DynChain-Vul: Adaptive Blockchain Sharding and Secure Cross-Shard Vulnerability Data Sharing for Cloud-Based Vulnerability Lifecycle Governance

Zhangcai Sun*, **Jun Li, Bo Wu**
Puer Power Supply Bureau of Yunnan Power Grid Co., Ltd., Puer, Yunnan, China
Corresponding author: Zhangcai Sun (e-mail: 1014258769@qq.com)

ABSTRACT Cloud-based vulnerability lifecycle governance requires continuous sharing of vulnerability intelligence, security evidence, remediation records, exploit verification results, and risk assessment information among cloud service providers, security operation centers, software vendors, and regulatory agencies. These security data are highly distributed, dynamically generated, and privacy-sensitive, demanding scalable, trustworthy, and auditable cross-domain data sharing. Although blockchain provides decentralized trust, tamper resistance, and traceability, conventional blockchain architectures suffer from limited throughput, excessive storage redundancy, and inefficient cross-shard data exchange, making them difficult to support large-scale cloud vulnerability governance involving heterogeneous computing resources.

This paper proposes DynChain-Vul, an adaptive blockchain sharding architecture for secure vulnerability data sharing in cloud-based vulnerability lifecycle governance. DynChain-Vul first profiles node computing capability, network bandwidth, communication latency, and storage resources, and then employs Gaussian mixture modeling together with dynamic weighted consistent hashing to allocate virtual nodes according to heterogeneous resource characteristics, thereby improving shard balance and overall system scalability. To accommodate continuously changing workloads, a dynamic shard mapping index supports adaptive shard splitting and merging while minimizing data migration overhead. For secure cross-shard vulnerability information exchange, DynChain-Vul integrates multi-path redundant indexing based on Kademlia routing with a threshold-signature verification mechanism, enabling reliable vulnerability data retrieval without relying on a centralized trusted coordinator.

Experimental results on a blockchain emulation platform demonstrate that DynChain-Vul increases system throughput from 725.98 TPS to 1493.17 TPS as the number of shards increases from 2 to 8, reduces transaction confirmation latency from 39.23 ms to 5.92 ms, maintains shard-load covariance within 0.12–0.15, and keeps aggregated cross-shard signature storage nearly constant. These results indicate that adaptive blockchain sharding combined with verifiable cross-shard vulnerability data sharing provides an efficient and trustworthy foundation for cloud-based vulnerability lifecycle governance, supporting scalable collaborative vulnerability intelligence sharing and secure security operations.

INDEX TERMS Power Internet of Things, blockchain sharding, dynamic weighted consistent hashing, cross-shard data sharing, threshold signature, Kademlia routing

I. INTRODUCTION

Cloud-native infrastructures are rapidly becoming the foundation of modern digital power systems and critical information infrastructures. Cloud platforms, virtualization environments, container orchestration systems, microservice applications, and distributed edge nodes continuously generate large volumes of security-related information, including asset inventories, vulnerability reports, exploit verification results, remediation records, and security audit logs. These security data must be securely exchanged among cloud service providers, security operation centers (SOCs), software vendors, enterprise users, and regulatory agencies.

Compared with conventional industrial Internet of Things (IoT) applications, cloud-based vulnerability governance

imposes much stricter requirements on data authenticity, timeliness, traceability, and accountability. An inaccurate vulnerability report, delayed remediation record, or unverifiable exploit validation result may directly affect vulnerability assessment, risk analysis, security operation, and regulatory auditing. Existing studies on cloud security and cyber-physical systems have emphasized the importance of scalable, trustworthy, and interoperable security data sharing in such collaborative environments [1, 2, 3].

Blockchain has emerged as a promising solution for trustworthy vulnerability information management because it provides decentralized storage, immutable ledgers, distributed consensus, and auditable data provenance. It enables secure vulnerability intelligence sharing, tamper-resistant evidence storage, collaborative remediation management, and trusted security auditing without relying on a centralized authority. However, directly applying conventional blockchain architectures to cloud-based vulnerability governance remains challenging. Public blockchains require every node to maintain the complete ledger and participate in global consensus, resulting in considerable communication, computation, and storage overhead. Although permissioned blockchain platforms such as Hyperledger Fabric alleviate some of these limitations, large-scale vulnerability lifecycle governance involving multiple organizations still suffers from heterogeneous computing resources, dynamically changing workloads, and inefficient cross-domain data sharing [4, 5, 6].

