International Journal of Safety and Security Engineering
Vol. 16, No. 7, July, 2026, pp. 1521-1534
Journal homepage: http://iieta.org/journals/ijsse

A Secure SeDAS-Based Access Control and Authentication Mechanism for Cloud Environment

Karthik D U[1,2*], Basavesha D[1]

[1] Department of Computer Science and Engineering, Shridevi Institute of Engineering and Technology (Affiliated to Visvesvaraya Technological University (VTU), Belagavi 590018), Tumakuru 572106, India
[2] Department of Computer Science and Engineering, M. S. Ramaiah University of Applied Science (RUAS), Bengaluru 560058, India

Corresponding Author Email: karthiikdu@gmail.com

https://doi.org/10.18280/ijsse.160710

Received: 1 May 2026
Revised: 3 July 2026
Accepted: 13 July 2026
Available online: 31 July 2026

Keywords:
SSACAM, cloud security, data confidentiality, attack detection

ABSTRACT

Cloud computing environments are widely used in distributed data storage, remote service provisioning and multi-user resource sharing, but still vulnerable to identity spoofing, poor session control, unauthorised access and inadequate verification. Conventional security mechanisms like Multi-Factor Authentication (MFA), Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC) are used for cloud security but are often limited by inflexible policy structures, difficult rule management, verification inefficiencies, and reduced flexibility in fast-changing cloud environments. To overcome these issues, this study proposes a Secure SeDAS (Self-Destructing Data System)-Based Access Control and Authentication Mechanism (SSACAM) for cloud environments. The proposed framework integrates SeDAS-based data protection, adaptive authentication, context-aware authorisation, Distributed Hash Table (DHT) based key sharing, secure session key reconstruction, and self-destruction-controlled data recovery. In SSACAM, encrypted data is stored in the cloud storage and key shares are distributed separately through DHT nodes to prevent unauthorised recovery. Compared with RBAC, ABAC and MFA separately, SSACAM improved the authentication accuracy by 21.8%, 15.2%, and 10.9%, the access-control efficiency by 22.7%, 13.6%, and 5.1%, the data confidentiality by 23.6%, 13.9%, and 7.4%, and the attack-detection rate by 24.7%, 19.7%, and 5.9%, respectively. It also reduced the response time by 2.8%, 14.7%, and 17.5% compared to RBAC, ABAC, and MFA, respectively, in comparison with conventional methods. The performance of SSACAM demonstrates improved security and efficiency under the evaluated conditions. The results indicate that the method is suitable for reliable and scalable cloud security applications that are also privacy-preserving.

1. INTRODUCTION

Using shared network resources, cloud computing has become a key way to store, manage, and access data and services. Because it is flexible, scalable, cheap, and easy to access from anywhere, it is very useful for modern uses in healthcare, banking, education, e-commerce, smart industry, and government services. But because so many people are using the cloud so quickly, security has become harder, especially when it comes to user authentication and access control [1, 2].

Because cloud platforms are open, spread out, and used by many people at once, it's easy for people to get into them without permission, steal identities, leak data, and abuse accounts. There must be a safe and reliable way to check identity and control access to protect sensitive cloud resources [3, 4].

Some of the most important trends in cloud security right now are smart identity verification, context-aware access management, adaptive authentication, zero-trust security, and lightweight cryptographic protection. Cloud services that connect to the Internet of Things (IoT), cloud storage for businesses, online payment systems, electronic health record systems, and virtual learning environments all use these trends a lot. There are still a lot of problems with traditional methods, even though cloud technology has a lot of benefits. Role-Based Access Control (RBAC) is simple to use and understand, but it does not work well in cloud environments that change all the time [5, 6]. Attribute-Based Access Control (ABAC) lets you set very specific permissions, but it's hard to make rules for it and it slows down computers. Multi-Factor Authentication (MFA) makes logging in safer, but it might take longer to prove who you are and make things less convenient for users. We need a better secure framework because of these problems. Therefore, the Secure SeDAS (Self-Destructing Data System)-Based Access Control and Authentication Mechanism (SSACAM) is being suggested. It will make cloud access safer, faster to check, and easier to

manage [7, 8].

1.1 Research gaps

A lot of work has been done on the security of cloud computing, but the access control and authentication systems still have some significant limitations. A lot of the models out there only make sure that users are who they say they are. They don't do a good job of coordinating authentication and dynamic access control in a shared cloud environment. Cloud platforms in the real world are always changing, with new users, devices, and apps coming and going. Because of this, static security rules are often not enough to make access decisions in real time. There is a difference between how security should work and how it really works in the cloud [9, 10]. Another big problem is that old ways don't work well in all situations. RBAC is easy to use, but it's not good for changing user permissions or permissions that depend on the situation. ABAC gives you a lot of control, but it can be hard to keep track of when there are a lot of users, policies, and resources. MFA makes it harder to verify users, but it can also make things slower, require more work from users, and make big cloud apps more challenging to deploy. These methods do make things safer, but they don't always strike a good balance between safety, flexibility, and ongoing protection [11].

Another issue is that there are no integrated frameworks that can make authentication stronger, access faster, data more private, and more challenging to deploy without permission all at the same time [12]. A lot of the methods that have been written about only look at one or two security factors and don't think about how well the cloud environment works as a whole. There also aren't many lightweight and scalable solutions that work with multi-tenant cloud platforms yet. To fix these problems and make cloud security more reliable, we require a safe and flexible system like the proposed SSACAM [13, 14].

1.2 Related work

SeDAS enables self-destructing cloud data through active storage, making both data and keys unreadable after a predefined expiration period; however, it lacks adaptive authentication and dynamic access control [15]. Vanish supports self-destructing data by distributing keys through DHT, allowing the keys to disappear automatically after a specified period, although its reliability may be affected by DHT node failures and network churn [16]. Other self-destructing schemes provide automatic deletion and privacy protection for sensitive electronic data but do not address cloud-based multi-user authentication [17]. Decentralised access control with anonymous authentication can protect user privacy in cloud storage, yet it does not support time-bound recovery based on SeDAS [18]. Existing studies have also compared RBAC, ABAC, DAC, MAC, and fine-grained access control mechanisms, but they are survey-based and do not propose a new secure cloud model [19].

ABAC provides fine-grained access permissions based on attributes, but it increases computational cost and key-management complexity [20]. Lightweight cloud data access control based on a hypergraph structure reduces access-control complexity; however, it does not support DHT-based key sharing or self-destruction control [21]. Existing cloud data security studies identify major security problems and corresponding protection methods, but they do not provide an integrated SeDAS-based authentication framework [22].

2. SEDAS-BASED CLOUD DATA PROTECTION ARCHITECTURE

Figure 1 shows how the SeDAS works in the cloud. In this configuration, the user utilizes a local system for data transmission and reception. The system encrypts data before sending it to cloud storage to keep the original content from being seen and to protect privacy.

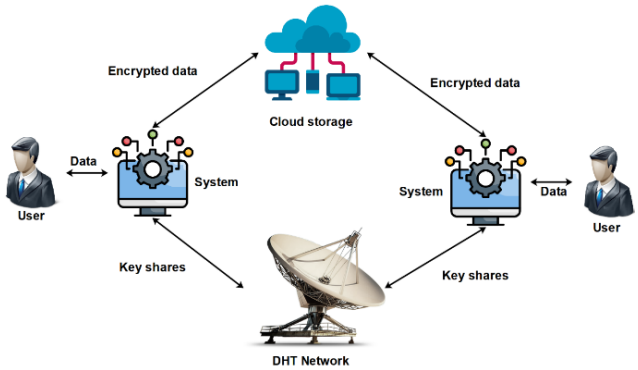

Figure 1. Architecture of Self-Destructing Data System (SeDAS) for secure cloud storage

The key shares are not saved in the cloud; they are spread out across the Distributed Hash Table (DHT) network [23]. This separation of encrypted data and key sharing makes things safer because an attacker can not get to the original data only by getting to the cloud storage. The system acquires the key shares it needs from the DHT network and puts them back together so that the data may be read. The SeDAS model protects your privacy better [24, 25].

(1) Encrypted cloud storage formation

The original user data is encrypted before it is uploaded to the cloud in the current SeDAS architecture. This keeps the stored information safe from direct unauthorised access, as expressed by Eq. (1).

$$C_{cloud} = Enc(D, K_m) \tag{1}$$

where, C_{cloud} denotes the ciphertext stored in the cloud, $Enc()$ represents the encryption function, D indicates the original user data, and K_m denotes the master encryption key used to protect the data [26, 27].

(2) Distributed key share construction in Distributed Hash Table

The current SeDAS method splits the master key into several parts and sends them through the DHT network instead of keeping the whole encryption key in one place. This is shown by Eq. (2).

$$K_m = \oplus_{i=1}^{n} KS_i \tag{2}$$

where, K_m represents the reconstructed master key, KS_i denotes the i^{th} key share stored in the DHT network, n indicates the total number of distributed key shares, and $\oplus$ represents the exclusive-OR operation used for key reconstruction [28, 29].

(3) Time-bound data recovery condition

The current SeDAS model allows for data recovery only when the necessary key shares are accessible within the designated validity period, as indicated by Eq. (3).

$$D^* = \begin{cases} Dec(C_{cloud}, K_m), & \text{if } t \le T_{exp} \\ \sum_{i=1}^{n} \delta(KS_i) \ge k[6pt]\emptyset, & \text{if } t > T_{exp} \end{cases} \quad (3)$$

where, D^* denotes the recovered data, $Dec(\cdot)$ represents the decryption function, C_{cloud} indicates the encrypted cloud data, K_m denotes the reconstructed master key, t is the current access time, T_{exp} represents the expiration time of the protected key shares, $\delta(KS_i)$ denotes the availability status of the i^{th} key share, k indicates the minimum number of valid key shares required for successful recovery, and $\emptyset$ means that the data cannot be recovered [30, 31].

