International Journal of Computational Intelligence in Control

STRATEGY ATTRIBUTE BASED ACCESS CONTROL WITH IMPROVED KEY GENERATION METHOD FOR CLOUD COMPUTING

D.Karthik
Research Scholar, PG & Research Department of Computer Science, Government Arts College(Autonomous) (Affiliated to Bharathidasan University, Tiruchirappalli), Karur, Tamilnadu, India.

Dr. A. Vinayagam
Assistant Professor, PG& Research Department of Computer Science, Government Arts College(Autonomous) (Affiliated to Bharathidasan University, Tiruchirappalli), Karur, Tamilnadu, India.

Received: 26th September 2021
Revised: 19th October 2021
Accepted: 17th November 2021

Abstract: Attribute based encryption is a promising technique that achieves flexible and fine-graineddata access control over encrypted data, which is very suitable for a secure data sharing environmentsuch as the currently popular cloud computing.Apache Hadoop is a predominant software framework for distributedcompute and storage with capability to handle hugeamounts of data, usually referred to as Big Data. This data collectedfrom different enterprises and government agencies often includesprivate and sensitive information, which needs to be secured fromunauthorized access.However, traditional attribute-based encryption failsto provide an efficient keyword-based search on encrypted data, which somewhat weakens the powerof this encryption technique, as search is usually the most important approach to quickly obtaindata of interest from large-scale dataset. In this paper, Strategy -Attribute Based Access Control is proposed which is based on the full-blown key-strategyattribute-based encryption scheme.

KEYWORDS: Attribute based Access Control, Encryption, Key-Strategy, Cloud computing, Optimization algorithm

1. INTRODUCTION

The three service delivery models for cloud computing are: (1) Software as a Service (SaaS) in which cloud customers use the provider's applications over the Internet; (2) Platform as a Service (PaaS) in which customers deploy their self-created applications on a development platform that a cloud service provider provides; and (3) Infrastructure as a Service (IaaS) in which cloud customers rent processing, storage, network capacity from cloud service provider[1] [2] [3] [4]. The cloud computing paradigm is associated with security concerns both at the providers' end and consumers' end. While providers want to ensure that their resources and services are utilized only by authorized users; consumers would like to ensure that their data is securely maintained in the cloud and that the servers are not compromised [17][18][19][20][21][22][23][24][25][26][27][28][29][30].

Access control is a fundamental aspect of information security that is directly tied to the primary characteristics such as confidentiality, integrity and availability. Cloud computing service providers should provide the following basic functionalities from the perspective of access control: (i) Control access to the service features of the cloud based on the specified policies and the level of service

purchased by the customer. (ii) Control access to a consumer's data from other consumers in multi-tenant environments. (iii) Control access to both regular user functionsand privileged administrative functions. (iv) Maintain accurate access control strategy and up to date user profile information.

Access control models can be traditionally categorized into three types: (1) Discretionary (2) Mandatory and (3) Role-based. In the discretionary access control (DAC) model, the owner of the object decides its access permissions for other users and sets them accordingly. The UNIX operating system is a classical example for discretionary access control model. For example, the subject (i.e., owner of an object) can specify what permissions (read/write/execute) members in the same group may have and also what permissions all others may have. DAC models are usually used only with legacy applications and will incur considerable management overhead in the modern multi-user and multi-application environment, characteristic of distributed systems such as cloud. The Mandatory access control (MAC) models abstract the need for resource-user mapping and hence are more adaptable for distributed systems, compared to DAC models. The MAC model is typically used in multi-level security systems. Here, the access permissions are decided by the administrator of the system, and not by the subject. In a multi-level MAC model, each subject as well as object is identified with a security level of classification (e.g., Unclassified, Classified, Secret and Top Secret). The Bell La Padula model recommends the "no read-up" rule and "no-write-down" rule for maintaining confidentiality of information. The Biba model recommends the "no-write-up", "no-read-down" and "no-execute-up-or-down" rules for maintaining the integrity of information. In a Role-based access control model (RBAC), a user has access to an object based on his/her assigned role in the system. Roles are defined based on job functions. Permissions are defined on job authority and responsibilities of the job. Operations on the object are invoked based on the permissions. RBAC models are more scalable than the discretionary and mandatory access control models, and more suitable for use in cloud computing environments, especially when the users of the services cannot be tracked with a fixed identity[5] [6] [7].

