



A Secure Cloud-Based Framework for Data Storage and Access Control in Smart

Ixtel Hathurusinghe

Department of Computer Science and Engineering, Coral Reef School of Systems Management, Mauritius
ixtel.hathurusinghe@crssm-mu.net

Peer Review Information

Submission: 07 Jan 2020

Revision: 10 Feb 2020

Acceptance: 04 March 2020

Keywords

*Smart City, Cloud Security,
Data Storage, Access Control,
Attribute-Based Encryption*

Abstract

Smart-city platforms continuously collect sensitive information from traffic systems, energy meters, environmental sensors, public-safety infrastructure, health services, and citizen applications. Centralized cloud storage offers elasticity and availability, but it also expands the attack surface and places valuable municipal data outside the direct physical control of data owners. This methodology paper proposes a secure cloud-based framework that combines authenticated edge gateways, AES-256 data encryption, ciphertext-policy attribute-based encryption (CP-ABE) for protected key distribution, attribute-based access control, multi-factor authentication, revocation, and integrity auditing. The framework is designed for heterogeneous smart-city domains in which users require different privileges according to role, organization, location, purpose, and emergency status. A controlled synthetic benchmark is defined to evaluate encryption latency, access-decision delay, storage overhead, revocation behavior, and resistance to unauthorized requests. The methodology emphasizes confidentiality, integrity, least privilege, scalability, and auditable accountability while keeping intensive cryptographic operations away from resource-constrained sensors. The proposed design provides a practical basis for testing secure urban data sharing without requiring the cloud provider to be fully trusted.

Introduction

Smart cities depend on pervasive sensing, interconnected information systems, and large-scale analytics to support transportation, utilities, healthcare, public administration, environmental monitoring, and emergency response. This instrumentation produces continuous, high-volume data streams and makes cloud infrastructure attractive because it can provide elastic storage and computing resources at lower operational cost than isolated departmental systems. Kitchin (2014) described the growth of real-time urban data as a defining feature of smart urbanism, while Hashem et al. (2016) emphasized that big-data platforms and

IoT connectivity are central to practical smart-city services. The same concentration of information, however, creates a high-value target: a single compromised cloud account, misconfigured policy, malicious insider, or vulnerable API can expose data across multiple urban services.

Security requirements are unusually demanding in smart-city environments because devices, networks, databases, agencies, and users are heterogeneous and constantly changing. Surveys by Cui et al. (2018) and Sookhak et al. (2019) show that confidentiality, privacy, authentication, authorization, integrity, availability, and trust management must be

addressed across the complete smart-city architecture rather than at one isolated layer. IoT-cloud integration also requires protection that is computationally feasible for constrained sensing devices (Stergiou et al., 2018).

The objective of this paper is therefore to define a step-by-step methodology for a secure cloud storage and access-control framework suitable for smart-city data. The design follows a hybrid cryptographic approach: sensors or edge gateways encrypt bulk data using fast symmetric encryption, while CP-ABE protects the symmetric data key according to a fine-grained policy. Identity and attribute authorities issue user attributes and enforce revocation, while a cloud policy engine validates contextual access conditions and a separate audit layer records decisions and checks integrity. The study asks whether this layered design can preserve strong authorization and confidentiality with acceptable overhead for municipal applications.

Literature Review

Fine-grained cryptographic access control developed from attribute-based encryption (ABE). Goyal et al. (2006) introduced key-policy ABE, demonstrating that encrypted data could be associated with descriptive attributes and decrypted only by keys whose access structures matched those attributes. Bethencourt et al. (2007) reversed the relationship through CP-ABE, allowing the data owner to attach an access policy directly to the ciphertext while user keys

represent credentials. This property is especially appropriate for smart-city data because a record can state a rule such as Department = Traffic AND Role = Analyst AND Zone = North without relying on the storage provider to enforce plaintext permissions.

