



Blockchain-Based Secure Data Sharing Framework for Internet of Things Applications

Haruto Ongprasert

Department of Computer Science and Engineering, Siam Delta Engineering Institute, Thailand
haruto.ongprasert@sdei-th.edu

Peer Review Information	Abstract
<i>Submission: 30 April 2017</i> <i>Revision: 09 May 2017</i> <i>Acceptance: 26 May 2017</i> Keywords <i>Blockchain, Internet of Things, Secure Data Sharing, Access Control, Data Integrity</i>	The Internet of Things (IoT) enables large numbers of heterogeneous devices to collect and exchange operational, environmental, and personal data, but conventional cloud-centered sharing models create concentrated points of trust and failure. This paper presents a methodology for a blockchain-based secure data sharing framework designed for IoT applications while respecting the computational limits of edge devices. The framework separates data from trust management: IoT payloads are encrypted and stored off-chain, whereas hashes, ownership records, access policies, and audit events are committed to a lightweight permissioned blockchain. An edge gateway performs device authentication, digital signing, encryption, and transaction submission so that constrained sensors are not required to maintain the ledger. A smart-contract layer evaluates access requests using subject, resource, operation, and time constraints and creates an immutable authorization trail. The methodology defines the architecture, transaction sequence, threat model, cryptographic controls, prototype assumptions, and evaluation metrics. A reproducible illustrative simulation is used to compare the proposed method with a conventional centralized access-control baseline in terms of tamper detection, access-decision latency, and service continuity under a single infrastructure failure. The simulated results show that the blockchain design adds validation latency but provides deterministic detection of modified audit records and avoids dependence on one authorization server. The study therefore supports a hybrid design in which blockchain is used for decentralized trust, policy enforcement, and accountability, while bulk sensor data remain outside the chain. The work is positioned against IoT and blockchain research published up to 2017 and identifies scalability, key management, metadata privacy, and validator governance as the principal issues for further empirical study.

Introduction

The Internet of Things links physical objects, sensors, actuators, gateways, applications, and service platforms into a persistent information environment. Early IoT surveys described this paradigm as an integration of identification, sensing, networking, and distributed intelligence, while also emphasizing interoperability, security, and privacy as

unresolved requirements [1]. Subsequent architectural work highlighted that IoT systems may generate large volumes of continuously changing data that must be stored, processed, and shared across organizational boundaries [3]. This broad connectivity creates value in smart homes, industrial monitoring, logistics, healthcare, energy systems, and cities, but it also

increases the number of entities that can request, process, or misuse data.

Secure data sharing in this environment is more complicated than ordinary client-server protection. IoT devices often have limited memory, energy, and processing capacity; they use heterogeneous protocols; ownership of a device may differ from ownership of its data; and services may involve several administrative domains. Roman, Zhou, and Lopez showed that distributed IoT architectures offer useful resilience and collaboration properties, but also introduce difficult authentication, authorization, privacy, and trust-management problems [2]. Sicari et al. similarly identified confidentiality, authentication, access control, privacy, trust, and policy enforcement as core security requirements that must operate at IoT scale [4].

Most deployed IoT sharing models historically relied on a centralized cloud service or authorization server. Such a model is straightforward to administer but places substantial trust in the service operator. If the central database, policy engine, or audit log is compromised, an attacker may modify records, grant unauthorized privileges, suppress evidence, or interrupt access for all participants. Centralization also becomes difficult when independent organizations need a common and verifiable history of data ownership and authorization decisions. These limitations motivate the use of distributed ledgers as a trust coordination layer rather than as a replacement for every IoT component.

Blockchain technology offers an append-oriented replicated ledger in which transactions are cryptographically linked and collectively validated. Christidis and Devetsikiotis argued that blockchains can enable verifiable peer-to-peer interactions and smart-contract automation without requiring every participant to trust one intermediary [7]. For IoT, however, the direct use of cryptocurrency-style proof-of-work is generally unsuitable because of computational cost, transaction delay, bandwidth use, and ledger growth. Dorri et al. therefore explored a lightweight blockchain architecture for smart-home IoT that removed mining incentives and reduced proof-of-work overhead while assigning more demanding tasks to a capable local device [10]. The present methodology adopts the same general principle: constrained things sense and communicate, whereas gateways and validator nodes provide ledger functions.