Blockchain sharding has been widely recognized as an effective approach for improving scalability by partitioning nodes, transactions, and ledger states into multiple parallel shards. Such an architecture naturally fits collaborative vulnerability governance, where different organizations or security domains can independently process local vulnerability information while periodically exchanging shared security evidence. Nevertheless, most existing sharding mechanisms assume homogeneous computing environments or static shard configurations. In practical cloud-security deployments, participating nodes exhibit substantial heterogeneity in computing capability, network bandwidth, communication latency, storage resources, and service availability. High-performance cloud servers can undertake more consensus and storage responsibilities, whereas lightweight edge nodes and departmental servers possess limited computational capacity. Furthermore, large-scale vulnerability disclosures, emergency security incidents, or coordinated attack campaigns may generate temporary workload hotspots, causing severe shard imbalance under static sharding strategies.

Efficient cross-shard vulnerability information sharing remains another critical challenge. Cloud-based vulnerability lifecycle governance frequently requires security evidence distributed across multiple organizations. For example, vulnerability verification may require software bill of materials (SBOM), vulnerability intelligence, exploit validation reports, remediation records, and security audit logs maintained by different entities. Conventional notary-based cross-chain solutions introduce centralized trust assumptions, whereas relay-chain and hash-lock mechanisms often incur considerable communication overhead during frequent cross-shard data access. Consequently, a scalable vulnerability governance platform should support efficient cross-shard routing together with decentralized and verifiable security data authentication without depending on any single trusted intermediary.

To address these challenges, this paper proposes DynChain-Vul, an adaptive blockchain architecture for secure vulnerability data sharing in cloud-based vulnerability lifecycle governance. The proposed framework combines resource-aware adaptive blockchain sharding with verifiable cross-shard vulnerability information exchange. The main contributions are summarized as follows:

- We propose DynChain-Vul, a blockchain architecture for cloud-based vulnerability lifecycle governance that profiles heterogeneous computing nodes and dynamically allocates shard participation weights according to computing capability, communication bandwidth, latency, and storage resources.
- We develop a dynamic weighted consistent hashing mechanism supported by Gaussian Mixture Model (GMM)-based node classification to allocate virtual nodes adaptively and improve shard load balancing under heterogeneous cloud environments.
- We introduce a Shard Mapping Index Table (SMIT) to support adaptive shard splitting and merging, significantly reducing data migration overhead and routing interruption under dynamically changing vulnerability workloads.
- We integrate Multi-Path Redundant Indexing (MPRI) with threshold-signature verification to achieve efficient, decentralized, and trustworthy cross-shard vulnerability information sharing without relying on centralized coordinators.
- We conduct comprehensive performance evaluations against representative blockchain sharding and cross-shard data sharing approaches. Experimental results demonstrate that DynChain-Vul achieves superior throughput, lower confirmation latency, improved load balancing, reduced storage overhead, and more efficient cross-shard vulnerability data retrieval.

II. RELATED WORK

A. BLOCKCHAIN FOR IOT AND POWER DATA SHARING

Blockchain has been widely explored for IoT access control, product traceability, edge data management, and cross-domain authentication because it can record events in a tamper-resistant and auditable form [5, 7, 8, 9]. For Power IoT, these properties are attractive for meter-data notarization, device identity, distributed energy resource coordination, and multi-party operational auditing. Nevertheless, IoT and grid-edge devices are constrained by computation, bandwidth, storage, and energy budgets. Lightweight nodes,

edge-assisted consensus, and permissioned ledgers reduce some resource pressure, but they do not fully solve large-scale shard management or cross-shard data verification.

B. BLOCKCHAIN SHARDING

Sharding protocols such as Elastico, OmniLedger, RapidChain, Chainspace, and Monoxide improve blockchain throughput by dividing consensus and state management among parallel groups [10, 11, 12, 13, 14]. Later studies examine transaction placement, load balance, state sharding, and adaptive scaling [15, 16, 17, 18, 19]. These studies provide important foundations, but many schemes still rely on static or weakly adaptive shard allocation. In Power IoT, workload and device resources change over time, so shard layouts must adapt while minimizing migration cost and preserving query availability.

C. CROSS-SHARD AND CROSS-CHAIN VERIFICATION

Cross-chain and cross-shard interaction mechanisms allow data or assets to move across ledger domains. Notary schemes are easy to deploy but depend on trusted intermediaries. Relay, side-chain, and smart-contract approaches can reduce centralization but often require additional communication or storage overhead [20, 21]. Threshold signatures provide a compact way for multiple validators to jointly endorse a message while storing only one aggregate signature [22, 23]. Kademlia-style distributed hash tables provide efficient decentralized routing based on XOR distance [24]. DynChain combines these ideas to support verifiable cross-shard data access in a dynamic sharding architecture.

III. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a permissioned Power IoT blockchain network composed of heterogeneous devices and edge nodes. Devices include smart meters, protection and control terminals, phasor measurement units, distributed energy resource gateways, charging-station controllers, and substation servers. Each device may generate data records, submit transactions, participate in local consensus, or serve as a validator for cross-shard queries. The network is partitioned into multiple shards, and each shard maintains a local ledger interval and a local consensus group.

