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

Secure Data Transmission and Privacy-Preserving Path Control for Distributed IoT Based on SDN and Mathematical Optimization

Jing Su[1], and Zilin Guo[1,*]

[1]Dalian University of Science and Technology, Dalian City, 116036, Liaoning Province, China

Abstract

INTRODUCTION: Distributed Internet of Things (IoT) communication must jointly satisfy low-latency forwarding, data confidentiality, integrity, access control, and lifecycle protection. Performance-oriented routing may expose sensitive traffic to malicious relays, high-risk links, unauthorized domains, and tampered flow entries.
OBJECTIVES: This study reformulates route optimization as a secure data-transmission and privacy-preserving path-control problem.
METHODS: A three-layer SDN framework combines robust node-reputation and link-risk scoring, security-label matching, attribute-based access control, flow isolation, AES-256-GCM authenticated encryption with ephemeral keys, signed rules, and a diversity-aware improved genetic algorithm. A composite objective integrates delay, loss, utilization, route risk, privacy exposure, and hard policy constraints. A reproducible Python/NetworkX discrete-event simulation compares Dijkstra, ECMP, AODV, OLSR, RPL, classical GA, SDN-TE, trust-aware routing, and two proposed variants.
RESULTS: Across eight independent runs, the full scheme achieved 16.68 ± 1.35 ms normal-operation delay and 98.54 ± 0.21% packet delivery. With 20% malicious nodes, it maintained 98.18 ± 0.39% packet delivery, reduced malicious-link exposure from 20.83% for Dijkstra to 6.67%, eliminated observed plaintext leakage of sensitive payloads in the main runs, and reduced attack-response latency from 722.22 ± 23.53 ms to 130.92 ± 6.34 ms. Paired comparisons against Dijkstra were significant for malicious-link exposure, abnormal-path selection, leakage, packet delivery, and response latency ($p < 0.05$).
CONCLUSION: The framework converts security and privacy requirements into enforceable routing, cryptographic, access-control, isolation, auditing, and lifecycle policies, enabling reproducible security-performance co-optimization for distributed IoT.

Keywords: Distributed IoT; secure data transmission; privacy-preserving path control; software-defined networking; threat modeling; trust evaluation; access control; improved genetic algorithm

Received on 30 March 2026, accepted on 16 August 2026, published on 26 August 2026

doi: 10.4108/eetsis.12426

*Corresponding author. Email: guozilin2005@163.com

1. Introduction

Mathematical modeling provides a formal basis for jointly coordinating communication efficiency, data security, and privacy protection in distributed IoT systems [1]. Sensitive telemetry, industrial commands, and cross-domain service data may be exposed through eavesdropping, malicious forwarding nodes, man-in-the-middle interception, unauthorized access, weak tenant isolation, flow-table tampering, or improper key and retention management [2,3]. Conventional shortest-path and performance-oriented routing methods primarily optimize network cost or reachability and do not, by themselves, guarantee that a selected path is appropriate for the sensitivity of the transmitted data [4].

Trust-aware IoT routing protocols such as SecTrust-RPL and metric-based RPL trust schemes use node trust and behavioral evidence to detect or avoid compromised relays while maintaining routing performance [5,6]. Complementary studies address IoT data encryption and transport-layer confidentiality [7], controller-side DDoS threats in SDN [8], and broader IoT authentication and key-security mechanisms [2]. However, route selection alone cannot protect payload content or control how sensitive data are accessed, isolated, audited, rotated, and deleted over their lifecycle. Conversely, encryption and access-control studies often protect data independently of route risk. The resulting separation between network optimization and privacy governance leaves sensitive flows vulnerable to compromised relays and unauthorized cross-domain forwarding.

This study develops an SDN-based secure transmission framework in which node trust, link risk, traffic sensitivity, cryptographic protection, access authorization, and isolation policies are jointly considered during path selection and rule deployment. The controller assigns signed sensitivity labels, selects routes that satisfy risk and access constraints, and activates differentiated protection for sensitive flows. Security scores are calculated primarily from controller-observed evidence and updated using robust aggregation to limit manipulation.

The principal contributions are: (1) a multi-objective secure path-control model that internalizes communication cost, route risk, privacy exposure, and flow sensitivity; (2) a lifecycle protection mechanism integrating authenticated encryption, ephemeral keys, attribute-based access control, tenant/domain isolation, signed rule installation, auditing, key rotation, and retention-aware deletion; (3) robust node-reputation and link-risk models resistant to contaminated reports; and (4) a diversity-aware improved genetic algorithm evaluated against relevant routing baselines using numerical attack metrics, confidence intervals, significance tests, convergence analysis, scalability analysis, and component-level ablation.

2. Related Work

2.1. SDN and Distributed IoT Routing

SDN separates the control and data planes and provides logically centralized topology awareness, programmable flow rules, and policy-driven traffic engineering [9–12]. Distributed IoT processing also increasingly spans cloud and edge resources to improve scalability and responsiveness [19,20,23]. Dijkstra and ECMP remain common performance baselines, while AODV, OLSR, and RPL represent shortest-path, on-demand, proactive, and low-power/low-lossy routing paradigms [4,13–15]. Genetic algorithms have also been applied to network-selection problems [16], while SDN enables programmable route and rule management [9,17]. Most of these approaches, however, optimize performance, topology, or isolated security functions without explicitly coupling route risk with sensitive-data protection.