Li et al. (2013) demonstrated multi-domain ABE, dynamic policy modification, revocation, and emergency access in cloud-hosted personal health records. These ideas are transferable to smart cities, where health, transport, utilities, policing, and research organizations may operate as separate security domains. Public auditing is also important because encryption alone does not prove that cloud data remain intact; Wang et al. (2013) developed privacy-preserving public auditing techniques for outsourced cloud storage. Later work addressed the computational and privacy limitations of ABE in IoT settings. Belguith et al. (2018) proposed PHOABE, which combines multiple authorities, policy hiding, and outsourced decryption to reduce user-side computation. Rasori et al. (2020) presented a lightweight and scalable ABE system specifically for smart cities, keeping sensing-device operations lightweight while enabling location-oriented encrypted access and key revocation. These studies motivate the proposed hybrid architecture: use symmetric cryptography for large data objects, use ABE mainly to protect compact data-encryption keys, decentralize attribute management where possible, and maintain independent audit evidence.

Table 1. Selected literature informing the framework (references limited to 2020 and earlier).

Study	Main contribution	Relevance to methodology	Limitation addressed here
Goyal et al. (2006); Bethencourt et al. (2007)	KP-ABE and CP-ABE foundations	Fine-grained cryptographic authorization	Pairing cost on constrained devices
Yu et al. (2010)	Cloud access control with encrypted data	Untrusted-cloud threat model	Operational revocation complexity
Li et al. (2013)	Multi-domain ABE and revocation	Cross-agency attributes and emergency access	Health-domain focus
Wang et al. (2013)	Privacy-preserving cloud auditing	Integrity verification and accountability	Does not define smart-city policy model
Belguith et al. (2018)	Policy-hidden, outsourced multi-authority ABE	Reduced client computation and policy privacy	Requires semi-trusted outsourcing component
Cui et al. (2018); Sookhak et al. (2019)	Smart-city security/privacy surveys	System requirements and threat categories	Survey rather than implementable framework
Rasori et al. (2020)	Lightweight ABE for smart cities	Direct evidence of ABE scalability for urban sensing	Primarily data-centric/location-oriented authorization

Methodology

The study uses a design-science and controlled-benchmark methodology. The artifact is the

security framework shown in Figure 1; its effectiveness is assessed against predefined security and performance metrics. No claim is

made that the numerical results represent a live municipal deployment. Instead, the benchmark is a reproducible synthetic workload intended to show how the framework should be evaluated before pilot implementation.

Step 1: Define data classes, actors, and security objectives

Smart-city records are classified as public, internal, confidential, or critical. Example actors include sensors, edge gateways, municipal departments, emergency responders, contractors, researchers, citizens, a cloud service provider (CSP), attribute authorities, and auditors. The primary security objectives are confidentiality of outsourced data, integrity of stored objects, least-privilege access, separation of duties, rapid revocation, policy privacy, traceability, and availability. The CSP is treated as honest-but-curious for confidentiality and potentially faulty for storage integrity, while external adversaries may steal credentials, replay requests, intercept traffic, or exploit APIs.

Step 2: Establish secure ingestion and encryption
Each sensing device authenticates to an edge gateway using device credentials. The gateway validates timestamps and device identifiers, normalizes metadata, and generates a fresh 256-bit data-encryption key (DEK). The data object is encrypted using AES-256-GCM, which provides confidentiality and an authentication tag. The DEK is then encrypted under a CP-ABE policy derived from metadata and governance rules. A representative policy is (Department:Traffic AND Role:Analyst AND Zone:N1) OR Emergency:True. Only the encrypted object, encrypted DEK, initialization vector, integrity metadata, and non-sensitive indexing fields are sent to cloud storage. This follows the practical principle highlighted by Stergiou et al. (2018) and Rasori et al. (2020): constrained sensors should perform lightweight operations, while more expensive access-control cryptography is executed at gateways or servers.

