

Blockchain-Enabled Secure Data Sharing for Internet of Things Networks: A Decentralized Trust and Access Control Framework

Dr. Neeraj Mehta¹

Research Scholar¹

Department of Computer Science and Engineering
Institute of Advanced Technology, Jaipur, Rajasthan, India
neeraj.mehta@iatjaipur.ac.in

Abstract

The rapid expansion of the Internet of Things (IoT) has resulted in the deployment of billions of interconnected devices across smart homes, healthcare, transportation, agriculture, manufacturing, energy systems, and urban infrastructure. These devices continuously generate and exchange information, creating significant opportunities for intelligent automation and data-driven decision-making. However, the highly distributed nature of IoT networks introduces serious challenges related to data security, device authentication, access control, trust management, and centralized points of failure. Conventional IoT architectures frequently depend on centralized servers to authenticate devices and manage data access. Such architectures can become difficult to scale and may create attractive targets for attackers. Blockchain technology provides a distributed mechanism for maintaining tamper-resistant records and establishing trust among entities that may not completely trust one another. This paper proposes a conceptual blockchain-enabled framework for secure IoT data sharing. The proposed architecture combines IoT devices, edge gateways, blockchain-based identity management, smart-contract-driven access control, encrypted off-chain storage, and reputation-based trust evaluation. To avoid storing large IoT datasets directly on a blockchain, the framework stores sensitive data in controlled off-chain repositories while recording integrity-related metadata on the distributed ledger. A lightweight access-control mechanism is proposed to reduce the computational burden on constrained IoT devices. The paper also examines important challenges including blockchain scalability, transaction latency, storage growth, consensus overhead, privacy leakage, key management, compromised devices, and interoperability. A conceptual security model and performance evaluation methodology are presented using metrics such as authentication latency, transaction throughput, storage overhead, energy consumption, access-control accuracy, and attack detection capability. The

study concludes that blockchain can strengthen trust and accountability in IoT data-sharing environments, but practical deployment requires lightweight consensus mechanisms, edge-assisted processing, privacy-aware ledger design, efficient identity management, and careful separation between on-chain metadata and off-chain sensitive information.

Keywords: Internet of Things, Blockchain, Data Sharing, Access Control, Smart Contracts, IoT Security, Decentralized Identity, Edge Computing, Trust Management, Distributed Ledger

1. Introduction

The Internet of Things has changed the way physical objects communicate and interact with digital systems. Sensors, embedded controllers, smart appliances, industrial machines, vehicles, wearable devices, and environmental monitoring systems can now exchange information through wired and wireless networks.

An IoT environment can be represented as:

IoT=Devices+Networks+Data+Applications

IoT devices continuously generate information.

For NN devices, total data generation can be expressed as:

$$D_{total} = \sum_{i=1}^N D_i$$

As the number of connected devices increases, the amount of information exchanged among different organizations and applications also increases.

This creates an important requirement for secure data sharing.

A traditional architecture may use:

Device → Central Server → Application

The central server performs:

- Authentication

- Authorization
- Data management
- Logging
- Policy enforcement

Although centralized architectures are relatively straightforward to implement, they introduce a central trust point.

If the central server is compromised:

Securitysystem↓Security_{system}↓

Furthermore, organizations participating in an IoT ecosystem may not completely trust a single authority.

Blockchain introduces a distributed ledger in which transactions can be recorded and verified by multiple participating entities.

A simplified blockchain-based architecture is:

IoT→EdgeGateway→BlockchainIoT\rightarrow EdgeGateway\rightarrow Blockchain

The blockchain can maintain:

- Device identities
- Access permissions
- Transaction records
- Data hashes
- Policy information
- Audit information

The actual IoT data can remain outside the blockchain.

2. Motivation

IoT security is complicated by several characteristics of IoT environments.

These include:

1. Large numbers of devices.
2. Heterogeneous hardware.
3. Limited device resources.
4. Distributed ownership.
5. Continuous data generation.
6. Different communication protocols.
7. Long device lifetimes.
8. Physically exposed devices.

A single IoT ecosystem may contain devices manufactured by different organizations.

Consequently:

$\text{Trust}(\text{DeviceA}, \text{DeviceB}) \text{Trust}(\text{Device_A}, \text{Device_B})$

cannot always be assumed.

A decentralized trust mechanism can therefore be useful.

Blockchain provides an immutable transaction history in which participants can independently verify recorded events.