1.1 Attribute based Access Control

Attribute-based encryption (ABE) is more suitable (compared to the traditional public-key infrastructure based or identity-based encryption) to protect the privacy and secrecy of data in a cloud computing environment. ABE is useful when the source of the data knows neither the identity of the recipient nor their public key; but only knows certain attributes of the recipient. For example, imagine user Alice wishing to communicate with her former classmates, but she does not know their email addresses. ABE identifies a user with a set of attributes. In [15], Sahai and Waters (SW) propose ABE as follows: Given a secret key on a set of attributes ω, one can decrypt a ciphertext encrypted with a public key based on a set of attributes ω', only if the sets ω and ω' overlap sufficiently as determined by a threshold value t. The SW scheme also proposes the use of an access tree-based strategy to decide on the attributes required to decrypt a message.

2. RELATED WORKS

Xie, Xingxing, et al [8] proposed a new ciphertext-strategy ABE (CP-ABE) construction with efficient attribute and user revocation. Besides, an efficient access control mechanism is given based on the CP-ABE construction with an outsourcing computation service provider.

Ruj, Sushmita, and Amiya Nayak [9] propose a decentralized security framework for smart grids that supports data aggregation and access control. The proposed access control mechanism uses attribute-based encryption (ABE) which gives selective access to consumer data stored in data

repositories and used by different smart grid users. RTUs and users have attributes and cryptographic keys distributed by several key distribution centers (KDC).

Wang, Changji, and Jianfa Luo [10] proposed a new Key-strategy attribute-based encryption (KP-ABE) construction with constant ciphertext size. In our construction, the access strategy can be expressed as any monotone access structure. Meanwhile, the ciphertext size is independent of the number of ciphertext attributes, and the number of bilinear pairing evaluations is reduced to a constant.

Hu, Vincent C., et al [11] This document provides Federal agencies with a definition of attribute-based access control (ABAC). ABAC is a logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against strategy, rules, or relationships that describe the allowable operations for a given set of attributes. This document also provides considerations for using ABAC to improve information sharing within organizations and between organizations while maintaining control of that information.

Choi, Chang, Junho Choi, and Pankoo Kim [12] proposed Onto-ACM (ontology-based access control model), a semantic analysis model that can address the difference in the permitted access control between service providers and users. The proposed model is a model of intelligent context-aware access for proactively applying the access level of resource access based on ontology reasoning and semantic analysis method.

Chen, Hongsong, Bharat Bhargava, and Fu Zhongchuan[13] proposed a multilabels- based access control model that provides flexible security protection to big data. Our scalable access control model uses labels to provide scalable granularity access protection to a big data application in the healthcare area.

Su, Jinshu, et al [14] described ePASS, a novel ABS scheme that uses an attribute tree and expresses any strategy consisting of AND, OR threshold gates under the computational Diffie–Hellman problem. Users cannot forge signatures with attributes they do not possess, and the signature provides assurance that only a user with appropriate attributes satisfying the strategy can endorse the message, resulting in unforgeability. However, legitimate signers remain anonymous and are indistinguishable among all users whose attributes satisfy the strategy, which provides attribute privacy for the signer.