2.2. Trust-Aware Routing and Data Protection

Trust-aware routing uses historical behavior, packet-forwarding evidence, and trust metrics to reduce reliance on compromised nodes [5,6]. Related studies have examined IoT intrusion detection [18], secure SDN flow-rule installation [17], scalable distributed IoT architectures [19,20], distributed-system testability [21], and blockchain-related security threats [22]. Within EAI Endorsed Transactions on Scalable Information Systems, Nazir et al. developed an OpenFlow/Ryu-based SDN testbed [10], Wang investigated IoT network-layer data-encryption standards [7], and Vaghela et al. reviewed controller-side DDoS risks in SDN [8]. These studies provide important components, but route security, cryptographic content protection, access control, and lifecycle governance are typically evaluated separately.

2.3. Research Gap

The unresolved problem is how to convert flow sensitivity and privacy obligations into measurable path costs and enforceable SDN actions while retaining predictable communication performance. The proposed framework therefore treats security and privacy as endogenous decision variables and links route selection to payload protection, access authorization, isolation, audit, and rule-recovery actions.

3. Threat Model and Security Assumptions

3.1. System Model

The system consists of IoT terminals, edge/aggregation switches, a logically centralized trusted SDN controller, a

trust-and-risk evaluator, authorized application domains, and cryptographic/access-control services. The controller collects topology and QoS features, computes node and link scores, evaluates candidate paths, verifies policy compliance, and installs signed OpenFlow-equivalent rules. Application payloads remain encrypted end to end; the controller uses signed metadata and scores but does not decrypt business data.

3.2. Adversary Capabilities and Scope

An attacker may compromise up to 20% of terminals or forwarding nodes, eavesdrop on adjacent links, selectively drop or modify packets, inject abnormal requests, conduct man-in-the-middle interception, manipulate locally submitted trust observations, and attempt unauthorized flow-table updates. The attacker cannot compromise the trusted controller, forge controller signatures, break AES-256-GCM or ECDH-derived session keys, or alter the authenticated southbound channel. Global passive traffic analysis, physical side-channel extraction, controller compromise, large-scale jamming, and denial of service against the complete control plane are outside the current scope.

3.3. Security Assumptions and Threat–Defense Mapping

The controller and key-management service are isolated and trusted; control messages are authenticated and integrity protected; clocks are sufficiently synchronized for replay protection; keys are generated from a secure random source and rotated periodically; and trust scores are determined primarily from controller-observed evidence. Table 1 maps the evaluated threats to metrics, defenses, and experiments.

Table 1. Threat–defense–validation correspondence

Threat / capability	Security metric	Defense mechanism	Validation
Eavesdropping / MITM	Data-leakage probability	AES-256-GCM, ECDH keys, labels, low-risk route	5–20% malicious-node sweep
Selective drop / local DoS	PDR; delay; response latency	Risk-aware rerouting, path diversity, re-optimization	Attack-period packet delivery and recovery
Malicious relay	Malicious-link hit rate	Robust reputation, link risk, hard avoidance	20% malicious-node main experiment
Flow-table tampering	Attack-response latency	Signed rules, digest verification, clearing and reprogramming	Unauthorized-rule injection
Score manipulation	MLHR and PDR under contamination	Clipping, trimmed aggregation, signed controller evidence	0–20% contaminated observations

4. Secure Data Transmission and Privacy-Preserving Path-Control Model

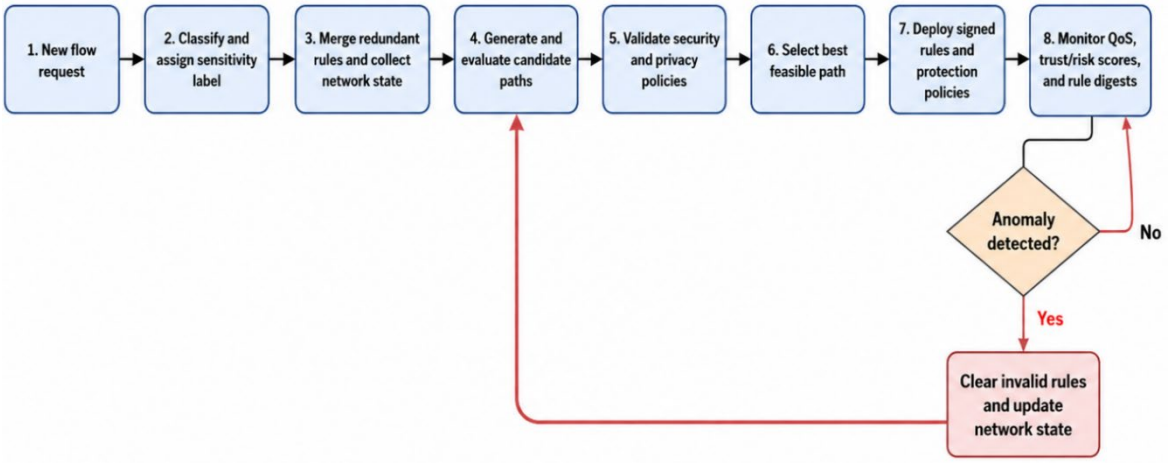

Figure 1. Closed-loop security-aware SDN path-control and rule-execution cycle

Figure 1 illustrates the closed-loop execution cycle of the proposed security-aware SDN path-control framework, covering flow classification, sensitivity labeling, candidate-path evaluation, policy validation, signed rule deployment, continuous monitoring, and anomaly-triggered re-optimization.