The fundamental idea of the proposed system is:

$\text{IoTData} + \text{BlockchainTrust} \rightarrow \text{SecureDataSharing}$

3. Problem Statement

The primary problem addressed by this research is:

How can blockchain technology be integrated with IoT networks to provide secure, decentralized, privacy-aware, and scalable data sharing while minimizing computational, storage, and communication overhead?

A practical system must simultaneously address:

- Device authentication
- Access control
- Data integrity
- Privacy
- Trust
- Scalability
- Performance
- Resource limitations

A security mechanism that is highly secure but computationally expensive may not be suitable for constrained IoT devices.

Therefore:

$Security \neq Only Protection$

Instead:

$Effective IoT Security = Security + Efficiency + Scalability$

4. Research Objectives

The objectives of this research are:

1. To investigate blockchain technology for IoT security.
2. To analyze security requirements for IoT data sharing.
3. To propose a blockchain-based IoT architecture.
4. To design decentralized device authentication.
5. To introduce smart-contract-based access control.
6. To separate sensitive data from blockchain metadata.
7. To investigate trust management.
8. To analyze scalability challenges.
9. To define performance metrics for evaluation.

5. Research Contributions

This paper proposes:

- A conceptual blockchain-IoT architecture.
- Decentralized identity management.
- Smart-contract-based authorization.
- Off-chain encrypted data storage.
- Blockchain-based integrity verification.
- A lightweight trust-management mechanism.
- Edge-assisted blockchain processing.
- A comprehensive performance evaluation framework.

6. Related Conceptual Technologies

The proposed framework combines four major technologies:

IoT+Blockchain+EdgeComputing+SmartContractsIoT+Blockchain+EdgeComputing+SmartContract
s

Each technology has a specific role.

IoT

Generates and consumes data.

Edge Computing

Processes information closer to IoT devices.

Blockchain

Maintains distributed trust and transaction records.

Smart Contracts

Automatically enforce predefined access policies.

7. Proposed System Architecture

The proposed architecture consists of six layers.

Layer 1: IoT Device Layer

Contains:

- Sensors
- Actuators
- Cameras
- Smart meters
- Wearable devices
- Embedded controllers

Layer 2: Edge Gateway Layer

Performs:

- Data aggregation
- Protocol conversion
- Preliminary processing
- Device verification
- Communication optimization

Layer 3: Security Layer

Provides:

- Authentication
- Encryption

- Key management
- Intrusion detection

Layer 4: Blockchain Layer

Maintains:

- Device identities
- Access transactions
- Data hashes
- Trust information

Layer 5: Smart Contract Layer

Controls:

- Data access
- Permission management
- Policy enforcement
- Authorization decisions

Layer 6: Application Layer

Provides IoT services to authorized users.

The overall architecture can be represented as:

IoTDevices→EdgeGateway→Security→Blockchain→SmartContracts→Applications
IoTDevices \rightharpoonup EdgeGateway \rightharpoonup Security \rightharpoonup Blockchain \rightharpoonup SmartContracts
Applications

8. Device Registration

Before an IoT device participates in the network, it must be registered.

Each device is assigned an identity:

ID_i

and cryptographic credentials:

$KeyPair_i = (K_{pub}, K_{priv})$

The device registration process is:

Device $\rightarrow$ IdentityVerification $\rightarrow$ CredentialGeneration $\rightarrow$ BlockchainRegistration
IdentityVerification $\rightarrow$ CredentialGeneration $\rightarrow$ BlockchainRegistration

The blockchain stores appropriate identity metadata rather than sensitive device information.

9. Decentralized Identity

Traditional IoT systems may maintain device identities in a centralized database.

The proposed framework distributes identity verification across participating blockchain nodes.

A simplified identity record may contain:

$Identity_i = (ID_i, Status_i, PublicKey_i, Policy_i)$

The device's private key remains under its control.

This reduces dependence on a single centralized identity database.

10. Device Authentication

When a device requests access, it must prove its identity.

A challenge-response mechanism can be used.

The verifier generates:

Challenge=CChallenge=C

The device signs it:

Signature=Sign(Kpriv,C)Signature=Sign(K_{priv},C)

The verifier checks:

Verify(Kpub,C,Signature)Verify(K_{pub},C,Signature)

If verification succeeds:

Authentication=TrueAuthentication=True

Otherwise:

Authentication=FalseAuthentication=False

11. Access Control

Authentication determines who a device or user is.

Access control determines what that entity can access.

