Preprints.org

Concept Paper

Not peer-reviewed version

A Proposed Framework for Blockchain Security Evaluation Using Graph Models

Divyasree Bellary *

Posted Date: 9 April 2026

doi: 10.20944/preprints202604.0637.v1

Keywords: blockchain security; graph theory; graph models; smart contract vulnerability; consensus mechanisms; anomaly detection; transaction graph analysis; peer-to-peer networks; temporal graph analysis; security evaluation framework

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Concept Paper

A Proposed Framework for Blockchain Security Evaluation Using Graph Models

Divyasree Bellary

School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India; divyabellary7@gmail.com

Abstract

Blockchain networks have emerged as foundational infrastructure for decentralised finance, supply chain management, healthcare data exchange, and digital identity systems. Despite their cryptographic foundations and distributed consensus mechanisms, blockchain deployments remain susceptible to a growing spectrum of security threats—including Sybil attacks, selfish mining, eclipse attacks, smart contract vulnerabilities, and routing-layer exploits [14]. This paper explores applications of graph theory for modeling blockchain networks to evaluate decentralization, security, privacy, scalability and NFT Mapping. We use graph metrics like degree distribution and betweenness centrality to quantify node connectivity, identify network bottlenecks, trace asset flows and detect communities.Traditional security evaluation methodologies, largely inherited from centralised network security assessment, fail to capture the topological, temporal, and game-theoretic complexity inherent to blockchain architectures. This paper proposes the Blockchain Security Evaluation Framework using Graph Models (BSEG), a structured five-component framework that applies graph-theoretic formalisms to model, analyse, and evaluate security properties across the full blockchain protocol stack. The proposed framework integrates transaction graph analysis, peer-to-peer network topology modelling, smart contract dependency graphs, consensus mechanism simulation through game-theoretic overlays, and a temporal anomaly detection layer that identifies structural deviations indicative of adversarial behaviour. A formal pseudo-algorithm details the core vulnerability propagation scoring pipeline, and two architectural diagrams illustrate the framework's layered structure and end-to-end evaluation workflow. The framework is evaluated against fifteen representative studies spanning blockchain se- curity, graph-based anomaly detection, smart contract analysis, and distributed system resilience. Key contributions include a unified graph-model vocabulary applicable across heterogeneous blockchain platforms, a scoring function that integrates structural centrality, edge entropy, and consensus devi- ation metrics into a composite security index, and a governance model that accommodates privacy- preserving audit mechanisms. Critical challenges including computational scalability over billion- edge transaction graphs, adversarial graph poisoning, and cross-chain interoperability are discussed alongside directions for empirical validation. This work provides a replicable architectural blueprint for security auditors, protocol designers, and researchers seeking to apply rigorous graph-theoretic methods to blockchain security assessment.

Keywords: blockchain security; graph theory; graph models; smart contract vulnerability; consensus mechanisms; anomaly detection; transaction graph analysis; peer-to-peer networks; temporal graph analysis; security evaluation framework

1. Introduction

Blockchain technology has undergone a remarkable trans- formation from its origins as the settlement layer for Bit- coin to a general-purpose distributed computation substrate supporting smart contracts, decentralised autonomous or- ganisations, non-fungible token ecosystems, and enterprise consortium networks. This proliferation of use cases has substantially expanded the attack