Let $D = \{d_1, d_2, \ldots, d_n\}$ denote the device set. Each device d_i has a feature vector

$$\mathbf{x}_i = [c_i, b_i, l_i, s_i], \tag{1}$$

where c_i denotes computing capability, b_i denotes available bandwidth, l_i denotes network latency, and s_i denotes storage capacity. These features are normalized to remove scale differences:

$$\bar{x}_{ij} = \frac{x_{ij} - \min(\mathbf{x}_j)}{\max(\mathbf{x}_j) - \min(\mathbf{x}_j) + \epsilon}. \tag{2}$$

The objective is to assign devices and transactions to shards such that (i) high-capability devices undertake more consensus work, (ii) low-capability devices remain usable without excessive storage or computation burden, (iii) shard load remains balanced under workload fluctuation, and (iv) cross-shard data queries remain verifiable and efficient.

For a shard S_k, let L_k be its transaction load, C_k its aggregate consensus capacity, and M_k the amount of data that must be migrated during adjustment. DynChain aims to minimize

$$\mathcal{J} = \lambda_1 \operatorname{Var}(L_1, \ldots, L_m) + \lambda_2 \sum_{k=1}^{m} \frac{L_k}{C_k} + \lambda_3 M_k, \tag{3}$$

where m is the shard number and $\lambda_1, \lambda_2, \lambda_3$ are deployment-dependent weights. In Power IoT, λ_1 and λ_2 reflect operational latency requirements, while λ_3 captures the cost of moving historical or auditable data.

IV. DYNCHAIN ADAPTIVE SHARDING

A. ARCHITECTURE OVERVIEW

Fig. 1 summarizes the proposed architecture. DynChain first profiles Power IoT devices and classifies them into resource classes. The class weight determines the number of virtual nodes assigned to each physical device. These virtual nodes are mapped to a consistent hash ring and then partitioned into shards. Each shard independently processes local transactions and maintains the corresponding ledger interval. When transaction load changes, SMIT coordinates split or merge operations and updates routing indexes. For data requests spanning multiple shards, MPRI and threshold signatures provide efficient retrieval and verifiable authenticity.

B. GMM-BASED DEVICE CLASSIFICATION

Power IoT devices differ significantly in computation, communication, and storage resources. DynChain converts these heterogeneous characteristics into a resource score. Let $\bar{t}_i$ represent the normalized transaction-processing cost derived from CPU time, bandwidth, and latency, and let $\bar{s}_i$ represent normalized storage capacity. A tunable score is defined as

$$r_i = \alpha(1 - \bar{t}_i) + (1 - \alpha)\bar{s}_i, \tag{4}$$

where $\alpha \in [0, 1]$ controls whether the deployment prioritizes transaction speed or long-term storage capacity. A dispatching-oriented system may choose a larger α to reduce processing latency, whereas an archival metering system may choose a smaller α to emphasize storage stability.

DynChain then uses a Gaussian mixture model to cluster devices into K resource classes [25]:

$$p(r_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(r_i \mid \mu_k, \sigma_k^2), \tag{5}$$

where π_k is the mixture weight and $\mathcal{N}(\cdot)$ is a Gaussian component. The expectation-maximization procedure estimates posterior responsibility γ_{ik}, class mean μ_k, and variance σ_k^2. The weight of class k is