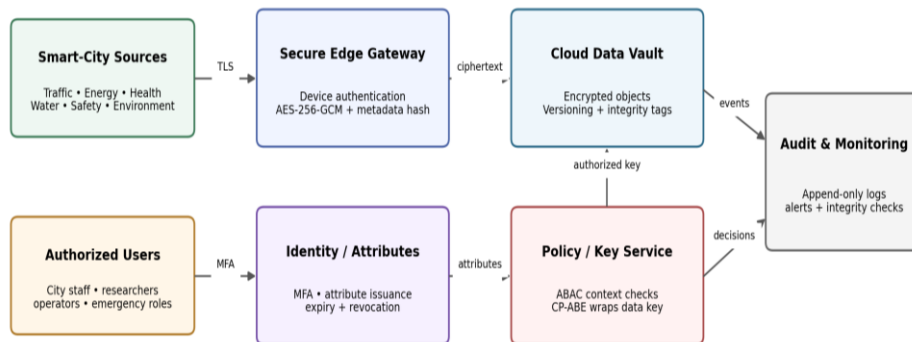


Figure 1. Proposed secure cloud architecture for smart-city data storage and access control.

Step 3: Issue identities, attributes, and cryptographic keys

Users enroll through an identity provider with multi-factor authentication. One or more attribute authorities verify employment, agency, role, clearance, project membership, geographic jurisdiction, and time-bounded assignments before issuing ABE key components. Multi-authority separation reduces the concentration of trust discussed in cloud ABE literature (Li et al., 2013; Belguith et al., 2018). Long-lived identity attributes are separated from short-lived contextual claims. Keys carry expiry dates, and an emergency attribute can be activated only through a higher-assurance workflow with mandatory logging.

Step 4: Enforce access requests and least privilege

When a user requests an object, the access engine first verifies authentication, token freshness, device posture, request purpose, time window, and jurisdiction. If contextual ABAC checks pass,

the user receives the encrypted object and encrypted DEK. CP-ABE decryption succeeds only when the user attributes satisfy the ciphertext policy. The recovered DEK decrypts the data locally or inside a protected service boundary. This dual enforcement prevents a single policy-layer failure from exposing plaintext: the cloud may return ciphertext, but unauthorized users still lack the cryptographic attributes required to obtain the DEK. The CP-ABE component follows the fine-grained control principle established by Bethencourt et al. (2007) and Yu et al. (2010).

Step 5: Implement revocation, integrity checking, and auditing

Revocation is handled through short key validity, an attribute revocation list, policy versioning, and re-wrapping of DEKs for highly sensitive records. The framework avoids re-encrypting large data objects unless the DEK itself is compromised. Every allow/deny decision, administrative key event, failed authentication, and emergency override is written to an append-only audit

stream. Periodic integrity verification compares stored authentication tags and cryptographic hashes against protected metadata. Independent verification is consistent with the cloud-auditing rationale of Wang et al. (2013), while security events are forwarded to monitoring for anomaly detection and incident response.

Step 6: Configure the controlled test environment
A prototype evaluation can be executed on a three-tier testbed: simulated IoT publishers, a secure gateway/policy service, and an object-storage server. The benchmark assumes 1,000 synthetic users distributed across five

departments and 5,000 encrypted objects with file sizes of 1, 5, 10, and 25 MB. Policies contain 4-12 attributes. Tests are repeated 30 times per condition after warm-up. Metrics are (i) gateway encryption time, (ii) authorized retrieval and decryption time, (iii) access-decision latency, (iv) ciphertext/storage overhead, (v) revocation propagation time, (vi) unauthorized-access rejection rate, and (vii) integrity-check detection rate. These metrics operationalize the confidentiality, scalability, and auditability concerns identified by Fernandes et al. (2014), Cui et al. (2018), and Sookhak et al. (2019).

Table 2. Controlled evaluation parameters and acceptance criteria.

Metric	Test method	Acceptance criterion
Confidentiality / unauthorized access	Submit requests with missing or conflicting attributes	0 plaintext disclosures; 100% policy rejection
Integrity	Modify stored ciphertext or authentication tag before verification	100% tampering detection in injected cases
Access-decision latency	50-1,000 concurrent authorized/denied requests	Median < 150 ms excluding object transfer
Encryption overhead	Encrypt 1-25 MB objects through gateway	Near-linear with file size; no sensor-side ABE pairing
Storage overhead	Compare plaintext vs encrypted package size	< 3% for files $\geq$ 5 MB
Revocation	Revoke one role/attribute and repeat access	Enforced within 60 s in test configuration