A permission can be represented as:

Permission=(Subject,Object,Action,Condition)Permission=(Subject,Object,Action,Condition)

where:

- Subject = requesting entity
- Object = requested data
- Action = requested operation
- Condition = policy requirement

For example:

DeviceA→Read→SensorDataBDevice_A→Read→SensorData_B

may be allowed only when:

Role(DeviceA)=AuthorizedServiceRole(Device_A)=AuthorizedService

12. Smart Contracts

Smart contracts are programmable rules recorded on the blockchain.

A conceptual access policy can be represented as:

Policy: If Authorized(User) ∧ Valid(Request) ⇒ Access=True
Policy: If ¬ Authorized(User) ∧ Valid(Request) ⇒ Access=False

Otherwise:

Access=False

Smart contracts can automatically enforce access policies without requiring manual approval for every transaction.

13. Data Storage Strategy

Large IoT datasets are unsuitable for direct storage on many blockchain implementations.

Therefore, the proposed framework uses hybrid storage.

On-Chain

Stores:

- Data hash
- Metadata
- Access policy



- Transaction record
- Device identity reference

Off-Chain

Stores:

- Sensor data
- Images
- Videos
- Large files
- Detailed records

The architecture becomes:

Data → EncryptedOffChainStorageData → EncryptedOffChainStorage

while:

Hash(Data) → BlockchainHash(Data) → Blockchain

14. Data Integrity

Suppose the original data is DD.

A cryptographic hash is calculated:

$H = \text{Hash}(D)$

The hash is recorded on the blockchain.

When the data is retrieved later:

$H' = \text{Hash}(D')$

The system verifies:

$H' = HH' = H$

If:

$H' \neq HH' \neq H$

the data may have been modified.

This provides a mechanism for detecting unauthorized modification.

15. Data Encryption

Sensitive IoT information should be encrypted before being stored.

Let:

$C = \text{Encrypt}(K, D)$

where:

- DD = original data
- KK = encryption key
- CC = encrypted data

Only authorized entities possessing the appropriate key should be able to decrypt the information.

16. Key Management

Cryptographic keys are fundamental to the proposed architecture.

A secure key-management system should address:

- Key generation
- Key distribution
- Key rotation
- Key revocation

- Key recovery

If a private key is compromised:

Security(Device)↓Security(Device)\downarrow

Therefore, the blockchain system should support rapid credential revocation.

17. Access Request Workflow

The proposed access process consists of the following steps.

Step 1

A user or device sends an access request.

Step 2

The identity is verified.

Step 3

The smart contract checks the access policy.

Step 4

The system evaluates the request context.

Step 5

Access is approved or rejected.

Step 6

The transaction is recorded.

Step 7

Authorized data is retrieved from off-chain storage.

The workflow is:

Request→Authentication→Authorization→DataAccessRequest \rightarrow Authentication
 \rightarrow Authorization \rightarrow DataAccess

18. Trust Management

Blockchain provides transaction history, but transaction history alone does not necessarily establish whether a device is trustworthy.

The proposed framework introduces a device trust score.

Let:

- BB = historical behavior
- AA = authentication reliability
- VV = transaction validity
- SS = security status

Then:

$$Trust_i = w_1B + w_2A + w_3V + w_4S$$

where:

$$\sum w_i = 1$$

A device with repeated abnormal activity can receive a lower trust value.

19. Reputation Management

Reputation can evolve over time.

Let:

$R_i(t)$

represent the reputation of device i at time t .

After valid behavior:

$R_i(t+1) > R_i(t)$

After suspicious behavior:

$R_i(t+1) < R_i(t)$

The exact update mechanism should be designed according to the application's security requirements.

20. Malicious Device Detection

A compromised IoT device may generate abnormal traffic or access patterns.

Possible indicators include:

- Excessive requests
- Unexpected destinations
- Unusual communication frequency
- Invalid credentials
- Repeated access violations

An anomaly score can be calculated:

$A_i = f(\text{Traffic}, \text{Behavior}, \text{AccessPattern})$

If:

$A_i > \tau A_i > \tau$

the device may be classified as suspicious.

21. Blockchain-Based Audit Trail

Every significant security event can generate a transaction.

Examples include:

- Device registration
- Authentication
- Access request
- Permission modification
- Data retrieval
- Device revocation

The resulting history can be represented as:

$T = \{t_1, t_2, \dots, t_n\}$

This creates an auditable record of system activities.

22. Privacy Considerations

Although blockchain can improve integrity and accountability, its transparency characteristics can create privacy concerns.

If sensitive information is permanently recorded on a public ledger, privacy may be compromised.