$$w_k = \frac{\sum_i \gamma_{ik} r_i}{\sum_i \gamma_{ik}}. \tag{6}$$

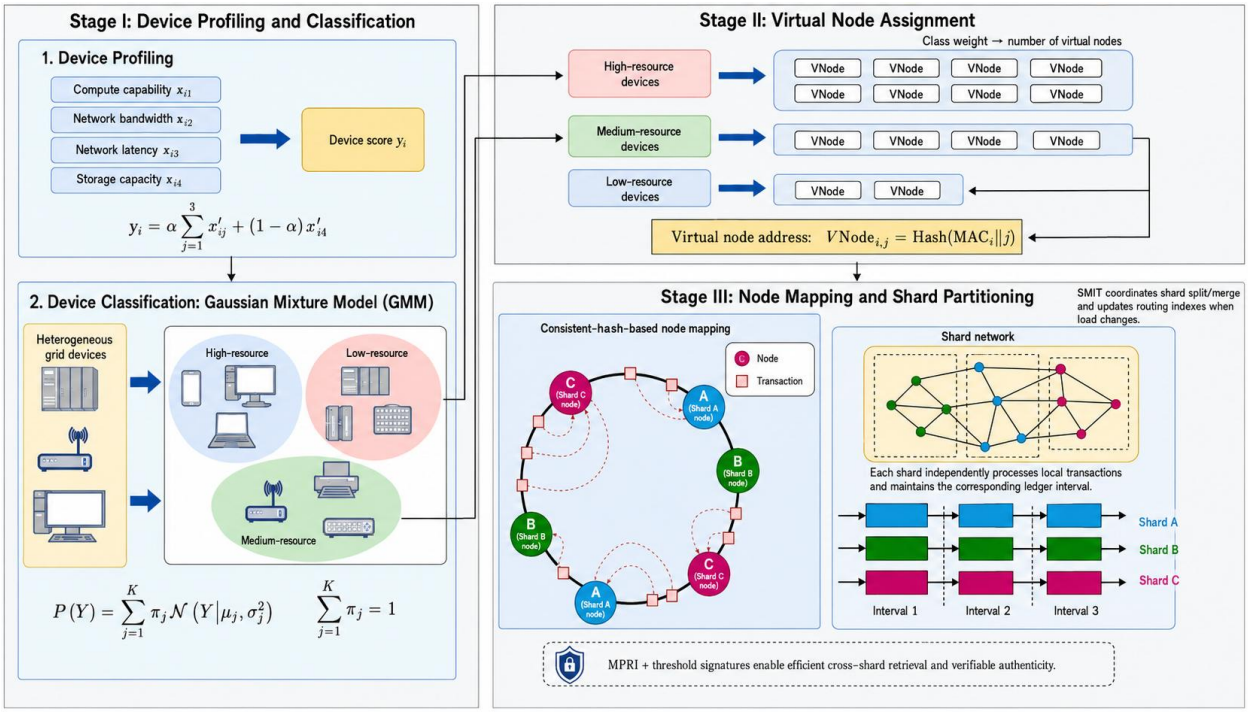

FIGURE 1. Overview of the DynChain architecture for Power IoT data sharing. Heterogeneous grid devices are profiled, classified, and mapped to weighted virtual nodes before adaptive sharding and verifiable cross-shard data exchange.

Each device in class k receives

$$v_i = \lceil \eta w_k \rceil \tag{7}$$

virtual nodes, where η is a scaling factor. Thus, higher-capability devices appear more frequently on the hash ring and participate in more consensus tasks, while low-power terminals participate with smaller load.

C. DYNAMIC WEIGHTED CONSISTENT HASHING

Consistent hashing maps both transactions and virtual nodes into the same circular hash space. If $H(\cdot)$ is a hash function, the q-th virtual node of device d_i is placed at

$$h_{i,q} = H(d_i \parallel q), \qquad q = 1, \ldots, v_i. \tag{8}$$

Transactions are assigned to the first clockwise virtual node or shard interval. Compared with modulo-based placement, consistent hashing limits remapping when devices join, leave, or change weight. Weighted virtual nodes further improve resource-aware balance because a powerful edge gateway owns more hash intervals than a constrained terminal.

D. SMIT-BASED SHARD SPLIT AND MERGE

Static sharding cannot handle the highly variable workload of Power IoT. Fault events, meter-reading windows, DER control campaigns, and charging-station bursts may create temporary hot shards. DynChain maintains a shard mapping index table:

$$\text{SMIT} = \{sid, [h_l, h_u), V_s, I_t, ver\}, \tag{9}$$

where sid is the shard identifier, $[h_l, h_u)$ is the hash interval, V_s is the node set, I_t is the transaction index, and ver is the index version. When L_k exceeds an upper threshold, SMIT triggers a split operation and divides the corresponding hash interval into two balanced intervals. When neighboring shards remain below a lower threshold, SMIT triggers a merge operation.

Fig. 2 illustrates the split and merge process. The core design principle is to update index pointers and hash intervals before moving data. This reduces bulk migration and allows historical data to remain queryable during adjustment. External shards periodically synchronize SMIT versions so that cross-shard requests are routed to the current target shard.

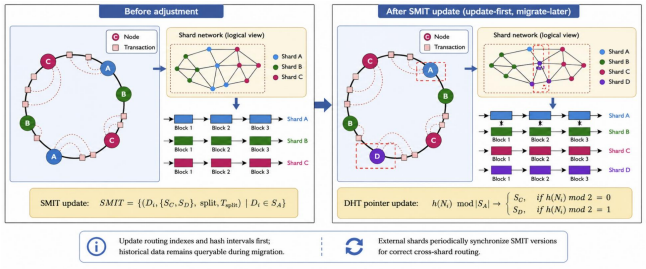

FIGURE 2. SMIT-driven shard split and merge. DynChain updates hash intervals and routing indexes to reduce data migration and avoid long service interruption during workload fluctuation.

