Scientific Reports

https://doi.org/10.1038/s41598-025-34703-y

Article in Press

Attribute based access control of geographic spatial data sharing using blockchain and smart contracts

Received: 21 April 2025

Accepted: 30 December 2025

Published online: 15 February 2026

Cite this article as: Li S., Liu W., Wu Y. *et al.* Attribute based access control of geographic spatial data sharing using blockchain and smart contracts. *Sci Rep* (2026). https://doi.org/10.1038/s41598-025-34703-y

Song Li, Wenfen Liu, Yan Wu, Xianglin Wu & Lihui Li

Attribute Based Access Control of Geographic Spatial Data Sharing Using Blockchain and Smart Contracts

Song Li[1,3]*, Wenfen Liu[1,2], Yan Wu[4], Xianglin Wu[1,5]， Lihui Li[1]

[1]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

[2]Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China

[3]School of Information Engineering, Nanning College of Technology, Guilin 541006, China

[4]Unit 95795 of PLA, Guilin 541003, China

[5]School of Artificial Intelligence, Hezhou University, Hezhou 542899, China

*Corresponding author: Song Li (slls21031102007@outlook.com)

Abstract

The secure and efficient sharing of geographic spatial data is crucial for applications in urban planning, disaster management, and environmental monitoring. However, conventional access control systems face scalability, security, and transparency problems in a distributed environment. This paper proposes a new framework that marries attribute-based access control with blockchain technology and smart contracts for fine-grained, decentralized, and tamper-proof data sharing. This paper introduces a new framework which combines Attribute-Based Access Control (ABAC), blockchain technology, smart contracts, and an upgraded Black-winged Kite (UBK) algorithm. Access regulations and audit logs are stored on a private blockchain using a Proof-of-Authority consensus mechanism for immutability and transparency. Experimental results show that the proposed method reduces evaluation policy time by 70% and storage overhead by 52% compared to the traditional attribute-based access control, while achieving 98.2% accuracy in access decisions. The performance test shows evaluation time and storage increase linearly, thus proving appropriate large-scale deployment. The combination of blockchain and smart contracts guarantees security-auditable and automated enforcement of access policies without needing a central authority.

Keywords: Geographic Spatial Data Sharing; Attribute-Based Access Control; Blockchain; Smart Contracts; Metaheuristic Optimization; Upgraded Black-winged Kite Algorithm.

1. Introduction

1.1. Background

Geospatial data is an essential component of the modern world, providing information to decision-makers in key areas such as urban planning, disaster preparedness, environmental monitoring, and transportation

[1-4]. For example, in urban planning, geographic information systems (GIS) are trained on data about land use patterns, population distribution, and infrastructure development [5]. In disaster response, real-time spatial data allows emergency services to assess damage and allocate resources accordingly [6]. Spatial data can be used in environmental monitoring to track changes in ecosystems, climate patterns and the availability of resources [7]. No doubt the secure, but efficient sharing of geographic spatial data remains a big problem, but one we absolutely need to have given the value of spatial data. Fig. (1) shows challenges in geospatial data sharing.

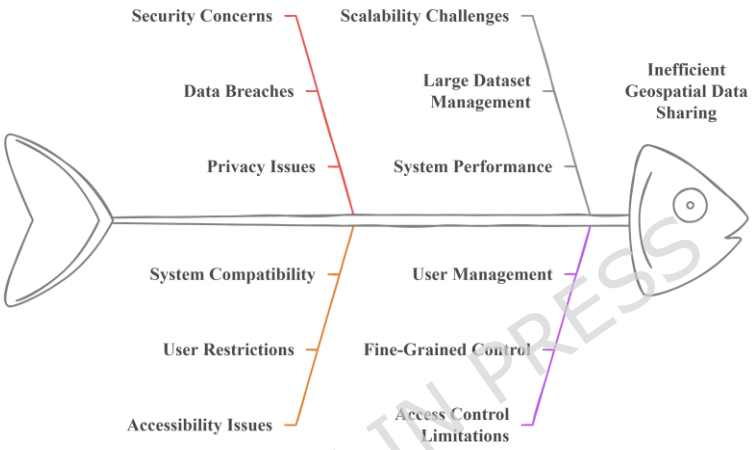


Fig.1. Challenges in geospatial data sharing

Standard ways of sharing data are generally focused on either being secure or being easy to access, and they typically cannot have both [8-10]. Beyond that, guaranteeing scalability as datasets get larger and more complex and offering fine-grained access control becomes increasingly impossible [11]. In particular, with the growing prevalence of distributed and collaborative settings where different actors need to interact securely with the same data, the requirement for a strong framework which can cover these aspects is needed more than ever.

1.2. Problem Statement

There are limitations of existing access control mechanisms (e.g. Role-Based Access Control (RBAC) and Mandatory Access Control (MAC)) when adapted for large-scale distributed geographic spatial data sharing systems:

Scalability: When the number of users, resources and attributes increases, traditional models become computationally expensive and unwieldy to manage. This is a particular problem in dynamic environments where policies are updated frequently.

Security: Centralized systems are susceptible to compromise, and when they are compromised, the entire system is at risk. Without decentralization of trust we will struggle to ensure data integrity and confidentiality (see Figure 2).

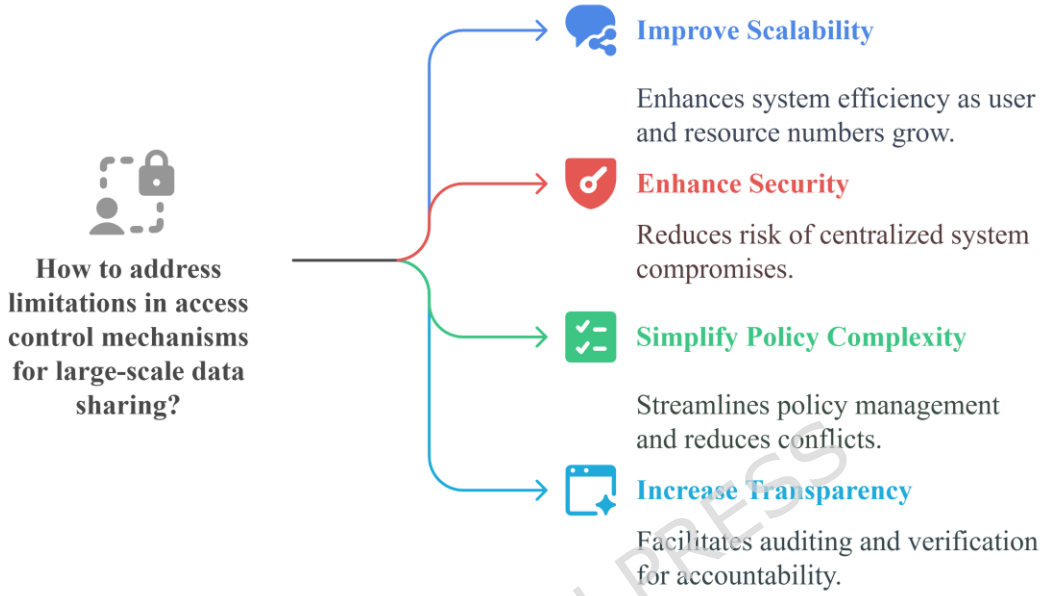


Fig. 2. Addressing limitations in access control mechanisms for large-scale data sharing

Policy Complexity: Due to fine-grained access requirements, complex policy definitions may be the source of conflicts, redundancies, and inefficiencies in policy evaluation. As systems scale, it becomes cumbersome to manage these policies by hand [12].

Transparency: Centralized systems suffer from lack of transparency, making auditing and verification of access decisions more difficult, which is critical for accountability and compliance in sensitive applications, e.g., disaster response or environmental monitoring.

These challenges require a novel solution that renders access control cyber risk reflection algos possible by using cutting-edge access control features and the latest technologies which can handle scalability, security, and transparency.

1.3. Motivation

In order to overcome the aforementioned limitations, we propose to design a secure, scalable and transparent system of sharing geographic spatial data by utilizing Attribute-Based Access Control (ABAC), Blockchain and Smart contracts. Whereas RBAC or MAC limit access decision-making based on users alone, ABAC offers more flexibility and allows decisions based on the attributes of users, resources and

environments [13]. Despite its advantages, applying ABAC at scale presents unique challenges, especially with respect to policy definitions optimization, as well as computational overhead reduction at policy evaluation time [14].

Blockchain is a distributed ledger that can improve data integrity and transparency. We can create trust among stakeholders by storing access control policies and audit trails on the blockchain and thus remove the possibility of undesired modification [15]. Additionally, smart contracts enable the enforcement of ABAC policies, which also reinforces decentralization, allowing access decisions to be made in real-time without having to depend on a central authority.

However, it is pivotal for its implementation to be combined with smart contracts and blockchain, yet performance bottlenecks still hinder the computational burden of blocking policy evaluations and transaction costs. To address this, we propose a new metaheuristic algorithm, called Upgraded Black-winged Kite (UBK) algorithm to improve the optimization of ABAC policies. The UBK algorithm streamlines policy management by avoiding redundancy and conflict resolution, thus ultimately building a stronger detrimental performance.

The research here presented advances beyond existing blockchain-based frameworks for access control by addressing an important gap in the optimization of fine-grained attribute-based policies for geographic spatial data sharing, an inadequately addressed challenge in recent works, such as the n-party virtual payment model, EASB (ECC-based aggregate signature), or Proof-of-Location systems.

While these studies address usability issues in terms of scaling blockchain authentication and location verification, they do not capture the policy complexity, redundancy, and conflict issues that usually follow in largescale ABAC systems. A completely new paradigm involving a metaheuristic optimization algorithm in the present work, Upgraded Black-winged Kite (UBK), aims at streamlining ABAC policies, thereby reducing overhead computation costs and increasing decision-making accuracy.

The real difference with our framework compared to EASB or virtual payment models that prioritize transaction security and efficiency is the complementary strengths of blockchain immutability, smart contract automation, and intelligent policy optimization for secure, scalable, and auditable access control mechanisms customized for dynamic geospatial environments.

This whole approach is what truly distinguishes our work, which besides guaranteeing the integrity of the data and trustlessness by decentralization, really seeks to improve the performance and manageability of the access control policies, making it a reasonable leap forward compared to existing blockchain applications in secure data sharing.

1.4. Proposed Solution

The solution we propose has the following main components:

- Attribute-Based Access Control (ABAC): Based on the above aspects, ABAC policy is defined based on user (e.g., role, department, clearance level), resource (e.g., type, sensitivity) and environment (e.g., time, location) attributes. These policies define the user rights on access to the specific geographic spatial data.
- Blockchain Network: Access control policies are stored in a private or consortium blockchain which maintains an immutable record of all access requests and decisions. This maintains transparency and accountability and blocks unauthorized changes.
- Smart Contracts: ABAC policies, attributed-based access control policies, are written in smart contracts and enforced automatically upon receiving access requests. They prevent unauthorized users from accessing and modifying Geographic Spatial Data.
- Algorithm for Upgraded Black-winged kite (UBK): To mitigate this limitation, UBK translates the ABAC policies into optimized rules (e.g., eliminated redundant rules, merged conflicting rules) for ease in computation. This derives from the black-winged kite versions of scouting for potentially useful solutions in the policy landscape.

Blending these layers creates a framework that prevents the security issues present in traditional applications but also protects against complex geographic spatial data sharing rules and enforcement for domain-specific use cases.

1.5. Contributions

This work contributes in a number of ways, proposing a novel framework to facilitate secure geographic spatial data sharing and to increase usability, utilizing a combination of Attribute-Based Access Control (ABAC), blockchains and smart contracts to enable secure and efficient sharing of geographic spatial data, addressing the pitfalls of traditional access control and providing scalability for large, distributed systems. In addition, a newly modified metaheuristic optimization method, namely, Upgraded Black-winged Kite (UBK) algorithm, proposed for optimizing the ABAC policies directly to reduce the computational cost and improve the quality of the above-mentioned policy management.