3. PROPOSED SEDAS-BASED ACCESS CONTROL AND AUTHENTICATION FRAMEWORK FOR CLOUD ENVIRONMENT

Figure 2 shows the proposed SSACAM for a cloud environment. The first step in this framework is user registration, where the user gives the authentication module their credentials and other information [32, 33].

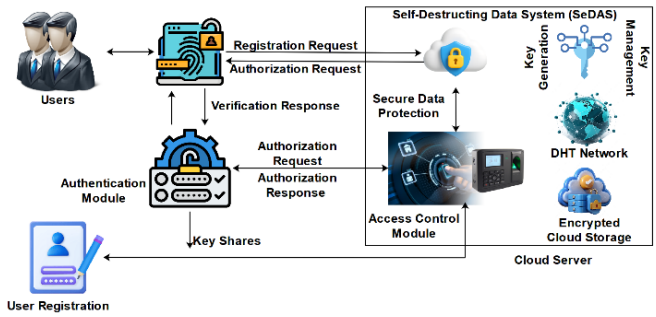

Figure 2. Block diagram of the proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

The user sends an access request through the authentication stage after registering. The user's identity is verified by the authentication module, which then responds to confirm it. An authorization request is submitted to the access control module following user verification. The availability of the requested cloud resource is then determined by this module.

The SeDAS combines key generation, key management, and a DHT network inside the cloud server to provide secure data protection. The data are stored in encrypted cloud storage, while the corresponding key shares are managed separately. This separation improves confidentiality and reduces the chance of unauthorized recovery of sensitive information. The access control module communicates with both the SeDAS unit and the encrypted cloud storage to validate permissions and retrieve protected resources securely. Thus, the proposed framework strengthens authentication accuracy, access control security, data confidentiality, and resistance to unauthorized access in cloud environments [34, 35].

3.1 Mathematical assumptions, parameter ranges, and decision logic

The SSACAM mathematical model proposed in this paper is developed assuming that each cloud access request is evaluated using three security layers: user authentication, context-aware authorisation and SeDAS-based secure data recovery. All trust parameters are normalised between 0 and 1, where 0 represents completely untrusted/invalid behaviour and 1 indicates fully trusted/valid behaviour. Thus, the authentication confidence, contextual trust, role privilege, permission compatibility, device fingerprint consistency, behavioural profile score and session freshness are considered as bounded security scores.

The weighting coefficients used in the equations indicate the relative importance of each security factor. The weights are chosen so that the total contribution is normalised, to avoid biasing any single parameter. Parameters that have a greater influence on access security (such as credential matching, contextual trust and resource sensitivity) are assigned higher weights. For practical purposes, these weights are used to balance user identity, device reliability, behavioural consistency and cloud resource risk in the decision-making process for access.

The threshold values determine the minimum acceptable level of security for granting the access. The authentication threshold is a measure of the reliability of the user identity. The authorisation threshold determines if the requested user-resource interaction is allowed. The contextual trust threshold validates the access request, given the current environment. The key reconstruction threshold is the number of valid DHT key shares required to recover encrypted data. If any of the threshold conditions is not met, then the request is rejected and secure data recovery is not permitted.

Table 1. Parameter ranges used in the proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM) mathematical formulation

Sl. No.	Parameter	Practical Meaning	Range
1	Identity score	Validity of registered user identity	0 to 1
2	Device fingerprint score	Trust level of user device	0 to 1
3	Behavioural profile score	Consistency of user behaviour	0 to 1
4	Session freshness	Validity of current login session	0 to 1
5	Contextual trust	Trust level based on access context	0 to 1
6	Resource sensitivity	Risk level of requested cloud resource	0 to 1
7	Weight coefficients	Relative importance of security factors	0 to 1 and normalized
8	Authentication threshold	Minimum confidence required for login approval	0.5 to 1
9	Authorization threshold	Minimum score required for access approval	0.5 to 1
10	Contextual trust threshold	Minimum trusted access condition	0.5 to 1
11	Key reconstruction threshold	Minimum valid key shares required	$k \le n$
12	Time validity	Data recovery before self-destruction time	$t < Tsd$

The major parameter ranges used in the proposed

formulation are summarized in Table 1. These ranges define the normalized values, decision limits, and practical constraints used for authentication, authorization, key reconstruction, and secure data recovery in the proposed SSACAM framework.

Eq. (4) computes the registration trust of a user by combining identity verification, device fingerprint information, behavioral profile, and time validity [36, 37].

$$RTV_u = \alpha_1 I_u + \alpha_2 D_f + \alpha_3 B_p + \alpha_4 T_v \tag{4}$$

where, RTV_u denotes the registration trust vector of the user, I_u represents the verified identity score, D_f indicates the device fingerprint consistency score, B_p denotes the behavioral profile score, T_v represents the temporal validity factor, and $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are weighting coefficients [38].

Eq. (5) determines the authentication confidence by using credential strength, session freshness, trusted device status, and anomaly resistance [38, 39].

$$AC_u(t) = \frac{\beta_1 C_r + \beta_2 S_f + \beta_3 D_t + \beta_4 (1 - A_n)}{1 + \exp(-\lambda RTV_u)} \tag{5}$$

where, $AC_u(t)$ denotes the authentication confidence of the user u at time t, C_r represents the credential matching ratio, S_f indicates the session freshness factor, D_t denotes the trusted device score, A_n represents the anomaly probability, λ denotes the authentication sensitivity parameter, and $\beta_1, \beta_2, \beta_3, \beta_4$ are adaptive weights [40, 41].

Eqs. (6) and (7) evaluate whether access should be granted by combining role privilege, permission matching, contextual trust, authentication confidence, and resource sensitivity [42, 43].

$$AZ_{u,r} = \Psi! \left[\gamma_1 R_u + \gamma_2 P_{u,r} + \gamma_3 C_x + \gamma_4 AC_u(t) - \gamma_5 S_r\right] \tag{6}$$

$$\Psi(z) = \begin{cases} 1, & z \geq \theta_a \\ 0, & z < \theta_a \end{cases} \tag{7}$$

where, $AZ_{u,r}$ denotes the authorization decision for user u on resource r, R_u represents the role privilege score, $P_{u,r}$ denotes the permission compatibility value, C_x indicates the contextual trust score, $AC_u(t)$ represents the authentication confidence, S_r denotes the resource sensitivity, $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5$ are control weights, θ_a indicates the authorization threshold, and $\Psi(\cdot)$ is the decision function [44, 45].

Eqs. (8) and (9) generate a secure session key and divides it into distributed key shares for protected storage in the DHT network [46, 47].

$$K_s = H! \left(U_{id}, |, N_s, |, AZ_{u,r}, |, S_r, |, t\right) \tag{8}$$

$$KS_i = K_s \oplus H! \left(Node_i, |, t_i, |, \mu_i\right), i = 1,2, \dots, n \tag{9}$$

where, K_s denotes the secure session key, $H(\cdot)$ represents the hash function, U_{id} indicates the user identity, N_s denotes the session nonce, $AZ_{u,r}$ represents the authorization result, S_r denotes the resource sensitivity, t indicates the current timestamp, KS_i denotes the i^{th} key share, $Node_i$ represents the DHT node identity, t_i denotes the local share time, μ_i indicates the node randomization factor, and n is the number of shares [48, 49].

Eqs. (10) and (11) reconstruct the secure session key from valid key shares and verifies its integrity before permitting data access [50, 51].

$$\widehat{K} * s = \oplus \ * i = 1^n [\delta_i KS_i] \tag{10}$$

$$\sum_{i=1}^{n} \delta_i \geq k \text{ and } H(\widehat{K} * s, |, U * id, |, N_s) = \eta \tag{11}$$

where, $\widehat{K} * s$ denotes the reconstructed secure session key, $\delta * i$ indicates the validity of the i^{th} key share, KS_i represents the i^{th} key share, k denotes the minimum required number of valid shares, $H(\cdot)$ represents the verification hash function, U_{id} denotes the user identity, N_s indicates the session nonce, and η represents the integrity verification token [52, 53].

Eqs. (12) and (13) allow data recovery only when authentication, authorization, and time validity are satisfied; otherwise, the protected data path is destroyed [54, 55].

$$\left\{D_{rec} = \begin{cases} Dec!\left(C_e, \widehat{K} * s\right), & \text{if } AC * u(t) \geq \theta_c \\ AZ_{u,r} = 1 \ \wedge \ t < T_{sd} \ \emptyset, \text{if } t \geq T_{sd} \ \vee \ AZ_{u,r} = 0 \end{cases}\right\} \tag{12}$$

$$T_{sd} = t_0 + \frac{\rho_1 TL + \rho_2 (1 - S_r) + \rho_3 C_x}{\rho_4 R_{th}} \tag{13}$$

where, D_{rec} denotes the recovered data, $Dec(\cdot)$ represents the decryption function, C_e indicates the encrypted cloud data, $\widehat{K} * s$ denotes the reconstructed session key, $AC * u(t)$ represents the authentication confidence, θ_c denotes the confidence threshold, $AZ_{u,r}$ indicates the authorization decision, t denotes the current time, T_{sd} represents the self-destruction time, t_0 denotes the initial storage time, TL indicates the time-to-live, S_r denotes the resource sensitivity, C_x represents the contextual trust level, R_{th} denotes the threat ratio, $\rho_1, \rho_2, \rho_3, \rho_4$ are destruction control coefficients, and $\emptyset$ means no data recovery [56, 57].