The objective of this paper is to define and evaluate a secure data-sharing methodology in which (i) IoT data are encrypted before leaving

the local trust boundary, (ii) a permissioned blockchain records integrity evidence and authorization state, (iii) smart contracts enforce fine-grained access rules, (iv) raw high-volume data remain off-chain, and (v) every authorized or denied access request produces an auditable event. The contribution is methodological rather than a claim of a production deployment. Accordingly, the results section reports an illustrative simulation with explicit assumptions so that the trade-off between additional ledger validation and stronger auditability can be examined without fabricating real-world measurements.

Literature Review

IoT research before the widespread adoption of blockchain already established the security context for data sharing. Atzori, Iera, and Morabito described the IoT as an environment in which heterogeneous objects cooperate through networked services and noted that cryptographic protection can be difficult on resource-constrained devices [1]. Gubbi et al. proposed a cloud-centric IoT vision and emphasized the need to manage large-scale sensing data while supporting ownership, privacy, and information sharing [3]. Al-Fuqaha et al. reviewed enabling protocols and architectures and observed that conventional Internet security mechanisms do not map cleanly to constrained IoT platforms [5]. Together, these works indicate that secure data sharing must minimize device-side overhead while still providing scalable authentication and policy enforcement.

Security-focused studies sharpened these requirements. Roman et al. compared centralized and distributed IoT architectures and noted that distributed settings can reduce concentration of control but complicate authentication and authorization [2]. Sicari et al. reviewed confidentiality, access control, privacy, trust, and policy enforcement, concluding that flexible security mechanisms are necessary because IoT combines heterogeneous standards and dynamic relationships [4]. This background makes decentralized authorization particularly attractive when data must cross application or organizational boundaries.

Blockchain research introduced mechanisms directly relevant to ownership and access control. Zyskind, Nathan, and Pentland demonstrated that a blockchain can act as an automated access-control manager for personal data while the data themselves are kept outside the ledger [6]. Christidis and Devetsikiotis extended the discussion to IoT, explaining how

smart contracts can automate multi-step interactions between non-trusting parties [7]. A systematic review by Conoscenti, Vetrò, and De Martin found early blockchain-IoT use cases and highlighted integrity, anonymity, and adaptability as important but competing concerns [8].

Access-control frameworks published by 2017 provide more concrete design cues. Ouaddah, Abou Elkalam, and Ait Ouahman introduced FairAccess, a decentralized authorization framework in which blockchain transactions support grant, get, delegate, and revoke

operations while preserving pseudonymous control over IoT resources [9]. Dorri et al. proposed a smart-home blockchain model that shifts heavier security tasks to a local miner/gateway and reported that a lightweight design can reduce the overhead associated with conventional blockchain mechanisms [10]. Kshetri examined how blockchain can strengthen IoT security through tamper resistance, identity management, provenance, and targeted containment of compromise [11]. These studies collectively support using blockchain as a control and evidence layer rather than storing every IoT payload on-chain.

Table 1. Selected literature used to derive the proposed methodology

Ref.	Year / Focus	Contribution	Methodological implication	Main limitation for IoT
[1]	2010 / IoT survey	IoT enabling technologies and open security issues.	Keep cryptographic work lightweight at devices.	Resource constraints.
[2],[4]	2013–2015 / security	Distributed IoT security, privacy, trust and access control challenges.	Threat model must include compromised services and cross-domain authorization.	Heterogeneity and scale.
[6]	2015 / personal data	Blockchain used as decentralized access-control manager.	Store control state on-chain; keep payload data external.	Privacy of public metadata.
[7]	2016 / blockchain-IoT	Smart contracts enable verifiable automated interactions.	Use contract logic for authorization decisions and audit events.	Ledger scalability and cost.
[8]	2016 / systematic review	Early blockchain-IoT cases; integrity/anonymity/adaptability trade-offs.	Evaluate both security gain and performance overhead.	Limited IoT evidence base.
[9]	2017 / FairAccess	Grant, get, delegate and revoke transactions for IoT authorization.	Model lifecycle of permissions explicitly.	Prototype and policy complexity.
[10],[11]	2017 / IoT blockchain	Lightweight gateway-centered blockchain and security analysis.	Offload ledger work to gateway/validators and avoid proof-of-work.	Governance and deployment assumptions.

The pre-2018 literature establishes the feasibility of blockchain-assisted IoT access control, but no single early approach removes all practical constraints. Public chains can expose metadata and impose unacceptable processing delay; purely on-chain storage is incompatible with continuous sensor streams; and a decentralized ledger does not automatically solve key distribution, device compromise, or poor policy design. The gap addressed here is therefore a balanced methodology that combines

off-chain encrypted storage, permissioned validation, smart-contract authorization, cryptographic auditability, and gateway-assisted computation. The evaluation focuses on whether this separation of responsibilities improves integrity evidence and service resilience at a tolerable access-control latency.