V. SECURE CROSS-SHARD DATA SHARING

A. THRESHOLD-SIGNED CROSS-SHARD VERIFICATION

For cross-shard or cross-chain data sharing, DynChain uses a threshold-signature mechanism. Consider a validator committee of n nodes, where any t valid partial signatures can produce an aggregate signature. For a message m representing a data request or data response, validator j computes

$$\sigma_j = H(m)^{sk_j}, \tag{10}$$

where sk_j is the validator's private key share. After at least t partial signatures are collected, the smart contract or aggregation component computes

$$\sigma = \prod_{j \in T} \sigma_j^{\lambda_j}, \tag{11}$$

where T is the participating validator set and λ_j is the Lagrange coefficient. The target shard verifies the aggregate signature through a bilinear pairing equation:

$$e(\sigma, g) \stackrel{?}{=} e(H(m), PK), \tag{12}$$

where PK is the aggregate public key. If the equation holds, the data response is accepted; otherwise, it is rejected.

This mechanism removes the need to store and verify every individual validator signature on chain. It also avoids a single notary because no validator alone can forge a valid endorsement. In a Power IoT deployment, validators may correspond to substations, regional edge gateways, or authorized utility nodes.

B. MULTI-PATH REDUNDANT INDEXING

Threshold signatures ensure authenticity, but they do not by themselves reduce query latency. DynChain therefore introduces MPRI, a multi-path redundant index based on Kademlia-style routing. For a data object with key k_d, shards maintain distributed index entries according to XOR distance. A query starts from the source shard and greedily forwards the request to neighbors closer to k_d. Instead of relying on a single path, MPRI keeps multiple candidate paths, which improves robustness when a shard is busy, offline, or recently adjusted.

The query process contains four steps. First, the source shard locates candidate next hops through the DHT routing table. Second, greedy forwarding selects the path with the smallest expected latency. Third, redundant candidate paths are preserved to avoid single-path failure. Finally, the target shard returns the data and threshold signature, allowing the source shard to verify authenticity before serving the application. This design is suitable for Power IoT because data requests often need both low latency and auditability.

ALGORITHM 1. MPRI QUERY WITH THRESHOLD VERIFICATION

Input: source shard r_s, data key κ, threshold τ
Output: verified data response R or rejection $\perp$

$P \leftarrow$ candidate paths from the Kademlia routing table
Select the lowest-latency path by greedy forwarding
Forward the request to target shard S_t
Validators in S_t compute partial signatures σ_j
Aggregate $\sigma \leftarrow \prod_{j \in T} \sigma_j^{\lambda_j}$
if $e(\sigma, g) = e(H(m), PK)$ then
  return verified response R
else
  return rejection $\perp$
end if

VI. EXPERIMENTAL EVALUATION

A. EXPERIMENTAL SETUP

DynChain was implemented on block-emulator and evaluated against representative sharding and query baselines. The test environment used an Intel Xeon Silver 4214 CPU at 2.20 GHz, 256 GB DDR4 memory, a 16 TB disk, and CentOS 7.9. The transaction workload used the first three million Ethereum mainnet transactions, represented through XBlock-ETH-style account records [26, 27]. In the Power IoT adaptation, account addresses are treated as grid terminals or data producers with heterogeneous resource profiles. This replayed workload stresses shard balance, transaction processing, confirmation latency, and cross-shard routing.

The sharding baselines include Monoxide [14], BrokerChain [28], and CLPA-style graph partitioning [17]. Cross-shard data query is compared with random-walk search and skip-list based retrieval [29]. For cross-chain verification, DynChain's threshold-signature mechanism is compared with a notary-style ECDSA scheme [26]. Table I summarizes the experimental configuration.

TABLE I. EXPERIMENTAL CONFIGURATION

Item	Setting
Prototype	Block-emulator based DynChain implementation
Processor	Intel Xeon Silver 4214 CPU @ 2.20 GHz
Memory	256 GB DDR4
Operating system	CentOS 7.9
Main workload	3,000,000 Ethereum transactions
Shard counts	2, 4, 6, 8 for throughput/latency tests
Query shard counts	16, 32, 64, 128 for MPRI tests
Baselines	Monoxide, BrokerChain, CLPA, ECDSA, random walk, skip-list

B. THROUGHPUT AND CONFIRMATION LATENCY

Fig. 3 compares throughput and confirmation latency under different shard counts with 96 virtual nodes. DynChain shows a clear scaling trend: throughput increases from 725.98 TPS with 2 shards to 1493.17 TPS with 8 shards. Under the same settings, BrokerChain improves from 569.17 TPS to 1027.02 TPS. The advantage of DynChain comes from its resource-aware virtual-node allocation and lower cross-shard contention. Confirmation latency also decreases from 39.23 ms to 5.92 ms as shard parallelism increases. In contrast, Monoxide becomes less stable when the shard