Therefore:

$SensitiveData \rightarrow \neg PublicBlockchain$

The proposed framework stores sensitive information off-chain and records only necessary metadata.

This follows the principle:

MinimizeOnChainDataMinimizeOnChainData

23. Edge-Assisted Blockchain

Blockchain operations may be too demanding for highly constrained IoT devices.

Edge gateways can therefore act as intermediaries.

Instead of:

IoTDevice→BlockchainIoTDevice\rightarrow Blockchain

the proposed architecture uses:

IoTDevice→EdgeGateway→BlockchainIoTDevice\rightarrow EdgeGateway\rightarrow Blockchain

The gateway can perform:

- Transaction aggregation
- Signature verification
- Data filtering
- Protocol conversion
- Temporary caching

24. Lightweight Blockchain

A conventional consensus mechanism may require substantial computational resources.

IoT systems therefore require lightweight distributed consensus.

Possible approaches include:

- Proof of Authority
- Practical Byzantine Fault Tolerance
- Delegated mechanisms
- Consortium blockchain consensus

The appropriate choice depends on:

Security+Latency+Energy+Scalability Security+Latency+Energy+Scalability

25. Consortium Blockchain

The proposed architecture is particularly suitable for consortium environments.

Participants may include:

- Manufacturers
- Service providers
- Municipal organizations
- Hospitals
- Utility providers
- Industrial organizations

Only authorized organizations operate blockchain nodes.

This can provide greater control than a completely open blockchain.

26. Consensus and IoT Performance

Let:

$$T_{\text{transaction}} = T_{\text{validation}} + T_{\text{consensus}} + T_{\text{confirmation}} \quad T_{\text{transaction}} = T_{\text{validation}} + T_{\text{consensus}} + T_{\text{confirmation}}$$

If consensus becomes expensive:

$$T_{\text{transaction}} \uparrow \quad T_{\text{transaction}} \uparrow$$

For real-time IoT applications, excessive transaction latency may be unacceptable.

Therefore, blockchain should primarily manage events requiring distributed trust rather than every individual sensor reading.

27. Transaction Optimization

Suppose an IoT device generates:

10,00010,000

sensor readings per hour.

Recording every reading directly on a blockchain would create unnecessary overhead.

Instead, the system can aggregate data.

For example:

Data_{hour} → Hash(Data_{hour}) Data_{hour} → Hash(Data_{hour})

Only the aggregate hash can be recorded.

This can significantly reduce transaction volume.

28. Data Aggregation

The edge gateway can combine multiple measurements.

Suppose:

$D = \{d_1, d_2, \dots, d_n\}$ $D = \{d_1, d_2, \dots, d_n\}$

The gateway generates:

$D_{agg} = \text{Aggregate}(D)$ $D_{\{agg\}} = \text{Aggregate}(D)$

Then:

$H = \text{Hash}(\text{Dagg})$

The blockchain records HH.

This approach reduces blockchain storage requirements.

29. Secure Data Sharing

Suppose organization A owns data DAD_A.

Organization B requests access.

The process is:

$B \rightarrow \text{AccessRequest}(DA) \rightarrow \text{AccessRequest}(D_A)$

The smart contract verifies:

$\text{Policy}(B, DA)$

If permitted:

$\text{Access} = \text{True}$

The encrypted data can then be retrieved from off-chain storage.

30. Conditional Access

Access policies can depend on multiple conditions.

For example:

$\text{Access} = \text{Identity} \wedge \text{Role} \wedge \text{Time} \wedge \text{DeviceStatus}$

A user may be authorized only during a specific period.

Similarly, a device may be denied if its security status becomes invalid.

31. Dynamic Access Control

IoT environments change continuously.

Therefore, access control should not always remain static.

Let:

$Policy(t)=f(Context(t))$

Context may include:

- Device location
- Security status
- Network state
- User role
- Time
- Trust score

Dynamic policies can respond to changing conditions.

32. Revocation

If a device becomes compromised:

$Status(Device)=Revoked$

The blockchain can record the revocation event.

Subsequent requests can be rejected:

$Status=Revoked \Rightarrow Access=False$

This provides an auditable mechanism for device lifecycle management.

33. Device Lifecycle Management

IoT devices generally pass through several stages:

Registration→Activation→Operation→Update→Suspension→Revocation
Registration \rightarrow Activation \rightarrow Operation \rightarrow Update \rightarrow Suspension \rightarrow Revocation

The proposed blockchain can maintain lifecycle metadata.

34. Firmware Integrity

Firmware modification can compromise an IoT device.