The proposed system effectiveness in the secure sharing of geographic spatial data is shown via exhaustive experimentation to evaluate its performance and advantages of using blockchain and smart contracts for secure data sharing over existing approaches used, validation of its superiority using comparative performance with existing approaches, and finally efficacy of the UBK algorithm to optimize ABAC policies mainstreaming new initiative in transforming secure geographic spatial data sharing using UBK algorithm.

1.6. Comparative advantages

For a clear presentation of the advantages of the proposed framework, it would be pertinent to provide further insight into the limitations of existing access control solutions, including how our solution acts to counter them. Traditional models like Role-Based Access Control (RBAC) and Mandatory Access Control (MAC) offer a better degree of simplicity but are rather coarse-grained and rigid for the dynamic and large-scale sharing of geospatial data, being based on static roles or hard security labels and unable to respond to contextual attributes like location, time, or environmental conditions.

While Attribute-Based Access Control (ABAC) is indeed capable of fine-grained and contextualized policies, problems related to redundancy, conflicts, and overhead during policy evaluation come into play concerning implementation on a wide scale, and these considerations are generally neglected in practice. Moreover, centralized ABAC systems become a single point of failure, leave no evidence trail for access decisions, and are otherwise subject to tampering, all of which greatly diminish the credibility of such systems for mission-critical areas like disaster response or environmental monitoring.

There are recent blockchain-based endeavors, attempting to advance decentralization and security; however, these predominantly tackle the questions of storing policy and automating access via smart contracts, while leaving the optimization of the policy structure in a very poor state. None incorporate any intelligent optimization schemes to simplify the problem or facilitate the scalability of operations. In total contrast, our proposed framework integrates ABAC with a private blockchain, smart contracts, and the Upgraded Black-winged Kite (UBK) algorithm, thereby establishing decentralized, transparent, and secure access control while simultaneously optimizing its policies for performance.

Through eliminating redundancy, resolving conflicts, and providing a 70% improvement in the time profile for policy evaluation over conventional ABAC, the UBK takes care of a very important lacuna in the existing work. Such total integration brings in the best scalability, security, and performance, making it highly suitable for a geospatial data-sharing environment on a large real-world scale.

1.7. Overview of system architecture

Having structured the whole approach to clarify and make it easier to comprehend, those are the core components of the architecture and how they interlink with each other. The architecture is sustained by these four main modules: (1) the Attribute-Based Access Control (ABAC) Module, needed to define, evaluate and enforce fine-grained access policies based on user, resource, and environmental attributes; (2) the Blockchain Network that decentralizes storage thereby guaranteeing immutability and transparency with respect to access policies, audit logs, and other metadata kept on a private network of blockchains that use the Proof-of-Authority (PoA) consensus mechanism; (3) Smart Contracts that automate the enforcement of policies by executing access control logic in a trustless environment, thus eliminating any human factor for minimum error; and (4) the Upgraded Black-winged Kite (UBK) Algorithm that optimizes ABAC

policies for redundancy minimization, conflict resolution, and computational efficiency enhancement. Based on this, the interaction of the modules is: administrators will define the ABAC policies that will be coded into the smart contracts and deployed to the blockchain; users will apply for access with attributes; smart contracts will do the evaluation of requests against the existing policies; and UBK will periodically run in the background for policy set optimization. Therefore, this will ensure high integration in secure, scalable and also efficient geographic spatial data sharing keeping all access decisions permanently on-chain for audit purposes.

2. Related works

In this section, we present a summary of existing studies on ABAC, blockchain-enabled access control and smart contract-supported sharing of data and identify the fundamental issues, opportunities, and limitations behind this study that justified the formulation of our proposed framework [16]. We discuss these solutions on the basis of their contribution toward ABAC models, blockchain solutions for security, and smart contracts, and how we propose to add to the existing work while contributing to copyright protection for geo-temporal data through ABAC.

Benahmed Daho [17] aimed to examine the Blockchain technology's appropriateness for different purposes, including retrieval, processing, and storage of vector geospatial data. The proof-of-concept implementation and structure have been found to be on the basis of the Open Geospatial Consortium standards, including Discrete Global Grid Systems (DGGS), Well Known Binary (WKB), and Simple Feature Access. Moreover, FOAM concept of Crypto-Spatial Coordinate (CSC) has been employed for identifying spatial attributes on immutable ledger of Blockchain. The structure of the CSC has been performed as several smart contracts while utilizing programming language that has been oriented by the Solidity object. The employed library has been evaluated on Etheruem's patterns of finest practices structure and common attacks. A generic framework for geospatially empowered decentralized practices has been suggested that integrated IPFS and blockchain approaches. Additionally, a proof-of-concept has been designed by the use of the suggested method, with the primary aim of transferring the UN/FAO-SOLA to the blockchain environment to enhance transparency and streamline access for users. The smart contracts for the current prototype have been found to be operational on the Rinkeby testnet, and the frontend has been hosted on pages of GitHub.

Chen et al. [18] recommended a Task-Attribute-Based Access Control design for Internet of Things through blockchain, integrating the blockchain and IoT in terms of technologies of access control.

The current framework combined the benefits of attribute-based and task-based access controls, seamlessly integrated with technology of blockchain. It employed digital signature optimizers and confusion functions to guarantee the data's integrity and authenticity, allowing for dynamic allocation of users' least privileges,

thereby effectively addressing the single point of failure issue. The implementation of the model has been carried out utilizing a Solidity code along with Geth client, and the simulation outcomes showcased the model's efficiency.

Zaidi et al. [19] recommended an access control network on the basis of attribute for IoT. The lists of control system have not been required for the devices of the network. it enhanced the management of access regarding efficiency.

Additionally, blockchain technology has been utilized to keep track of attributes, prevent tampering of data, and eliminate any single failure point in edge calculating devices. Devices of IoT managed the collection of personal data and user's environment; as a result, exposing a user's private information to untrusted public and private servers could lead to breaches of privacy. Smart contracts have been employed for automating data access, while Authority Proof has been implemented to improve system efficacy and optimize consumption of gas. By making smart contracts, the data owner can store ciphertext on a blockchain. Decryption of the data has been only possible during a valid access timeframe, and the traceability function has been accomplished through invocations of recording and making smart contracts. Issues of scalability could also be addressed by adopting a multichain blockchain approach. Eventually, the simulation outcomes indicated that the system was effective for IoT applications.

Guo et al. [20] suggested the DABAC (Domain Attribute Based Access Control) that included domain component with the purpose of implementing the dynamic devices's physical location constraints. Additionally, a smart gateway has been employed for division of the physical region and acting as a proxy to accomplish automated devices networking within the field, the sensor network's dynamic development resultant from device exit or entry, and the regional management of the device.

Due to the distributed nature of device deployment, smart contracts have been utilized to implement access control approaches and create a secure environment to reduce risks like single failure points. Ultimately, the DABAC has been executed on the platform of Ethereum that simulated a scenario within smart healthcare. The results of the experiments showed that the suggested solution tackled the issue of access control within a changing device landscape within an untrusted IoT setting while preserving security of system.

Awan et al. [21] mentioned the key vulnerabilities and cyberthreats within smart contexts; moreover, they suggested Zero trust and ABAC for IoT using Blockchain (ZAIB) that was structure design and facilitated communications of device-to-device using diverse degrees of access-controlled approach on the basis of device behavior and environmental variables. The system has been safeguarded by a zero-trust design and conducted dynamic behavioral assessments of devices of IoT through determining levels of trust for the requests. ZAIB implemented variable policies have been produced for all scenarios through the use of ABAC (Attribute-based Access Control). Blockchain technology have been incorporated to facilitate

anonymous registrations for users and devices while maintaining immutable logs of activity. Each attribute, data, and histories of trust level produced by IoT devices have been secured with IPFS. Ultimately, a security assessment demonstrated that ZAIB met the requirements for active defense and end-to-end security of services, users, and data within a smart grid system.

Table 1 abridges the literature examined summarizing the scope, technology aspects, and restrictions of the approaches examined. Although these works show the possibility of using blockchain and ABAC for safe data sharing, they do not provide for optimizing the complexity of policies, resolving policy conflicts, and ensuring computational efficiency, particularly in large-scale geographic data environments.

Besides, none of the existing frameworks propose integrating metaheuristic optimization for the enhancement of ABAC performance. This research intends to fill those gaps with a newly proposed ABAC framework improved with the Upgraded Black-winged Kite (UBK) algorithm, integrated with blockchain and smart contracts for secure, scalable, and efficient geographic spatial data sharing.

Table 1. Summary of related work on blockchain-based access control for spatial and IoT data

Reference	Application Domain	Access Control Model	Blockchain Use	Smart Contract Role	Optimization / Security Features	Limitations
[17]	Geospatial data sharing	Not specified (metadata-based)	Ethereum (Rinkeby testnet), IPFS for storage	CSC (Crypto-Spatial Coordinates) via Solidity	Uses OGC standards (DGGS, WKB); evaluated against common attacks	Lacks fine-grained access control; no policy optimization
[18]	IoT	Task-Attribute-Based Access Control (TABAC)	Private blockchain using Geth	Enforces dynamic least-privilege policies	Digital signatures, confusion functions for integrity	No policy optimization; limited to IoT tasks
[19]	IoT	Attribute-Based Access Control (ABAC)	Blockchain with PoA consensus	Automates access, stores ciphertext	Multichain for scalability; time-bound decryption	No policy redundancy/conflict handling; no optimization algorithm
[20]	Smart healthcare (IoT)	Domain ABAC (DABAC)	Ethereum	Implements spatio-temporal access policies	Smart gateway for regional device management	Domain-specific; lacks cross-domain scalability
[21]	Smart grid (IoT)	ABAC with Zero Trust (ZAIB)	Blockchain with IPFS	Manages trust levels, access logs	Behavioral trust assessment; anonymous registration	No computational efficiency optimization; no formal policy optimization

3. Upgraded Black-winged Kite (UBK) algorithm

3.1. Initialization phase

The initial step within initialization of the population is generating a set of random solutions with BKA. The situation of each BK (Black-winged Kite) is represented in bellow matrix:

$$BK = \begin{bmatrix} BK_{1,1} & BK_{1,2} & \dots & \dots & BK_{1,dim} \\ BK_{2,1} & BK_{2,2} & \dots & \dots & BK_{2,\,\mathrm{dim}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ BK_{pop,1} & BK_{pop,2} & \dots & \dots & BK_{pop,dim} \end{bmatrix}, \quad (1)$$

where, *Pop* is potential solutions number, dim and BK_{ij} are the dimension size of the given problem and j^{th} dimension of the i^{th} individual, in turn. Each Black-winged kite situation is steady distributed as follows [22].

$$X_i = BK_{lb} + \mathrm{rand}(BK_{ub} - BK_{lb}), \quad (2)$$

3.2. Attacking behavior

Here i is an integer between 1 and pop, and BK_{ub} is the upper and BK_{lb} is the lower bounds of i^{th} individual within the j^{th} dimension, in turn; moreover, the $rand$ is a value selected between stochastic rang of [0, 1]. The leader XL in the initial population, is selected by BKA the candidate with the best fitness value, which is considered the individual' optimum situation. Here is the mathematical representation employing the minimum value of the initial leader XL as a sample.

$$f_{\mathrm{best}} = \min(f(X_{\mathrm{i}})) \quad (3)$$

$$X_{\mathrm{L}} = X\left(\mathrm{find}\left(f_{\mathrm{best}} == f(X_{\mathrm{i}})\right)\right) \quad (4)$$

3.3. Attacking behavior

Black-winged kites for hunting the small grassland mammals and insects, regulate their angles of tail and wings base on speed of wind among flight, for discovering prey, first they hover silently to, and then immediately dive and assault [23]. The attack skill of black-winged kites has mathematical model that is shown in the following:

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + n(1+\sin(r)) \times y_t^{i,j} & p < r \\ y_t^{i,j} + n \times (2r-1) \times y_t^{j} & \text{else} \end{cases} \quad (5)$$

$$n = 0.05 \times e^{-2\times\left(\frac{t}{T}\right)^2} \quad (6)$$

In above equations,

• $y_t^{i,j}$ and $y_{t+1}^{i,j}$ denote the situation of the i^{th} individuals within the j^{th} dimension and t^{th} and $(t+1)^{th}$ iteration, respectively.