TABLE II. SUMMARY OF REPRESENTATIVE EXPERIMENTAL RESULTS

Metric	DynChain result	Reference comparison
Throughput, 2 to 8 shards	725.98 to 1493.17 TPS	BrokerChain: 569.17 to 1027.02 TPS
Confirmation latency	39.23 to 5.92 ms	Monoxide becomes unstable beyond 6 shards
Shard-load covariance	0.12–0.15	Monoxide: 0.21 to 1.38
Aggregate signature storage	33–48 bytes	ECDSA grows linearly; $>$ 1200 bytes at 21 nodes
Cross-chain verification time	about 5.53 ms at 31 nodes	ECDSA increases with validators
MPRI query hops	3.98 to 6.91	Random walk: 4.43 to 11.72
MPRI query time	26.17 to 38.17 ms	Random walk: 59.93 ms at 16 shards and grows rapidly

count is large because cross-shard transactions introduce additional coordination overhead.

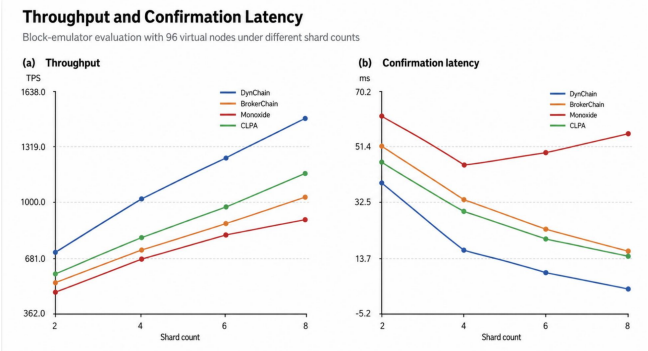

FIGURE 3. Throughput and confirmation latency under different shard counts. DynChain achieves higher throughput and lower latency due to adaptive weighted sharding.

C. LOAD BALANCE

Shard-load covariance is used to measure whether transactions are evenly distributed across shards. Smaller covariance indicates better balance. As shown in Fig. 5(a), DynChain maintains covariance within 0.12–0.15 when the shard count increases from 2 to 8. Monoxide performs acceptably at small scale, with covariance around 0.21, but grows to 1.38 at 8 shards because its fixed mapping is less able to adapt to workload concentration. BrokerChain and CLPA improve over Monoxide but still exhibit larger imbalance than DynChain. The result confirms that dynamic weighted consistent hashing and virtual-node allocation can smooth shard load under heterogeneous resources.

D. THRESHOLD-SIGNATURE OVERHEAD

Table II and Fig. 4 report signature overhead. A single ECDSA verification is faster than a pairing-based threshold signature, but ECDSA storage grows with the number of endorsing validators because every signature must be stored and verified separately. In contrast, DynChain stores only one aggregate threshold signature. The measured BLS threshold signature size is 33 bytes at the algorithm level and 48 bytes in the encoded cross-chain experiment, both remaining constant as validator count grows. When the cross-chain validator count reaches 31, verification time remains around 5.53 ms, showing stable scalability for large cross-shard data endorsement.

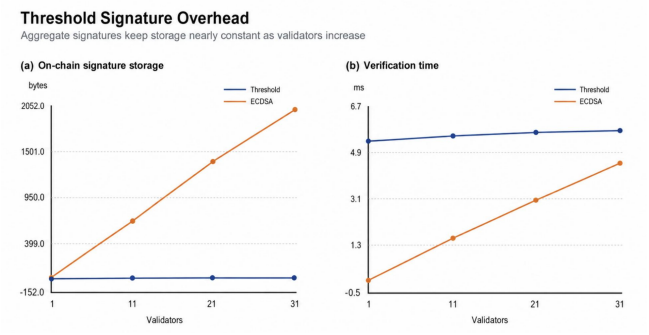

FIGURE 4. Threshold-signature overhead compared with ECDSA. Aggregate signatures keep on-chain storage nearly constant as validator count increases.

E. CROSS-SHARD QUERY EFFICIENCY

Fig. 5(b) evaluates query latency for 16, 32, 64, and 128 shards. MPRI keeps the average hop count low, increasing from 3.98 hops at 16 shards to 6.91 hops at 128 shards. Query time rises moderately from 26.17 ms to 38.17 ms. Random-walk search grows much faster: its hop count increases from 4.43 to 11.72, and its 16-shard query time is already 59.93 ms. Skip-list retrieval is closer to MPRI in hop count but has higher latency because it lacks greedy path optimization and redundant candidate selection. These results show that Kademlia-based multi-path indexing is suitable for large-scale Power IoT data lookup.