4.1. Robust Node Reputation and Link-Risk Evaluation

Node reputation combines the previous score with controller-observed anomaly, selective-drop, authentication/compliance, and signed historical evidence. All component values are normalized to [0,1], and the update is clipped to prevent extreme reports from dominating.

$$T_i(t) = \mathrm{clip}_{[0,1]}\{\beta T_i(t-1) + (1-\beta)E_i(t)\},$$
$$E_i(t) = \omega_1(1-a_i) + \omega_2(1-d_i) + \omega_3 c_i + \omega_4 s_i, \quad \sum_{k=1}^{4}\omega_k = 1 \tag{1}$$

In Eq. (1), a_i is the anomaly rate, d_i is the selective-drop ratio, c_i is authentication and rule-compliance evidence, s_i is signed historical reputation, β controls temporal smoothing, and the nonnegative weights ω_k sum to one. End devices cannot directly assign their own final reputation.

Link risk aggregates normalized abnormal-traffic alarms, packet loss, rule-tampering evidence, encryption weakness, and cross-domain exposure:

$$r_{ij}(t) = \lambda_1 z_{ij}^{\mathrm{anom}} + \lambda_2 z_{ij}^{\mathrm{loss}} + \lambda_3 z_{ij}^{\mathrm{tam}} + \lambda_4\left[1-e_{ij}(t)\right] + \lambda_5 g_{ij}(t),$$
$$\sum_{m=1}^{5}\lambda_m = 1 \tag{2}$$

The controller updates scores every five equivalent seconds. Observations are clipped and aggregated using a 10% trimmed mean; unsigned reports and values outside the median-absolute-deviation threshold are rejected.

The route-level risk penalizes both the average and worst link as well as the least-trusted forwarding node:

$$R(P,t) = \alpha\frac{1}{|P|}\sum_{(i,j)\in P} r_{ij}(t) + (1-\alpha)\max_{(i,j)\in P} r_{ij}(t)$$
$$+\eta\left[1-\min_{v\in P} T_v(t)\right], \quad 0\le\alpha\le 1, \eta\ge 0 \tag{3}$$

4.2 Privacy Exposure and Composite Path Cost

For a flow f, privacy exposure increases with sensitivity S_f, missing or insufficient encryption E_f, unauthorized cross-domain transitions X(P), and low node trust:

$$Q(P) = q_1 S_f \overset{(1)}{\left(1-E_f\right)} + q_2 X(P) + q_3\left[1-\min_{v\in P} T_v(t)\right],$$
$$\sum_{\ell=1}^{3} q_\ell = 1, \quad q_\ell \ge 0 \tag{4}$$

The complete path cost is the weighted sum of normalized route risk, end-to-end delay, packet loss, maximum utilization, and privacy exposure:

$$C(P) = w_1 R(P) + w_2 D(P) + w_3 L(P) + w_4 U(P) + w_5 Q(P), \quad \sum_{k=1}^{5} w_k = 1, \quad w_k \geq 0 \tag{5}$$

Weights are selected according to the signed flow label. Confidential and highly sensitive traffic receive larger w_1 and w_5 values, stricter trust and risk thresholds, mandatory authenticated encryption, and stronger domain-isolation rules. Public traffic uses a more balanced profile to limit overhead.

4.3 Diversity-Aware Improved Genetic Algorithm

A chromosome is a loop-free node sequence from source to destination. The initial population combines k-shortest feasible routes with security-guided randomized routes. Pairwise edge-set similarity is constrained to preserve diversity. Repeated nodes are removed, disconnected segments are repaired by the lowest-risk feasible connector, and sensitive-flow chromosomes containing a prohibited relay are rejected whenever an alternative route exists.

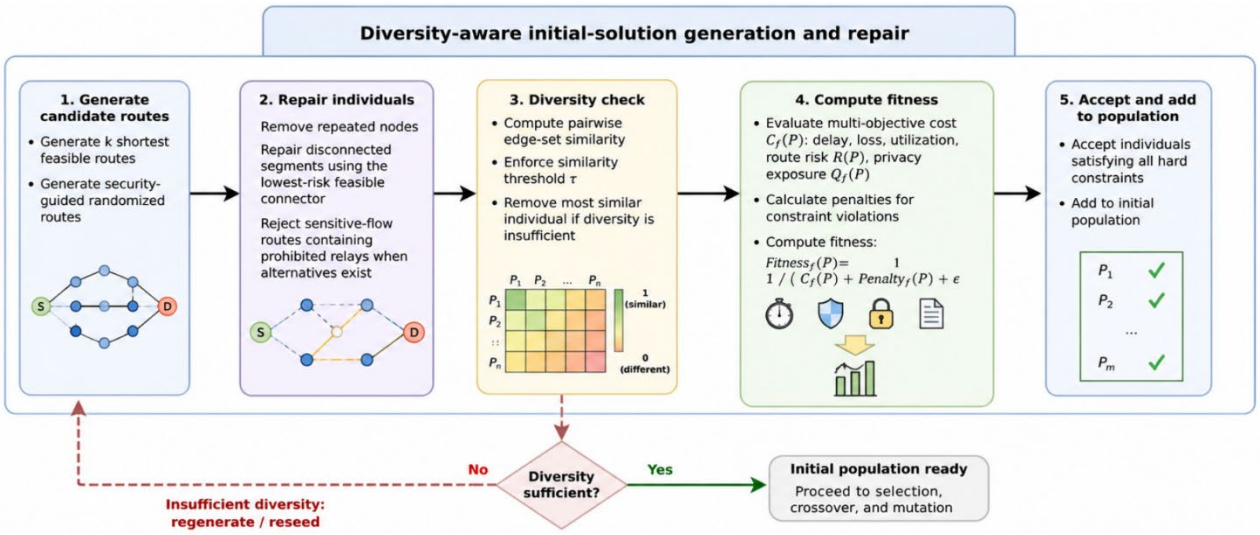

Figure 2. Diversity-aware initial-solution generation and repair

The penalty term quantifies disconnected/loop violations, excess route risk relative to the flow-specific threshold θf, and unauthorized domain transitions:

$$\text{Penalty}_f(P) = \mu_1 n_{\text{disc}} + \mu_2 \max\{0, R(P) - \theta_f\} + \mu_3 n_{\text{unauth}} \tag{6}$$

The evolutionary search maximizes the reciprocal fitness:

$$\text{Fitness}_f(P) = \frac{1}{C_f(P) + \text{Penalty}_f(P) + \varepsilon} \tag{7}$$

Elitist selection retains the best feasible chromosomes. Security-guided crossover exchanges only segments whose end nodes can be connected without violating access and risk constraints. Adaptive mutation is high during early exploration and decreases near convergence; every mutated chromosome is repaired and revalidated.