3. DRAGON FLY OPTIMIZATION ALGORITHM

Xin-She Yang developed the Dragon Fly algorithm [15] in the year 2008 by the inspiration gotfrom the fireflies. Three main assumptions were made here; they are (a) all FF are unisex (b)Attractiveness is directly proportional to brightness, and attractiveness is inverselyproportionally to distance. (c) The objective function defines the brightness of FF. Each FFhas its attractiveness, which is represented as ρ, and it decreased with distance x. Equation (3) represents the attractiveness between two FF in whichρ_0 denotes maximum attractiveness, and it is referred to as the light absorption coefficient. Further, g and h are the two FF at position K_g and K_h, their distance is evaluated using the mathematical equation (4) in which b represents the count of dimensions. The movement of FF is represented in Eq. (5). The light intensity M_h of FF is evaluated based on the distance between the

fireflies. The mathematical equation of FF is shown in Eq. (2) in which M_0 represents the original light intensity [16].

$$M = M_0 e^{-x} \quad (2)$$

$$\rho(x) = \rho_0 e^{-x}, v \geq 1 \quad (3)$$

$$x_{gh} = \left\| K_g - K_h \right\| = \sqrt{\sum_{w=1}^{b} (K_{g,w} - K_{h,w})^2} \quad (4)$$

$$K_{best} = K_g + \rho_0^{-\gamma x_{gh}^2} \left(K_h - K_g\right) + \omega \left(rand - \frac{1}{2}\right) \quad (5)$$

The first term denotes the current position of FF, and the second term denotes the attractiveness of FF. The last term describes the random movement of FF. Theinitial position of FF is denoted as per Eq. (5).

3. PROPOSED STRATEGY ATTRIBUTE BASED ACCESS CONTROL APPROACH FOR BIG DATA SECURITY

The proposed Strategy – Attribute based Access Control approach composed of the following Algorithms:

Step 1: Initialization of the Parameters

Step 1.1: Input: Number of Authorities and security parameters.

Step 1.2: Output: The generation of Master key and Public Key.

Step 1.3:The authority chooses the master key as the secret key.

Step 1.4:The authority chooses the prime order and the bilinear group.

Step 1.5: The number of attributes in the authority are generated by the bilinear group.

Step 1.6: The cryptographic hash function is defined.

Step 2: Encryption

In this stage, the number of attributes and security parameter as the input. The output is generated by this stage is Master Key and the Public Key. Here the master key is considered as the secret key by the authority. Then the authority can choose a prime order and bilinear group.So the bilinear group generates the attributes of the authority is given by $h_1, \ldots, h_U \rightarrow G$. The cryptographic hash function is defined.

Algorithm 1: Step up $(\lambda, U) \rightarrow$ (PK, MK). The setup algorithm takes the security parameter λ and the number of attributes as input and output public parameters PK and master key MK. The authority keeps the MK as its secret.

The system parameters are generated asfollows. The algorithm takes as input the securityparameter $\lambda$and the number of attributes in the system. The authority will choose a bilinear group G of prime order ρ. The generator of the G is g and U random group elements $h_1, \ldots, h_U \rightarrow G$ that are associated with the U attributesin the system.Define a cryptographic hash

functionH.Furthermore, it randomly chooses values$a, \alpha, \beta, \gamma \in Z_p$and $h \in G$.The public parametersPK and master secret key MK are

$$PK = e(g,g)^{a}, g^{\alpha}, H, g, g^{\beta}, g^{\gamma}, h, h_1, \ldots., h_U$$

$$MK = \alpha, \beta, \gamma$$

Encrypt(PK, M,A) → (CT): The encryption algorithm takes the public parameters PK, a message M, and an access structure A as input, and outputs the ciphertext CT.