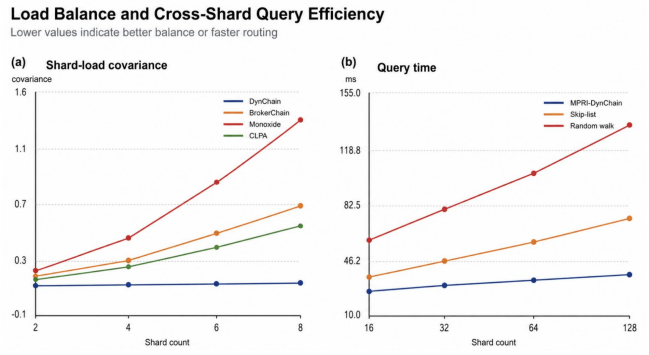

FIGURE 5. Load balance and cross-shard query efficiency. DynChain maintains lower shard-load covariance and lower query time than baseline approaches.

VII. DISCUSSION

The results suggest that resource-aware adaptive sharding is valuable for Power IoT data-sharing platforms. Unlike homogeneous sharding, DynChain allows substation servers

and edge gateways to undertake more work while keeping constrained terminals lightly loaded. The SMIT mechanism further reduces adjustment cost during workload bursts by updating routing and hash-interval metadata before large-scale data migration. This behavior is important for grid applications because service interruption during topology change or fault-event bursts may reduce the usefulness of the ledger.

The threshold-signature and MPRI components address a complementary requirement: a Power IoT blockchain must not only process local transactions efficiently but also support auditable cross-domain queries. Aggregate signatures reduce storage growth, and Kademlia-style routing reduces query latency. Together, they allow different grid participants to verify data without relying on a single notary.

Several limitations remain. First, the current prototype uses replayed transaction traces and heterogeneous-device profiles rather than a fully instrumented production grid dataset. Second, split and merge thresholds are manually configured; future work should learn thresholds adaptively from workload and device-health signals. Third, threshold-signature verification introduces pairing overhead. Although the overhead is millisecond-level in the prototype, large-scale deployments may benefit from batching, hardware acceleration, or lighter verifiable-computation techniques.

VIII. CONCLUSION

This paper presents DynChain, an adaptive blockchain sharding and secure cross-shard data-sharing architecture for Power IoT. DynChain uses GMM-based device classification and dynamic weighted consistent hashing to allocate consensus work according to heterogeneous device capability. SMIT supports shard split and merge with reduced migration, while MPRI and threshold signatures provide efficient and verifiable cross-shard data access. Experimental results show improved throughput, lower confirmation latency, stable load balance, compact signature storage, and efficient cross-shard query performance. DynChain therefore provides a practical foundation for trustworthy data sharing among smart-grid devices, substations, and cross-domain energy service platforms.

ACKNOWLEDGEMENTS

This work was supported by the Yunnan Power Grid Co., Ltd., Science and Technology Project, Research on Vulnerability Lifecycle Governance and Security Protection Technologies in Cloud Environments (Project No. YNKJXM20240458), specifically under the subproject Research on Intelligent Cross-Domain Cloud Asset Inventory and Vulnerability Governance Technologies; the Technical Service Project of Puer Power Supply Bureau, Yunnan Power Grid Co., Ltd., Research on Distributed Information Asset Vulnerability Risk Governance Technologies (Contract No. 05080020250303010500373).

DECLARATIONS

- **Funding:** No specific funding information is provided in this manuscript. This field should be completed before journal submission if project funding is to be reported.
- **Conflict of interest:** The authors declare no conflict of interest.
- **Data availability:** The evaluation uses public blockchain transaction traces and a block-emulator prototype. Detailed scripts and processed traces can be made available by the authors upon reasonable request.
- **Author contributions:** Peihao Zhang contributed conceptualization, methodology, implementation, experiments, and writing. Ying Gao contributed supervision, validation, and manuscript review.

REFERENCES

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid—The New and Improved Power Grid: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012, doi: https://doi.org/10.1109/SURV.2011.101911.00087.

[2] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart Grid Technologies: Communication Technologies and Standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, 2011, doi: https://doi.org/10.1109/TII.2011.2166794.

[3] W. Wang and Z. Lu, "Cyber security in the Smart Grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013, doi: https://doi.org/10.1016/j.comnet.2012.12.017.

[4] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. Weed Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15, doi: https://doi.org/10.1145/3190508.3190538.

[5] L. Da Xu, Y. Lu, and L. Li, "Embedding Blockchain Technology Into IoT for Security: A Survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10452–10473, 2021, doi: https://doi.org/10.1109/JIOT.2021.3060508.

[6] A. Reyna, C. Martin, J. Chen, E. Soler, and M. Diaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018, doi: https://doi.org/10.1016/j.future.2018.05.046.

[7] S. Pal, A. Dorri, and R. Jurdak, "Blockchain for IoT access control: Recent trends and future research directions," *Journal of Network and Computer Applications*, vol. 203, p. 103371, 2022, doi: https://doi.org/10.1016/j.jnca.2022.103371.