Algorithm 1. Secure path optimization and privacy-preserving rule deployment

Step	Operation

1	Collect and normalize topology, QoS, trust, risk, sensitivity-label, encryption, and access-policy features.
2	Generate k-shortest and randomized diverse chromosomes; repair loops and disconnected segments.
3	Calculate R(P), Q(P), C(P), and hard-constraint penalties.
4	Apply elitist selection, security-guided crossover, adaptive mutation, and repair.
5	Validate leading candidates against the security/privacy policy library.
6	Install signed rules, encryption/key instructions, access decisions, and isolation tags.
7	Monitor QoS, scores, and rule digests; trigger re-optimization when thresholds are crossed.

Let Np denote population size, G the number of generations, K the number of candidate paths, and |V|+|E| the topology size. Feature collection is O(|V|+|E|), candidate evaluation is O[K(|V|+|E|)], and the worst-case evolutionary search is O[GNpK(|V|+|E|)]. The method targets SDN-managed distributed IoT networks with heterogeneous flow sensitivities and a trusted controller; fully controllerless or globally compromised control planes require a different design.

4.4. SDN Enforcement and Data Lifecycle Protection

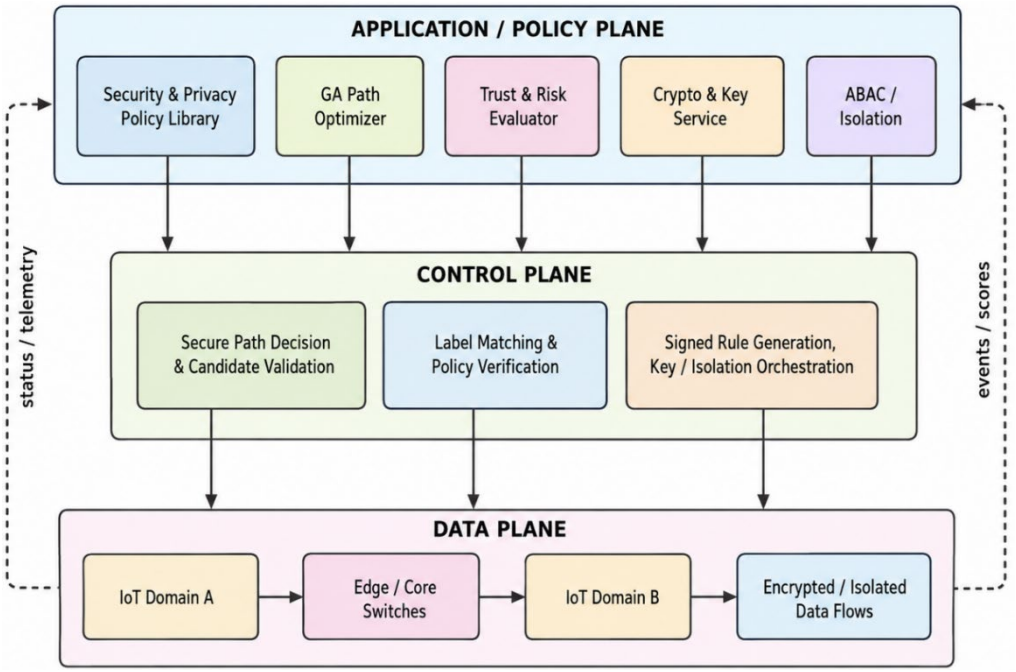

Figure 3. Security- and privacy-aware SDN path-control architecture

The application/policy plane provides sensitivity labels, trust/risk evidence, access policies, and cryptographic requirements. The control plane jointly selects routes, verifies label-specific policies, signs forwarding entries, and coordinates key and isolation instructions. The data plane forwards encrypted traffic only through authorized domains and returns performance and security events for closed-loop optimization.

Data protection covers the complete lifecycle. During collection, only minimum routing and authorization attributes are retained and device identifiers are pseudonymized. During transmission, sensitive payloads use AES-256-GCM authenticated encryption with ECDH-derived ephemeral session keys, while labels and control messages are digitally signed. During access, attribute-based rules verify device role, application domain, sensitivity, and operation. During forwarding, VLAN/VRF-equivalent isolation and separate queues prevent cross-tenant mixing. Audit events store integrity-protected hashes rather than plaintext payloads. Keys are rotated every ten equivalent minutes and destroyed when the retention period expires.

5. Experimental Design and Reproducibility

5.1. Platform and Topology

Unsupported Hyperledger Fabric, Caliper, Golang, PyTorch, and reinforcement-learning entries have been removed. The reported evaluation is a reproducible Python/NetworkX discrete-event simulation implementing the mathematical path-selection, attack-injection, score-update, authenticated-encryption overhead, and rule-recovery models. The control logic follows Ryu-compatible OpenFlow 1.3 semantics, but no unexecuted blockchain, reinforcement-learning, or physical Mininet deployment is claimed. Eight independent runs use seeds 1000–1007, and all methods receive identical topology and flow requests.