The encryption algorithm takes asinput the public parameters PK, a message M toencrypt, and an LSSS access structure(M, ρ). The functionρassociates rows of M to attributes. Let M be a l3n matrix.The algorithmfirst chooses a random vector$V = (s, y_2, \ldots, y_n) \in Z_p^n$. These values will be used to share the encryption exponent s. For i=1 to l, it calculates $\lambda_i = v.M_i$, where M_i is the vector corresponding to the ith row of M. In addition, thealgorithm chooses random$r_1, \ldots, r_l \in Z_p$. The ciphertext is published as CT along with adescription of(M, ρ).

$$CT = \begin{Bmatrix} C = Me(g,g)^{\alpha s}, C' = g^{s}, C'' = g^{\beta s}, C''' = \\ g^{\gamma s}, \left(C_1 = g^{a\lambda_1} h_{p(1)}^{-r_1}, D_1 = g^{r_1}\right), \ldots, \\ \left(C_1 = g^{a\lambda_1} h_{p(l)}^{-r_1}, D_1 = g^{r_1}\right) \end{Bmatrix}$$

Key Gen(MK, S, GID) → (CT): The key generationalgorithm takes the master key MK, a set of attributesS that describe the key and the global identifierGID as input, and outputs a private key SK.

The authority takes as input the master key MK, a set of attributesS_1 that describes the key and user's global identifier $GID = u$. It Dragon Fly Optimization (DFO) algorithm chooses $t \in Z_p^*$ and computes the private key SK as

$$K = g^{\alpha} g^{at} h^{u\beta}, L_2 = h^{\gamma}, K_x = h_x^t, \forall x \in S_1$$

Key Generation by Dragon Fly Optimization

Step 1: Key Generation by Dragon Fly Algorithm

Step 1.1: The random values are selected using the DragonFly Optimization algorithm. Initialize Maximum generation Max_g and intensity of light M_g Light Absorption coefficient is defined.

Step 1.2:While (t<Max_g)

Step 1.3: For g=1:n_1 for all DF.

Step 1.4: For h=1:n_2 for all DF

Step 1.5: IF ($M_h > M_g$)

Step 1.6: FF g is moved towards h

Step 1.7: End if

Step 1.8: Attractiveness varies with distance x.

Step 1.9:New solutions are evaluated, and light intensity is updated

Step 1.10:End for h

Step 1.11:End for g

Step 1.12: DF are ranked, and the best DF is predicted

Step 1.13:End while

Step 1.14: Similarly, the receiver selects a private key dB and generates its public keyPB =dB*G.

Step 1.15: The sender generates the security key "SK= dA*PB," and the receiver also generates the security key "SK= dB*PA".

Step 1.16: Return SK

Out Key Gen(SK) → (RK, OutK). The outsourced keygeneration algorithm takes the private key SK asinput and outputs a retrieve key RK and an out sourced key Out K.

To create a outsourced key for theprivate key SK.The user chooses a random value$z \in Z_p^*$ and gets the retrieve key RK=z. The outsourced key Out K are published as

$$K' = g^{\alpha/z} g^{at/z} h^{u\beta/z}, L_1' = g^{t/z}, L_2' = h^{\gamma/z}$$

$$L_3' = h^{u/z}, K_x' = h_x^{t/z}, \forall x \in S_1$$

Trans Key Gen(MK, S2) → (TK). It first calls the Key Gen algorithm, and then calls the Encrypt algorithmto encrypt d_1, d_2 under the attributes set S_2,and finally outputs the transform key TK.

The authority calls the Key Gen algorithm, then chooses random$t^1, d_1, d_2 \in Z_p$, and compute $H(d_1), H(d_2)$. Then, it encrypts $d_1, d_2$with the access structureA_2 using the Encrypt $(PK, (d_1, d_2), A_2)$ algorithm. It outputs the TK as

$$T = g^{\alpha} g^{at'} h^{H(d_1),H(d_2)}, T' = g^{t'}, T_x = h_x^{t'}, \forall x \in S_1$$

$$T'' = En_{A_2}(d_1, d_2)$$

Re Enc(TK, CT) → (CT_2): This algorithm takes asinput the TK and CT that is associated with A.Finally, it outputs the updated ciphertext CT_2.

