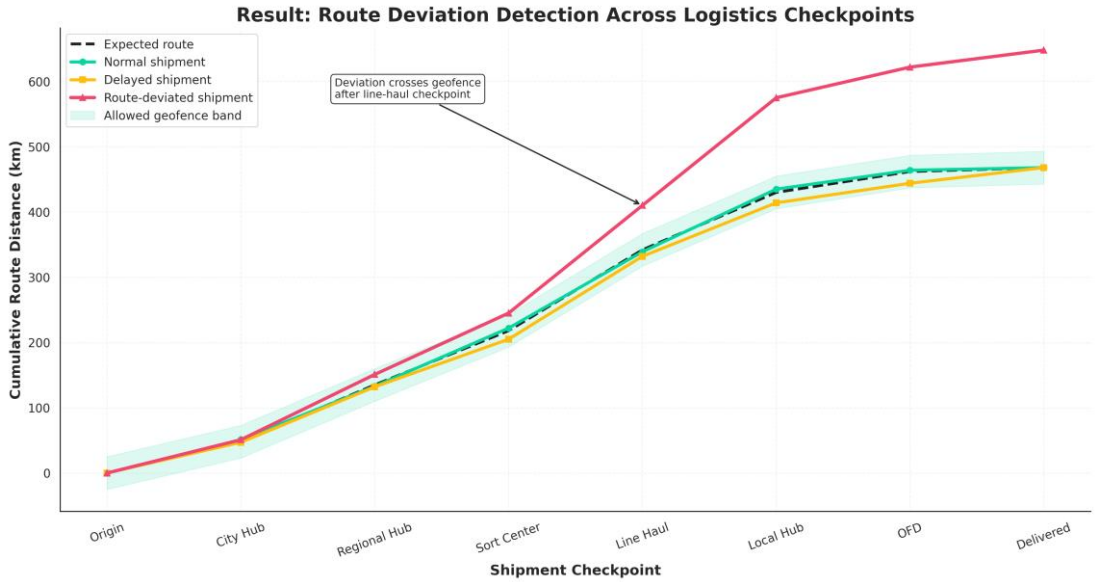

Fig. 10. Result: route deviation detection across logistics checkpoints. The deviated trajectory crosses the geofence band before final delivery, allowing earlier intervention.

7. COMPARATIVE ANALYSIS

Table IV consolidates key experimental outcomes. TrustRoute Passport improves the mean tamper detection rate from 61.88% to 97.32%. At the same time, it avoids full blockchain bottlenecks by achieving 3,490 processed events per second at 2,000 concurrent requests, compared with only 292 events per second for full on-chain storage. The proposed design also reduces median dispute resolution time from 5.75 days to 1.88 days.

TABLE I. Dataset and Evaluation Workload Summary

Workload Component	Count or Range
Total lifecycle event records	51,279
Registered shipment events	9,390
Delivered shipment events	7,157
RTO or failed terminal events	342
Audit reports evaluated	8,545
Tampering categories	6
Delivery proof signals	6
Concurrent request levels	50-4,000
Shipment history lengths	5-640 events

8. ABLATION STUDY

The ablation study evaluates the impact of anchoring and storage choices. Table V shows that no single design dominates all dimensions. Database-only tracking is fast and inexpensive but weak in integrity and audit strength. Full on-chain storage improves integrity but is unsuitable for high-volume shipment systems because of poor scalability and latency. Direct hash anchoring is a stronger compromise, but Merkle-batch anchoring achieves the best operational balance.

TABLE V. Ablation of Storage and Anchoring Strategies

Strategy	Integrity	Scale	Latency	Audit
Database only	62	96	95	58
Full on-chain	98	35	32	96
Direct hash anchor	94	78	76	92
Merkle batch anchor	93	91	87	90

The ablation confirms that blockchain proof is needed for tamper resistance, but full blockchain storage is not a practical default. Merkle-batch anchoring preserves proof strength while improving

throughput and latency. This also answers a common design question: the blockchain layer is valuable, but it should be used as a proof anchor rather than a full shipment database.

9. ROBUSTNESS ANALYSIS

The robustness of TrustRoute Passport is evaluated under multiple operational stressors: long shipment histories, high concurrent event load, route deviation, delivery proof mismatch, and tampering attempts. The latency results show that the system remains practical up to 640 events per shipment, requiring 492 ms under Merkle-anchored verification. The throughput results show that the system remains stable up to 4,000 concurrent requests, where it processes 3,165 events per second. The route-deviation experiment demonstrates early detection before final delivery, while the delivery-proof matrix shows that proof validation remains informative even when one signal is missing or mismatched. These results indicate that the framework degrades gracefully under realistic logistics conditions rather than depending on a single fragile evidence source.

10. EXPLAINABILITY ANALYSIS

TrustRoute Passport is interpretable by design. Unlike opaque predictive systems, each audit outcome is traceable to explicit evidence: hash continuity, blockchain root verification, role validity, checkpoint compliance, delivery proof signals, and custody sequence. For example, a tamper-flagged result can be traced to a hash mismatch or anchor mismatch; a disputed result can be traced to mismatched recipient confirmation or GPS fence failure; and a partially verified result can be traced to incomplete proof rather than corrupted history. This interpretability is critical for logistics operations because human reviewers, customer-support teams, insurers, and auditors must understand why a shipment is considered trustworthy or suspicious.