Methodology

1. Research Design and Objectives

The study uses a design-science methodology. First, security and performance requirements are derived from the literature. Second, a layered framework is specified to satisfy those requirements. Third, the transaction and access-control procedures are defined. Fourth, an illustrative prototype simulation compares the proposed architecture with a centralized baseline. The dependent variables are access-decision latency, detection of modified audit records, and continuity of authorization service under a single infrastructure failure. The independent architectural condition is centralized versus permissioned-blockchain authorization. Security properties are evaluated against defined attacker capabilities rather than assumed from the presence of blockchain alone.

2. Proposed Framework Architecture

The framework contains five logical elements: IoT devices, an edge gateway, a permissioned blockchain, encrypted off-chain storage, and authorized applications. Devices collect measurements and transmit them through a protected local channel to the gateway. The gateway maintains device identities, verifies message freshness, encrypts payloads, computes hashes, and submits signed blockchain transactions. Validator peers operated by trusted stakeholders replicate the ledger and execute smart-contract logic. The data store holds only encrypted sensor objects. Applications receive decryption material only after the contract authorizes the requested subject, resource, and operation. This design follows the off-chain personal-data principle in [6], the smart-contract model in [7], the decentralized access-control direction of FairAccess [9], and gateway offloading in [10].

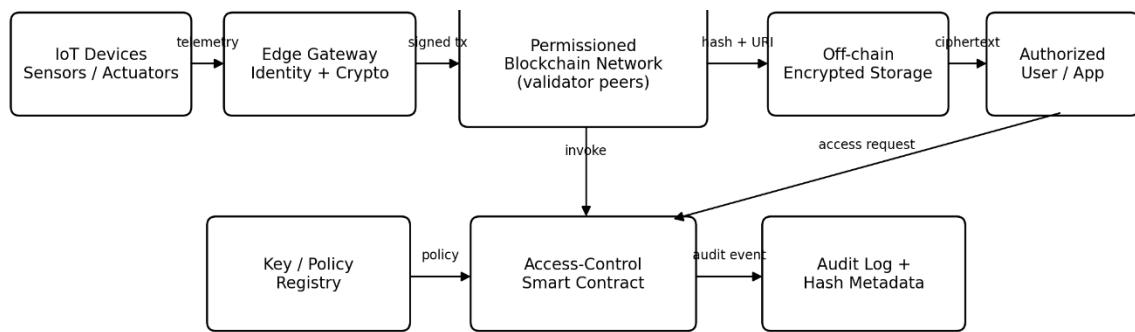


Figure 1. Proposed blockchain-based secure data sharing framework for IoT applications.

3. Data Registration and Sharing Procedure

Each IoT device is provisioned with a unique identifier and a cryptographic credential during enrollment. The gateway binds the device identifier to an owner account and records only the minimum identity metadata required for later verification. When a sensor produces data, the gateway performs four operations: (1) validate the source and timestamp; (2) generate a content identifier; (3) encrypt the data object with a fresh symmetric data key; and (4) calculate a SHA-256 digest of the ciphertext. The encrypted object is written to off-chain storage, while the blockchain transaction contains the object identifier, digest, storage pointer, owner identifier, policy identifier, and timestamp. Consequently, later retrieval can verify that the off-chain ciphertext has not been replaced without forcing every validator to store the sensor stream.

An access request is represented as $\text{Request} = \{\text{subject, object, action, context, nonce}\}$. The smart contract retrieves the current policy state and evaluates whether the subject is permitted to perform the requested action on the specified object. Context can include time window, device domain, purpose code, or delegated role. A positive decision creates an authorization event and allows the gateway or a key service to release the wrapped data key to the requester. A negative decision produces a denial event without exposing the key. Grant, delegation, revocation, and access operations are modeled as distinct state transitions, consistent with the transaction lifecycle described for FairAccess [9].

4. Security Controls and Threat Model

The threat model assumes an adversary may eavesdrop on external links, submit unauthorized access requests, compromise an application credential, modify data stored outside the blockchain, or attempt to alter an audit history. The model does not assume that

blockchain can protect a sensor after physical capture; therefore, device attestation and hardware tamper resistance remain outside the study scope. It also assumes a permissioned validator set in which the number of malicious validators remains below the consensus fault threshold. This is appropriate for consortium-style IoT applications where organizations are identifiable but do not wish to depend on one operator.