This algorithm takes an input the TK and CT that is associated with A. Suppose that S_1 satisfies the access structure (M_1, ρ_1) and let $I_1 \subset \{1,2, \dots, l_1\}$ be defined as $I_1 = \{i: \rho_1(i) \in S_1\}$. Then, let $\{\omega_i \in Z_p\}_{i \in I_1}$ be a set of constants such that if λ_i is a valid share of any secret s according to M_1, then $\sum_{i \in I_1} \omega_i \lambda_i = s$. It outputs the updated ciphertext CT_2 as follows:

$$C_1' = Me(g,g)^{\alpha s}, C_3' = En_{A_2}(d_1, d_2), C_4' = g^s$$

$$C_2' = \frac{e(C', T)}{\prod_{i \in I} \left(e(C_i, T') e\left(D_i, T_{\rho(i)}\right)\right)^{\omega_i}}$$

$$= e(g,g)^{\alpha s} e(g,h)^{sH(d_1)H(d_2)}$$

Decrypt(CT, SK, CT2, SK2) → (M): The decrypt algorithmtakes in the updated ciphertext CT2, the privatekey SK. It outputs the plaintext M if decryptionsucceeds, and a rejected symbol ⊥ otherwise.

The recipient can decrypt the ciphertext if his key's attributes satisfy the access structure associated with the ciphertext. To decrypt CT using the private key SK, the recipient first checks whether the equation $e(L_2, C^{tu}) = e(h^u, C''')$ holds. If it cannot pass the verification, which means that the key comes from a malicious authority, the recipient will stop the process, which can avoid the waste of network resources due to invalid secret keys. Next, the recipient computes

$$\frac{e(C', K)}{e(h^u, C'') \prod_{i \in I} \left(e(C_i, L_1) e\left(D_i, K_{\rho(i)}\right)\right)^{\omega_i}}$$

Then the message M can be got by computing $C/e(g,g)^{\alpha s}$. To decrypt the CT_2using the private key SK_2, the recipient gets (d_1, d_2). Next, the recipient computes

$$\frac{C_1'}{\left(C_2'/e\ (\pi)(C_4', h^{H(d_1)H(d_2)})\right)} = M$$

Out Decrypt(RK, Out K, CT) → (M): The Out Decrypt algorithm takes as input a retrieve key RK, an outsourced key Out K and cipher text CT. It outputs themessage M if S ∈ A and a rejected symbol ⊥ otherwise.A is usually the access structure for aLinear Secret Sharing Scheme (LSSS),which is a threshold.

The algorithm takes as input a outsourced key Out K for a set S, a ciphertext CT for access structure (M, ρ) and let $I \subset \{1,2,\dots,l,l+1,\dots,2l\}$ be defined as $I = \{i: \rho(i) \in S\}$, where $\rho(l+i) = \rho(i)$ and $\rho(0) = \rho(1)$. Then, let $\{\omega_i \in Z_p\}_{i \in I}$ be a set of constants such that if λ_i is a valid share of any secret s according to M, then $\sum_{i \in I_1} \omega_i \lambda_i = s$.

Strategy Updating: When the data owner wants to change the access strategy from previous strategy A to a new strategy A, he first runs the update-key generation algorithm and then sends the update keys to the cloud server. After receiving update keys, the cloud server executes the ciphertext-update algorithm to update the ciphertext.

4. RESULT AND DISCUSSIONS

In this paper, we have presented a strategy-attribute based access control system of the big data architecture security for the cloud storage systems, which is both efficient and secure. Table 1 depicts the Encryption computing time taken in seconds for the varying number of authorities involved in the strategy-attribute based access control system. Figure 1 represents the graphical representation of the encryption computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems. From table 1 and figure 1, it is clear that the proposed SA-BAC performs the encryption in less time than the existing A-BAC system.