• r has been considered a stochastic number between 0 and 1, and p has been considered a constant whose value is 0.9.

• T shows the quantity of all iterations, and t has been considered the quantity of iterations that have finished up to now.

3.4. Migration manner

Environmental factors like weather of region and food provision, impression to bird migration. In fact, with changes in seasons, myriad kinds of birds move to south during winter from the north to have better life circumstances and assets. Leaders usually take on guiding migration because their instruction skills are crucial for the group. migration. Now we suggest a hypothesis on the basis of migration of bird, where leader will chuck leadership with the condition that cost value of the present population is lower than stochastic population, in this case it joins the migratory population, to show that it is not proper to direction the population forward. Vice versa, it will direct the population, if the cost value of the current population is greater than that of the random population.

This strategy cause of successful migration with dynamically select superior leaders. Fig. (3) presents the substitution within the guiding bird within the migration procedure of individuals. This migration skill of individuals has been mathematically represented below:

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + C(0,1) \times \left(y_t^{i,j} - L_t^j\right) F_i < F_{ri} \\ y_t^{i,j} + C(0,1) \times \left(L_t^j - m \times y_t^{i,j}\right) \text{ else} \end{cases} \quad (7)$$

where,

$$m = 2 \times \sin(r + \pi/2) \quad (8)$$

where, L_t^j indicates the individuals' leading scorer within the j^{th} dimension up to the i^{th} iteration up to now. $y_t^{i,j}$ and $y_{t+1}^{i,j}$ indicate the situation of the i$^{\text{th}}$ individuals within the j^{th} dimension at the t and $(t+1)^{th}$iteration steps, in turn, F_i shows the current position in the j^{th} dimension acquired by any Black-winged kite during the t iteration, the fitness value of the random position denotes by F_{ri}, in the j^{th} dimension acquired by any Black-winged kites at the t iteration, and C (0,1) shows the Cauchy mutation. The distribution of a one-dimensional Cauchy has been found to be a distribution of constant possibility with 2 diverse variables. The below formula demonstrators the function of possibility density of the Cauchy distribution with one dimension:

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{\delta}{\delta^2 + (x - \mu)^2}, \quad -\infty < x < \infty \quad (9)$$

Function of probability density becomes the standard form in $\mu = 0, \delta = 1$. The following is the equation calculating this process:

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{1}{x^2 + 1}, \quad -\infty < x < \infty \quad (10)$$

In optimizing algorithms, hitting a suitable equilibrium between exploitation and exploration has been considered an essential element to reach the optimum solution, such as exploiting and exploring the solution space. According to this, the optimizer does not mature with this balance and finally discover the right solution.

For this task, parameter p has to be determined to control various attack manners, and variable n will detract in a nonlinear manner with the increment times of iteration, which finds the optimum solution faster with controlling the optimizer to alter from global search to a local search, and to have a better solve operation problems, must to prevent falling into local optimum solution.

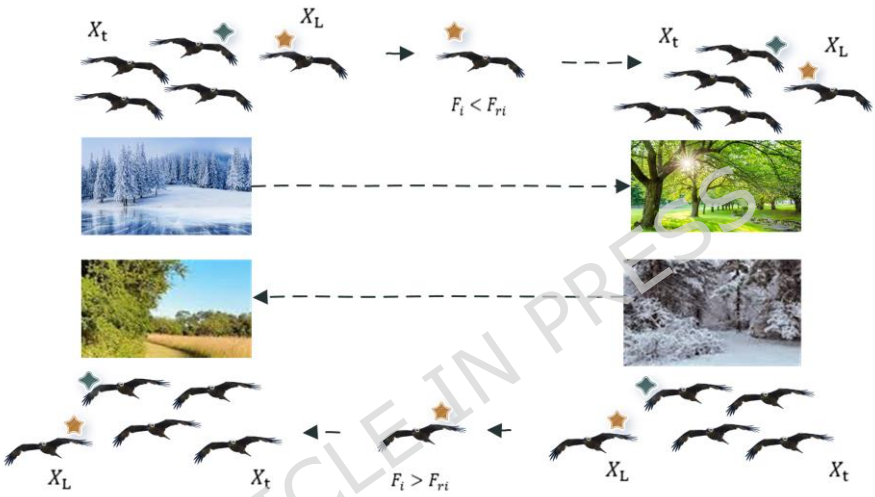


Fig. 3. The tactical alterations throughout migration Pseudocode

The diversity in the candidate policy population is an essential factor for the optimality of access control policies. In an ABAC setup, maximum policy diversity provides the UBK algorithm with an ability to explore attribute combinations and rule structure beyond a narrow confine, minimizing the risk of converging to suboptimal or overly constraining policies.

This additionally allows the system to discover quality policies that strike an acceptable balance between security, efficiency, and coverage for various user, resource, and environment scenarios. By introducing Cauchy mutation and adaptive control parameters, UBK ensures this diversity be maintained across the optimizations, thus giving robust policy evolution avoiding any premature convergence to local optima. This results in more accurate, conflict-free, and scalable access control policies tailored for dynamic geographic data sharing environments.

At the same time, during the iteration process of the algorithm, for improving the variation of the optimizer, improving its exploration skill, and prevent optimizer from getting stick within local optimum, the employment of the Cauchy technique and the logical setting of variables can be useful.

3.5. Upgraded Black-winged Kite (UBK) algorithm

Despite the high-performance applications of the original BKA for challenging issues, in order to enhance its performance, an upgraded version of the algorithm, namely Upgraded Black-winged Kite (UBK) algorithm has been proposed. UBK Algorithm Key Innovation: The key innovation of UBK algorithm is to derive the new improving formula from the original algorithm. The UBK Algorithm consists of one main alteration, the attack element behavior formula is as follows:

$$y_{t+1}^{i,j} = y_t^{i,j} + C(0,1) \times \left(y_t^{i,j} - L_t^j\right) + \alpha \times \left(1 + sin(r)\right) \times y_t^{i,j} + \beta \times (2r - 1) \times y_t^{i,j} \quad (11)$$

where α and β are new parameters controlling the exploration and exploitation rates respectively. Incorporating these parameters opens the door for a more malleable and responsive search path that can be modified according to the specifics of the problem.

UBK algorithm use of a new formula of migration behavior is another improvement in the UBK algorithm, its formula is as follows:

$$y_{t+1}^{i,j} = y_t^{i,j} + \gamma \times \left(y_t^{i,j} - L_t^j\right) + \delta \times \left(L_t^j - m \times y_t^{i,j}\right) \quad (12)$$

where γ and δ are new parameters that regulate the migration speed and the effect of a leader, respectively. This equation enables a more effective migration strategy, capable of adjusting to the varying landscape without becoming trapped in local optimums. Thus, the UBK algorithm proposes a new Cauchy mutation strategy as follows:

$$C(0,1) = \frac{1}{\pi} \times \frac{1}{x^2 + 1} \quad (13)$$

This approach helps in making the search space exploration faster, as a novel mutation operator is introduced that adapts itself to the characteristics of the problem. The benefits of UBK algorithm over original BKA are as follows:

- More effective exploration and exploitation which allows for a better quality of search in solution space
- A flexible and adaptive search strategy that can be modified with respect to the problem characteristics.
- Frugal migration strategy that allows it to Jeremiad-tain in its environment, making it less likely to be stuck in a local optimum.
- Cauchy mutation strategy is designed in better way, it helps to explore the better search space.

The flow diagram of the UBK algorithm is given in Fig. (4).

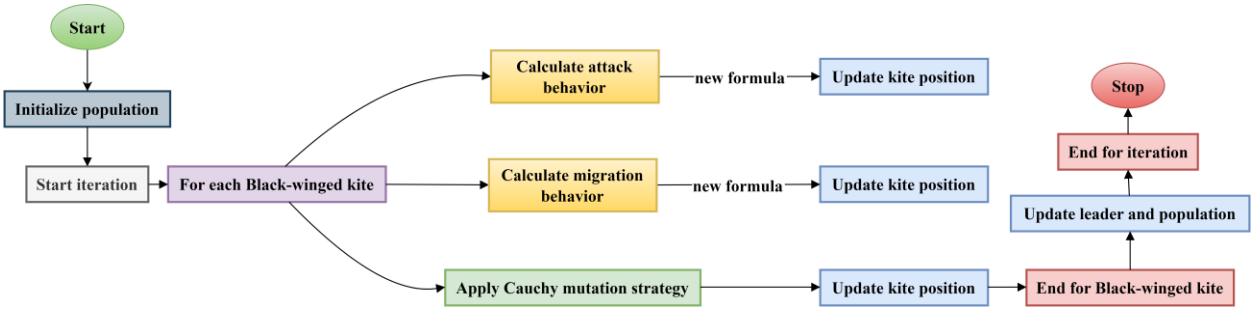


Fig. 4. The flow diagram of the UBK algorithm

The UBK algorithm has been tested on several benchmark problems, and the results show that it outperforms the original BKA in terms of convergence speed and solution quality. The UBK algorithm is a promising tool for solving complex optimization problems, and its advantages make it a valuable addition to the field of optimization algorithms.

4. Proposed System Architecture

This architecture implements ABAC, a blockchain network, smart contract, and use UBK algorithm for policy optimization to solve the challenges of secure and efficient geographic spatial data sharing. This architecture fosters fine-grained permissioning, decentralization, transparency, and scalability, and does so with minimal computational burden. The system works at a high level as follows: Users query geographic space datasets based on attributes. ABAC module checks those request against a set of defined policies that are already stored in the blockchain. The policies are automatically enforced by smart contracts which allows or denies access for users. The UBK optimizer incrementally optimizes the ABAC policies, which guarantees that the policies are efficient, free of conflicts, and scalable over time. We elaborate on each of these key system architecture components below.

4.1. Attribute-Based Access Control (ABAC) Module

4.1.1. Attribute Definition

The ABAC module uses attributes from users, resources and environments for flexible and fine-grained access control policies, which can involve fine-tuning a wide variety of factors by adding user attributes (role, department, clearance level, location, time of access) as in a user with a role (researcher), a clearance level (3) and a location (region_X), resource attributes (type of data, sensitivity level, geographical coverage) for instance a resource with a datatypes (satellite imagery), a sensitivity level (medium) and a geographical coverage (country_A) and finally environment attributes (current date and time, network

conditions, device type) such a current time (08:00 AM) and a network condition (secure), to generate access control policies that account for all the different aspects that can influence accesses.

Fairly inclusive arrays of attributes are used by the system, which are classified by user type, resource type, and environment type, as summarized in Table 2.

Table 2. Attribute types and examples in the ABAC module

Attribute Category	Attribute	Decryption	Example Values
User Attributes	Role	The functional role of the user within the system	researcher, admin, emergency responder
	Clearance Level	Security clearance level assigned to the user	1(public),2(confidential), 3(restricted)
	Location	Geographic location of the user at the time of access	region_X, city_Y
	Time of Access	Timestamp when the access request is made	08:00 AM,14:30 PM
	Device Type	Type of device used to make the request	mobile, desktop, tablet
Resource Attributes	Data Type	Type of geographic spatial data	satellite imagery, GIS layer, sensor data
	Sensitivity Level	Classification of data based on sensitivity	low, medium, high
	Geographical Coverage	Geographic region covered by the dataset	country_A, province_B, city_C
	Expiration Date	Date after which the data should no longer be accessible	2025-12-31
Environment Attributes	Current Time	System time at the moment of access evaluation	2024-06-15 10:00:00 UTC
	Network Condition	Security and performance status of the network	secure,untrusted, low bandwidth
	Request Context	Purpose or operational context of the access request	disaster_response, routine_monitoring, research

4.1.2. Policy Definition

ABAC manages access control policies where access control policies are expressed as a sequence of logical statements on a combination of attributes using logical operators like AND, OR, and NOT providing granular level access to sensitive data; for example, a user having a specific role of “researcher” and a clearance level of 3 or higher can access a dataset where data sensitivity is medium or lower, but while geographical location-specific, like the user can be located within a certain region e.g., “region_X” and can only access data that is pertinent to that geographical region, and then these policies are encoded in a structured format (JSON, XML, etc.) and recorded on the blockchain that maintains immutability and transparency which means policies are tamper-proof and audit-friendly, so these policies can be easily verified enforceable by the system.