- Confidentiality: sensor payloads are symmetrically encrypted before external storage; the blockchain contains a digest and pointer rather than plaintext data.
- Integrity: the stored ciphertext is checked against the on-chain digest, while ledger chaining and replicated validation make unauthorized historical modification evident.
- Authentication: devices, gateways, validators, and applications use public-key credentials; signed requests bind actions to identities.
- Authorization: smart-contract rules evaluate access based on current ownership, role, delegated permission, operation, and contextual restrictions.
- Accountability: each grant, revoke, allow, and deny event is written as a signed ledger event, supporting traceable post-incident review.
- Availability: authorization state is replicated across validators, reducing dependence on one policy server; off-chain storage may likewise be replicated in a complete implementation.

5. Smart-Contract Authorization Logic

The authorization procedure is deliberately simple so that its behavior is auditable. On receiving a request, the contract verifies that the resource exists, the requester credential is active, the nonce has not been processed, and the requested operation is defined. It then checks direct ownership, an explicit grant, or a valid delegated role. If a contextual constraint is attached, the request must also satisfy the permitted time or domain condition. The contract finally records ALLOW or DENY together with a request identifier and timestamp. Revocation takes precedence over an older grant. Policy changes require a transaction signed by the resource owner or an authorized administrator, preventing an application from silently modifying its own permissions.

6. Illustrative Evaluation Setup

A reproducible simulation was specified to examine methodology-level trade-offs rather than claim laboratory measurements. The logical test environment contains 20 IoT devices, 8 users/applications, one edge gateway, four permissioned validator peers, and one encrypted object store. A total of 250 access requests are generated from a fixed random seed, with 70% satisfying the policy and 30% intentionally unauthorized. For latency, the centralized baseline is modeled as network delay plus one policy lookup; the blockchain condition adds signature verification, smart-contract execution, and quorum validation. Random delays are sampled from bounded normal distributions to produce stable illustrative values. One hundred stored audit records are then modified to test tamper detection. Finally, the central authorization server or one of four validators is marked unavailable to evaluate whether access decisions can continue.

The baseline represents a typical centralized ACL service with a mutable database log. It is not designed as the strongest possible centralized security architecture; rather, it provides a transparent reference for the specific properties that distributed ledgers are intended to add. The blockchain model uses no proof-of-work and does not replicate bulk payloads. This assumption is consistent with the early literature's warning that cryptocurrency-style blockchain overhead is unsuitable for constrained IoT and with lightweight proposals that move complex operations to capable nodes [8], [10].

Results

The illustrative simulation produced a mean access-decision latency of 18.1 ms for the centralized baseline and 47.4 ms for the proposed blockchain condition. The corresponding 95th-percentile values were 25.3 ms and 61.7 ms. Thus, consensus and ledger validation introduce an expected delay penalty. However, all 100 intentionally modified audit entries were detected by recomputing the blockchain-linked integrity evidence, whereas the mutable baseline log had no intrinsic cryptographic mechanism to reveal a change once an attacker was assumed to possess database write capability. Both approaches blocked all requests classified as unauthorized by the test policy; the advantage of the blockchain design is therefore not that it invents authorization rules, but that it decentralizes policy state and preserves evidence of decisions.

Table 2. Illustrative evaluation results (simulation, not production benchmarks)

Metric	Centralized baseline	Proposed framework	Interpretation
Unauthorized requests blocked	100%	100%	Same policy semantics in the simulation.
Modified audit records detected	0% intrinsic	100% (100/100)	Hash-linked ledger provides tamper evidence.
Mean access-decision latency	18.1 ms	47.4 ms	Ledger validation adds delay.
95th-percentile latency	25.3 ms	61.7 ms	Tail latency remains higher with consensus.
Authorization service after one infrastructure failure	Unavailable when the single policy server fails	Available with 3 of 4 validators	Replication removes one authorization-server failure point.
Raw IoT payload stored on-chain	No	No	Proposed method stores only hash/metadata on-chain.

Interpretation of Results

The results indicate a security-performance trade-off rather than unconditional superiority. Centralized authorization is faster because a single server can decide immediately after authentication and policy lookup. The blockchain framework requires additional cryptographic checks and agreement among validator peers, so it should be used where shared trust, traceability, and resilience justify that delay. For applications involving infrequent sharing of sensitive records, configuration data, machine logs, or cross-organization transactions, tens of milliseconds of additional authorization latency may be acceptable. For closed-loop control requiring sub-millisecond response, authorization should be pre-established and local real-time operation should not wait for a blockchain transaction.