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

surface associated with blockchain systems. As these systems grow in complexity, analyzing their topological structure and vulnerabilities re- quires robust mathematical frameworks. The security of a blockchain system is not a monolithic property but a mul- tidimensional one, spanning the cryptographic primitives underlying digital signatures and hashing, the consensus protocol governing agreement among participants, the peer- to-peer networking layer through which messages propagate, the smart contract execution environment, and the economic incentive structures that constrain rational participant be- haviour. Graph theory provides an exceptionally well-suited mathematical language for this multidimensional analysis. Blockchain systems are, at their structural core, graphs: the transaction ledger is a directed acyclic graph of in- puts and outputs; the peer network is an undirected or directed communication graph; smart contract calls form a directed call graph; and the evolution of consensus state can be modelled as a temporal graph where vertices repre- sent protocol states and edges represent valid state transi- tions. Graph-theoretic properties—centrality, connectivity, clustering coefficients, spectral gap, entropy—translate di- rectly into security-relevant quantities. High degree central- ity in the transaction graph identifies hub addresses that may represent exchange wallets, mixing services, or high- value targets. Low spectral gap in the peer network indicates vulnerability to network partitioning. High edge entropy in the call graph suggests unpredictable smart contract exe- cution paths that complicate formal verification. Despite the intuitive alignment between blockchain security and graph theory, existing security evaluation approaches remain fragmented. Most work addresses individual attack vectors or individual layers in isolation. A comprehensive, archi- tecturally grounded framework that applies graph-theoretic methods coherently across all layers of the blockchain pro- tocol stack does not currently exist in the literature. This gap is consequential: without such a framework, security auditors must synthesise methods from disparate research traditions, practitioners lack a common vocabulary for com- municating findings across layers, and researchers cannot make systematic comparative assessments of blockchain security postures. This paper addresses this gap by proposing the Blockchain Security Evaluation Framework using Graph Models (BSEG): a five-component architectural framework that applies graph-theoretic analysis to the transaction layer, network layer, smart contract layer, consensus layer, and temporal anomaly layer of blockchain systems. The frame- work is grounded in a systematic review of fifteen repre- sentative papers spanning blockchain security, graph-based anomaly detection, smart contract analysis, network topol- ogy analysis, and distributed system resilience. A formal pseudo-algorithm specifies the core vulnerability propaga- tion scoring pipeline. Two architectural diagrams illustrate the framework's structure and evaluation workflow. The paper also identifies open research challenges and directions for future empirical validation. The organisation of this paper is as follows. Section 2 reviews the relevant literature across the five thematic areas addressed by the framework. Section 3 presents the proposed BSEG framework in detail, including layer descriptions, workflow, algorithm, and eth- ical safeguards. Section 4 concludes with implications for practice and directions for future research.

2. Literature Review

2.1. Transaction Graph Analysis and Blockchain Forensics

Blockchain transaction histories are commonly modeled as directed graphs for forensic analysis. Reid and Harrigan [1] showed that graph clustering and flow analysis can link multiple addresses to a single entity using the common- input heuristic, establishing the foundation for transaction graph analysis. Meiklejohn et al. [2] further demonstrated that interaction with known services enables accurate entity identification and revealed that a small number of high- degree nodes control significant transaction volume. These power-law degree distributions indicate that targeting central nodes can severely impact transaction flow, highlighting a key structural vulnerability.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

2.2. Peer-to-Peer Network Topology and Eclipse Attacks

Blockchain peer-to-peer networks govern block and trans- action propagation. Decker and Wattenhofer [3] showed that propagation delays create windows of vulnerability that can be exploited for attacks such as double spending and selfish mining. Heilman et al. [4] introduced eclipse attacks, where an adversary isolates a node by controlling its peer connections, enabling censorship and manipulation. Their defenses—such as increasing peer diversity and randomiz- ing connections—are inherently graph-based and motivate topology analysis in security evaluation frameworks.

2.3. Smart Contract Vulnerability Analysis

Smart contracts have introduced significant security risks, with major incidents such as the DAO and Parity exploits. Luu et al. [5] developed OYENTE, a symbolic execution tool that detects vulnerabilities like reentrancy and timestamp dependence. Tsankov et al. [6] proposed SECURIFY, which uses dataflow and dependency graphs to analyze contract behavior. Graph-based representations improve scalability and interpretability, supporting dependency graph modeling for smart contract security analysis.

2.4. Consensus Mechanism Security and Game-Theoretic Analysis

Consensus protocols ensure agreement in blockchain sys- tems. Eyal and Sirer [7] demonstrated the selfish mining attack, showing that rational miners can gain disproportion- ate rewards by deviating from honest behavior. Garay et al. [8] provided formal guarantees for consistency and liveness in proof-of-work systems, defining adversarial thresholds. These works underpin graph-based modeling of consensus as state transition systems for evaluating stability and adver- sarial resilience.

2.5. Graph-Based Anomaly Detection in Distributed Systems

Graph-based anomaly detection methods are widely used in distributed systems. Akoglu et al. [9] categorized anomaly types and detection techniques including spectral meth- ods, community detection, and deep learning. Chen et al. [10] applied graph neural networks to control flow graphs for vulnerability detection, achieving strong performance. These approaches support temporal anomaly detection in blockchain systems using evolving graph structures and learned representations.

2.6. Routing Layer Attacks and Network Security

Blockchain networks are also vulnerable to routing-layer attacks. Apostolaki et al. [11] showed that BGP hijacking can partition the Bitcoin network and redirect traffic, enabling targeted attacks. Their findings highlight the importance of Internet-level topology, where a small number of au- tonomous systems can influence large portions of traffic. This motivates incorporating routing graphs into network- level security analysis.