The unified secure data recovery condition for the proposed SSACAM framework is given by Eq. (14). It combines authentication confidence, authorisation status, contextual trust, valid key-share ratio, resource sensitivity and time-risk factor into one access decision. Data recovery is allowed only when the combined security score is above the secure access threshold, the minimum number of valid DHT key shares is available and the current access time is within the self-destruction deadline. Otherwise, the recovery process is denied [58-60].

$$\Psi_{u,r}(t) = \alpha A_u + \beta Z_{u,r} + \gamma C_u + \delta \rho_K - \eta S_r - \mu R_t$$
$$D_{rec} = \begin{cases} Dec(C_e, K_r), & \Psi_{u,r}(t) \geq \Theta_s \\ \mathrm{K_V} \geq \mathrm{k}, ; \mathrm{t} < \mathrm{T_{sd}} \ \emptyset \ , \text{otherwise} \end{cases} \tag{14}$$

where, $\Psi_{u,r}(t)$ denotes the unified security score, A_u denotes authentication confidence, $Z_{u,r}$ denotes the authorization decision, C_u denotes contextual trust, $\rho_K = \frac{K_V}{n}$ denotes the valid key-share ratio, S_r denotes resource sensitivity, R_t denotes the time-risk factor, and Θ_s denotes the secure access threshold [61].

Figure 3 presents the authentication and authorization process of the proposed SSACAM framework. It starts with user registration, where credentials, device fingerprint, and

behaviour profile are collected. Then, the system computes the Registration Trust Vector (RTV) and stores the user profile. After receiving the login request, the system performs credential verification and session check and computes the Authentication Confidence (ACu). If ACu is below the authentication threshold, access is denied. Otherwise, the process proceeds to context-aware authorization evaluation for secure cloud access [62, 63].

Figure 4 shows the secure storage and data recovery procedure in the proposed SSACAM framework. Initially, the system generates a secure session key (Ks) and divides it into multiple key shares (Ksi) [64, 65]. These shares are stored in the DHT network, while the user data are encrypted and stored in the cloud server. During data access, the system verifies whether the number of valid key shares is greater than the minimum threshold (k). If the condition is not satisfied, the recovery process is denied. When the required key shares are available, the system checks the self-destruction time condition. If the time validity is satisfied, the encrypted cloud data are decrypted and the authorized data are returned to the user. Otherwise, the recovery request is rejected [66-68].

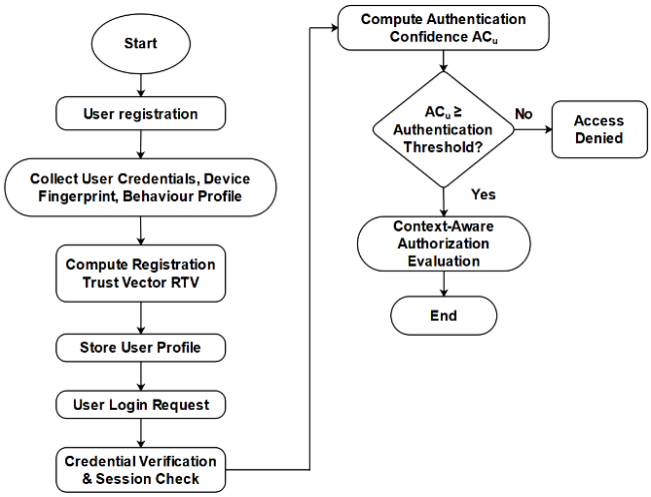

Figure 3. Flow chart of proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM) authentication and authorisation

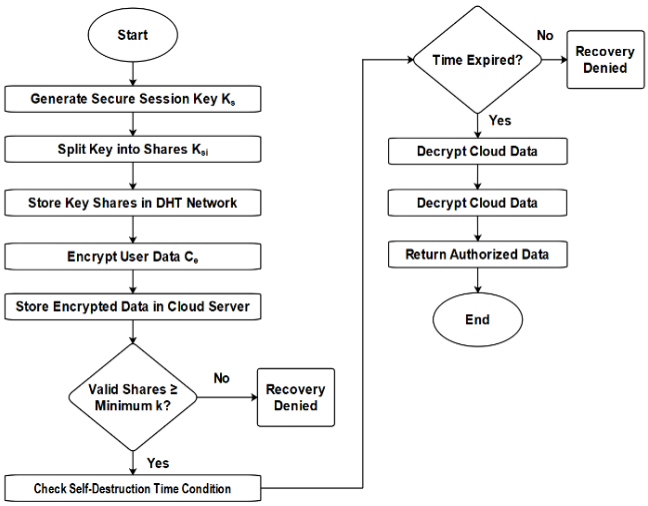

Figure 4. Flow chart of secure storage and data recovery in proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

Proposed SSACAM

Input: Uid, Cr, Df, Bp, Sf, Cx, Rr, Sr, TL, t
Output: AZ, Drec / Access Denied
Begin
STEP_1. Initialize thresholds and security parameters.
STEP_2. Collect Uid, Cr, Df, Bp, and Tv.
  Compute RTV.
  Store user profile.
STEP_3. Receive login request.
  Verify Cr, Sf, Dt, and An.
  Compute ACu.
STEP_4. If ACu < θc
    Return Access Denied
  End If
STEP_5. Compute Ru, Pur, Cx, and Sr.
  Evaluate authorization score.
STEP_6. If authorization is valid
    AZ = 1
  Else
    AZ = 0
    Return Access Denied
  End If
STEP_7. Generate nonce Ns.
  Compute session key Ks.
STEP_8. Split Ks into KSi shares.
  Store shares in DHT nodes.
STEP_9. Encrypt data and store Ce in cloud.
STEP_10. Collect valid shares from DHT.
  Compute KV.
STEP_11. If KV ≥ k
    Reconstruct Kr
  Else
    Return Access Denied
  End If
STEP_12. Compute Tsd.
STEP_13. If t < Tsd
    Decrypt Ce using Kr
    Recover Drec
  Else
    Return Data Expired
  End If
STEP_14. Return AZ and Drec
End

3.2 Attack model and security analysis

The common cloud security attacks are analysed against the proposed SSACAM framework to justify its security enhancement. Multi-factor registration trust, credential verification and authentication confidence threshold controls identity spoofing and invalid credential attacks. Session replay attack is mitigated through session freshness and time-validity checking. Context-aware authorisation prevents unauthorised access attempts by checking the user's role, permissions, contextual trust and the sensitivity of the resource before access is granted [69].

Device fingerprint scoring is used for managing device spoofing which helps to identify abnormal access from unknown devices. Since keeping the data in an encrypted form constrains cloud data leakage, the related key shares are shared separately via DHT nodes. The compromise of DHT key-shares is controlled, as data recovery is allowed only when the minimum number of valid key-shares is available. The self-destruction time condition prevents expired data recovery

attacks. Thus, SSACAM offers multi-layer protection via authentication, authorisation, encrypted storage, distributed key management, and time-bound secure recovery.

The adversary is assumed to observe cloud communication, generate fake login requests, replay old sessions, spoof device identity, and try to access unauthorised access, and compromise limited DHT key-share nodes. But the adversary can not learn valid credentials, full key shares, or the secret session key. The adversary's main goal is to impersonate users, access protected cloud resources, recover encrypted data, or utilise expired data. SSACAM prevents these attacks through adaptive authentication, context-aware authorisation, DHT-based key sharing, integrity verification, and self-destruction-based recovery control.

4. RESULTS AND DISCUSSION

In this section, we present the experimental evaluation of the proposed SSACAM framework for cloud security. The performance evaluation is done through the authentication accuracy, efficiency of access control, data confidentiality, rate of attack detection and response time. The proposed method is compared with conventional RBAC, ABAC and MFA techniques under identical test conditions. The results show that SSACAM provides better security, faster access decisions, better data protection, and higher attack detection capability in cloud environments.

4.1 Experimental test data, metric calculation, and statistical validation

The designed SSACAM framework is tested using the simulated cloud access request logs generated under normal and attack-based access scenarios. The test data set comprised legitimate user requests, invalid credential attempts, unauthorised role-based requests, expired session attempts, abnormal device fingerprint cases, contextual access violations, DHT key-share failure cases, and self-destruction time-expired recovery attempts. The same set of access requests are compared with RBAC, ABAC, MFA and the proposed SSACAM model. All the methods are evaluated on the same experimental settings with the parameters shown in Table 2 [70, 71]. The test data used for evaluation are summarized in Table 3.