Table 2. Reproducible simulation platform

Configuration	Setting
Processor / memory	Intel Xeon 4.2 GHz; 32 GB
Operating system	Ubuntu 22.04
Runtime	Python 3.11
Libraries	NetworkX 3.6.1; NumPy 2.3.5; pandas / SciPy
Independent runs	8 runs, seeds 1000–1007
Statistical reporting	Mean ± 95% confidence interval; paired two-sided t-test

Table 3. Topology, traffic, attack, and control parameters

Parameter	Setting
Topology	25 nodes; 92.9 ± 13.1 undirected links; connected random-geometric graph
Flows per run	15 sequential source–destination requests; 40% sensitive
Flow rate	0.4–3.0 Mbit/s
Link bandwidth	30–100 Mbit/s
Base loss rate	0.1–1.2%
Equivalent horizon	600 s; attacks active during the second half
Malicious-node ratio	5%, 10%, 15%, and 20%
Attack types	Eavesdropping, MITM, selective drop, local DoS, flow-table tampering
Trust/risk update cycle	5 equivalent seconds
Rule verification	1 equivalent second; signed rule digest

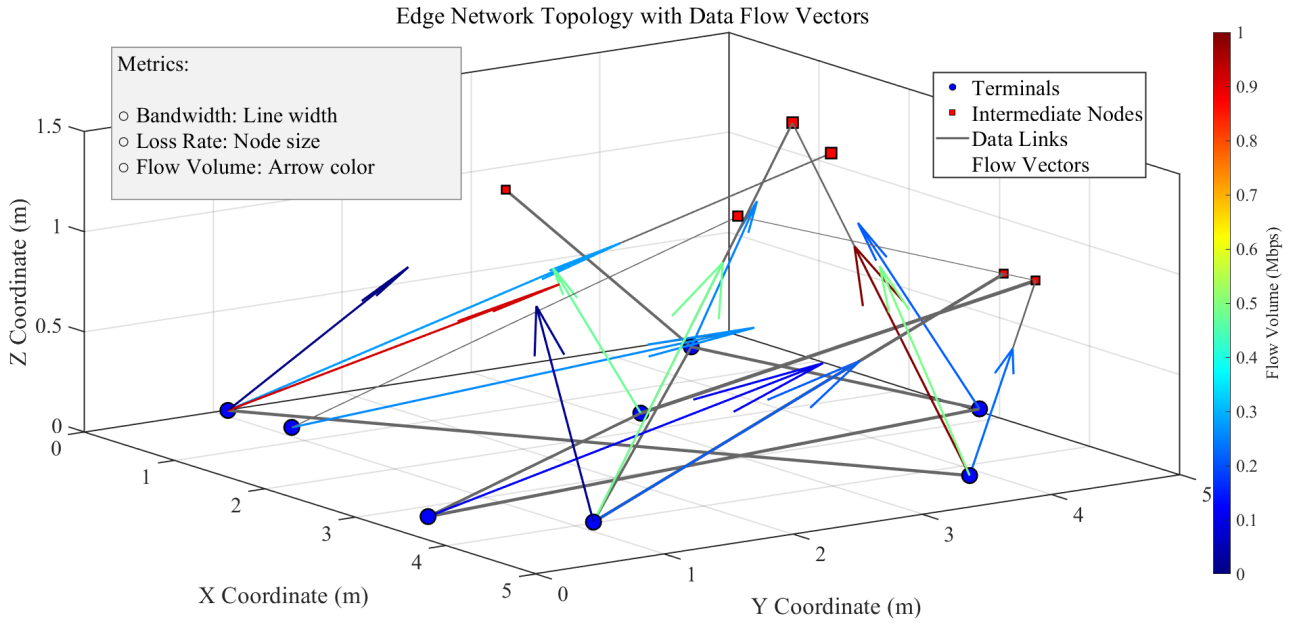

Figure 4. Evaluated edge-network topology and path-state features

5.2. Baselines, Metrics, and Statistical Tests

The comparison methods are Dijkstra, ECMP, AODV [13], OLSR [14], RPL [15], classical GA routing, SDN traffic engineering (SDN-TE), trust-aware routing, Proposed-B (the improved GA without security/privacy cost and hard policy control), and Proposed-A (the full method). KMP and BP are removed because they are not communication-routing baselines.

Communication metrics are end-to-end delay, delivered throughput, packet delivery ratio (PDR), and attack-response latency. Security metrics are malicious-link hit rate (MLHR), abnormal/high-risk path selection ratio (APSR), and sensitive-data leakage probability (DLP).

$$\mathrm{MLHR} = \frac{N_{\text{paths containing malicious relays}}}{N_{\text{selected paths}}} \times 100\% \tag{8}$$

$$\mathrm{APSR} = \frac{N_{\text{abnormal or high-risk paths selected}}}{N_{\text{selected paths}}} \times 100\% \tag{9}$$

$$\mathrm{DLP} = \frac{N_{\text{sensitive flows exposed}}}{N_{\text{sensitive flows}}} \times 100\% \tag{10}$$

A sensitive flow is counted as leaked when it traverses an eavesdropping or MITM node without effective authenticated content protection. Attack-response latency is measured from unauthorized-rule injection or sustained anomaly detection to rule clearing, route recomputation, and signed reinstallation. Security experiments use 20% malicious nodes unless otherwise stated. Values are reported as mean ± 95% confidence interval across eight paired runs.