3. Proposed Frameworks

3.1. Framework Overview

The Blockchain Security Evaluation Framework using Graph Models (BSEG) is a five-component architectural framework designed to provide systematic, graph-theoretic security evaluation across the full protocol stack of any blockchain system. The framework is premised on four de- sign principles: comprehensiveness—addressing all major

attack surfaces from the cryptographic transaction layer through the network and consensus layers to the smart contract execution environment; formalism—grounding all security metrics in mathematically defined graph prop- erties with well-understood interpretations; composabil- ity—

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

enabling modular deployment of individual compo- nents against specific evaluation objectives without requir- ing full-stack implementation; and transparency—producing interpretable, auditable security scores whose derivation can be explained to non-specialist stakeholders including regulators, auditors, and protocol governance bodies.

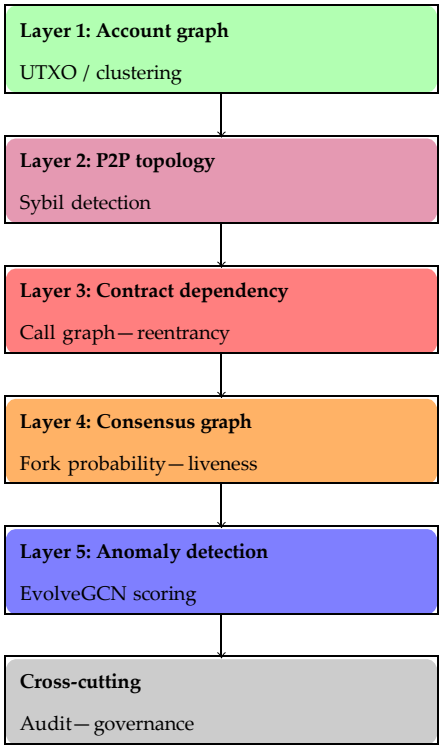

Figure 1.

The five layers of BSEG are the Transaction Graph Component, the Peer Network Topology Component, the Smart Contract Dependency Component, the Consensus State Graph Component, and the Temporal Anomaly Detec- tion Component. Each component exposes a standardised interface that accepts a graph representation of its target subsystem and returns a set of security metrics conforming to the BSEG Composite Security Index schema. A cross- cutting governance layer provides privacy-preserving audit capabilities, role-based access control, and model gover- nance services to all components.

3.2. Component Descriptions

The Transaction Graph Component models the transac- tion ledger as a directed graph where nodes represent ad- dresses and edges represent transactions. It computes degree distribution, clustering coefficients, centrality measures, and transaction flow entropy to detect anomalies. Metrics are normalized to [0,1] and contribute to the overall security index.

The Peer Network Topology Component analyzes the network graph of peer connections. It evaluates spectral gap, clustering coefficient, community structure, and between- ness centrality to assess robustness, detect Sybil regions, and identify critical nodes. External routing data can be incorporated to assess network partition risks.

The Smart Contract Dependency Component constructs call graphs, state dependency graphs, and execution flow graphs from contract code. It identifies reentrancy vulner- abilities, privilege escalation paths, and cascading risks, while also detecting flash loan attack surfaces through de- pendency analysis.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

The Consensus State Graph Component models the con- sensus protocol as a state transition graph. It evaluates selfish mining strategies, fork probabilities, and liveness properties, and applies game-theoretic analysis to identify conditions where participants may deviate from honest behavior.

The Temporal Anomaly Detection Component uses tem- poral graph models to identify structural anomalies across all lower components. It maintains a baseline of graph proper- ties such as degree distribution, clustering, community struc- ture, and spectral features, and computes deviation scores for new graph snapshots. An EvolveGCN-based graph neural network classifies deviations as normal or anomalous us- ing historical data. Alert thresholds are configurable, and the model supports automated retraining when performance drops below defined levels.

3.3. Evaluation Workflow

The end-to-end evaluation workflow of the BSEG frame- work proceeds through six phases, as illustrated in the dia- gram below.