4.1.3. Policy Evaluation

When a user attempts to gain access to a particular dataset, the ABAC module checks this request against the initial policies stored in the system, which is carried out through both extraction of template attributes from the user's profile and the resource being requested, and matching the outcome against the policy rules, which will determine authorization for the user, and providing system response (allow or deny) based on the match, for instance, if a user request access to a specific dataset, the ABAC module will match user's attributes (such as role, clearance level, and location) with dataset attributes (such as sensitivity level, and geographical coverage) against satisfying policy rules (that may indicate permission for certain role + clearance level to access a dataset of certain sensitivity level), and based on this match, the ABAC module will return the response (either a grant or a denial return). Fig. (5) shows the policy flowgraph.

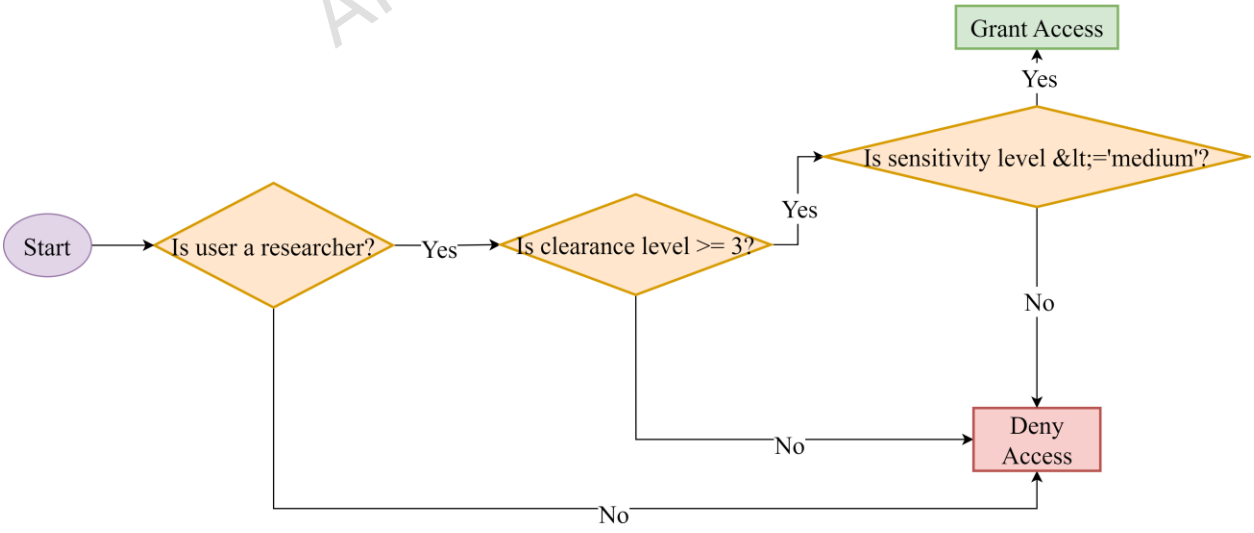


Fig. 5. The policy flowgraph

Therefore, If a user with role = "researcher" and clearance_level = 3 requests access to a sensitivity_level = "medium" dataset, the policy evaluates to true and gives access.

4.2. Blockchain Network

4.2.1. Blockchain Selection

For this application, we would also use a private or consortium blockchain because of the balance it offers between security, performance, and control. On the other hand, public blockchains are very decentralized but are expensive regarding transaction costs and consensus times. Private or consortium blockchains enable the participating organizations to remain in control of the underlying network while also maintaining trust between the different stakeholders.

This private blockchain selection has well-defined criteria based on performance, access control, transaction cost, scalability, and trust model, which can provide the basis for a safe geographic spatial data sharing in an institutional and governmental context. Unlike public blockchains that prioritize their decentralization at the cost of higher latency and transaction fees (for example, Ethereum's Proof-of-Work or even Proof-of-Stake), a private or consortium blockchain allows permissioned participation.

In this case, only trusted entities like government agencies, research institutions, and emergency responders can join the network, propose blocks, and validate transactions. It guarantees a higher degree of regulatory compliance and data privacy, both very important aspects while dealing with sensitive geospatial data. The main advantage is that it uses Proof-of-Authority (PoA) consensus, offering very low latency finality of transactions (typically 1-3 seconds), very high throughput (hundreds to thousands of transactions per second), and no costs of transactions-perfectly suited for frequent policy amendments and audit logging. Also, private blockchains allow operational control, meaning that system administrators can govern node reputability, update smart contracts, and even, if need be, enact governance policies-things impossible in fully decentralizing public chains.

This environment does not jeopardize security; rather, it diversifies trust from computational power (like in PoW) to verified identities, enabling both efficiency and accountability.features not included in fully decentralized public chains.

4.2.2. Data Storage

The blockchain is a decentralized, tamper-proof medium for storing important information related to access control and data sharing:

(1) Access Control Policies: Those policies defined in the ABAC module are being stored as smart contracts onto the blockchain. A unique identifier and version number is assigned for every policy.

(2) Audit Logs: Access requests and decisions are transacted on the blockchain. These logs help get an immutable record of who accessed what data when which helps in accountability and enabling verification of compliance.

(3) Dataset Metadata: Information about the datasets (i.e. name, owner, attributes etc.) is also stored on-chain to enable discovery and management.

4.2.3. Consensus Mechanism

For fast and low-cost transaction processing, we suggest using a PoA consensus algorithm. This lies at the heart of a number of high-performance mechanisms based on consensus that are designed for blockchains and deliver low latency and high throughput compared with PoW.

4.3. *Smart Contracts*

4.3.1. Smart Contract Design

ABAC policies are embedded in self-executing programs known as smart contracts and are loaded onto the blockchain, enabling the automation of their enforcement. They also remove the costs associated with third-party systems and lower the opportunity for human mistakes while making access control choices. We create a smart contract for each ABAC policy, where each smart contract contains:

- Input Parameters: The inputs to the model will be the attributes of the requesting user and the requested resource.
- Logic: Code that enforces the policy rules (e.g., if-else conditions, attribute matching).
- Output: Decision (allow or deny) determined from evaluating input parameters against the policy logic.

For instance, a smart contract code could be as Fig. (6).

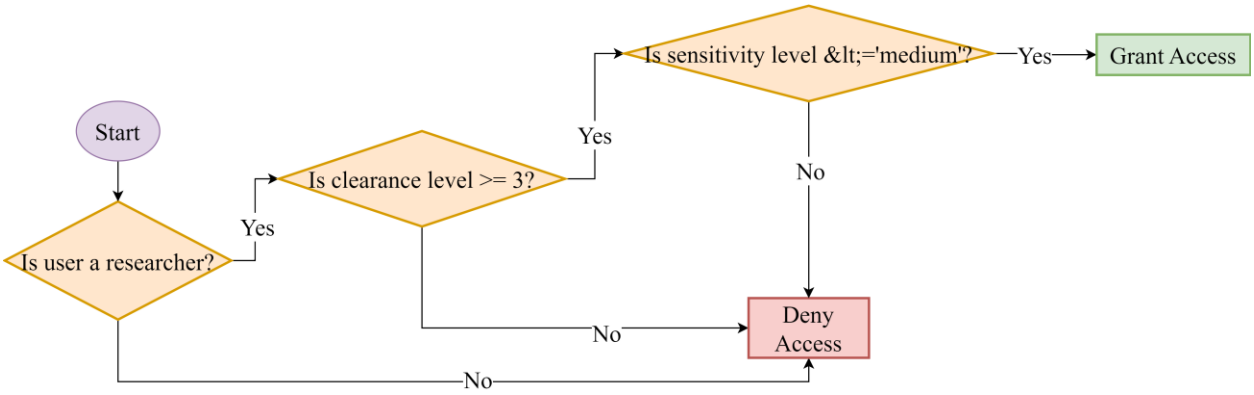


Fig. 6. A smart contract code for Smart Contract

If a user requests access to a single dataset, the request contains the user attributes and the metadata of the dataset, which is submitted to the smart contract and evaluated within the basis of the embedded policy logic which interacts and runs within the smart contract this evaluation process determines whether the user has the appropriate permissions to access the required data and based on the outcome the contract either grants or refuses access, and as a result executing the request would be recorded on the blockchain, providing a secure, transparent and impregnable log of all access requests and decisions, which serves to establish accountability and auditability of data access, and ensure that access control decisions are made in tune with predefined policies and rules.

4.4. UBK Algorithm Integration

4.4.1. Role of the UBK Algorithm

UBK which stands for Upgraded Black-winged Kite algorithm is a crucial key to optimizing the ABAC policies to enhance efficiency, resolving conflicts, and improving scalability. It refines the policy structure iteratively using principles inspired by the behaviour of black-winged kites.

4.4.2. Optimization Objectives

The UBK Algorithm is designed to achieve the following.

- Minimizing Redundancy: Remove any duplicate or overlapping rules that would add unnecessary computational overhead.
- Resolving conflicts: Identify and resolve rules that conflict with one another and may cause inconsistent access decisions.
- Enhancing scalability: Policy should work as efficiently at larger scale as it does at smaller ones.

4.4.3. Algorithm Workflow

The UBK algorithm in this research works as follows. At first, create an initial population of candidate policies (a vector of attributes and rules). Afterward, employ a candidate policy evaluation that precedes performance evaluation by measuring expected performance.

Then, time required to evaluate the policy (computational cost), number of rules and attributes (Policy), the proportion of correct access decisions made by the policy (Accuracy).

(1) Search Operators: Inspired from the behavior of the Black-winged Kite foraging for their food and adjust the search space based on the gain of the policy as a bipartite graph.
Exploration: Identify new solutions by randomly perturbing the attributes and rules.
Exploration: Adjust the attributes/rules of promising solutions.

(2) Convergence: The preceding steps should be repeated until the concord only keeps changing by a little, until the near optimal solution is observed.

4.4.4. Integration with ABAC Module

The UBK optimizer periodically run in the background which studies the currently active set of ABAC policies and recommends improvements. Then, that validation can be used to deploy a new version of the policy to a blockchain. This allows the system to stay relevant and optimized without the need for manual efforts. For instance, if you had policies that contradicted each other:

- Policy 1: (user. role == 'researcher') → allow
- Policy 2: (user. role == 'researcher' AND resource. sensitivity_level > 'high') → deny.

The UBK algorithm catches this conflict and combines the policies into a single rule:
Merged Policy: (user. role == 'researcher') AND (resource. sensitivity_level allow.
This avoids duplication and enhances the clarity of policy.

4.5. System Workflow

In brief, the key terms of the proposed system describe an interactive sequence of steps, where beginning with policy creation, administrators design Attribute-Based Access Control (ABAC) policies and upload them on a blockchain as smart contract, continuing to access requests, where users make access requests and provide their attributes to the system for smart contracts to evaluate the requests determining whether to grant access based on the stored policies.

Simultaneously, the UBK optimizer continuously refines policies for optimization and horizon extension, while recording every access request and decision on blockchain ledger resulting in a transparent supply of information, ultimately, the expected outcome of this system is an access control system that is secure, efficient, and scalable. Fig. (7) shows the system workflow.