Table 2. Experimental setup parameters for proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

Sl. No.	Parameter	Value
1	Session nonce length N_s	128 bit
2	Secure session key length K_s	256 bit
3	Current timestamp t	120 s
4	Local share generation time t_i	25 ms
5	Key share propagation delay t_p	40 ms
6	Authentication response time t_a	85 ms
7	Authorization decision time t_z	60 ms
8	DHT lookup time t_d	95 ms
9	Time-to-live TTL	300 s
10	Self-destruction deadline T_{sd}	600 s
11	Initial storage instant t_0	0 s
12	Secure data recovery time t_r	110 ms
13	Number of distributed key shares n	8 shares
14	Minimum shares for reconstruction k	5 shares
15	DHT storage nodes N_d	32 nodes

Table 3. Test data used for Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM) performance evaluation

Sl. No.	Test Scenario	Number of Requests	Purpose
1	Legitimate user access	3000	To verify correct authentication and access approval
2	Invalid credential attempt	1200	To test identity verification failure detection
3	Unauthorized role access	1000	To evaluate access control rejection accuracy
4	Session replay attempt	900	To test session freshness and replay resistance
5	Device fingerprint mismatch	800	To evaluate device-based trust verification
6	Contextual trust violation	900	To test location, time, and access-context validation
7	DHT key-share failure	700	To evaluate secure key reconstruction reliability
8	Self-destruction time-expired request	600	To test time-bound data recovery denial
9	Resource sensitivity violation	500	To evaluate high-risk resource protection
10	Mixed attack traffic	400	To test attack detection under combined threat cases

The performance metrics were calculated as follows:

$$\mathrm{AA} = \frac{\mathrm{TP_A} + \mathrm{TN_A}}{\mathrm{TP_A} + \mathrm{TN_A} + \mathrm{FP_A} + \mathrm{FN_A}} \times 100 \tag{15}$$

$$\mathrm{ACE} = \frac{\mathrm{N_{CD}}}{\mathrm{N_{TR}}} \times 100 \tag{16}$$

$$\mathrm{DC} = \frac{\mathrm{N_{UP}}}{\mathrm{N_{UA}}} \times 100 \tag{17}$$

$$\mathrm{ADR} = \frac{\mathrm{TP_D}}{\mathrm{TP_D} + \mathrm{FN_D}} \times 100 \tag{18}$$

$$\mathrm{RT} = \mathrm{T_{auth}} + \mathrm{T_{az}} + \mathrm{T_{DHT}} + \mathrm{T_{rec}} \tag{19}$$

$$\mathrm{Improvement} = \frac{\mathrm{M_{SSACAM}} - \mathrm{M_{baseline}}}{\mathrm{M_{baseline}}} \times 100 \tag{20}$$

For response time, lower value indicates better performance. Therefore, response-time improvement was calculated as given in Eq. (21).

$$\mathrm{RT_{Improvement}} = \frac{\mathrm{RT_{baseline}} - \mathrm{RT_{SSACAM}}}{\mathrm{RT_{baseline}}} \times 100 \tag{21}$$

where, AA denotes authentication accuracy, ACE denotes access control efficiency, DC denotes data confidentiality, ADR denotes attack detection rate, RT denotes response time,

TP_A and TN_A denote correctly accepted and correctly rejected authentication cases, FP_A and FN_A denote wrongly accepted and wrongly rejected authentication cases, N_{CD} denotes correct access decisions, N_{TR} denotes total requests, N_{UP} denotes unauthorized recovery attempts prevented, denotes total unauthorized attempts, TP_D denotes detected attacks, FN_D denotes undetected attacks, T_{auth} denotes authentication time, T_{az} denotes authorization time, T_{DHT} denotes DHT lookup time, and T_{rec} denotes secure recovery time.

The experimental dataset was generated by using a cloud access simulation framework developed to emulate realistic user authentication and authorisation activities in a multi-user cloud environment. The generated logs contained both valid and malicious access requests. Valid user credentials, authorised roles, trusted devices, valid session tokens and acceptable contextual access conditions were used to make legitimate requests. To generate attack scenarios, one or more security attributes were deliberately altered, such as invalid credentials, unauthorised role assignments, replayed sessions, device fingerprint mismatches, contextual trust violations, DHT key-share failures, and expired data-recovery requests.

The experimental parameters in Table 2 are chosen from the most published cloud security configurations and real world deployment settings. The session key length, nonce size, number of DHT nodes, propagation delay and time-to-live values were selected to represent realistic cloud storage and distributed key management environments.

To ensure that the evaluation is balanced, we controlled the attack ratio by setting the number of requests for each attack category as described in Table 3. The entire dataset consisted of 10,000 access requests. Among them, 3,000 were legitimate requests, and 7,000 were spread over multiple attack scenarios as attack-oriented requests. The attack distribution was kept the same across all experiments.

For reproducibility, we used a fixed random seed value of 42 for log generation, user-behavior simulation and attack injection. Additionally, the experiments were independently repeated 10 times, and the average values of authentication accuracy, access control efficiency, data confidentiality, attack detection rate, and response time were reported.

The generated dataset was split into 70% training data and 30% testing data. The training partition was used for parameter tuning, threshold calibration and trust-score optimisation. The testing partition was only used for performance evaluation and comparison with the RBAC, ABAC and MFA baseline methods.

4.2 Baseline configuration for comparative evaluation

For a fair comparison, the same workload of cloud access, number of requests, test scenarios, and the performance metrics were used to compare RBAC, ABAC, MFA, and the proposed SSACAM. RBAC, ABAC, and MFA are broad security models, fixed baseline configurations is represented in Table 4.

For a fair comparison, RBAC, ABAC, MFA, and SSACAM were implemented under the same cloud access workload. The simulation setup comprises 500 users, 100 cloud resources and 10,000 access requests. RBAC used 10 pre-defined roles with role-permission mapping. ABAC used six attributes namely user identity, role, device status, location, access time and resource sensitivity. Login approved MFA password and OTP verification All methods were exposed to the same attack scenarios, namely invalid credentials, unauthorised role access, session replay, device mismatch, contextual violation and expired recovery requests. Access was allowed only when the corresponding method satisfied its defined decision rule. This configuration guaranteed that all baseline methods were tested for the same workload, resource and attack conditions.

4.3 Raw performance data and improvement justification

The raw performance figures of RBAC, ABAC, MFA and the proposed SSACAM are captured under same test data, workload conditions and evaluation metrics to substantiate the reported improvement percentages. Baseline mean is calculated by taking the mean performance of RBAC, ABAC and MFA. The percentage improvement of SSACAM was then computed with respect to the baseline mean, as shown in Table 5.

Table 4. Baseline configurations used for fair comparison

Sl. No.	Method	Baseline Configuration	Parameter Setting	Evaluation Criteria
1	RBAC	Static Role-Based Access Control	User role and permission table	Correct access approval and rejection
2	ABAC	Attribute-Based Policy Control	User, device, resource, and context attributes	Policy matching accuracy and delay
3	MFA	Multi-factor login verification	Password and OTP-based verification	Authentication success and failure rate
4	SSACAM	Adaptive authentication with SeDAS recovery	Trust score, DHT key shares, and self-destruction time	Authentication, authorization, confidentiality, detection, and response time

Table 5. Raw performance data and improvement calculation for Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

Performance metric	RBAC	ABAC	MFA	Proposed SSACAM	Over RBAC	Over ABAC	Over MFA
Authentication accuracy (%)	74.6	78.9	82.0	90.9	21.8%	15.2%	10.9%
Access-control efficiency (%)	72.6	78.4	84.8	89.1	22.7%	13.6%	5.1%
Data confidentiality (%)	72.4	78.6	83.3	89.5	23.6%	13.9%	7.4%
Attack-detection rate (%)	73.0	76.0	85.9	91.0	24.7%	19.7%	5.9%
Response-time reduction (%)	360.0 ms	410.0 ms	424.0 ms	349.8 ms	2.8%	14.7%	17.5%

The improvement of SSACAM was calculated separately with respect to each baseline method, instead of calculating the average for RBAC, ABAC and MFA as a unified baseline. RBAC, ABAC and MFA are used for different security purposes. They cannot be considered as one combined technique. Authentication accuracy and attack detection rate were computed using TP, TN, FP and FN values. The efficacy of access control was calculated as the number of correct access approval and correct access rejection over total access requests. To determine the data confidentiality, the percentage of preventing unauthorised data recovery attempts successfully was calculated. Therefore, the revised comparison is a more clear and fair evaluation of SSACAM with respect to each baseline method.

All reported performance values are averaged over ten independent runs of simulations with a fixed random seed (42) and 70:30 training-testing split for reproducibility and statistical consistency.

4.4 Statistical significance analysis

To increase the statistical validity of the experimental results, all the methods were evaluated over 10 independent simulation runs using the same workload and attack distribution. Results are presented as mean, standard deviation (SD) and 95% confidence interval (CI). The 95% CI was computed as given in Eq. (22).

$$CI = \bar{x} \pm t_{0.975,9}\left(\frac{SD}{\sqrt{10}}\right) \tag{22}$$

where, $\bar{x}$ is the mean value, SD is the standard deviation, $n = 10$ is the number of experimental runs, and $t_{0.975,9} = 2.262$ is the critical value of the Student's t-distribution.

Table 6 presents the mean, standard deviation, and 95% confidence interval of each method across 10 independent simulation runs.

To examine whether the observed differences among RBAC, ABAC, MFA, and SSACAM are statistically significant, a one-way ANOVA test was conducted for each performance metric. The ANOVA results are presented in Table 7.

Further, paired t-tests were performed between SSACAM and each baseline method to verify the pairwise significance of the improvement. The results are shown in Table 8.