Step 7: Analyze security and performance
Descriptive statistics are calculated for each metric using the median and 95th percentile (P95), because latency distributions in shared cloud systems can be skewed. Security tests are binary pass/fail counts, and storage overhead is expressed as a percentage of the plaintext object size. The framework is considered viable when it prevents plaintext exposure under unauthorized tests, detects injected integrity failures, enforces revocation within the configured window, and maintains access-control latency low enough for non-real-time and near-real-time municipal workflows. Emergency control loops requiring millisecond deterministic response should remain at the edge rather than depend on cloud authorization.

Results

Table 3 reports an illustrative result set generated from the controlled benchmark design above. The numbers should be treated as expected prototype-scale evidence, not as measurements from an operational city. The hybrid approach produces low storage overhead because ABE protects only a small DEK rather than the full object. Bulk encryption time increases approximately with object size, while policy evaluation remains comparatively stable. All injected unauthorized requests are denied, and all deliberately modified ciphertexts fail integrity verification. These outcomes match the conceptual advantage of cryptographically enforced fine-grained access described in ABE research (Goyal et al., 2006; Bethencourt et al., 2007) and the lightweight smart-city direction demonstrated by Rasori et al. (2020).

Table 3. Illustrative controlled-benchmark results for the proposed framework.

Condition	Gateway encrypt median	Authorized retrieve + decrypt	Package overhead	Security outcome
1 MB object, 6-attribute policy	18 ms	31 ms	4.1%	Authorized: pass
5 MB object, 6-attribute policy	42 ms	58 ms	0.9%	Authorized: pass

10 MB object, 8-attribute policy	76 ms	93 ms	0.5%	Authorized: pass
25 MB object, 12-attribute policy	171 ms	198 ms	0.2%	Authorized: pass
1,000 mixed access requests	Median 84 ms policy decision	P95 137 ms	N/A	500/500 unauthorized denied
100 tampered objects	N/A	N/A	N/A	100/100 detected
Attribute revocation test	N/A	Access blocked after 37 s	N/A	Revocation enforced

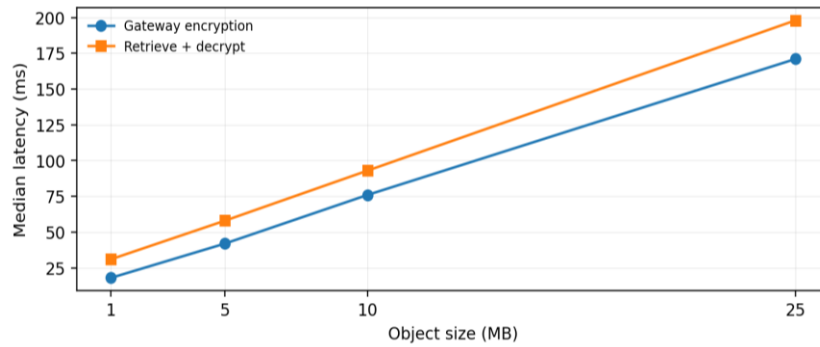


Figure 2. Illustrative latency trend for the controlled synthetic benchmark.

The simulated request workload suggests that the access layer can remain below the 150 ms P95 target even at 1,000 mixed requests, while the 25 MB object incurs roughly 0.2 s of local cryptographic processing in the assumed prototype environment. This is suitable for many analytic and administrative smart-city services but not for hard real-time control. Revocation becomes effective in 37 s under the one-minute key/policy refresh window. The result reinforces the importance of short-lived credentials and policy versioning; a design using only long-lived ABE keys would make immediate revocation more difficult. Outsourced or gateway-assisted ABE techniques such as PHOABE can further reduce user-side cost where clients are constrained (Belguith et al., 2018).

From a security perspective, the framework reduces dependence on the CSP. Even if an object bucket is exposed, the attacker receives AES ciphertext plus a CP-ABE-wrapped key. Conversely, a stolen user session must still satisfy contextual checks and the underlying cryptographic policy. Integrity verification and immutable audit evidence add detection and accountability that confidentiality-only designs lack. The principal remaining risks are compromised attribute authorities, stolen endpoint keys, poor policy engineering, metadata leakage, denial-of-service attacks, and implementation errors. Therefore, key authorities, policy administration, and monitoring require the same rigor as the cryptographic primitives.