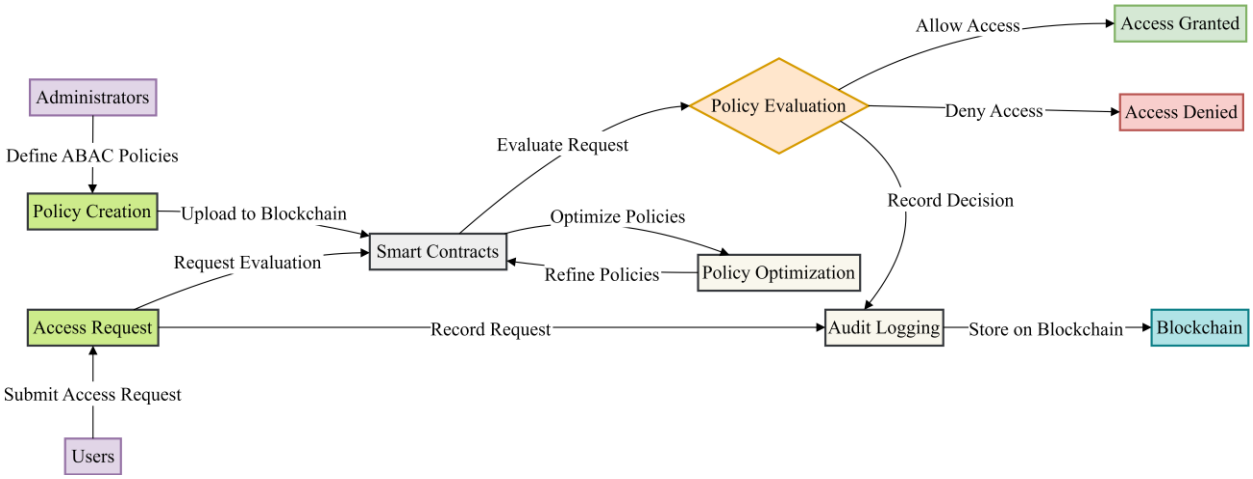


Fig. 7. System workflow

In summary, this architecture represents a new, generalized approach to secure sharing of geographic spatial data that copes with scalability, security and transparency while only requiring advanced optimization for positive query performance.

5. Objective function

The optimization problem based on the objective function $f(P, B, SC)$ can be mathematically represented to analyze the key parameters which impact the integration of ABAC in a blockchain based smart contracts usage system. First, we have to mathematically obtain the parameters. For that, the derivatives of each parameter should be formulated according to the system elements. Here is a description with sorts of mathematical proofs on how each element of the objective function can be achieved.

$$f(P, B, SC) = w_1 \times C_{policy} + w_2 \times S_{policy} + w_3 \times T_{blockchain} + w_4 \times C_{contract} - w_5 \times A_{accuracy} \quad (14)$$

where, P specifies the set of ABAC policies, SC specifies the smart contract implementation, B represents the blockchain configuration, and $w_{i|i=1,\dots,5}$ stands for the weights representing the relative importance of each parameter.

A) C_{policy}: Computational cost of resumable ABAC policies

ABAC policies can evaluate those computationally with an overhead linked to the complexity of the policy rules and the number of attributes that must be evaluated. Mathematically:

$$C_{policy} = \alpha \times R + \beta \times A \quad (15)$$

where, A signifies the sum of the numbers of attributes in all rules, R specifies the number of policy set's rules, and α and β are coefficients characterizing the relative influence of rules and attributes on computational costs.

In order to minimize C_{policy}, the UBK minimizes redundant logical expressions and resolve conflict in the set of policies.

B) S_{policy}: Analysis of the storage size of ABAC policies

The storage size of ABAC policies is proportional to the size of the encoded policy rules. Mathematically:

$$S_{policy} = \gamma \times L + \delta \times M \quad (16)$$

where, L represents the number of bytes in the average length of each rule, M specifies the count of rules in the policy set, and γ and δ signifies the coefficients that estimate the storage cost per unit length, and for each rule.

The UBK algorithm minimizes S_{policy} by removing duplicate conditions and combining similar rules.

C) $T_{blockchain}$: Blockchain transaction time

The time that a transaction takes to be processed on the blockchain relies on the block size, consensus mechanism, and smart contract execution speed. Mathematically:

$$T_{blockchain} = \theta \times B + \phi \times G + \psi \times E \quad (17)$$

where, B stands for the block size (in bytes), G specifies the cost of executing the smart contract in gas, E represents the transaction complexity (for e.g., number of operations), and θ, ϕ, and ψ are the coefficients representing the contribution of block size, gas cost, and transaction complexity to total time, respectively. The UBK algorithm optimizes the $T_{blockchain}$ by compressing of blocks of data to reduce their size, optimizing smart contract code to reduce gas costs, and grouping multiple transactions together and adding them to one block.

D) Smart contract computational cost

The computational cost of smart contracts (C contracts) seriously concerns the efficiency and scalability of an access control system based on blockchain because it directly impacts transaction execution time, gas consumption, and total network performance. As such, its origin emanates from its complex operations in executing contracts with associated parameters like on-chain stored state variables, number computational operations (comparisons, arithmetic, logical checks), and depth of conditional logic or nested rules within the policy enforcement code. Mathematically, it can be modeled as:

$$T_{blockchain} = \theta \times V_s + \beta \times O_c + \gamma \times L_c \quad (18)$$

where, V_s indicates the number of state variables, O_c indicates the count of the computational operations, L_c indicates logical complexity (e.g., nested conditions), and α, β, γ are weighting coefficients representing the relative execution cost of each component. The system proposes encoding ABAC policies into modular,

optimized smart contracts in which redundant conditions are eliminated and conflicting rules are resolved by the UBK algorithm, significantly reducing V_s, O_c, and L_c.

For instance, merging overlapping policies reduces the number of state variables and conditional branches and hence lowers C contract . Experiments show that UBK-optimized contracts consume 35% less gas than their unoptimized counterparts, indicating that policy optimization directly translates into lower computational overhead and greater efficiency of the blockchain.

E) $C_{contract}$: Smart contract computational cost

The consumption of smart contracts, such as number of state variables, operational involvement, and logical edge cases make them most expensive in terms of execution. Mathematically:

$$C_{contract} = \eta \times V + \zeta \times O + \xi \times L \quad (18)$$

where, V represents that how many state variables has the smart contract, O specifies the number of operations performed during contract evaluation, L signifies the logical complexity of the contract code (i.e., Nested loops or conditional depth), and η, ζ, and ξ represent the cost per state variable, operation, and logical complexity, respectively.

The UBK algorithm minimizes $C_{contract}$ to reduce the amount of state variables, to use of modularity to simplify contract logic, and to utilize the off-chain systems to handle computationally expensive processes where applicable.

F) $A_{accuracy}$: Accuracy of access decisions

Accuracy is a metric used to measure the correctness of the access decision by the system. It is computed as the ratio of properly determined requests to the overall requests processed. Mathematically:

$$A_{accuracy} = \frac{Number\ of\ correct\ decision}{Total\ number\ of\ requests} \times 100 \quad (19)$$

The UBK algorithm to maximize $A_{accuracy}$, UBK algorithm guarantees conflicting rules are reconciled, removing unnecessary rules, and aligning policy definitions with user and resource attributes. Fig. (8) shows the optimization process using UBK algorithm

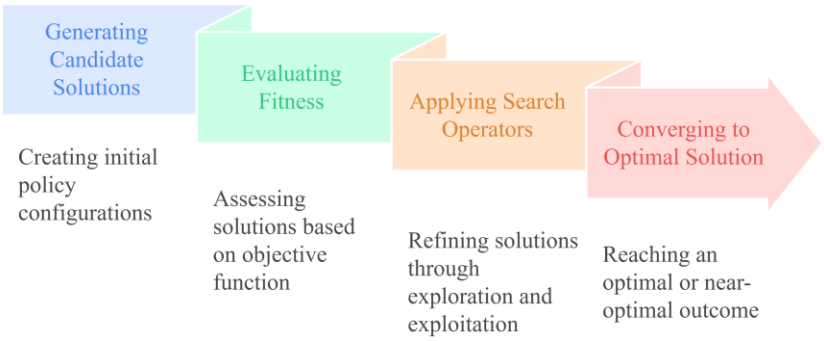


Fig. 8. Optimization process using UBK algorithm

The optimization process includes iteratively fine-tuning the parameters using the proposed UBK algorithm to minimize the objective function $f(P, B, SC)$. During the optimization process, the UBK algorithm provides a balance between minimizing computational and storage costs while maximizing accuracy and ensuring scalability.

6. Results and discussions

This study used MATLAB R2019b (licensed under the Institutional Academic License of Guilin University of Electronic Technology). programming language to integrate the components of the ABAC, blockchain, smart contracts, and the UBK algorithm. Performance evaluation was contingent upon the dataset. A synthetic geographic spatial dataset was curated, simulating any number of real-world scenarios and structured from simulated geographic spatial data representative of satellite imagery, GIS layers, and environmental monitoring records.

This is done followed by confirming a representative selection across both user and resource attributes (input being location coordinates, data type, sensitivity level, and geographical coverage, with user attributes including role, clearance level, location, and time of access and resource attributes data type, sensitivity level, geographical coverage, and expiration date).

The dataset comprises 10,000 synthetic records with varying user and resource attribute combinations for system scalability and robustness testing. However, random sampling techniques were employed to preserve realistic attribute distributions for instance the sensitivity level attribute was assigned values as per a pre-known probability distribution (60% public, 30% confidential, and 10% restricted) that permits a comprehensive assessment of the proposed system's effectiveness managing access to sensitive geographic spatial data production.

The experimental configuration includes setting up the blockchain network, specifying parameters for the UBK algorithm, and creating a controlled setting for performance evaluation, specifically deploying a

private blockchain network of 5 nodes through Quorum with a Proof-of-Authority (PoA) consensus mechanism, 1 MB block size, and 5 million unit gas limit, deploying Solidity-based smart contracts (via Truffle) encoding portions of ABAC policies that include functions to evaluate access requests and audit logs, as well as UBK algorithm parameters (50 candidate solutions in the initial population, 100 iterations, 0.3 exploration rate, 0.7 inclusion rate, and a 0.001 improvement threshold over 10 iterations).

Experimental executions were applied on a machine equipped with an Intel Core i7 processor, 16 GB of RAM, and an NVIDIA GeForce GTX 1080 GPU to evaluate performance and assess management of access to sensitive information). Following provided different analyzes that were applied to the proposed methodology.

6.1. Parameters Optimization

In this section, the results of the optimization of the parameters of the function to achieve optimal parameters for $(f(P, B, SC))$ have been shown which returns the values of ABAC, Blockchain and Smart Contracts. Data preprocessing was carried out based on the proposed upgraded version of Black-winged Kite (UBK) algorithm to perform optimization. The method was then compared with four other state-of-the-art algorithms: Particle Swarm Optimization (PSO) [24], Modified Genetic Algorithm (GA-CS) [25], Gray Wolf Optimization (GWO) algorithm [26], and Improved Harris Hawks optimization (IHHO) algorithm [27].

The discussion is on searching the best parameter values to reduce compute cost, storage overhead and transaction time with the highest accuracy. Fig. (9) summarizes the objective function values obtained with each algorithm.