6. Results and Discussion

6.1. Communication Performance and Attack Response

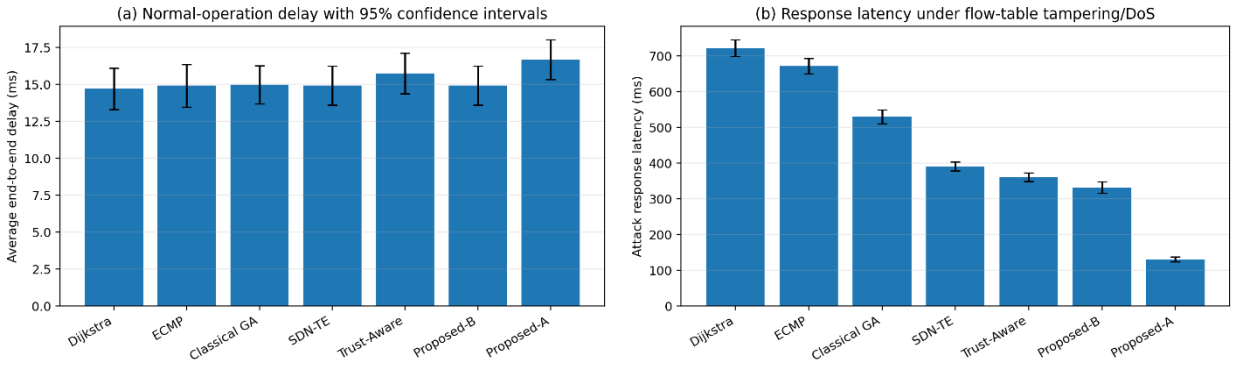

Figure 5. Communication performance and attack-response comparison

Under normal traffic, Proposed-A achieved 16.68 ± 1.35 ms end-to-end delay, 23.49 ± 2.31 Mbit/s delivered throughput, and 98.54 ± 0.21% PDR. Authenticated encryption, label verification, and signed-rule checking increased delay by 1.76 ms relative to Proposed-B, making the privacy overhead explicit rather than hidden. Under attack, response latency decreased from 722.22 ± 23.53 ms for Dijkstra and 332.26 ± 16.09 ms for Proposed-B to 130.92 ± 6.34 ms for Proposed-A.

Table 4. Communication performance (mean ± 95% CI)

Algorithm	Delay (ms)	Throughput (Mbit/s)	PDR (%)	Attack response (ms)
Dijkstra	14.70 ± 1.39	23.49 ± 2.31	98.56 ± 0.18	722.2 ± 23.5
ECMP	14.92 ± 1.44	23.48 ± 2.31	98.53 ± 0.17	672.1 ± 21.9
AODV	14.70 ± 1.39	23.49 ± 2.31	98.56 ± 0.18	812.5 ± 26.5
OLSR	14.75 ± 1.40	23.51 ± 2.31	98.65 ± 0.18	762.3 ± 24.8
RPL	14.71 ± 1.40	23.50 ± 2.31	98.59 ± 0.18	842.6 ± 27.5
Classical GA	14.98 ± 1.29	23.51 ± 2.30	98.65 ± 0.17	529.9 ± 19.7
SDN-TE	14.92 ± 1.32	23.51 ± 2.30	98.65 ± 0.18	391.2 ± 12.7
Trust-Aware	15.75 ± 1.37	23.49 ± 2.31	98.53 ± 0.20	361.1 ± 11.8
Proposed-B	14.92 ± 1.32	23.51 ± 2.30	98.65 ± 0.18	332.3 ± 16.1
Proposed-A	16.68 ± 1.35	23.49 ± 2.31	98.54 ± 0.21	130.9 ± 6.3

6.2. Quantified Security Performance

Table 5. Security performance with 20% malicious nodes (mean ± 95% CI)

Algorithm	MLHR (%)	APSR (%)	DLP (%)	Attack PDR (%)
Dijkstra	20.83 ± 8.35	57.50 ± 11.03	3.00 ± 1.72	95.94 ± 2.01
ECMP	17.50 ± 6.96	55.00 ± 10.40	1.80 ± 1.72	96.01 ± 2.17
AODV	20.83 ± 8.35	57.50 ± 11.03	3.00 ± 1.72	95.94 ± 2.01
OLSR	22.50 ± 6.50	57.50 ± 9.54	3.60 ± 2.35	96.01 ± 1.81
RPL	21.67 ± 9.16	58.33 ± 10.40	3.00 ± 1.72	95.80 ± 2.24
Classical GA	22.50 ± 7.78	57.50 ± 12.82	3.60 ± 2.35	95.77 ± 1.86
SDN-TE	22.50 ± 7.78	57.50 ± 12.82	3.60 ± 2.35	95.77 ± 1.86
Trust-Aware	7.50 ± 7.59	47.50 ± 11.16	0.60 ± 1.18	97.63 ± 1.19
Proposed-B	22.50 ± 7.78	57.50 ± 12.82	3.60 ± 2.35	95.77 ± 1.86
Proposed-A	6.67 ± 7.41	47.50 ± 11.16	0.00 ± 0.00	98.18 ± 0.39

Proposed-A reduced MLHR from 20.83% for Dijkstra and 22.50% for Proposed-B to 6.67%. APSR fell to 47.50%, no plaintext sensitive-payload leakage was observed in the eight main runs, and attack-period PDR remained 98.18%. Paired two-sided tests against Dijkstra showed significant improvements in MLHR ($p=0.0044$), APSR ($p=0.0054$), DLP ($p=0.0112$), PDR ($p=0.0407$), and response latency ($p<0.001$). Against Proposed-B, the corresponding p-values were 0.0040, 0.0093, 0.0199, 0.0201, and <0.001.