Table 1: Encryption Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Authorities

Number of Authorities	Encryption time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
2	12	22
3	18	30
4	22	41
5	25	52
6	29	63
7	38	81
8	52	97
9	64	105
10	78	128
11	85	146

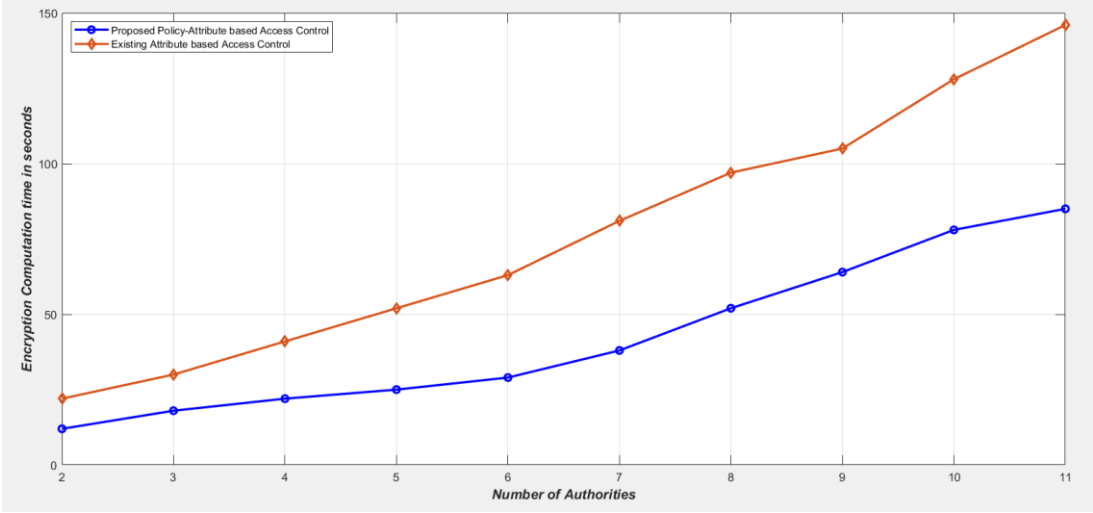

Figure 1: Graphical Representation of the encryption computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems

Table 2 depicts the key generation computation time in seconds using proposed SA-BAC and existing A-BAC systems for varying number of authorities. Figure 2 represents the graphical representation of the key generation computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems. From table 2 and figure 2, it is clear that the proposed SA-BAC performs the key generation in less time than the existing A-BAC system.

Table 2: Key Generation Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Authorities

Number of Authorities	Key Generation time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
2	18	25
3	28	39
4	37	51
5	49	78
6	54	89
7	65	99
8	72	108
9	78	122
10	85	131
11	92	139