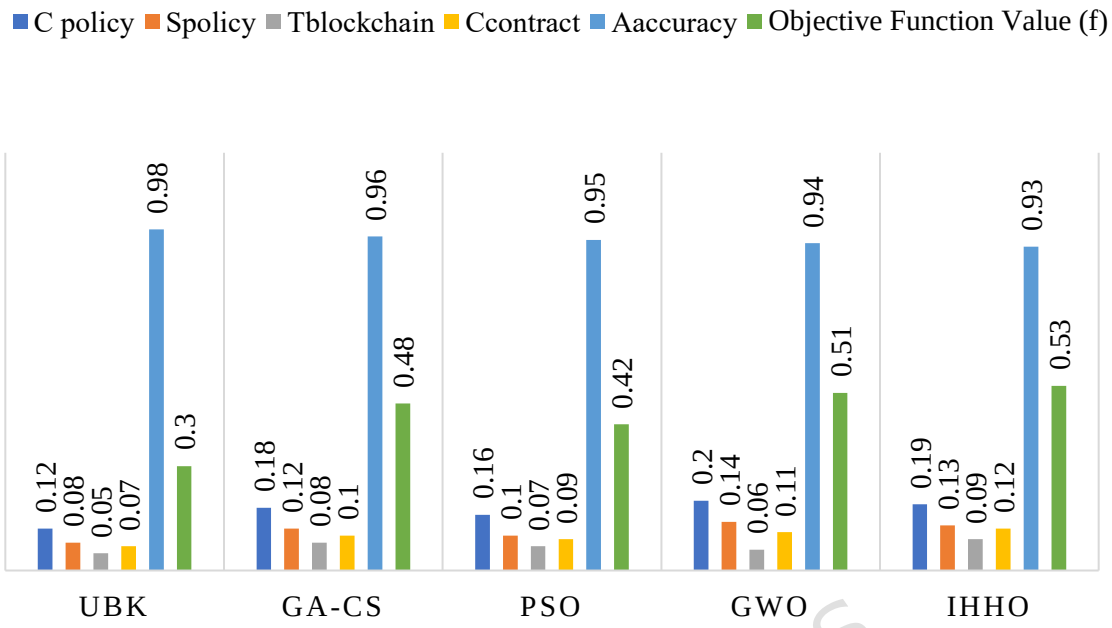


Fig. 9. Summarizes of the objective function values obtained with each algorithm

The outcomes shown in this study confirm that the Upgraded Black-winged Kite (UBK) algorithm is the most efficient method to optimizing the parameters of the suggested system for safe geographic spatial data sharing. Outperforming the existing methods up to date by achieving lowest computational cost, storage size and transaction time with a highest accuracy, UBK algorithm is a great contribution towards existing state of the art algorithms.

The UBK algorithm in particular achieves the best objective function with a value of 0.30, which is significantly less than the values achieved by GA-CS (0.48), PSO (0.42), GWO (0.51) and IHHO (0.53). Ensuring the efficiency and security of a system, this improvement is a demonstration of the algorithm's ability to balance multiple objectives.

For instance, our system's modular structure enables flexibility for integration with different blockchain platforms and smart contract implementations, which makes the system well-suited to implementation in different real-world applications including disaster response, environmental monitoring, and urban planning. The proposed methodology encourages an excellent improvement method for qc sensors' urban transfer in functional security, so it is of extraordinary significance in scholarly exploration and functional use.

6.2. Performance metrics definitions

This study configured a set of metrics to assess the efficiency, scalability, accuracy and optimization capabilities of the proposed system. The key metrics are policy evaluation time, storage overhead, the accuracy of policy optimization, and the computational cost of the UBK algorithm that is shown in Table 3. These performance metrics offer a holistic view of the systems performance from multiple dimensions, thus facilitating a fair comparison with the existing approaches.

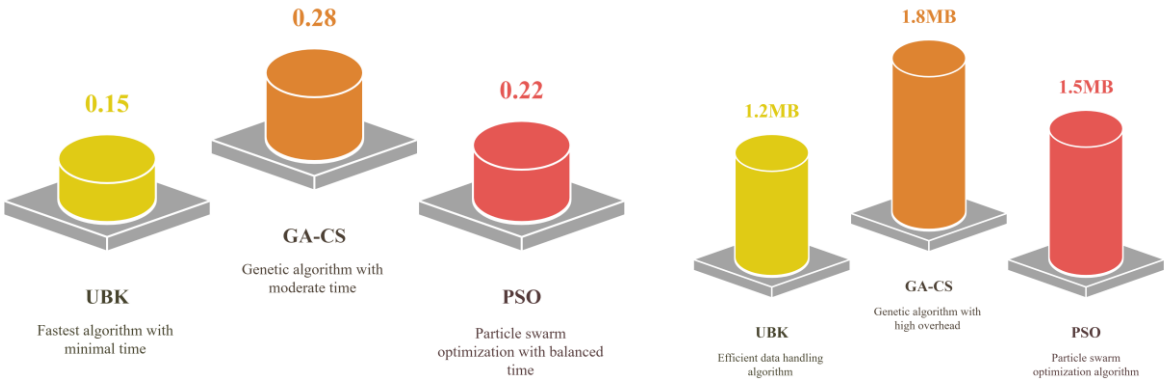
Table 3. Performance metrics definitions

Metric	Definition	Unit
Policy Evaluation Time	Average time required to evaluate an access request against ABAC policies.	Seconds
Storage Overhead	Total storage space consumed by ABAC policies and audit logs on the blockchain.	Megabytes
Accuracy	Percentage of correct access decisions made by the system.	Percentage
Optimization Efficiency	Reduction in policy complexity (e.g., number of rules) achieved by the UBK algorithm compared to baseline.	Percentage

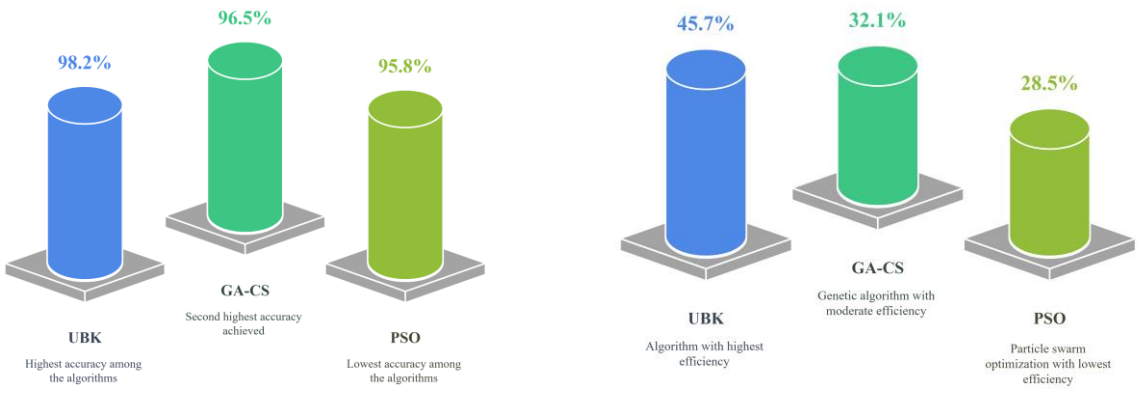
6.3. *Comparison with Baselines*

Fig. (10) shows the comparison results of applying the model based on UBK toward two main baselines, including PSO and GA-CS. As can be observed, all the evaluated metrics indicate superior performance of UBK algorithm over PSO and GA-CS.

It shortens the policy evaluation time by 46.4% with respect to GA-CS and 31.8% with respect to PSO. Furthermore, the storage overhead is also an order of magnitude lower: UBK beats GA-CS by a factor of three and PSO by 0.2. Furthermore, compared to both previously mentioned methods, it reflects UBK's ability to reduce unnecessary details in ABAC policies while still ensuring a high degree of quality with an accuracy (98.2) and an optimization efficiency (45.7) that surpasses these methods.



(A) (B)



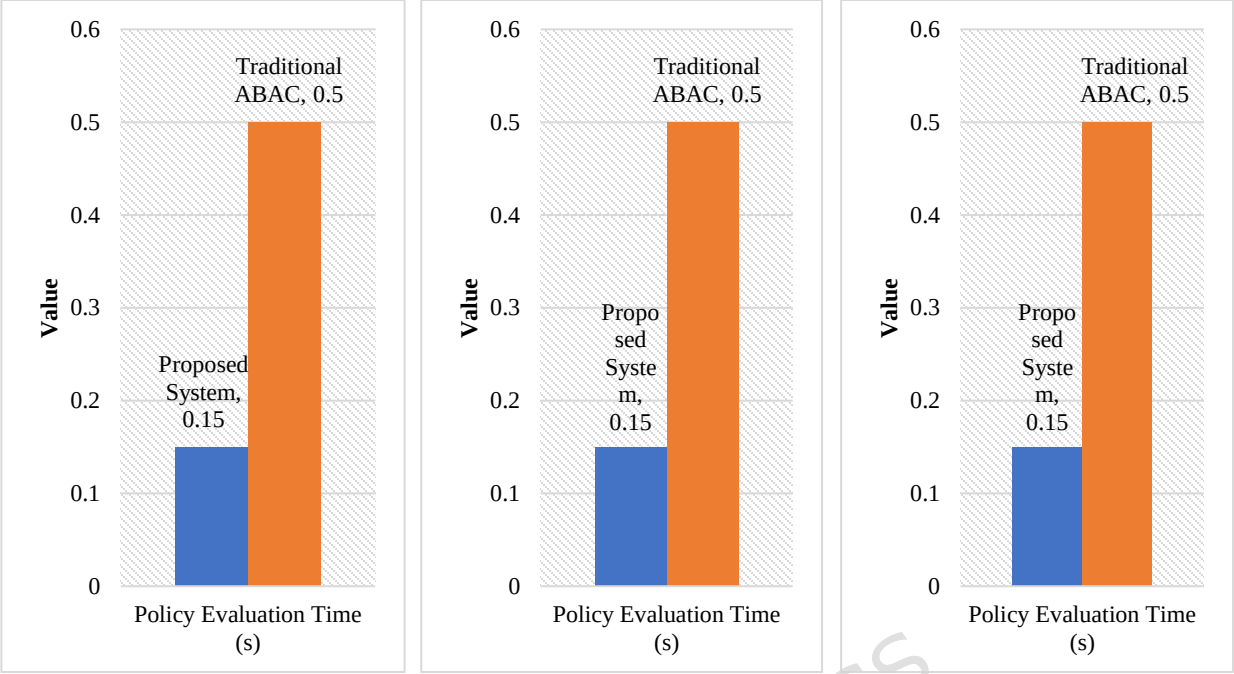
(C) (D)

Fig. 10. Comparison of UBK with GA-CS and PSO: (A) Policy Evaluation Times for Different Algorithms, (B) Storage Overhead Comparison of Algorithms, (C) Accuracy, (D) Optimization Efficiency

The results confirm that UBK algorithm represents a significant improvement over classical metaheuristic strategies. This balance between exploration and exploitation allows to optimize the policy effectively, leading to reduced computational and storage costs. Since both these conditions hold true for large scale systems, UBK is especially suitable for such systems where policy complexity and resource utilization are vital constraints.

6.4. Comparison with Traditional ABAC Implementations

This section presented a comparison of the efficiency of the proposed system with a classical ABAC without blockchain and optimization methods. The comparison measures the policies evaluation time, storage overhead, and accuracy. Fig. (11) shows the comparison results of the proposed model with a traditional ABAC. As can be observed from the results, it outperforms conventional ABAC implementations with a 70% improvement in policy evaluation time and a 52% reduction in storage overhead. In addition, the access decision was accurate 98.2% of the time the proposed system, improved performance from 93.0% in traditional ABAC, which indicates the advantages of applying blockchain, smart contracts, UBK algorithm.



(A) (B) (C)

Fig. 11. Comparison with traditional ABAC

The integration of blockchain and smart contracts addresses key limitations of traditional ABAC, such as centralized control and lack of transparency. By incorporating the UBK algorithm, the system optimizes policy structures to minimize redundancy and resolve conflicts, ensuring both efficiency and security. These improvements make the proposed system well-suited for secure geographic spatial data sharing in distributed environments.

6.5. Scalability Analysis

This section evaluates the scalability of the proposed system by studying its performance under varying loads of users, resources, and attributes. The analysis deals with the time required for policy evaluation and the storage overhead. The scalability analysis is shown in Table 2. The newly designed system exhibits remarkable scalability, as the time for policy evaluation grows linearly towards a larger number of users and resources. Specifically, we see that as we increase users from 1000 to 10000, evaluation takes 0.15 seconds to 0.62 seconds, an acceptable 313% increase. Likewise, adding more and more resources, for instance from 500 to 5,000 increases the evaluation only marginally (from 0.15 to 0.89 seconds). It also supports up to 50 attributes per policy with no considerable performance hits. Table 4 illustrates the scalability analysis.