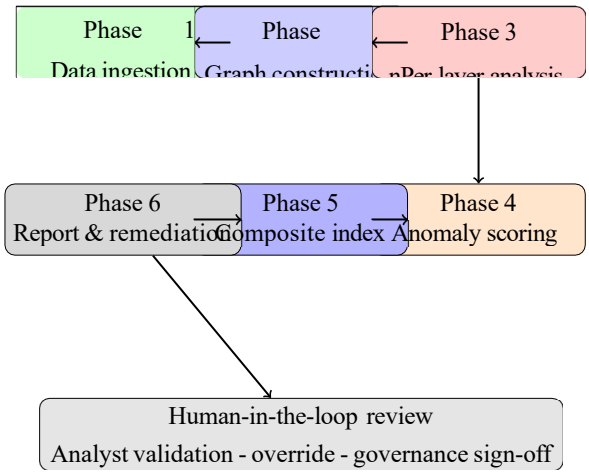

Figure 2.

Phase 1 involves data ingestion, where raw blockchain data (transactions, peer tables, smart contract bytecode, and consensus logs) is collected and normalized into standard graph formats. Phase 2 focuses on graph construction, converting this data into the five graph representations used by BSEG. Phase 3 performs per-layer analysis, where each component extracts security metrics from its respective graph. Phase 4 applies anomaly scoring by comparing current graph snapshots with historical baselines to compute deviation scores. Phase 5 computes the composite security index by aggregating component metrics using a weighted scoring function. Phase 6 generates reports with security assessments and remediation recommendations, while a feedback loop updates the anomaly detection model for continuous improvement.

3.4. Core Algorithm: Vulnerability Propagation Scoring

The following pseudo-algorithm formalises the core vul- nerability propagation scoring computation executed across the BSEG component stack. The algorithm accepts a blockchain system descriptor as input and produces a composite security index and component-level vulnerability vectors as output.

Algorithm 1: BSEG Vulnerability Propagation Scoring

Input: Blockchain system descriptor S = {ledger_data, \ \ peer_snapshot,

contract_bytecodes, consensus_log, temporal_window} Output: Composite Security Index CSI [0,1],

Component vectors V = {V_tx, V_net, V_sc, V_cons, V_temp}

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1


```
1: G_tx ← ConstructTransactionGraph(S.ledger_data) 2: G_net ← ConstructPeerGraph(S.peer_snapshot)
3: G_sc ← ConstructContractCallGraph(S.contract_bytecodes) 4: G_cons ←
ConstructConsensusStateGraph(S.consensus_log)

5: // Layer 1: Transaction graph metrics
6: V_tx.centrality ← ComputeDegreeCentrality(G_tx) 7: V_tx.entropy ← ComputeEdgeEntropy(G_tx)
8: V_tx.clustering ← ComputeClusteringCoefficient(G_tx)

9: // Layer 2: Network topology metrics
10: V_net.spectral_gap ← ComputeSpectralGap(Laplacian(G_net)) 11: V_net.sybil_score ← DetectSybilCommunities(G_net)
12: V_net.partition_risk ← ComputePartitionProbability\ \
(G_net, BGP_graph)

13: // Layer 3: Smart contract dependency metrics
14: V_sc.reentrancy_paths ← DetectCycles(G_sc, type=REENTRANCY)
15: for each contract c in G_sc do
16: V_sc.cascade_score(c) ← ComputeReachability(G_sc, source=c)
17: end for

18: // Layer 4: Consensus state metrics
19: V_cons.fork_prob ← SimulateForkProbability\ \
(G_cons, adversary_fraction)
20: V_cons.selfish_gain ← ComputeSelfishMiningPayoff(G_cons)
21: if V_cons.selfish_gain > RATIONAL_THRESHOLD then 22: FlagForConsensusReview(S)
23: end if

24: // Layer 5: Temporal anomaly scoring
25: baseline ← LoadHistoricalBaseline(S.temporal_window) 26: V_temp.deviation ← EvolveGCN_Score(G_tx, G_net, G_sc, \ \
baseline)
27: if V_temp.deviation > ALERT_THRESHOLD then 28: TriggerAnalystAlert(V_temp)
29: end if

30: // Composite index computation
31: CSI ← WeightedAggregate(V_tx, V_net, V_sc, V_cons, V_temp,
weights=GovernanceConfig.weights) 32: NormalisedCSI ← Normalise(CSI, range=[0,1])

33: StoreInAuditLog(NormalisedCSI, V, S.id) 34: return NormalisedCSI, V
```

4. Conclusions

This paper presented BSEG, a five-component frame- work for blockchain security evaluation using graph models. The framework integrates transaction graph analysis, peer network topology evaluation, smart contract dependency analysis, consensus state graph simulation, and temporal anomaly detection into a unified architecture for systematic and reproducible security assessment. Based on a review of representative studies in blockchain security, graph-based analysis, and distributed systems, BSEG consolidates exist- ing knowledge into a practical evaluation framework appli- cable across diverse blockchain platforms.