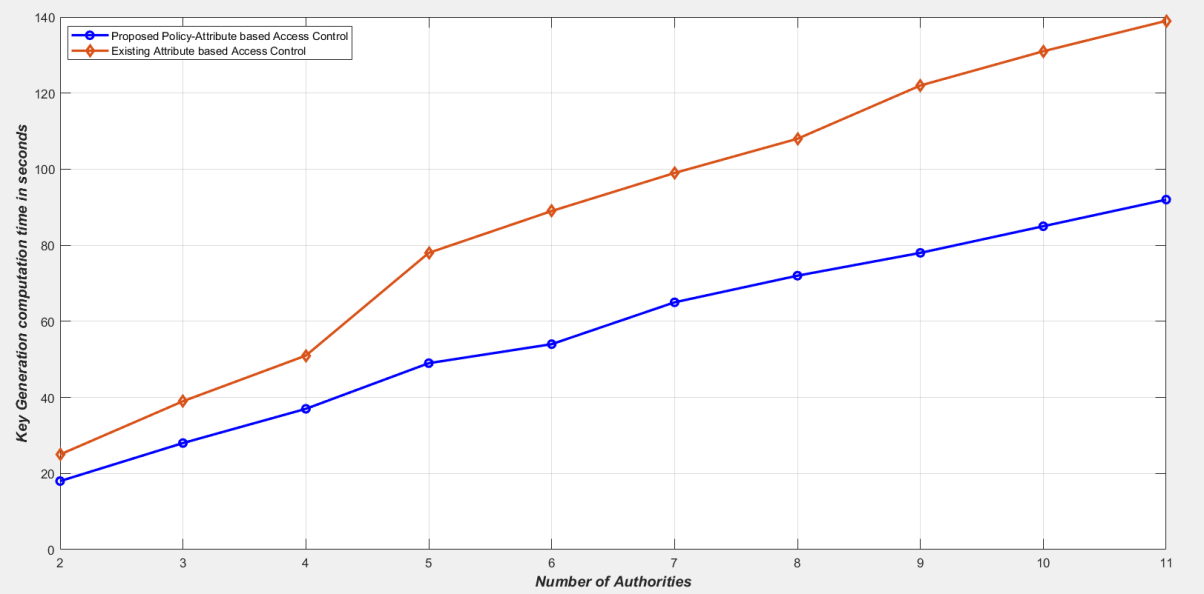

Figure 2: Graphical Representation of the Key Generation computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems

Table 3 depicts the Decryption computation time in seconds using proposed SA-BAC and existing A-BAC systems for varying number of authorities. Figure 3 represents the graphical representation of the decryption computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems. From table 3 and figure 3, it is clear that the proposed SA-BAC performs the decryption in less time than the existing A-BAC system.

Table 3: Decryption Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Authorities

Number of Authorities	Decryption time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
2	16	28
3	21	35
4	32	48
5	39	56
6	49	68
7	56	75
8	68	89
9	75	95
10	82	109
11	93	115

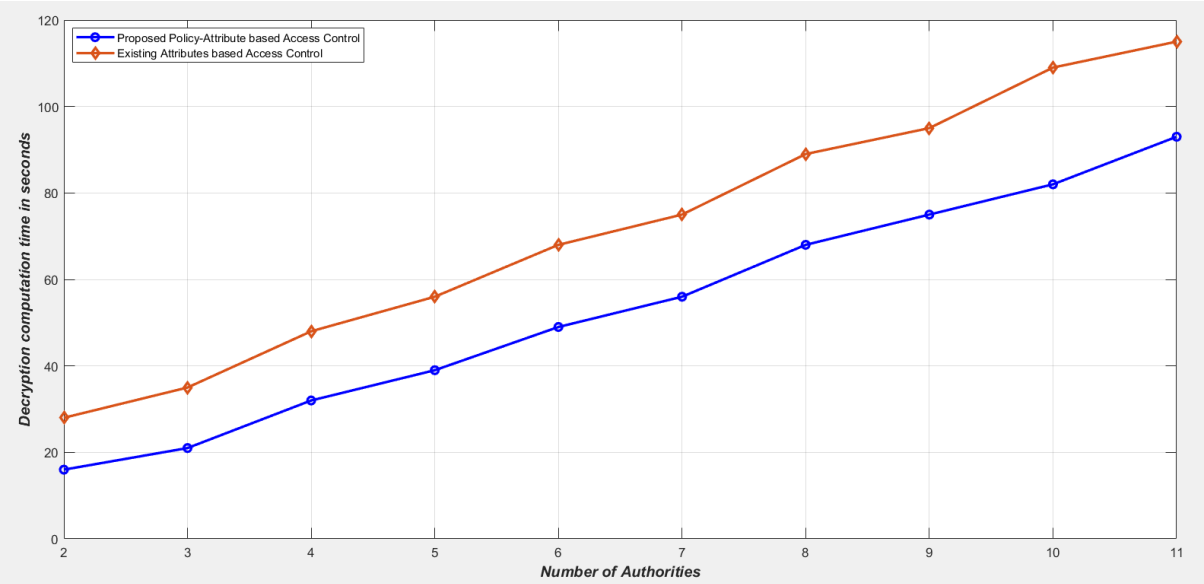

Figure 3: Graphical Representation of the Decryption computation time in seconds with number of authorities using proposed SA-BAC and existing A-BAC systems

Table 4 depicts the Encryption computation time in seconds using proposed SA-BAC and existing A-BAC systems for varying number of attributes per authority.Figure 4 represents the graphical representation of the encryption computation time in seconds with number of attributes per authorities using proposed SA-BAC and existing A-BAC systems. From the table 4 and figure 4, it is clear that the proposed SA-BAC performs the encryption in less time than the existing A-BAC system for the varying number of attributes per authority.