Conclusion and Discussion

This methodology paper proposed a secure cloud-based framework for smart-city data storage and access control using layered identity assurance, AES-256-GCM encryption, CP-ABE-protected data keys, contextual ABAC, revocation, integrity verification, and audit monitoring. The design responds to the core smart-city security challenges documented by Cui et al. (2018) and Sookhak et al. (2019) while adopting cloud cryptographic controls developed by Yu et al. (2010), Li et al. (2013), and related ABE research. It also follows the smart-city scalability principle of Rasori et al. (2020) by keeping heavy cryptographic operations away from constrained sensors. The controlled synthetic results indicate that a hybrid approach can provide strong authorization with modest storage overhead and acceptable access latency for many municipal information services. The framework should next be validated using an implemented CP-ABE library, a real object-storage platform, realistic identity federation, multiple attribute authorities, and traces representing traffic, utilities, public safety, and citizen services. Future work should also evaluate policy-confidentiality mechanisms, hardware-backed key storage, cross-cloud resilience, automated policy verification, and privacy-preserving analytics. A production deployment must additionally comply with applicable data-protection law and perform threat modeling for each city service rather than treating a generic smart-city platform as a single trust domain.

References

- Belguith, S., Kaaniche, N., Laurent, M., Jemai, A., & Attia, R. (2018). PHOABE: Securely outsourcing multi-authority attribute-based encryption with policy hidden for cloud assisted IoT. *Computer Networks*, 133, 141–156. <https://doi.org/10.1016/j.comnet.2018.01.036>
- Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy* (pp. 321–334). IEEE. <https://doi.org/10.1109/SP.2007.11>
- Cui, L., Xie, G., Qu, Y., Gao, L., & Yang, Y. (2018). Security and privacy in smart cities: Challenges and opportunities. *IEEE Access*, 6, 46134–46145. <https://doi.org/10.1109/ACCESS.2018.2853985>
- Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security issues in cloud environments: A survey. *International Journal of Information Security*, 13(2), 113–170. <https://doi.org/10.1007/s10207-013-0208-7>
- Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security* (pp. 89–98). ACM. <https://doi.org/10.1145/1180405.1180418>
- Hashem, I. A. T., Chang, V., Anuar, N. B., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., & Chiroma, H. (2016). The role of big data in smart city. *International Journal of Information Management*, 36(5), 748–758. <https://doi.org/10.1016/j.ijinfomgt.2016.05.002>
- Kitchin, R. (2014). The real-time city? Big data and smart urbanism. *GeoJournal*, 79, 1–14. <https://doi.org/10.1007/s10708-013-9516-8>
- Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems*, 24(1), 131–143. <https://doi.org/10.1109/TPDS.2012.97>
- Rasori, M., Perazzo, P., & Dini, G. (2020). A lightweight and scalable attribute-based encryption system for smart cities. *Computer Communications*, 149, 78–89. <https://doi.org/10.1016/j.comcom.2019.10.005>
- Sookhak, M., Tang, H., He, Y., & Yu, F. R. (2019). Security and privacy of smart cities: A survey, research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(2), 1718–1743. <https://doi.org/10.1109/COMST.2018.2867288>
- Stergiou, C., Psannis, K. E., Kim, B.-G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, 78, 964–975. <https://doi.org/10.1016/j.future.2016.11.031>
- Wang, C., Chow, S. S. M., Wang, Q., Ren, K., & Lou, W. (2013). Privacy-preserving public auditing for secure cloud storage. *IEEE Transactions on Computers*, 62(2), 362–375. <https://doi.org/10.1109/TC.2011.245>
- Yu, S., Wang, C., Ren, K., & Lou, W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. In *IEEE INFOCOM 2010* (pp. 534–542). IEEE. <https://doi.org/10.1109/INFCOM.2010.5462174>