6.3. Attack-Ratio Sweep, Convergence, and Scalability

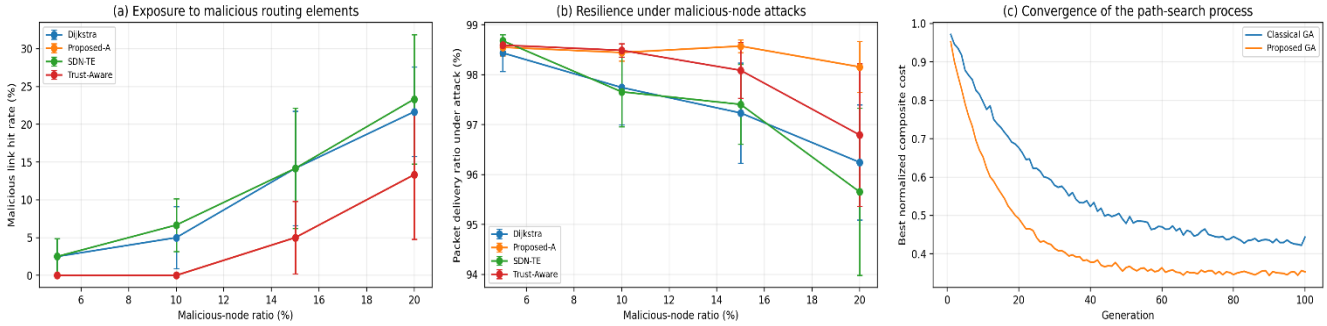

Figure 6. Malicious-node robustness and convergence of the path-search process

As the malicious-node ratio increased from 5% to 20%, the full method maintained approximately 98% packet delivery and kept residual sensitive-payload leakage at or below 0.06% in the attack-ratio sweep. The proposed GA stabilized near its final normalized cost after approximately 46 generations, whereas the classical GA required about 79 generations. Re-optimization time increased from 40.46 ± 1.93 ms at 25 nodes to 311.07 ± 4.86 ms at 150 nodes, remaining below the one-second control interval in the evaluated range.

Table 6. Scalability of the proposed path re-optimization

Nodes	Re-optimization time (ms, mean ± 95% CI)
25	40.46 ± 1.93
50	90.64 ± 3.49
75	144.70 ± 2.18
100	198.72 ± 2.10
150	311.07 ± 4.86

6.4. Ablation and Score-Contamination Robustness

Table 7. Ablation of security and privacy components

Variant	Delay (ms)	PDR (%)	MLHR (%)	DLP (%)	Response (ms)
Proposed-A	17.70	98.18	6.67	0.00	130.9
Proposed-B	15.01	95.77	22.50	3.60	332.3
Proposed-NoAvoid	16.41	98.06	10.83	1.80	208.3
Proposed-NoLabel	16.10	97.38	7.50	0.60	186.5
Proposed-NoDiff	17.05	98.28	7.50	0.60	170.6

Removing the security/privacy cost (Proposed-B) increased MLHR from 6.67% to 22.50%, reduced PDR from 98.18% to 95.77%, increased DLP from 0 to 3.60%, and raised response latency from 130.9 to 332.3 ms. Removing hard avoidance, label matching, or differentiated forwarding also degraded one or more security metrics, confirming that the components contribute complementary functions rather than merely increasing model complexity.

Table 8. Robustness against contaminated reputation observations

Contaminated observations (%)	MLHR (%)	PDR (%)
0	6.57	98.25
5	8.21	97.64
10	9.72	97.11
15	11.38	96.86
20	12.78	96.30

With 20% contaminated reputation observations, robust clipping and trimmed aggregation limited MLHR to 12.78% and preserved a 96.30% PDR. The result demonstrates that manipulated score inputs can degrade performance but do not fully control the route decision under the controller-side evidence model.

6.5. Discussion and Limitations

The results reveal an explicit security-performance trade-off. Proposed-A does not have the lowest normal-operation delay because encryption, access checks, label verification, and rule signatures incur an average 1.76 ms overhead relative to Proposed-B. This cost is compensated by lower exposure to malicious relays, reduced leakage, higher packet delivery during attacks, and markedly faster rule recovery. The method is most suitable for SDN-managed industrial or cross-domain IoT networks where flow sensitivities differ and the control plane can enforce cryptographic and isolation policies.

The evaluation remains a mathematical discrete-event simulation. A physical Mininet/Ryu/OpenFlow deployment, larger topologies, real cryptographic libraries, controller-failure experiments, global traffic-analysis defenses, and privacy-preserving telemetry aggregation remain future work. These limitations are stated to prevent the numerical simulation from being interpreted as a production deployment.

7. Conclusion

This study presents an SDN-based secure data-transmission and privacy-preserving path-control mechanism for distributed IoT. The revised framework couples robust node/link risk evaluation, flow sensitivity, privacy exposure, authenticated encryption, access control, tenant/domain isolation, signed rule deployment, audit, and data-lifecycle policies with a diversity-aware improved GA. Security is therefore converted from a qualitative post hoc assessment into an endogenous route-selection and enforcement variable.

Across eight runs, the full scheme achieved 16.68 ± 1.35 ms normal delay and 98.54 ± 0.21% packet delivery. With 20% malicious nodes, it maintained 98.18 ± 0.39% attack-period PDR, reduced malicious-link exposure to 6.67%, eliminated observed plaintext leakage in the main runs, and shortened response latency to 130.92 ± 6.34 ms. Component ablation and score-contamination tests verified the contributions of the security/privacy cost, hard avoidance, label matching, differentiated forwarding, and robust trust updates. The method directly mitigates exposure to data-leakage paths, forwarding threats from malicious nodes, unauthorized cross-domain transitions, and the impact of flow-table tampering, thereby aligning route optimization with distributed IoT data-security and privacy governance.