Table 6. Statistical summary of performance results over 10 independent runs

Metric	Method	Mean	SD	95% CI
Authentication accuracy (%)	RBAC	74.6	1.21	73.73–75.47
Authentication accuracy (%)	ABAC	78.9	1.10	78.11–79.69
Authentication accuracy (%)	MFA	82.0	0.95	81.32–82.68
Authentication accuracy (%)	SSACAM	90.9	0.82	90.31–91.49
Access-control efficiency (%)	RBAC	72.6	1.28	71.68–73.52
Access-control efficiency (%)	ABAC	78.4	1.03	77.66–79.14
Access-control efficiency (%)	MFA	84.8	0.91	84.15–85.45
Access-control efficiency (%)	SSACAM	89.1	0.78	88.54–89.66
Data confidentiality (%)	RBAC	72.4	1.31	71.46–73.34
Data confidentiality (%)	ABAC	78.6	1.08	77.83–79.37
Data confidentiality (%)	MFA	83.3	0.96	82.61–83.99
Data confidentiality (%)	SSACAM	89.5	0.84	88.90–90.10
Attack-detection rate (%)	RBAC	73.0	1.35	72.03–73.97
Attack-detection rate (%)	ABAC	76.0	1.22	75.13–76.87
Attack-detection rate (%)	MFA	85.9	0.90	85.26–86.54
Attack-detection rate (%)	SSACAM	91.0	0.77	90.45–91.55
Response time (ms)	RBAC	360.0	4.80	356.57–363.43
Response time (ms)	ABAC	410.0	5.30	406.21–413.79
Response time (ms)	MFA	424.0	5.70	419.92–428.08
Response time (ms)	SSACAM	349.8	4.20	346.80–352.80

Table 7. ANOVA results for statistical significance testing

Metric	*F*-Statistic	*p*-Value	Significance
Authentication accuracy	$F(3,36) = 448.54$	$p < 0.001$	Significant
Access-control efficiency	$F(3,36) = 506.69$	$p < 0.001$	Significant
Data confidentiality	$F(3,36) = 464.92$	$p < 0.001$	Significant
Attack-detection rate	$F(3,36) = 599.96$	$p < 0.001$	Significant
Response time	$F(3,36) = 527.54$	$p < 0.001$	Significant

Table 8. Pairwise paired t-test results between Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM) and baseline methods

Metric	SSACAM vs RBAC	SSACAM vs ABAC	SSACAM vs MFA
Authentication accuracy	$t(9) = 39.05, p < 0.001$	$t(9) = 31.88, p < 0.001$	$t(9) = 26.81, p < 0.001$
Access-control efficiency	$t(9) = 37.20, p < 0.001$	$t(9) = 28.20, p < 0.001$	$t(9) = 13.10, p < 0.001$
Data confidentiality	$t(9) = 38.30, p < 0.001$	$t(9) = 27.50, p < 0.001$	$t(9) = 17.20, p < 0.001$
Attack-detection rate	$t(9) = 39.80, p < 0.001$	$t(9) = 35.10, p < 0.001$	$t(9) = 14.40, p < 0.001$
Response time	$t(9) = 6.45, p < 0.001$	$t(9) = 34.60, p < 0.001$	$t(9) = 41.20, p < 0.001$

In Figure 5, the authentication accuracy of RBAC, ABAC, MFA and the proposed SSACAM are compared under the same experimental conditions. The authentication accuracies of RBAC, ABAC and MFA were 74.6%, 78.9% and 82.0% respectively, while the proposed SSACAM achieved 90.9%. Therefore, SSACAM achieves relative improvements of 21.8%, 15.2%, and 10.9% against RBAC, ABAC and MFA respectively. The enhancement is mainly due to the use of credential verification, device-fingerprint validation, behavioural-profile assessment, session-freshness checking and adaptive authentication-confidence evaluation. These mechanisms help SSACAM to identify the legitimate users more accurately and reduce the false acceptance and false rejection cases [72-74].

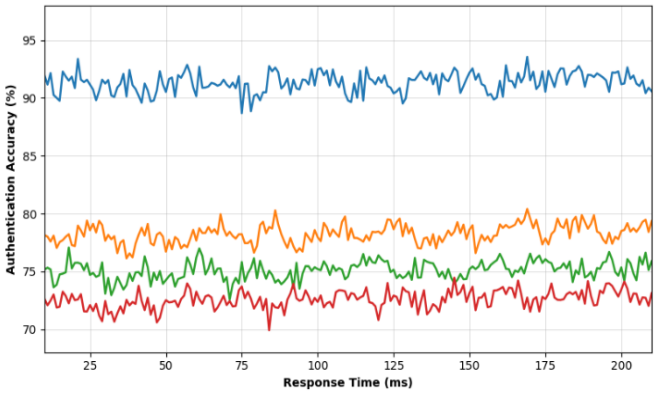

Figure 5. Authentication accuracy comparison of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), and proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

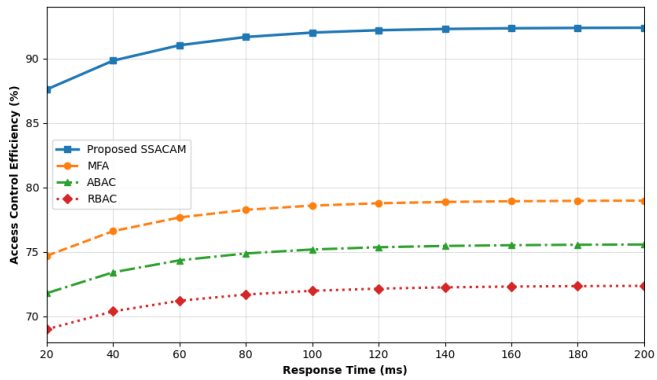

Figure 6. Access control efficiency comparison of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), and proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

Figure 6 shows the access-control efficiency of RBAC, ABAC, MFA and the proposed SSACAM. The efficiencies of access control in RBAC, ABAC and MFA were 72.6%, 78.4% and 84.8% respectively, whereas SSACAM achieved 89.1%. Thus, the proposed framework improves the access-control efficiency by 22.7%, 13.6% and 5.1% in comparison with RBAC, ABAC and MFA, respectively. The enhanced performance can be attributed to the context-aware authorisation mechanism, which takes user roles, permission compatibility, contextual trust, authentication confidence, and resource sensitivity into account simultaneously. Unlike the static or authentication-based baseline methods, SSACAM dynamically evaluates the user-resource interaction before granting access [75, 76].

Figure 7 shows the data-confidentiality performance of the evaluated security methods. The confidentiality values of RBAC, ABAC, and MFA were 72.4%, 78.6%, and 83.3%, respectively. The proposed SSACAM achieved 89.5%. Thus, SSACAM shows a relative improvement of 23.6% over RBAC, 13.9% over ABAC and 7.4% over MFA. The added privacy is achieved by encrypting the user data before storing it in the cloud and distributing the corresponding key shares individually among DHT nodes. Data recovery is only allowed when the required number of valid key shares are available, integrity verification passes, authorisation is granted and the request is received before the self-destruction deadline. This isolation prevents unauthorised data retrieval even if the cloud storage is compromised [77, 78].

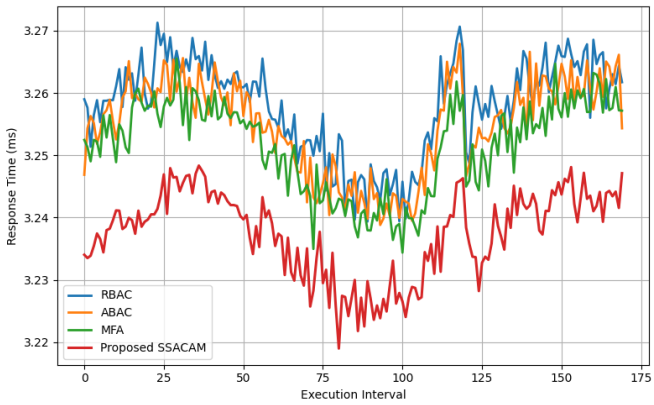

Figure 7. Data confidentiality comparison of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), and proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

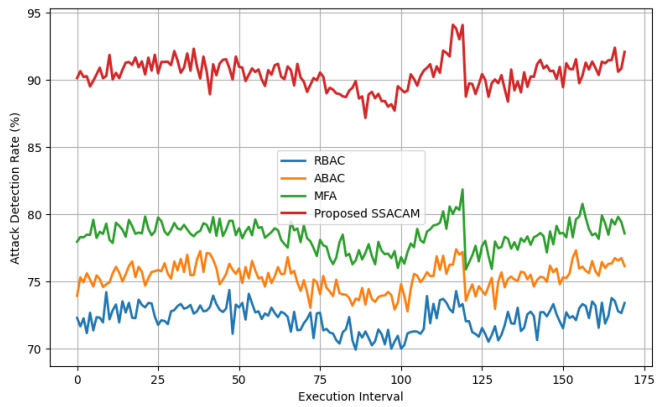

Figure 8. Attack detection rate comparison of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), and proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

Figure 8 shows the comparison of the attack-detection rates of RBAC, ABAC, MFA and the proposed SSACAM. The detection rates of the baseline methods are 73.0%, 76.0% and 85.9%, respectively, while SSACAM is 91.0%. Thus, SSACAM enhances the attack detection rate by 24.7%, 19.7%,

and 5.9% compared to RBAC, ABAC, and MFA, respectively. The enhancement is attributed to the multi-layer security analysis conducted by the suggested method such as detection of invalid-credentials, session-replay, device-fingerprint mismatch, contextual-trust, unauthorised role-access, DHT key-share and expired data-recovery. Therefore, SSACAM can identify a wider spectrum of cloud-security threats as compared to the individual baseline mechanisms [79-81].

Figure 9 shows the performance of RBAC, ABAC, MFA, and the proposed SSACAM across multiple security parameters including authentication accuracy, access control efficiency, data confidentiality, and attack detection rate. The proposed SSACAM achieves the highest performance for all parameters, demonstrating improved security and reliability compared with the conventional methods.