Tamper evidence is the clearest methodological gain. The blockchain does not prevent an attacker from altering an off-chain ciphertext object, but the modification becomes detectable because its digest no longer matches the committed ledger value. Likewise, replicated access events make it more difficult for one compromised administrator to erase evidence from every validator. This observation is consistent with the decentralized personal-data model of Zyskind et al. [6], the verifiable interaction model of Christidis and Devetsikiotis [7], and Kshetri's argument that blockchain can strengthen IoT provenance and resistance to manipulation [11].

Availability also improves only within the stated assumptions. A four-validator permissioned network can continue when one validator is unavailable if the consensus policy still has quorum, but failures in the gateway, key service,

network, or off-chain storage can still block data access. The design therefore reduces one central dependency; it does not eliminate all single points of failure unless the remaining components are also replicated. Similarly, privacy is not guaranteed by pseudonymous blockchain identifiers alone. Transaction timing, resource identifiers, and access patterns can leak metadata, a limitation already recognized by early blockchain-IoT research [8], [9].

Conclusion and Discussion

This methodology paper proposed a blockchain-based secure data sharing framework for IoT applications using research available up to 2017. The central design decision is to separate high-volume data storage from distributed trust management. IoT payloads are encrypted and retained off-chain, while a permissioned blockchain stores cryptographic digests, ownership and policy state, authorization events, and audit evidence. Constrained devices are protected from ledger overhead by an edge gateway that performs encryption, signing, hashing, and transaction submission. Smart contracts implement explicit grant, delegate, revoke, allow, and deny transitions so that access-control state can be inspected and verified by multiple stakeholders rather than one service operator. The illustrative simulation shows why this hybrid structure is useful and where it is costly. The proposed architecture provides strong tamper evidence for modified records and can preserve authorization service when one validator fails, but it increases decision latency compared with a centralized policy server. These results support a selective use of blockchain: the ledger should protect trust-sensitive state and audit history, while real-time

sensing, actuation, and bulk storage remain in specialized IoT components. This conclusion is consistent with the resource constraints identified in foundational IoT surveys [1], [5], the distributed security challenges described by Roman et al. [2] and Sicari et al. [4], and the lightweight blockchain direction reported in early IoT studies [10]. Several limitations define the next steps for a real implementation. First, the current results are simulated and must be validated on physical gateways, sensors, and validator nodes. Second, permissioned consensus introduces governance questions: organizations need rules for validator admission, key recovery, software updates, and dispute resolution. Third, the ledger itself can reveal metadata even when payloads are encrypted, so privacy-enhancing identifiers and careful minimization of on-chain fields are required. Fourth, compromised endpoints can still submit false sensor data; blockchain preserves the history of submitted information but cannot prove that a physical measurement was truthful. Finally, revocation and key rotation must be coordinated with cached data keys to prevent a formerly authorized user from retaining long-term access. Future work should therefore implement the framework on Raspberry Pi-class gateways, compare multiple permissioned consensus mechanisms, measure throughput and energy use, study metadata leakage, and evaluate policy management under device ownership transfer. The proposed methodology provides a reproducible foundation for that empirical work without assuming that blockchain alone solves every IoT security problem.

References

- L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. DOI: 10.1016/j.comnet.2010.05.010.
- R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013. DOI: 10.1016/j.comnet.2012.12.018.
- J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013. DOI: 10.1016/j.future.2013.01.010.
- S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015. DOI: 10.1016/j.comnet.2014.11.008.
- Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. DOI: 10.1109/COMST.2015.2444095.
- G. Zyskind, O. Nathan, and A. "Sandy" Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, pp. 180–184, 2015. DOI: 10.1109/SPW.2015.27.
- K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016. DOI: 10.1109/ACCESS.2016.2566339.
- M. Conoscenti, A. Vetrò, and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–6, 2016. DOI: 10.1109/AICCSA.2016.7945805.
- Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: A new blockchain-based access control framework for the Internet of Things," *Security and Communication Networks*, vol. 9, pp. 5943–5964, 2016/2017. DOI: 10.1002/sec.1748.
- Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 618–623, 2017. DOI: 10.1109/PERCOMW.2017.7917634.
- N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017. DOI: 10.1109/MITP.2017.3051335.