Suppose the approved firmware hash is:

H_{FH_F}

The device reports:

$H_{F'H'_F}$

The system verifies:

$H_F = H_{FH'_F} = H_{F'}$

If the hashes differ, the device may require investigation.

35. Secure Software Updates

A blockchain record can maintain the authorized version of device firmware.

For example:

FirmwareVersion=4.2FirmwareVersion=4.2

A device reporting an unexpected version can be flagged.

This creates an auditable software-update history.

36. IoT Attack Scenarios

The proposed framework considers several threats.

36.1 Device Spoofing

An attacker attempts to impersonate a legitimate device.

36.2 Unauthorized Data Access

An unauthorized entity attempts to retrieve protected data.

36.3 Data Modification

Stored IoT data is altered.

36.4 Replay Attack

A previously valid authentication message is reused.

36.5 Sybil Attack

An attacker creates multiple false identities.

36.6 Denial of Service

The attacker attempts to make IoT services unavailable.

36.7 Malicious Insider

An authorized participant abuses legitimate privileges.

37. Defense Against Replay Attacks

A timestamp or nonce can be incorporated into authentication.

Let:

$\text{Request} = (\text{ID}, \text{C}, \text{Timestamp})$

The system checks whether the timestamp is valid.

If:

$|T_{\text{current}} - T_{\text{request}}| > \tau$

the request can be rejected.

38. Defense Against Unauthorized Access

Authorization can combine multiple security factors.

$\text{AccessDecision} = f(\text{Identity}, \text{Role}, \text{Trust}, \text{Context})$

This is stronger than relying on identity alone.

39. Defense Against Data Manipulation

Data integrity is verified using cryptographic hashes.

$D \rightarrow \text{Hash}(D)$

The blockchain preserves the reference hash.

Any modification creates:

$\text{Hash}(D') \neq \text{Hash}(D)$

allowing the system to detect possible tampering.

40. Defense Against Sybil Attacks

A permissioned blockchain can reduce Sybil risk by requiring authorized membership.

A new participant must satisfy:

$\text{IdentityVerification} = \text{True}$

before obtaining permission to participate.

41. Privacy-Preserving Blockchain

The proposed framework adopts several privacy principles.

Principle 1

Do not store raw sensitive data on-chain.

Principle 2

Store only essential metadata.

Principle 3

Encrypt off-chain information.

Principle 4

Use controlled blockchain membership.

Principle 5

Minimize unnecessary transaction information.

42. Performance Metrics

The proposed system should be evaluated using several metrics.

42.1 Authentication Latency

$$TA = T_{request} + T_{verification} \quad T_A = T_{\{request\}} + T_{\{verification\}}$$

42.2 Access Latency

$$T_{Access} = T_{Auth} + T_{Policy} + T_{Data} \quad T_{\{Access\}} = T_{\{Auth\}} + T_{\{Policy\}} + T_{\{Data\}}$$

42.3 Transaction Throughput

$$TPS = \frac{\text{Transactions}}{\text{Second}} \quad TPS = \frac{\{Transactions\}}{\{Second\}}$$

42.4 Storage Overhead

$$O_s = \frac{\text{BlockchainStorageTotalData}}{\text{TotalData}}$$

42.5 Energy Consumption

$$E = E_{\text{Device}} + E_{\text{Gateway}} + E_{\text{Blockchain}}$$

42.6 Availability

$$\text{Availability} = \frac{\text{Uptime}}{\text{TotalTime}}$$

43. Security Evaluation

Security evaluation should measure:

- Unauthorized-access prevention
- Authentication success rate
- Attack detection rate
- False-positive rate
- Data-integrity verification
- Device revocation effectiveness

A detection system can be evaluated using:

$$\text{DetectionRate} = \frac{\text{DetectedAttacks}}{\text{TotalAttacks}}$$

and:

$$\text{FalsePositiveRate} = \frac{\text{FalseAlerts}}{\text{TotalNormalEvents}}$$

44. Comparative Analysis

Feature	Centralized IoT	Traditional Distributed IoT	Proposed Blockchain IoT
Central Authority	High	Moderate	Reduced
Data Integrity	Server dependent	Variable	Strong ledger-based verification
Auditability	Moderate	Variable	High
Device Authentication	Centralized	Distributed/varied	Decentralized
Access Control	Server based	Mixed	Smart-contract based
Tamper Evidence	Moderate	Variable	High
Scalability	High initially	Moderate	Requires optimization
Transaction Overhead	Low	Moderate	Higher
Privacy	Depends on provider	Variable	Requires careful ledger design

45. Proposed Experimental Methodology

A future implementation can create an IoT test environment containing multiple device categories.