The results indicate that SSACAM outperforms due to the combination of adaptive authentication, context-aware authorisation, encrypted cloud storage, DHT-based key sharing and self-destruction-controlled recovery. RBAC is mostly about static roles. ABAC is about complex policy checking. MFA is mostly about login verification. SSACAM, on the other hand, verifies the user's identity, trust in the device, freshness of the session, context, key-share availability, and time validity to grant access.

The enhancement of authentication accuracy is because of multi-factor trust verification. Context-aware permission checking enhances the efficiency of access control. Storing encrypted data and key shares separately improves data confidentiality. The higher rate of attack detection is due to the detection of invalid credentials, session replay, device mismatch, contextual violation, and expired recovery attempts. Hence, SSACAM can provide stronger and more reliable cloud security than traditional ones.

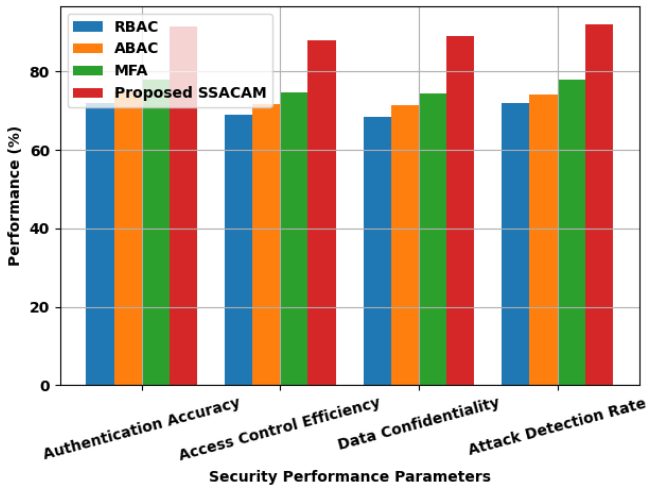

Figure 9. Overall performance comparison of Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), and proposed Secure SeDAS-Based Access Control and Authentication Mechanism (SSACAM)

5. CONCLUSION

The study proposed a SSACAM to improve security, privacy and access management in cloud environments. The proposed framework includes adaptive authentication, context-aware authorisation, encrypted cloud storage, DHT-based key-share distribution, secure session-key reconstruction, integrity verification, and self-destruction-controlled data recovery. On the contrary, although ABAC, RBAC and MFA primarily address individual security functions, SSACAM takes into account the user's identity, device's trust, session freshness, access context, resource sensitivity, key-share availability and time validity in an integrated manner. Experimental results demonstrate that SSACAM achieves 90.9% authentication accuracy, 89.1% access-control efficiency, 89.5% data confidentiality, and 91.0% attack-detection rate. The authentication accuracy increased by 21.8%, 15.2%, and 10.9%, the access-control efficiency increased by 22.7%, 13.6%, and 5.1%, the data confidentiality increased by 23.6%, 13.9%, and 7.4%, and the attack-detection rate increased by 24.7%, 19.7%, and 5.9%, respectively, compared individually with RBAC, ABAC, and MFA. Moreover, SSACAM outperformed RBAC, ABAC and MFA by reducing the response time by 2.8%, 14.7% and 17.5%, respectively. The results show that SSACAM offers a safe and efficient solution in the considered cloud access and attack scenarios. Future work will involve real-cloud deployment, larger multi-tenant datasets, dynamic threshold optimisation and testing against advanced coordinated attacks.

ACKNOWLEDGMENT

The authors sincerely thank the Department of Computer Science and Engineering, Shridevi Institute of Engineering and Technology (Affiliated to Visvesvaraya Technological University (VTU), Belagavi-590018), Tumakuru, and M. S. Ramaiah University of Applied Sciences, Bengaluru, for their support and encouragement in completing this research work.

REFERENCES

[1] Ruan, L., Bai, Y., Li, S.N., et al. (2023). Cloud workload turning points prediction via cloud feature-enhanced deep learning. IEEE Transactions on Cloud Computing, 11(2): 1719-1732. https://doi.org/10.1109/TCC.2022.3160228

[2] Motai, Y., Henderson, E., Siddique, N.A., Yoshida, H. (2020). Cloud colonography: Distributed medical testbed over cloud. IEEE Transactions on Cloud Computing, 8(2): 495-507. https://doi.org/10.1109/TCC.2015.2481414

[3] Eljak, H., Ibrahim, A.O., Saeed, F., et al. (2024). E-learning-based cloud computing environment: A systematic review, challenges, and opportunities. IEEE Access, 12: 7329-7355. https://doi.org/10.1109/ACCESS.2023.3339250

[4] Pourmajidi, W., Zhang, L., Steinbacher, J., Erwin, T., Miranskyy, A. (2025). A reference architecture for governance of cloud native applications. IEEE Transactions on Cloud Computing, 13(3): 935-952. https://doi.org/10.1109/TCC.2025.3578557

[5] Wu, C., Toosi, A.N., Buyya, R., Ramamohanarao, K. (2021). Hedonic pricing of cloud computing services. IEEE Transactions on Cloud Computing, 9(1): 182-196. https://doi.org/10.1109/TCC.2018.2858266

[6] Pragati, P., Mallikarjunaiah, K., Srikantaswamy, M. (2025). Intelligent modulation recognition using a hybrid CNN-Random Forest framework. Ingénierie des

Systèmes d'Information, 30(11): 3013-3019. https://doi.org/10.18280/isi.301118

[7] Basavaraju, N., Mahadevachar, V.K., Srikantaswamy, M. (2025). An enhanced secure framework for detecting and rectifying unauthorized access in cloud computing environments using the elliptic curve digital signature algorithm. Engineering, Technology & Applied Science Research, 15(4): 24188-24195. https://doi.org/10.48084/etasr.9764

[8] Sheela, S., Harshith, D., Raikar, G., et al. (2025). Securing pharmaceutical supply chains using AI-integrated blockchain technology. In 2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3), Bhubaneswar, India, pp. 1-6. https://doi.org/10.1109/ISAC364032.2025.11156788

[9] Wu, X.T., Xiong, Y.Y., Chen, B., Yang, C.N., Yan, W.Q., Peng, Q.Y. (2025). On the design of distributed multi-user secret image sharing for general access structures. IEEE Transactions on Dependable and Secure Computing, 22(6): 8025-8042. https://doi.org/10.1109/TDSC.2025.3603223

[10] Reddy, A.N.K., Anand, V., Khonina, S.N., Podlipnov, V.V., Juodkazis, S. (2021). Robust demultiplexing of distinct orbital angular momentum infrared vortex beams into different spatial geometry over a broad spectral range. IEEE Access, 9: 143341-143348. https://doi.org/10.1109/ACCESS.2021.3120836

[11] Lee, K., Tan, Y.H., Chuah, J.H., Youn, C.H. (2024). Privacy aware feature level domain adaptation for training deep vision models with private medical stethoscope dataset. IEEE Access, 12: 148201-148215. https://doi.org/10.1109/ACCESS.2024.3466226

[12] Chen, Y.F., Wang, C., Yu, J.J., Long, J.Y., Sang, B.H., Li, F. (2025). An efficient phase noise elimination method to facilitate cost-effective 100/200G Coherent PON. Journal of Lightwave Technology, 43(15): 7075-7082. https://doi.org/10.1109/JLT.2025.3569298

[13] Kul, S., Arslan, B., Sagiroglu, S. (2025). Explainable AI in smart grid applications. In 2025 IEEE 19th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), Antalya, Turkiye, pp. 1-6. https://doi.org/10.1109/CPE-POWERENG63314.2025.11027305

[14] Karkacıer S., Ozseven T. (2025). Dorsal hand vein biometric systems: A survey on methods, applications, and future directions. In 2025 3rd Cognitive Models and Artificial Intelligence Conference (AICCONF), Prague, Czech Republic, pp. 1-7. https://doi.org/10.1109/AICCONF64766.2025.11064271

[15] Zeng, L.F., Chen, S.B., Wei, Q.S., Feng, D. (2012). SeDas: A self-destructing data system based on active storage framework. In 2012 Digest APMRC, Singapore, pp. 1-8. https://ieeexplore.ieee.org/document/6407530.

[16] Geambasu, R., Kohno, T., Levy, A.A., Levy, H.M. (2009). Vanish: Increasing data privacy with self-destructing data. In Proceedings of the 18th Conference on USENIX Security Symposium, Montreal Canada, pp. 299-316. https://dl.acm.org/doi/10.5555/1855768.1855787.