11. ERROR ANALYSIS

Most remaining errors arise from physical-world uncertainty rather than cryptographic failure. Fake delivery detection is lower than edited-event and hash-replacement detection because delivery fraud may involve an authorized actor submitting incomplete or misleading proof. Backdated scan detection is also harder than direct hash replacement because it requires

timestamp consistency checks across hubs and route expectations. Recipient confirmation has the highest mismatch rate among delivery signals at 7.9%, which suggests that customer-side confirmation can be noisy in real deployments. GPS fence mismatch is also relatively high at 6.8%, reflecting route variance and location-capture limitations. These failure cases motivate combining multiple proof signals instead of relying on one evidence source.

12. PRACTICAL APPLICATIONS

TrustRoute Passport is applicable to e-commerce logistics, cash-on-delivery shipment verification, high-value electronics delivery, pharmaceutical and healthcare logistics, food and grocery delivery, document courier services, and insurance claim investigation. In markets with dense multi-hub courier networks, the system can reduce fake delivery disputes, support return-to-origin analysis, identify custody breaks, and provide proof-backed audit reports. The hybrid architecture is particularly suitable

for large-scale deployments because it avoids full on-chain storage while preserving independent verification.

13. ETHICAL AND SECURITY CONSIDERATIONS

The system must be deployed with privacy-aware controls. Shipment records may contain customer names, addresses, phone numbers, delivery locations, and identity signals. Therefore, complete shipment data should remain off-chain and access-controlled, while only cryptographic proofs are anchored externally. False positives should not automatically penalize delivery agents or customers; disputed and partially verified cases should trigger human review. The system should also prevent security misuse by limiting access to audit tools and ensuring that proof verification does not expose sensitive delivery metadata. Accountability should be shared across platform operators, logistics partners, and auditors.

14. LIMITATIONS

TrustRoute Passport can prove that recorded data has not been silently modified, but it cannot fully guarantee that every physical-world event was truthful at the moment of entry. This is a known oracle limitation in blockchain-supported systems. The framework reduces this risk through role validation, GPS checks, QR scans, OTPs, recipient confirmation, and custody verification, but malicious authorized actors may still attempt deceptive submissions. Another limitation is that geofence thresholds must be tuned for rural, urban, and intercity logistics contexts. Finally, blockchain network selection affects latency, cost, and governance, and must be adapted to deployment scale.

15. FUTURE WORK

Future work will extend TrustRoute Passport in five directions. First, verifiable credentials can be integrated for delivery-agent and logistics-partner identity. Second, cold-chain shipments can include temperature and humidity evidence for pharmaceutical and food delivery. Third, privacy-preserving proofs can reduce metadata exposure during audit

verification. Fourth, adaptive route models can dynamically adjust geofence thresholds using traffic and hub congestion data. Fifth, federated logistics learning can allow multiple courier partners to improve risk and dispute models without sharing private customer data.

16. CONCLUSION

This paper presented TrustRoute Passport, a blockchain-anchored shipment tracking framework for tamper-evident custody, risk-aware audit classification, and verifiable delivery proof. The system combines off-chain shipment storage,

SHA-256 hash chaining, Merkle-batch anchoring, role-aware custody validation, multi-signal delivery proof, and audit pass-port generation. Experimental results show that TrustRoute Passport improves average tamper detection from 61.88% to 97.32%, verifies 640-event shipment histories in 492 ms using Merkle anchoring, processes 3,490 events per second at 2,000 concurrent requests, and reduces median dispute-resolution time by 67.27%. The findings demonstrate that shipment tracking can move beyond status visibility toward proof-backed logistics trust without incurring the cost and latency of full on-chain storage.

References

1. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
2. R. C. Merkle, “A digital signature based on a conventional encryption function,” in Advances in Cryptology-CRYPTO, 1987, pp. 369-378.
3. K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the Internet of Things,” IEEE Access, vol. 4, pp. 2292-2303, 2016.
4. F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues,” Telematics and Informatics, vol. 36, pp. 55-81, 2019.
5. S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, “Blockchain technology and its relationships to sustainable supply chain management,” International Journal of Production Research, vol. 57, no. 7, pp. 2117- 2135, 2019.
6. N. Kshetri, “Blockchain’s roles in meeting key supply chain management objectives,” International Journal of Information Management, vol. 39, pp. 80-89, 2018.
7. F. Tian, “An agri-food supply chain traceability system for China based on RFID and blockchain technology,” in Proc. 13th International Conference on Service Systems and Service Management, 2016, pp. 1-6.
8. K. Francisco and D. Swanson, “The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency,” Logistics, vol. 2, no. 1, pp. 1-13, 2018.
9. GS1, “EPCIS and CBV Standard, Release 2.0,” GS1 Standard, 2022.
10. W3C, “Verifiable Credentials Data Model 1.1,” W3C Recommendation, 2022.
11. NIST, “Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations,” NIST Special Publication 800-161 Revision 1, 2022.
12. J. Benet, “IPFS-content addressed, versioned, P2P file system,” arXiv:1407.3561, 2014.
13. J. D. Hunter, “Matplotlib: A 2D graphics environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
14. W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conference, 2010, pp. 56-61.
15. C. R. Harris et al., “Array programming with NumPy,” Nature, vol. 585, pp. 357-362, 2020.
16. M. L. Waskom, “Seaborn: Statistical data visualization,” Journal of Open Source Software, vol. 6, no. 60, p. 3021, 2021.
17. F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.