For example:

Devices=Sensors+Actuators+Gateways+Applications
 Devices=Sensors+Actuators+Gateways+Applications

The system can be evaluated under different workloads.

Experiment 1

Normal device operation.

Experiment 2

High-frequency sensor transmission.

Experiment 3

Unauthorized access attempts.

Experiment 4

Compromised-device behavior.

Experiment 5

High blockchain transaction volume.

Experiment 6

Network delay.

Experiment 7

Node failure.

46. Experimental Comparison

The proposed architecture can be compared against:

1. Centralized authentication.
2. Conventional database-based access control.
3. Blockchain without edge processing.
4. Blockchain with edge-assisted processing.
5. Proposed adaptive architecture.

Metrics can include:

Latency, Throughput, Energy, Storage, Security

47. Expected Outcomes

The proposed architecture is expected to provide:

- Improved auditability.
- Stronger data-integrity verification.
- Reduced dependence on a single trust authority.
- More flexible access control.
- Better device lifecycle management.
- Improved transparency among participating organizations.

However, these represent expected architectural benefits rather than experimentally established numerical improvements.

48. Scalability Challenge

Blockchain scalability is one of the most important limitations.

Suppose:

$$N=1,000,000$$

IoT devices continuously generate transactions.

If every device generates r transactions per second:

$$TPS_{required} = N \times r$$

Even a small r can produce a very large transaction requirement.

Therefore, not every sensor event should be placed directly on-chain.

49. Hierarchical Architecture

A hierarchical architecture can improve scalability.

Devices→EdgeNodes→RegionalBlockchain→CloudDevices \rightarrow EdgeNodes \rightarrow RegionalBlockchain \rightarrow Cloud

Edge nodes aggregate local transactions.

Regional nodes handle blockchain operations.

The cloud performs large-scale analytics.

50. Edge-Blockchain Collaboration

The combination of edge computing and blockchain can distribute processing.

The edge can handle:

- Local filtering
- Authentication
- Aggregation
- Preliminary anomaly detection

Blockchain can handle:

- Trust
- Integrity
- Access policy
- Audit

Cloud infrastructure can handle:

- Long-term analytics
- Machine learning
- Large-scale storage

Thus:

Edge+Blockchain+CloudEdge+Blockchain+Cloud

creates a layered architecture.

51. Energy Efficiency

IoT devices often operate under energy constraints.

Therefore:

$$\text{Energy}_{\text{total}} = \text{Energy}_{\text{sensing}} + \text{Energy}_{\text{communication}} + \text{Energy}_{\text{processing}}$$
$$\text{Energy}_{\text{total}} = \text{Energy}_{\{\text{sensing}\}} + \text{Energy}_{\{\text{communication}\}} + \text{Energy}_{\{\text{processing}\}}$$

Blockchain operations can introduce additional computation and communication.

The proposed framework reduces device-level blockchain participation by using edge gateways.

52. Fault Tolerance

A decentralized blockchain network can potentially continue functioning when one participating node fails.

If:

$\text{Node}_1 = \text{Failure}$ $\text{Node}_{-1} = \text{Failure}$

other nodes can continue maintaining the ledger, depending on the consensus protocol and fault assumptions.

This can reduce dependence on a single infrastructure component.

53. Interoperability

IoT systems frequently use different communication protocols.

Examples include:

- MQTT
- CoAP
- HTTP
- Modbus
- Zigbee
- Bluetooth

The edge gateway can provide protocol translation.

ProtocolA → Gateway → ProtocolB Protocol_A \rightarrow Gateway \rightarrow Protocol_B

This helps integrate heterogeneous devices.

54. Smart Contract Security

Smart contracts themselves can contain vulnerabilities.

Possible issues include:

- Incorrect authorization rules
- Logic errors
- Improper input validation
- Key-management weaknesses

Therefore:

BlockchainSecurity ≠ SmartContractSecurity BlockchainSecurity \neq SmartContractSecurity

Both layers must be tested independently.

55. Privacy Versus Transparency

Blockchain provides transparency and auditability.

However:

Transparency ↑ Transparency \uparrow

may sometimes result in:

Privacy ↓ Privacy \downarrow

if too much information is recorded on the ledger.

Therefore, the system must determine:

Minimum Information Required Minimum Information Required

for auditability.