[17] Yue, F.S., Wang, G.J., Liu, Q. (2010). A secure self-destructing scheme for electronic data. In 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Hong Kong, China, pp. 651-658. https://doi.org/10.1109/EUC.2010.104

[18] Ruj, S., Stojmenovic, M., Nayak, A. (2014). Decentralized access control with anonymous authentication of data stored in clouds. IEEE Transactions on Parallel and Distributed Systems, 25(2): 384-394. https://doi.org/10.1109/TPDS.2013.38

[19] Karatas, G., Akbulut, A. (2018). Survey on access control mechanisms in cloud computing. Journal of Cyber Security and Mobility, 7(3): 1-36. https://doi.org/10.13052/jcsm2245-1439.731

[20] Xie, B., Song, R., Li, Z., Deng, X., Xiao, B. (2026). Flexible and privacy-preserving access control framework for decentralized identity systems. IEEE Transactions on Information Forensics and Security, 21: 4205-4218. https://doi.org/10.1109/TIFS.2026.3676653

[21] Mythili, R., Venkataraman, R., Raj, T.S. (2020). An attribute-based lightweight cloud data access control using hypergraph structure. The Journal of Supercomputing, 76: 6040-6064. https://doi.org/10.1007/s11227-019-03119-7

[22] Rao, R.V., Selvamani, K. (2015). Data security challenges and its solutions in cloud computing. Procedia Computer Science, 48: 204-209. https://doi.org/10.1016/j.procs.2015.04.171

[23] Wang, C.M., Lu, J.H., Li, X.L., Cao, P., Zhou, Z.J., Wen, Q.L. (2023). A personal privacy data protection scheme for encryption and revocation of high-dimensional attribute domains. IEEE Access, 11: 82989-83003. https://doi.org/10.1109/ACCESS.2023.3296781

[24] Mann, Z.Á., Kunz, F., Laufer, J., Bellendorf, J., Metzger, A., Pohl, K. (2021). RADAR: Data protection in cloud-based computer systems at run time. IEEE Access, 9: 70816-70842. https://doi.org/10.1109/ACCESS.2021.3078059

[25] Gupta, R., Gupta, I., Singh, A.K., Saxena, D., Lee, C.N. (2023). An IoT-centric data protection method for preserving security and privacy in cloud. IEEE Systems Journal, 17(2): 2445-2454. https://doi.org/10.1109/JSYST.2022.3218894

[26] Huang, Y.S. (2025). Research on cloud data security computing framework based on fusion of homomorphic encryption and differential privacy. Journal of Cyber Security and Mobility, 14(4): 927-954. https://doi.org/10.13052/jcsm2245-1439.1447

[27] Chen, Z.Q., Song, Z.H., Zhang, T., Wei, Y. (2024). Design and performance of privacy protection model for big data transmission based on mixed encryption. Journal of Cyber Security and Mobility, 13(6): 1425-1448. https://doi.org/10.13052/jcsm2245-1439.1369

[28] Chang, V., Ramachandran, M. (2016). Towards achieving data security with the cloud computing adoption framework. IEEE Transactions on Services Computing, 9(1): 138-151. https://doi.org/10.1109/TSC.2015.2491281

[29] Ni, J.B., Zhang, K., Yu, Y., Yang, T.T. (2022). Identity-based provable data possession from RSA assumption for secure cloud storage. IEEE Transactions on Dependable and Secure Computing, 19(3): 1753-1769. https://doi.org/10.1109/TDSC.2020.3036641

[30] Ding, C.C., Shen, L., Zhang, X.Y., Lu, Y.D., Li, Y.F., Liang, Q.Y. (2025). A privacy protection system for consumer electronics data storage devices based on a

cloud computing–edge computing collaborative mechanism. IEEE Transactions on Consumer Electronics, 71(4): 10185-10195. https://doi.org/10.1109/TCE.2025.3600463
[31] Gupta, I., Saxena, D., Singh, A.K., Lee, C.N. (2023). SeCoM: An outsourced cloud-based secure communication model for advanced privacy preserving data computing and protection. IEEE Systems Journal, 17(4): 5130-5141. https://doi.org/10.1109/JSYST.2023.3272611
[32] Rawat, D.B., Doku, R., Garuba, M. (2021). Cybersecurity in big data era: From securing big data to data-driven security. IEEE Transactions on Services Computing, 14(6): 2055-2072. https://doi.org/10.1109/TSC.2019.2907247
[33] Liu, J.N., Luo, X.Z., Weng, J., et al. (2022). Enabling efficient, secure and privacy-preserving mobile cloud storage. IEEE Transactions on Dependable and Secure Computing, 19(3): 1518-1531. https://doi.org/10.1109/TDSC.2020.3027579
[34] Mann, Z.Á., Metzger, A., Prade, J., Seidl, R., Pohl, K. (2023). Cost-optimized, data-protection-aware offloading between an edge data center and the cloud. IEEE Transactions on Services Computing, 16(1): 206-220. https://doi.org/10.1109/TSC.2022.3144645
[35] Awaysheh, F.M., Aladwan, M.N., Alazab, M., Alawadi, S., Cabaleiro, J.C., Pena, T.F. (2022). Security by design for big data frameworks over cloud computing. IEEE Transactions on Engineering Management, 69(6): 3676-3693. https://doi.org/10.1109/TEM.2020.3045661
[36] Gai, C., Shen, W.T., Yang, M., Yu, J. (2023). PPADT: Privacy-preserving identity-based public auditing with efficient data transfer for cloud-based IoT data. IEEE Internet of Things Journal, 10(22): 20065-20079. https://doi.org/10.1109/JIOT.2023.3282939
[37] Wu, J.H., Mu, N.K., Lei, X.Y., Le, J.Q., Zhang, D., Liao, X.F. (2022). SecEDMO: Enabling efficient data mining with strong privacy protection in cloud computing. IEEE Transactions on Cloud Computing, 10(1): 691-705. https://doi.org/10.1109/TCC.2019.2932065
[38] Xu, Z.Y., He, D.B., Vijayakumar, P., Gupta, B.B., Shen, J. (2023). Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical WSNs. IEEE Journal of Biomedical and Health Informatics, 27(5): 2334-2344. https://doi.org/10.1109/JBHI.2021.3128775
[39] Lu, J.N., Qin, C., Zeng, Y., Wu, G.L., Chen, H. (2025). Collaborative recovery method for cyber-physical distribution system considering multiple coupling constraints. Journal of Modern Power Systems and Clean Energy, 13(5): 1752-1762. https://doi.org/10.35833/MPCE.2024.000925
[40] Wang, L.P., Guan, Z., Chen, Z., Hu, M.S. (2023). Enabling integrity and compliance auditing in blockchain-based GDPR-compliant data management. IEEE Internet of Things Journal, 10(23): 20955-20968. https://doi.org/10.1109/JIOT.2023.3285211
[41] Zhao, L.B., Li, B.H., Yuan, H.T. (2024). Cloud edge integrated security architecture of new cloud manufacturing system. Journal of Systems Engineering and Electronics, 35(5): 1177-1189. https://doi.org/10.23919/JSEE.2024.000112
[42] Henze, M., Matzutt, R., Hiller, J., et al. (2022). Complying with data handling requirements in cloud storage systems. IEEE Transactions on Cloud Computing, 10(3): 1661-1674. https://doi.org/10.1109/TCC.2020.3000336
[43] Zhu, Y.S., Zhou, Y.W., Wang, J., Yang, B., Zhang, M.W. (2025). Revocable-hierarchical-identity-based inner product function encryption in smart healthcare. IEEE Internet of Things Journal, 12(11): 15319-15332. https://doi.org/10.1109/JIOT.2025.3527556
[44] Shan, X.H., Yu, H.Y., Chen, Y.R., Chen, Y.W., Yang, Z. (2025). S2A-P2FS: Secure storage auditing with privacy-preserving flexible data sharing in cloud-assisted industrial IoT. IEEE Transactions on Mobile Computing, 24(7): 5699-5715. https://doi.org/10.1109/TMC.2025.3538057
[45] Ali, M., Sadeghi, M.R., Liu, X.M., Vasilakos, A.V. (2023). Anonymous aggregate fine-grained cloud data verification system for smart health. IEEE Transactions on Cloud Computing, 11(3): 2839-2855. https://doi.org/10.1109/TCC.2022.3229269
[46] Goldstein, A., Fink, L., Ravid, G. (2022). A cloud-based framework for agricultural data integration: A top-down-bottom-up approach. IEEE Access, 10: 88527-88537. https://doi.org/10.1109/ACCESS.2022.3198099
[47] Zhang, Z., Yang, Z., Du, X.H., Li, W.F., Chen, X.Y., Sun, L. (2021). Tenant-led ciphertext information flow control for cloud virtual machines. IEEE Access, 9: 15156-15169. https://doi.org/10.1109/ACCESS.2021.3051061
[48] Xin, Y.S., Ma, H., Zhang, R. (2025). Hodor: Robust fine-grained information flow control with full data traffic protection for cloud-edge computing. IEEE Transactions on Information Forensics and Security, 20: 3074-3087. https://doi.org/10.1109/TIFS.2025.3546846
[49] Song, Y., Mu, T.T., Wang, B. (2023). HV-SNSP: A low-overhead data recovery method based on cross-checking. IEEE Access, 11: 5737-5745. https://doi.org/10.1109/ACCESS.2023.3235787
[50] Jeong, W., Park, H., Park, C. (2023). KeyScrub: A reliable key backup and recovery method for blockchain. IEEE Access, 11: 91747-91755. https://doi.org/10.1109/ACCESS.2023.3308208
[51] Tian, Z.J., Zhang, Z.Y., Hanzo, L. (2023). Distributed multi-view sparse vector recovery. IEEE Transactions on Signal Processing, 71: 1448-1463. https://doi.org/10.1109/TSP.2023.3267995
[52] Qian, R.R., Qi, Y., Xue, Y., Zhou, T.Z., Zhang, J.Y. (2023). Federated consensus-based algorithm for stable recovery of sparse signals. IEEE Transactions on Vehicular Technology, 72(12): 15719-15731. https://doi.org/10.1109/TVT.2023.3289942
[53] Song, Y., Zhao, W.X., Wang, B. (2023). BPR: An erasure coding batch parallel repair approach in distributed storage systems. IEEE Access, 11: 44509-44518. https://doi.org/10.1109/ACCESS.2023.3257404
[54] Deng, L., Liu, X.Y., Zheng, H.F., Feng, X.X., Zhu, M., Tsang, D.H.K. (2025). Graph-tensor FISTA-Net: Edge computing-aided deep learning for distributed traffic data recovery. IEEE Transactions on Network Science and Engineering, 12(4): 2835-2847. https://doi.org/10.1109/TNSE.2025.3554634
[55] Zhao, Y., Liao, X.F., Zhou, M.L., Xing, H.X. (2024). Distributed computation for sparse recovery via continuous-time neurodynamic approach. IEEE Transactions on Consumer Electronics, 70(1): 3372-3383. https://doi.org/10.1109/TCE.2023.3281887