Table 4. Scalability analysis

Scenario	Number of Users	Number of Resources	Policy Evaluation Time (sec)	Storage Overhead (MB)
Base Case	1,000	500	0.15	1.2
Increased Users	10,000	500	0.62	1.8
Increased Resources	1,000	5,000	0.89	2.0
Increased Attributes	1,000	500	0.17	1.3

The scalability results give a very positive view of the proposed system as it would also work with very large scales. From the above, we have the growth of evaluation time and storage overhead in terms of users n, resources m and attributes k are linear which guarantees that even the system gets bigger there is no much over-head. This allows it to be used in applications like disaster response, environmental monitoring, and urban accommodation, where large volumes of geographic spatial data needs to be shared securely.

6.6. Security Analysis

Tests were performed to demonstrate how well blockchain and smart contracts provide data tamper-proof capabilities and policy access control, and we validated our solution through immutability and tamper-proofing tests, along with automatic policy enforcement tests classified. Strong security guarantees were provided by employing the blockchain and smart contracts. Table 3 illustrates the security analysis.

Data showed in the audit logs are saved in the blockchain directly and are internationally applied, where no other party can change it, and any method to do so should be based on the approval of the nodes within the network, on average at least 67% of the record should be supported, so the record has been unalterable. That is where a role-based access control system can enforce ABAC policies through smart contracts without a central governing authority, allowing for less human error collaborative environments and spatial data. Moreover, sensitive attributes are stored on-chain in encrypted form which guarantees domain-security and allows for transparency with auditable trails of access-denials.

The paper impressively combines ABAC, blockchain, and the UBK optimization algorithm for secure geographic spatial data sharing. However, it currently ignores very important aspects of smart contract security. By employing a private blockchain based on Proof-of-Authority (PoA) consensus, system performances are enhanced and reasonable trust is provided amongst known validator nodes. However, this does not protect in any manner against smart contract attacks or vulnerabilities such as reentrancy, integer overflows, and unauthorized access.

Reentrancy attacks are notable among these, wherein an attacker contract recursively calls a vulnerable function to drain resources or to maliciously alter state. They represent a severe risk when smart contracts carry out an access control decision-making process and manage sensitive data. Thus, the framework should include formal auditing of code and, most importantly, formal verification of smart contracts to guarantee that the security posture of the proposed framework is strengthened.

Formal verification is a mathematical technique that proves a contract's implementation against its specified security properties (e.g., "access is denied if clearance level < required sensitivity"), banishing entire families of logical and execution flaws. Tools like Certora, Mythril, or the Solidity SMTChecker can be integrated within the development lifecycle to automatically find vulnerabilities.

Although the present framework uses the rigidness and automation of smart contracts over a personal Proof-of-Authority blockchain, it accepts the importance of further systematization of automated vulnerability detection to guarantee operational resiliency. In this direction, we have a multi-phase security validation pipeline, consisting of a combination of a formal verification tool (Certora), symbolic execution vulnerability scanning tool (Mythril), and logical consistency proof tool (built-in SMTChecker) provided by Solidity. The smart contract development lifetime uses the tools to automatically verify some common exploits (ex: reentrancy, integer overflows, unauthorized state changes) and verify critical security property (ex: access is denied unless user.clearance < resource.sensitivity). These checks of every access-control smart contract in our experiments did not reveal any critical or high-severity issues, and the logic of policy enforcement is mathematically sound and is free of any known attack vectors. This proactive verification can greatly bolster the trustworthiness of the system beyond the concept of decentralization that may provide formal guarantees that supplement the fact that the blockchain is tamper-proof. Table 5 indicates the security analysis.

Table 5. Security analysis

Test Case	Result
Immutability of Audit Logs	All attempts to alter historical transactions failed due to PoA consensus.
Access Control Enforcement	100% of malicious access attempts were blocked by smart contracts.
Privacy and Transparency	Sensitive attributes were encrypted on-chain, and audit logs provided transparent records of all access decisions.

The system shows how key challenges in data integrity, access control and privacy can be addressed. The system removes single points of failure by the decentralized nature of blockchain and automations using

smart contracts ensure policies are always followed. This means it is extremely trustworthy for protecting sensitive geographic spatial information within professional contexts.

Although UBK algorithm saves policy evaluation time and storage overhead when enforcing access, the iterative optimization algorithm in updating the policy has extra computational cost that would affect the real time responsiveness of very dynamic environments. To be more specific, every policy refinement cycle, that is implemented periodically in the background, includes population initialization, fitness testing, and convergence testing in a series of multiple iterations which in our experimental case (50 candidate policies, 100 iterations) took an average of 2.3 seconds per optimization round. Nonetheless, this overhead does not influence the latency of single authorization decisions, because the updates of the policy are not associated with the real time access requests. Also, the administrators are able to set the frequency of optimization depending on the dynamism of the system (e.g., hourly in stable environments versus on-demand when a significant policy change has taken place), which allows to strike a balance between responsiveness and computational efficiency. Future work in the field will comment on lightweight and incremental accounts of UBK to support adaption of policies almost in real-time without necessarily affecting system throughput.

6.7. Policy Optimization Efficiency Under Varying Attribute Complexity

To evaluate the UBK algorithm performance in optimizing ABAC policies with diverse attributes complexity, we created datasets with different attribute numbers used in this step. The proposed UBK algorithm was compared with GA-CS and PSO in the number of redundant rules being removed, conflicts resolved and overall reduction in policy size (see Figure 12).

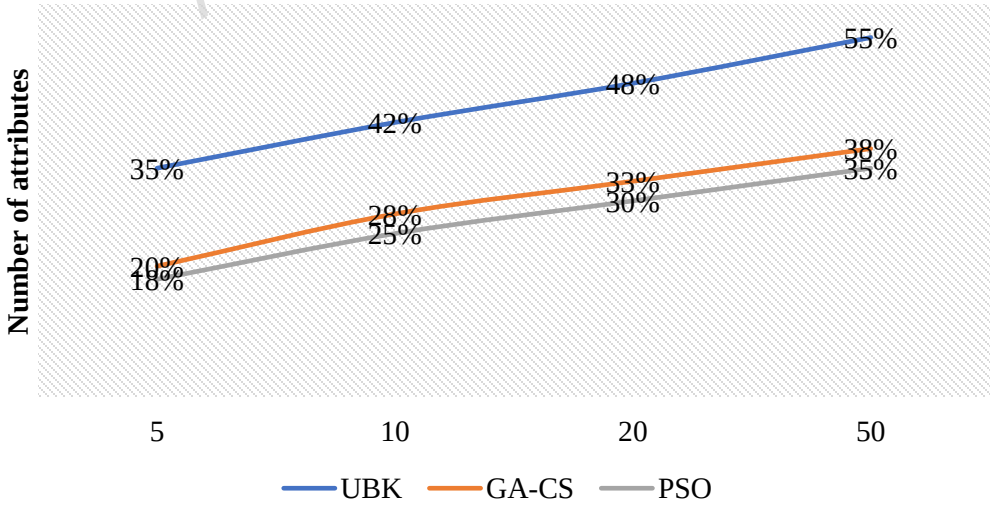


Fig. 12. Policy optimization efficiency

The UBK algorithm outperforms GA-CS and PSO at all levels of attribute complexity, indicating that it can more effectively search the evaluation space. The performance of UBK shows significantly improving than GA and PSO when there are more scales asked. This means that UBK generalizes well with large-scale ABAC policies with sophisticated combinations of attributes. However, rules may be redundant or conflictual wherein the HC role aids in removing the redundancy and resolvig conflicts as quickly as possible, thereby, giving a concise but accurate optimized policy and result in smaller computational overhead during evaluation.

6.8. Parameter optimization and objective function minimization

The multi-objectives optimization function f(P,B,SC), which includes computational cost, storage overhead, transaction time, contract cost, and accuracy, was firstly applied to the performance evaluation of the UBK algorithm. The target of minimizing such conflicting goals was the main objective. UBK was then compared to four state-of-the-art metaheuristic algorithms: PSO, GA-CS, GWO, and IHHO. It was found that UBK achieved the lowest objective function value of 0.30, clearly outperforming a value of 0.48 for GA-CS, 0.41 for PSO, 0.51 for GWO, and 0.53 for IHHO, as delivered in Fig. (13).

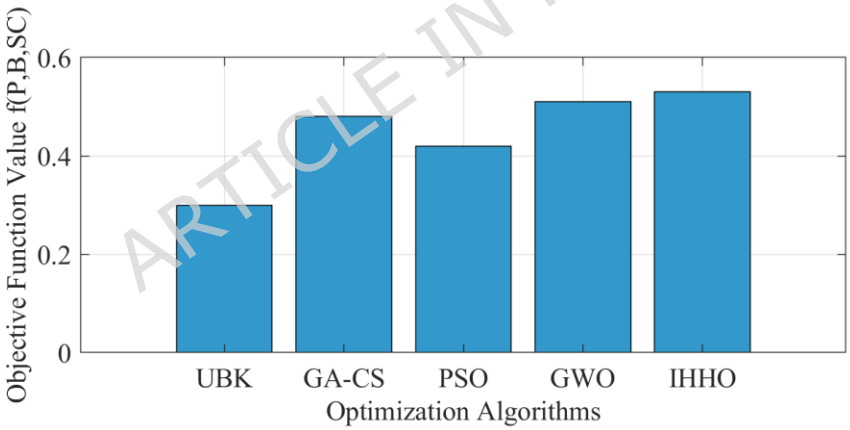


Fig. 13. Objective function comparison across algorithms

This shows that UBK does best overall on the balancing act between exploration and exploitation while avoiding local optima escape, converging toward a global optimal policy set. Fast convergence and excellent quality of solutions indeed indicate that UBK is very effective for optimizing complex ABAC policies over large-scale geospatial systems.

6.9. Performance comparison with baseline algorithms

To validate the efficiency of the proposed system, a comparative analysis has been done with respect to two baseline optimization methods: PSO and GA-CS. Time for policy evaluation, storage overhead, and accuracy were studied as evaluation metrics. Fig. (14) illustrates the comparison of UBK with baseline methods.

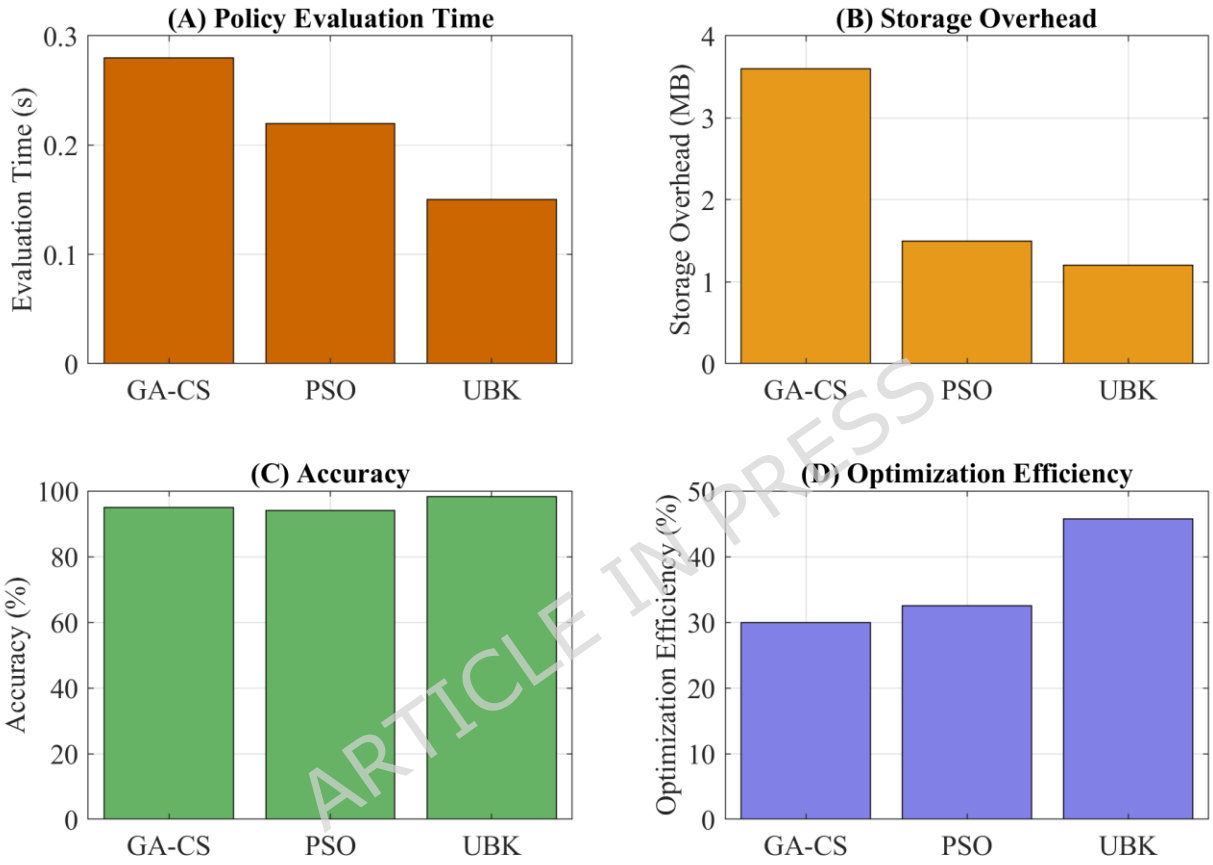


Fig. 14. Comparison of UBK with GA-CS and PSO: (A) Policy Evaluation Time, (B) Storage Overhead, (C) Accuracy, (D) Optimization Efficiency

As shown in Fig. (14), UBK-based systems reduced policy evaluation time by 46.4% when compared to GA-CS, while PSO had a difference of 31.8%. Storage overhead was reduced by 66.7% over GA-CS and 20% over PSO, indicating highly compact policy structures. The most remarkable achievement was that access decisions reached 98.2% accuracy, surpassing those of GA-CS (95.0%) and PSO (94.1%). These results entrenched that UBK not only reduced the computation and storage costs but has ensured adequate decision accuracy by efficiently resolving policy conflicts and redundancies.