56. Governance

A blockchain-IoT ecosystem may involve multiple organizations.

Governance rules should define:

- Who can join?
- Who can operate nodes?
- Who can create smart contracts?
- Who can revoke devices?
- Who can change policies?
- Who is responsible for incidents?

Without governance:

Technology ≠ Complete Solution Technology \neq Complete Solution

57. Trust Between Organizations

Consider three organizations:

A, B, CA, \ B, \ C

Each organization may have different policies.

Instead of requiring:

A Trusts BA \ Trusts \ B

and:

B Trusts CB \ Trusts \ C

the blockchain provides a shared transaction history.

The participants can independently verify recorded events.

This can reduce dependence on interpersonal or institutional trust.

58. Proposed Secure Data-Sharing Algorithm

A conceptual algorithm is presented below.

Input

Request=(User,Device,Data,Action)Request=(User,Device,Data,Action)

Procedure

1. Verify requester identity.
2. Verify device status.
3. Check access policy.
4. Calculate trust score.

5. Verify request context.
6. Evaluate smart contract.
7. If authorized, retrieve encrypted data.
8. Verify data integrity.
9. Record the access event.
10. Provide the requested data.

Output

AuthorizedDataAuthorizedData

or:

AccessDeniedAccessDenied

59. Algorithmic Representation

The decision process can be expressed as:

Decision={Allow, I\land S\land P\land T\land Deny, Otherwise Decision= \begin{cases} Allow, & I\land S\land P\land T\land \\ Deny, & Otherwise \end{cases}

where:

- II = valid identity
- SS = secure device status
- PP = valid policy
- TT = acceptable trust

60. Advantages

The proposed framework provides several potential advantages.

Decentralized Trust

Reduces dependence on one central authority.



Data Integrity

Blockchain records can provide strong tamper evidence.

Auditability

Access and security events can be recorded.

Flexible Access Control

Smart contracts can automate authorization.

Device Lifecycle Management

Registration and revocation can be recorded.

Hybrid Storage

Large sensitive datasets remain off-chain.

Edge Integration

Local processing can reduce blockchain workload.

61. Limitations

The proposed architecture also has important limitations.

Blockchain Overhead

Blockchain introduces additional computation and communication.



Scalability

Very large IoT deployments may exceed blockchain transaction capacity.

Storage Growth

Blockchain ledgers continue to grow.

Key Management

Compromised credentials can create severe security problems.

Smart Contract Vulnerabilities

Incorrect contract logic can introduce access-control failures.

Privacy

Metadata recorded on a ledger can potentially reveal behavioral information.

Deployment Complexity

Integrating IoT, edge, blockchain, and cloud systems requires significant engineering effort.

62. Future Research Directions

62.1 Lightweight Consensus

Future research should investigate consensus mechanisms designed specifically for IoT environments.

62.2 AI-Based Trust Management

Machine learning can analyze device behavior and dynamically calculate trust.

62.3 Privacy-Preserving Ledgers

Future blockchain systems should minimize metadata leakage.

62.4 Edge Intelligence

AI at edge gateways can identify attacks before suspicious transactions reach blockchain infrastructure.

62.5 Post-Quantum Security

Future IoT systems should investigate cryptographic mechanisms resistant to potential quantum attacks.

62.6 Automated Smart Contract Verification

Formal verification can help identify contract vulnerabilities before deployment.

62.7 Cross-Blockchain Interoperability

Future IoT environments may involve multiple blockchain networks.

62.8 Digital Twins

Digital twins can be used to simulate blockchain-IoT security scenarios.

63. Discussion

The integration of blockchain and IoT offers an alternative approach to managing trust in distributed environments.

Traditional IoT systems frequently depend on centralized infrastructure.

The proposed architecture changes this model:

CentralTrust → DistributedTrust

However, decentralization should not automatically be interpreted as improved performance.

Blockchain introduces:

Consensus+Storage+Communication
overhead.

Therefore, a blockchain architecture must be carefully designed for the specific IoT environment.

One of the most important principles of this research is that **blockchain should not be used to store every IoT measurement.**

If every sensor event becomes a blockchain transaction:

TransactionVolume ↑

and:

Performance ↓

Instead, edge gateways can aggregate information.

The blockchain can store hashes, access events, identity information, and policy metadata.

This produces:

IoTData→OffChainIoTData\rightarrow OffChain

and:

TrustMetadata→OnChainTrustMetadata\rightarrow OnChain

This separation can improve scalability while preserving data-integrity verification.

Another important consideration is privacy.