[8] Q. Lu and X. Xu, "Adaptable Blockchain-Based Systems: A Case Study for Product Traceability," *IEEE Software*, vol. 34, no. 6, pp. 21–27, 2017, doi: https://doi.org/10.1109/MS.2017.4121227.

[9] T. Nguyen, H. Nguyen, and T. N. Gia, "Exploring the integration of edge computing and blockchain IoT: Principles, architectures, security, and applications," *Journal of Network and Computer Applications*, vol. 226, p. 103884, 2024, doi: https://doi.org/10.1016/j.jnca.2024.103884.

[10] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol For Open Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 17–30, doi: https://doi.org/10.1145/2976749.2978389.

[11] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 583–598, doi: https://doi.org/10.1109/SP.2018.000-5.

[12] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling Blockchain via Full Sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948, doi: https://doi.org/10.1145/3243734.3243853.

[13] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A Sharded Smart Contracts Platform," in *Proceedings 2018 Network and Distributed System Security Symposium*, 2018, doi: https://doi.org/10.14722/ndss.2018.23241.

[14] J. Wang and H. Wang, "Monoxide: Scale out Blockchains with Asynchronous Consensus Zones," in *16th USENIX Symposium on Networked Systems Design and Implementation*, 2019, pp. 95–112. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping

[15] L. N. Nguyen, T. D. T. Nguyen, T. N. Dinh, and M. T. Thai, "OptChain: Optimal Transactions Placement for Scalable Blockchain Sharding," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 525–535, doi: https://doi.org/10.1109/ICDCS.2019.00059.

[16] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in Blockchains," *IEEE Access*, vol. 8, pp. 14155–14181, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2965147.

[17] C. Li, H. Huang, Y. Zhao, X. Peng, R. Yang, Z. Zheng, and S. Guo, "Achieving Scalability and Load Balance across Blockchain Shards for State Sharding," in *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, 2022, pp. 284–294, doi: https://doi.org/10.1109/SRDS55811.2022.00034.

[18] J. Xu, Q. Xie, S. Peng, C. Wang, and X. Jia, "AdaptChain: Adaptive Scaling Blockchain With Transaction Deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 6, pp. 1909–1922, 2023, doi: https://doi.org/10.1109/TPDS.2023.3267071.

[19] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A Survey on Blockchain Interoperability: Past, Present, and Future Trends," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–41, 2021, doi: https://doi.org/10.1145/3471140.

[20] R. Qiao, X.-Y. Luo, S.-F. Zhu, A.-D. Liu, X.-Q. Yan, and Q.-X. Wang, "Dynamic Autonomous Cross Consortium Chain Mechanism in e-Healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2157–2168, 2020, doi: https://doi.org/10.1109/JBHI.2019.2963437.

[21] S. Guo, F. Wang, N. Zhang, F. Qi, and X. Qiu, "Master-slave chain based trusted cross-domain authentication mechanism in IoT," *Journal of Network and Computer Applications*, vol. 172, p. 102812, 2020, doi: https://doi.org/10.1016/j.jnca.2020.102812.

[22] V. Shoup, "Practical Threshold Signatures," in *Advances in Cryptology – EUROCRYPT 2000*, Springer, 2000, pp. 207–220, doi: https://doi.org/10.1007/3-540-45539-6_15.

[23] Q. Feng, K. Yang, M. Ma, and D. He, "Efficient Multi-Party EdDSA Signature With Identifiable Aborts and its Applications to Blockchain," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1937–1950, 2023, doi: https://doi.org/10.1109/TIFS.2023.3256710.

[24] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Peer-to-Peer Systems*, Springer, 2002, pp. 53–65, doi: https://doi.org/10.1007/3-540-45748-8_5.

[25] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, 2000, doi: https://doi.org/10.1006/dspr.1999.0361.

[26] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001, doi: https://doi.org/10.1007/s102070100002.

[27] P. Zheng, Z. Zheng, J. Wu, and H. N. Dai, "XBlock-ETH: Extracting and Exploring Blockchain Data From Ethereum," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 95–106, 2020, doi: https://doi.org/10.1109/OJCS.2020.2990458.

[28] H. Huang, Z. Yin, Q. Chen, J. Zheng, X. Luo, G. Ye, X. Peng, Z. Zheng, and S. Guo, "BrokerChain: A Blockchain Sharding Protocol by Exploiting Broker Accounts," *IEEE Transactions on Networking*, vol. 33, no. 4, pp. 1930–1945, 2025, doi: https://doi.org/10.1109/TON.2025.3550502.

[29] J. Newell, S. ur Rehman, Q. Mamun, and M. Z. Islam, "EASL: Enhanced append-only skip list index for agile block data retrieval on blockchain," *Future Generation Computer Systems*, vol. 164, p. 107554, 2025, doi: https://doi.org/10.1016/j.future.2024.107554.