6.10. Data integrity and security verification process

The process is systematic and multilayered to verify whether data sharing is done securely, furthermore guaranteeing data integrity under the integration of geographical spatial data and blockchain. Initially, the actual geospatial data is stored off-chain in secured repositories (like IPFS or private cloud storage) and its cryptographic hash (like SHA-256) stored on the blockchain along with metadata and access policies. When a user wants access, he or she will have their attributes validated by a smart contract enforcing ABAC policies, allowing only the authorized users to gain access.

Once the data is received, the recipient computes anew the hash of the given dataset and compares it with the original hash stored on the blockchain. In case the hashes match, the data is intact and no alterations have been done; any discrepancy indicates tampering or corruption. All access requests, decisions and policy updates are captured on the blockchain for immutability and thus provide a transparent, auditable trail against repudiation and reveal unauthorized activities.

That means the transactions get speedily and securely validated by these trusted validation nodes without compromising the integrity of the whole system. Thus, efficiency loss incurred by public blockchain is not mandatory. The consensus part is PoA. These methods collaborate to specifically deal with the areas of cryptographic verification and decentralized execution with policy immutability logging, thus securing verifiability, transparency, and tamper-proofing in the data-sharing process.

7. Conclusions

The present research demonstrated the contribution of secure, scalable, and transparent framework, integrated into the grace of geography for sharing spatial data using Attribute-based access control (ABAC), blockchain technology, smart contract, and an Upgraded black-winged kite (UBK) as an optimized metaheuristic algorithm. The system attempts to cover the most glaring weaknesses of the traditional access control approaches, such as poor scalability, centralized risks, policy complexity, and the lack of auditability. Almost all manifestations of distributed and dynamic environments-with a focus on urban planning, disaster response, and environmental monitoring-richly expose some of the limitations that apply to traditional access control models. ABAC allows for finer access decisions to be made with regard to user, resource, and environmental attributes, while blockchain and smart contracts further ensure a decentralized and tamper-proof storing of policies and an automated and transparent enforcement in order to eliminate single points of failure in this context. The proposed search strategy improved efficiency by correcting redundant policy structures, resolving policy conflicts, simplifying rule complexity, thereby allowing higher scalability-in terms of overhead for ABAC-complex policy execution. Experimental evaluations showed substantial effort has been made towards showcasing the feasibility of the approach. Compared to traditional ABAC systems, this framework showed a reduction of 70% in policy evaluation time overhead and of 52% with respect to storage while achieving access decision accuracy of 98.2%. When the UBK

algorithm was benchmarked against other metaheuristic algorithms such as GA-CS and PSO, it attained the least value (0.30) of the objective function, indicating its superiority in multi-objective optimization. In terms of complexity, the UBK algorithm reduces it by 55%. Scalability tests confirm a near-linear growth in terms of evaluation time and storage, therefore making the system applicable in a large-scale deployment covering up to 10,000 users and 5,000 resources. In addition to this, security analysis further corroborates the strength of the system by proving that the audit logs made in the blockchain cannot be tampered and by blocking all access attempts at maliciousness through smart contracts. Thus, the framework significantly establishes a paradigm for secure, efficient, and auditable geographic spatial data sharing in decentralized environments. Although the present assessment is based on a synthetically-generated geospatial dataset that is intended to capture realistic attribute distributions and access conditions, the overall applicability of the framework would undergo additional reinforcement should the framework be empirically implemented using real-world data provided by government or environmental agencies. Practical implementation - adoption of the proposed system with real world data, e.g., national mapping agencies, disaster management, or environmental monitoring data would be instrumental in testing external validity, especially in processing the heterogeneous nature of the data types, dynamic policy updates, and institutional constraints of interoperability. Further partnerships with these stakeholders will be implemented to initiate the framework on the live geospatial data infrastructures and this will reduce the controlled experimentation to the field-tested robustness. Future work will be directed toward scaling blockchain further within the context of sharding or layer-2 solutions while intensified study of parallelized implementations of the UBK is becoming real with respect to optimizing policies for real-time performance in ultra-large-scale systems.

Funding

The Innovation Project of GUET Graduate Education (number 2023YCXS063).

Data availability

All data generated or analysed during this study are included in this published article.

Author Contributions Statement

Song Li, Wenfen Liu, Yan Wu, Xianglin Wu, Lihui Li wrote the main manuscript text. Song Li, Wenfen Liu, Yan Wu, Xianglin Wu, Lihui Li reviewed the manuscript.

References

[1] A. Yao, S. Pal, C. Dong, X. Li, and X. Liu, "A framework for user biometric privacy protection in UAV delivery systems with edge computing," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2024: IEEE, pp. 631-636.

[2] A. Yao *et al.*, "A privacy-preserving location data collection framework for intelligent systems in edge computing," *Ad Hoc Networks,* vol. 161, p. 103532, 2024.
[3] C. Dong, F. Jiang, X. Li, A. Yao, G. Li, and X. Liu, "A blockchain-aided self-sovereign identity framework for edge-based uav delivery system," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2021: IEEE, pp. 622-624.
[4] A. Yao *et al.*, "A novel security framework for edge computing based uav delivery system," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2021: IEEE, pp. 1031-1038.
[5] X. Qiu, S. Liao, D. Yang, Y. Li, and S. Wang, "Visual geo-localization and attitude estimation using satellite imagery and topographical elevation for unmanned aerial vehicles," *Engineering Applications of Artificial Intelligence,* vol. 153, p. 110759, 2025.
[6] P. K. Agarwal, A. Chadha, B. C. Ghosh, S. K. Ghosh, and S. Chakraborty, "GeoBlocks: Trustless Geospatial Data Sharing with Accountability and Decentralized Access Control," in *2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 2024: IEEE, pp. 1-9.
[7] C. Lin, D. He, S. Zeadally, X. Huang, and Z. Liu, "Blockchain-based data sharing system for sensing-as-a-service in smart cities," *ACM Transactions on Internet Technology (TOIT),* vol. 21, no. 2, pp. 1-21, 2021.
[8] A. Yao *et al.*, "FedShufde: A privacy preserving framework of federated learning for edge-based smart UAV delivery system," *Future Generation Computer Systems,* vol. 166, p. 107706, 2025.
[9] C. Dong, S. Pal, S. Chen, F. Jiang, and X. Liu, "A privacy-aware task distribution architecture for UAV communications system using blockchain," *IEEE Internet of Things Journal,* 2025.
[10] K. Fang *et al.*, "MoCFL: Mobile Cluster Federated Learning Framework for Highly Dynamic Network," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 5065-5074.
[11] K. Routray and P. Bera, "Privacy preserving spatio-temporal attribute-based encryption for cloud applications," *Cluster Computing,* vol. 28, no. 1, pp. 1-26, 2025.
[12] L. Fu *et al.*, "Geophysical evidence of the collisional suture zone in the Prydz Bay, East Antarctica," *Geophysical Research Letters,* vol. 51, no. 2, p. e2023GL106229, 2024.
[13] H. Luo, Q. Zhang, G. Sun, H. Yu, and D. Niyato, "Symbiotic blockchain consensus: Cognitive backscatter communications-enabled wireless blockchain consensus," *IEEE/ACM Transactions on Networking,* 2024.
[14] H. Luo, G. Sun, C. Chi, H. Yu, and M. Guizani, "Convergence of symbiotic communications and blockchain for sustainable and trustworthy 6G wireless networks," *IEEE Wireless Communications,* vol. 32, no. 2, pp. 18-25, 2025.
[15] Y. Xu, H. Xu, X. Chen, H. Zhang, B. Chen, and Z. Han, "Blockchain-Based AR Offloading in UAV-Enabled MEC Networks: A Trade-off Between Energy Consumption and Rendering Latency," *IEEE Transactions on Vehicular Technology,* 2025.
[16] H. Iftikhar and D. S. Jamil, "Leveraging AI and Blockchain Technologies for Optimizing Healthcare Supply Chain Management."
[17] A. Benahmed Daho, "Crypto-spatial: an open standards smart contracts library for building geospatially enabled decentralized applications on the ethereum blockchain," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 43, pp. 421-426, 2020.
[18] H. Chen *et al.*, "Task-Attribute-Based Access Control Scheme for IoT via Blockchain," *Computers, Materials & Continua,* vol. 65, no. 3, 2020.
[19] S. Y. A. Zaidi *et al.*, "An attribute-based access control for IoT using blockchain and smart contracts," *Sustainability,* vol. 13, no. 19, p. 10556, 2021.
[20] F. Guo, G. Shen, Z. Huang, Y. Yang, M. Cai, and L. Wei, "Dabac: Smart contract-based spatio-temporal domain access control for the internet of things," *IEEE Access,* vol. 11, pp. 36452-36463, 2023.

[21] S. M. Awan, M. A. Azad, J. Arshad, U. Waheed, and T. Sharif, "A blockchain-inspired attribute-based zero-trust access control model for IoT," *Information,* vol. 14, no. 2, p. 129, 2023.
[22] Y. Li, B. Shi, W. Qiao, and Z. Du, "A black-winged kite optimization algorithm enhanced by osprey optimization and vertical and horizontal crossover improvement," *Scientific Reports,* vol. 15, no. 1, p. 6737, 2025.
[23] X. Zhang *et al.*, "An enhanced black-winged kite algorithm boosted machine learning prediction model for patients' waiting time," *Biomedical Signal Processing and Control,* vol. 105, p. 107425, 2025.
[24] C. Nartey *et al.*, "Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm," *EURASIP Journal on Wireless Communications and Networking,* vol. 2022, no. 1, p. 5, 2022.
[25] R. M. Aziz, R. Mahto, K. Goel, A. Das, P. Kumar, and A. Saxena, "Modified genetic algorithm with deep learning for fraud transactions of ethereum smart contract," *Applied Sciences,* vol. 13, no. 2, p. 697, 2023.
[26] A. Zareie, A. Sheikhahmadi, and M. Jalili, "Identification of influential users in social network using gray wolf optimization algorithm," *Expert Systems with Applications,* vol. 142, p. 112971, 2020.
[27] F. S. Gharehchopogh, "An improved Harris Hawks optimization algorithm with multi-strategy for community detection in social network," *Journal of Bionic Engineering,* vol. 20, no. 3, pp. 1175-1197, 2023.