[56] Sun, X.C., Chen, J., Zhao, H.R., Zhang, W., Zhang, Y.C. (2023). Sequential disaster recovery strategy for resilient distribution network based on cyber–physical collaborative optimization. IEEE Transactions on Smart Grid, 14(2): 1173-1187. https://doi.org/10.1109/TSG.2022.3198696

[57] Yang, T., Li, H., Cai, S.T., Liu, Y.C. (2024). Distributed voltage control for microgrids against time-varying communication delay interference. IEEE Transactions on Smart Grid, 15(3): 2410-2423. https://doi.org/10.1109/TSG.2023.3321117

[58] Liu, Q., Zhang, L.P., Zhang, H.S., Wang, S.X., Ji, X.Q. (2025). Distributed secondary optimal control with fast voltage recovery and minimum generation cost for islanded DC microgrids. IEEE Transactions on Smart Grid, 16(1): 4-15. https://doi.org/10.1109/TSG.2024.3443242

[59] Zhang, H.C., Chen, C., Lei, S.B., Bie, Z.H. (2023). Resilient distribution system restoration with communication recovery by drone small cells. IEEE Transactions on Smart Grid, 14(2): 1325-1328. https://doi.org/10.1109/TSG.2022.3210771

[60] Shirko, O., Askar, S. (2023). A novel security survival model for quantum key distribution networks enabled by software-defined networking. IEEE Access, 11: 21641-21654. https://doi.org/10.1109/ACCESS.2023.3251649

[61] Zheng, Y.F., Zhou, M.L., Wang, S.L., et al. (2024). SecDR: Enabling secure, efficient, and accurate data recovery for mobile crowdsensing. IEEE Transactions on Dependable and Secure Computing, 21(2): 789-803. https://doi.org/10.1109/TDSC.2023.3262268

[62] Zhang, T., Tang, H., Mu, Y.F., Fang, D.H., Li, T., Jia, H.J. (2025). Iterative and accelerated techniques for distributed robust service restoration strategy in flexible distribution networks. IEEE Transactions on Smart Grid, 16(5): 3684-3699. https://doi.org/10.1109/TSG.2025.3581291

[63] Xavier, L.G.C., Meinhardt, C., Mendizabal, O.M. (2025). Beelog: Online log compaction for dependable systems. IEEE Transactions on Parallel and Distributed Systems, 36(4): 689-700. https://doi.org/10.1109/TPDS.2025.3541628

[64] Min, Y., Qi, N., Chen, Y.H., et al. (2025). A 56-Gb/s, 6.3-pJ/bit PAM-4 DFB laser driver incorporating asymmetric equalization and integrated CDR in 28 nm CMOS. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 33(10): 2749-2760. https://doi.org/10.1109/TVLSI.2025.3578353

[65] Miao, R.J., Zhang, Y.D., Zheng, Z.H., et al. (2023). CocoSketch: High-performance sketch-based measurement over arbitrary partial key query. IEEE/ACM Transactions on Networking, 31(6): 2653-2668. https://doi.org/10.1109/TNET.2023.3257226

[66] Kim, J., Lee, Y., Choi, D. (2025). Key-audit chain (KAC): A resilient and reliable key management framework for DID-based decentralized environments. IEEE Access, 13: 141871-141885. https://doi.org/10.1109/ACCESS.2025.3597593

[67] Poornima, M., Anitha, N., Mallikarjuna, S., Umashankar, L. (2025). An efficient internet of things based intrusion detection and optimization algorithm for smart networks. International Journal of Computing and Digital Systems, 17(1): 1-12. https://doi.org/10.12785/ijcds/1571001227

[68] Sheela S., Latha, A.P., Jyothi, S., et al. (2024). Enhancing stockpile management through deep learning with a focus on demand forecasting and inventory optimization. In 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), B G Nagara, Mandya, India, pp. 1-6. https://doi.org/10.1109/ICRASET63057.2024.10895608

[69] Deepak B.L., Sabin, T.T., Darshana, A., et al. (2024). Artificial Intelligence-driven power management system for enhanced efficiency in smart grids. In 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), B G Nagara, Mandya, India, pp. 1-5. https://doi.org/10.1109/ICRASET63057.2024.10895109

[70] Kumar N., Shivakumarswamy, P.M., Nikhil, N., et al. (2024). Optimal renewable energy wireless power management system for electric vehicles using predictive analytics. In 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, pp. 1-6. https://doi.org/10.1109/NMITCON62075.2024.10698816

[71] Liu, G.L., Sun, Q.Y., Wang, R., Hu, X.G. (2023). Nonzero-sum game-based voltage recovery consensus optimal control for nonlinear microgrids system. IEEE Transactions on Neural Networks and Learning Systems, 34(11): 8617-8629. https://doi.org/10.1109/TNNLS.2022.3151650

[72] Pei, J.H., Wang, J.Y., Wang, Z.Y., Shi, D.Y. (2023). Precise recovery of corrupted synchrophasors based on autoregressive Bayesian low-rank factorization and adaptive K-Medoids clustering. IEEE Transactions on Power Systems, 38(6): 5834-5848. https://doi.org/10.1109/TPWRS.2022.3221291

[73] Li, H.Q., Cheng, H.Q., Wang, Z., Wu, G.C. (2021). Distributed nesterov gradient and heavy-ball double accelerated asynchronous optimization. IEEE Transactions on Neural Networks and Learning Systems, 32(12): 5723-5737. https://doi.org/10.1109/TNNLS.2020.3027381

[74] Han, X.S., Liu, R., Li, Y., Yi, C., He, J.G., Wang, M. (2024). Accelerating neural BP-based decoder using coded distributed computing. IEEE Transactions on Vehicular Technology, 73(9): 13997-14002. https://doi.org/10.1109/TVT.2024.3391836

[75] Shen, X.C., Liu, Y., Li, F., Li, C.G. (2024). Privacy-preserving federated learning against label-flipping attacks on non-IID data. IEEE Internet of Things Journal, 11(1): 1241-1255. https://doi.org/10.1109/JIOT.2023.3288886

[76] Zhang, Z.Q., Peng, H.P., Li, L.X., Bao, S. (2025). Adaptive asynchronous federated learning for digital twin driven smart grid. IEEE Transactions on Smart Grid, 16(5): 4167-4183. https://doi.org/10.1109/TSG.2025.3579492

[77] Ye, J., Jiang, Y. (2026). Edge data auditing method supporting multi-keyword validation. IEEE Transactions on Mobile Computing, 25(1): 313-325. https://doi.org/10.1109/TMC.2025.3599724

[78] Xu, J.K., Zhang, Y.Q., Wang, L.B. (2023). Folded polynomial codes for coded distributed AAT-Type matrix multiplication. IEEE Transactions on Communications, 71(9): 5051-5064. https://doi.org/10.1109/TCOMM.2023.3286420

[79] Buldini, A., Mazzocca, C., Montanari, R., Uluagac, S.

(2025). Benchmarking selective disclosure mechanisms for verifiable credentials: A systematic comparison for security and privacy. IEEE Transactions on Information Forensics and Security, 20: 13205-13220. https://doi.org/10.1109/TIFS.2025.3636051

[80] Zhang, X., Wan, K., Sun, H., Wang, S.Q., Ji, M.Y., Caire, G. (2026). Optimal communication and key rate region for hierarchical secure aggregation with user collusion. IEEE Transactions on Information Theory, 72(2): 1030-1050. https://doi.org/10.1109/TIT.2025.3648767

[81] Mahendra, H.N., Pushpalatha, V., Mallikarjunaswamy, S., Sardar, T.H., Subramoniam, S.R. (2026). Hybrid quantum-classical convolutional neural network for multitemporal urban sprawl analysis using remote sensing data. Spatial Information Research, 34: 26. https://doi.org/10.1007/s41324-026-00687-8

NOMENCLATURE

U_{id}	User identity
Cr	User credential
Df	Device fingerprint
Bp	Behavioural profile
Sf	Session freshness
Cx	Contextual trust
Rr	Requested resource
Sr	Resource sensitivity
TL	Trust level
t	Current access time
C_e	Encrypted cloud data
D_{rec}	Recovered data
K_s	Secure session key
K_r	Reconstructed secure key
KS_i	Key share
KV	Valid key-share count
DHT	Distributed Hash Table
AZ	Authorization decision
RTV	Registration trust vector
AC_u	Authentication confidence of user
N_s	Session nonce
T_{sd}	Self-destruction time
TTL	Time-to-live

Greek Symbols

θ_c	Authentication confidence threshold
θ_a	Authorization threshold
θ_t	Contextual trust threshold
α	Registration trust weight coefficient
β	Device fingerprint weight coefficient
γ	Behavioural profile weight coefficient
λ	Authentication sensitivity parameter
μ	Access control weight coefficient

Subscripts

u	User
r	Resource
i	Key share index
s	Session
c	Confidence
a	Authorization
t	Time
rec	Recovered data
sd	Self-destruction
min	Minimum required value