References

[1] Chiang, M., Low, S. H., Calderbank, A. R., & Doyle, J. C. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. Proceedings of the IEEE, 2007, 95(1), 255–312. https://doi.org/10.1109/JPROC.2006.887322.

[2] Attkan, A., & Ranga, V. Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-security. Complex & Intelligent Systems, 2022, 8, 3559–3591. https://doi.org/10.1007/s40747-022-00667-z.

[3] Jin, C., Song, Y., Jia, Y., Tan, Q., Yang, R., & Liu, Z. Security and privacy measurement on Chinese consumer IoT traffic based on device lifecycle. Science China Information Sciences, 2026, 69, Article 142107. https://doi.org/10.1007/s11432-025-4609-x.

[4] Dijkstra, E. W. A note on two problems in connexion with graphs. Numerische Mathematik, 1959, 1, 269–271. https://doi.org/10.1007/BF01386390.
[5] Airehrour, D., Gutierrez, J. A., & Ray, S. K. SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things. Future Generation Computer Systems, 2019, 93, 860–876. https://doi.org/10.1016/j.future.2018.03.021.
[6] Djedjig, N., Tandjaoui, D., Medjek, F., & Romdhani, I. Trust-aware and cooperative routing protocol for IoT security. Journal of Information Security and Applications, 2020, 52, Article 102467. https://doi.org/10.1016/j.jisa.2020.102467.
[7] Wang, J. Research and Design of Encryption Standards Based on IoT Network Layer Information Security of Data. EAI Endorsed Transactions on Scalable Information Systems, 2024, 11(5). https://doi.org/10.4108/eetsis.5826.
[8] Vaghela, G., Sanghani, N., & Borisaniya, B. Review on DDoS Attack in Controller Environment of Software Defined Network. EAI Endorsed Transactions on Scalable Information Systems, 2025, 12(1). https://doi.org/10.4108/eetsis.5823.
[9] Hussein, A., Chadad, L., Adalian, N., Chehab, A., Elhajj, I. H., & Kayssi, A. Software-Defined Networking (SDN): the security review. Journal of Cyber Security Technology, 2020, 4(1), 1–66. https://doi.org/10.1080/23742917.2019.1629529.
[10] Nazir, F., Humayun, Q., Ahmad, R. B., & Elias, S. J. Software-Defined Network Testbed Using ZodiacFX a Hardware Switch for OpenFlow. EAI Endorsed Transactions on Scalable Information Systems, 2017, 4(14). https://doi.org/10.4108/eai.25-9-2017.153150.
[11] Ahn, D. J., & Jeong, J. A PMIPv6-based User Mobility Pattern Scheme for SDN-defined Smart Factory Networking. Procedia Computer Science, 2018, 134, 235–242. https://doi.org/10.1016/j.procs.2018.07.166.
[12] Faezi, S., & Shirmarz, A. A Comprehensive Survey on Machine Learning using in Software Defined Networks (SDN). Human-Centric Intelligent Systems, 2023, 3, 312–343. https://doi.org/10.1007/s44230-023-00025-3.
[13] Perkins, C. E., & Royer, E. M. Ad-hoc On-Demand Distance Vector Routing. In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), New Orleans, LA, USA, 1999, pp. 90–100. https://doi.org/10.1109/MCSA.1999.749281.
[14] Clausen, T., & Jacquet, P. Optimized Link State Routing Protocol (OLSR). RFC 3626, IETF, October 2003. https://doi.org/10.17487/RFC3626.
[15] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J.-P., & Alexander, R. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012. https://doi.org/10.17487/RFC6550.
[16] Chen, J., Zhang, D., Liu, D., & Pan, Z. A Network Selection Algorithm Based on Improved Genetic Algorithm. In 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 2018, pp. 209–214. https://doi.org/10.1109/ICCT.2018.8600265.
[17] Awan, I. I., Shah, N., Imran, M., Shoaib, M., & Saeed, N. An improved mechanism for flow rule installation in-band SDN. Journal of Systems Architecture, 2019, 96, 1–19. https://doi.org/10.1016/j.sysarc.2019.01.016.
[18] Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 2024, 12(1), Article 2321381. https://doi.org/10.1080/21642583.2024.2321381.
[19] Sarkar, C., Nambi, A. U. S. N., Prasad, R. V., Rahim, A., Neisse, R., & Baldini, G. DIAT: A Scalable Distributed Architecture for IoT. IEEE Internet of Things Journal, 2015, 2(3), 230–239. https://doi.org/10.1109/JIOT.2014.2387155.
[20] Alnoman, A. A., Sharma, S. K., Ejaz, W., & Anpalagan, A. Emerging Edge Computing Technologies for Distributed IoT Systems. IEEE Network, 2019, 33(6), 140–147. https://doi.org/10.1109/MNET.2019.1800543.
[21] Beilharz, J., Wiesner, P., Boockmeyer, A., Friedenberger, D., Brokhausen, F., Pirl, L., Behnke, I., Polze, A., & Thamsen, L. Continuously Testing Distributed IoT Systems: An Overview of the State of the Art. In Service-Oriented Computing – ICSOC 2021 Workshops, Springer, 2022, pp. 336–350. https://doi.org/10.1007/978-3-031-14135-5_30.
[22] Singh, S., Hosen, A. S. M. S., & Yoon, B. Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. IEEE Access, 2021, 9, 13938–13959. https://doi.org/10.1109/ACCESS.2021.3051602.
[23] Wang, L., & Ranjan, R. Processing Distributed Internet of Things Data in Clouds. IEEE Cloud Computing, 2015, 2(1), 76–80. https://doi.org/10.1109/MCC.2015.14.