A key contribution is the introduction of a unified graph- model representation across all layers of the blockchain stack. Unlike prior approaches that focus on isolated compo- nents or attack vectors,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

BSEG models each layer—transaction, peer network, contract, and consensus—as interconnected graph structures. This enables both detailed layer-wise anal- ysis and cross-layer security evaluation within a single framework.

The pseudo-algorithm defined in Section 3.4 formalizes the vulnerability propagation and scoring process, enabling reproducibility and practical implementation. The compos- ite security index aggregates component-level metrics into a single score while retaining detailed insights for targeted re- mediation. The modular design also supports flexible adop- tion, allowing organizations to integrate selected compo- nents based on their requirements.

However, several challenges remain. The framework has not yet been validated against real-world incidents, and future work should evaluate its effectiveness using historical attacks such as the DAO hack and major DeFi exploits. Scalability is another concern, as large blockchain networks require efficient or distributed graph processing techniques. Additionally, adversarial robustness must be addressed, as attackers may manipulate graph structures to evade detec- tion. Cross-chain interactions present further complexity, requiring extensions to model inter-chain dependencies.

Accessibility and trust are also critical considerations. While BSEG emphasizes interpretable outputs, effective adoption depends on user-friendly tools, clear documenta- tion, and transparent governance. Engagement with stake- holders and accountability in deployment are essential for building confidence in the framework.

Future work can extend BSEG through integration with advanced technologies. Large language models may support automated report generation, zero-knowledge proofs can enable privacy-preserving evaluation, and causal inference techniques may improve understanding of attack mecha-nisms. Overall, BSEG establishes a strong foundation for comprehensive and ethically grounded blockchain security evaluation using graph-based methods.

References

1. F. Reid, M. Harrigan, An analysis of anonymity in the bitcoin system, in: Security and Privacy in Social Networks, Springer, 2013, pp. 197– 223.
2. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy,
3. G. M. Voelker, S. Savage, A fistful of bitcoins: Characterising pay- ments among men with no names, in: Proceedings of the 2013 Internet Measurement Conference, ACM, 2013, pp. 127–140.
4. C. Decker, R. Wattenhofer, Information propagation in the bitcoin net- work, in: IEEE International Conference on Peer-to-Peer Computing, IEEE, 2013, pp. 1–10.
5. E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network, in: USENIX Security Symposium, USENIX, 2015, pp. 129–144.
6. L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, in: Proceedings of the 2016 ACM SIGSAC Con- ference on Computer and Communications Security, ACM, 2016, pp. 254–269.
7. P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli,
8. M. Vechev, Securify: Practical security analysis of smart contracts, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 67–82.
9. I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulner- able, in: Financial Cryptography and Data Security, Springer, 2014,
10. pp. 436–454.
11. J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: Analysis and applications, in: Advances in Cryptology—EURO- CRYPT 2015, Springer, 2015, pp. 281–310.
12. L. Akoglu, H. Tong, D. Koutra, Graph-based anomaly detection and description: A survey, Data Mining and Knowledge Discovery 29 (3) (2015) 626–688.
13. Z. Chen, Y. Cao, S. Wu, J. Fang, M. Zhang, Defining smart contract defects on ethereum, IEEE Transactions on Software Engineering 48 (1) (2022) 327–345.
14. M. Apostolaki, A. Zohar, L. Vanbever, Hijacking bitcoin: Routing attacks on cryptocurrencies, in: 2017 IEEE Symposium on Security and Privacy, IEEE, 2017, pp. 375–392.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 April 2026 doi:10.20944/preprints202604.0637.v1

15. K. Qin, L. Zhou, A. Gervais, Quantifying blockchain extractable value: How dark is the forest?, in: 2022 IEEE Symposium on Security and Privacy, IEEE, 2022, pp. 198–214.
16. J. R. Douceur, The sybil attack, in: International Workshop on Peer- to-Peer Systems, Springer, 2002, pp. 251–260.
17. G. Jayabalasamy, C. Pujol, K. L. Bhaskaran, Application of graph theory for blockchain technologies, Mathematics 12 (2024) 1133. doi:10.3390/math12081133.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.