Blockchain's permanent records can conflict with privacy requirements if personally sensitive information is recorded directly on the ledger.

Therefore:

PrivacyByDesignPrivacyByDesign

should be incorporated from the beginning.

The proposed architecture also emphasizes smart contracts.

Smart contracts can automate access decisions, but they must be carefully designed and tested.

A vulnerable smart contract could become an automated mechanism for unauthorized access.

Consequently:

SecureBlockchain=SecureConsensus+SecureContracts+SecureKeys+SecureApplicationsSecureBlockchain = SecureConsensus + SecureContracts + SecureKeys + SecureApplications

Finally, blockchain alone cannot protect physically compromised IoT devices.

If an attacker gains physical or software-level control of a sensor, blockchain can record transactions generated by that device, but it cannot automatically determine whether every piece of sensor data is truthful.

Therefore, blockchain should be combined with:

- Device attestation
- Behavioral monitoring
- AI-based anomaly detection
- Secure firmware
- Hardware security
- Continuous authentication

64. Conclusion

The rapid expansion of IoT networks has created new opportunities for intelligent data sharing while simultaneously increasing security, trust, privacy, and access-control challenges. Traditional centralized architectures can create single points of failure and may not provide an appropriate trust model for ecosystems containing multiple independent organizations.

This paper proposed a conceptual blockchain-enabled framework for secure IoT data sharing. The framework integrates IoT devices, edge gateways, blockchain-based identity management, smart contracts, encrypted off-chain storage, data-integrity verification, and trust management.

The architecture is based on a clear separation of responsibilities. IoT devices generate data, edge gateways perform local processing and aggregation, blockchain maintains trust and audit information, smart contracts enforce access policies, and off-chain storage maintains large datasets.

A major principle of the proposed framework is:

LargeData→OffChainLargeData\rightarrow OffChain

while:

Integrity+Identity+AccessEvents→BlockchainIntegrity+Identity+AccessEvents\rightarrow Blockchain

This separation can reduce blockchain storage and transaction requirements.

The paper also demonstrated that blockchain should not be considered a complete security solution by itself. IoT systems require complementary mechanisms for authentication, encryption, key management, secure firmware, anomaly detection, privacy protection, and smart-contract security.

Scalability remains one of the most significant challenges. A system containing millions of devices cannot necessarily place every sensor measurement directly on a distributed ledger. Edge-assisted aggregation and selective blockchain recording therefore represent important architectural strategies.

Future research should focus on lightweight consensus, AI-assisted trust management, privacy-preserving blockchain designs, secure smart contracts, post-quantum cryptography, interoperability, and real-world experimental evaluation.

The overall security architecture can be summarized as:

SecureIoT=Authentication+Encryption+AccessControl+Blockchain+EdgeSecurity+Monitoring
SecureIoT= Authentication+ Encryption+ AccessControl+ Blockchain+ EdgeSecurity+ Monitoring

Blockchain therefore represents not a replacement for existing IoT security mechanisms, but a potential trust and accountability layer within a broader secure IoT architecture.

References

1. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805.
2. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.
3. Dorri, A., Kanhere, S. S., & Jurdak, R. (2017). Towards an optimized blockchain for IoT. *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 173–178.
4. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 4, 2292–2303.
5. Reyna, A., Martín, C., Chen, J., Soler, E., & Díaz, M. (2018). On blockchain and its integration with IoT: Challenges and opportunities. *Future Generation Computer Systems*, 88, 173–190.
6. Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82, 395–411.
7. Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal*, 5(2), 1184–1195.
8. Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R. P., & Ni, W. (2019). Anatomy of threats to the Internet of Things. *IEEE Communications Surveys & Tutorials*, 21(2), 1636–1675.

9. Ferrag, M. A., Shu, L., Yang, X., Derhab, A., & Maglaras, L. (2020). Security and privacy for the Internet of Things: A review of blockchain solutions. *IEEE Access*, 8, 129953–129975.
10. Conoscenti, M., Vetrò, A., & De Martin, J. C. (2016). Blockchain for the Internet of Things: A systematic literature review. *IEEE/ACS 13th International Conference of Computer Systems and Applications*, 1–6.
11. Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36, 55–81.
12. Yaqoob, I., Salah, K., Jayaraman, R., & Al-Hammadi, Y. (2021). Blockchain for healthcare data management: Opportunities, challenges, and future directions. *IEEE Transactions on Engineering Management*.
13. Stankovic, J. A. (2014). Research directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1), 3–9.
14. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376.
15. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76, 146–164.