Table 4: Encryption Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Attributes per Authority

Number of attributes per authority	Encryption time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
6	21	35
8	29	48
10	35	68
12	51	79
14	63	92
16	75	118
18	89	129
20	97	135
22	101	147
24	112	163

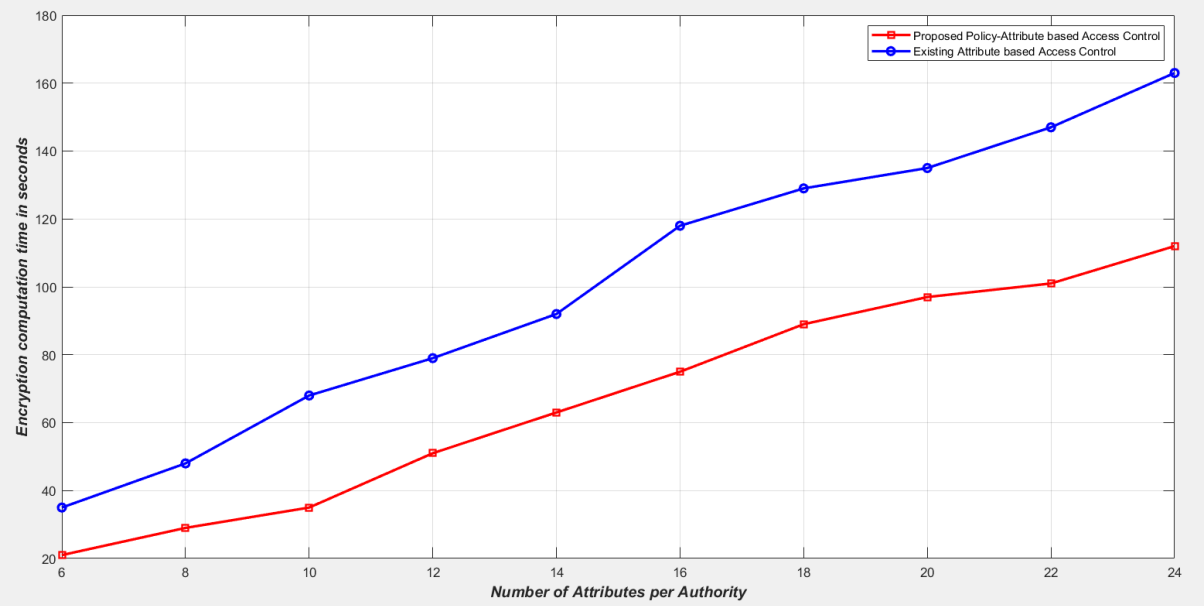

Figure 4: Graphical Representation of the encryption computation time in seconds with number of attributes per authority using proposed SA-BAC and existing A-BAC systems

Table 5 depicts the Key Generation computation time in seconds using proposed SA-BAC and existing A-BAC systems for varying number of attributes per authority. Figure 5 represents the graphical representation of the key generation computation time in seconds with number of attributes per authorities using proposed SA-BAC and existing A-BAC systems. From the table 5 and figure 5, it is clear that the proposed SA-BAC performs the key generation in less time than the existing A-BAC system for the varying number of attributes per authority.

Table 5: Key Generation Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Attributes per Authority

Number of attributes per authority	Key Generation Computation time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
6	21	38
8	32	54
10	46	71
12	59	92
14	70	105
16	89	118
18	97	126
20	101	138
22	119	145
24	121	167

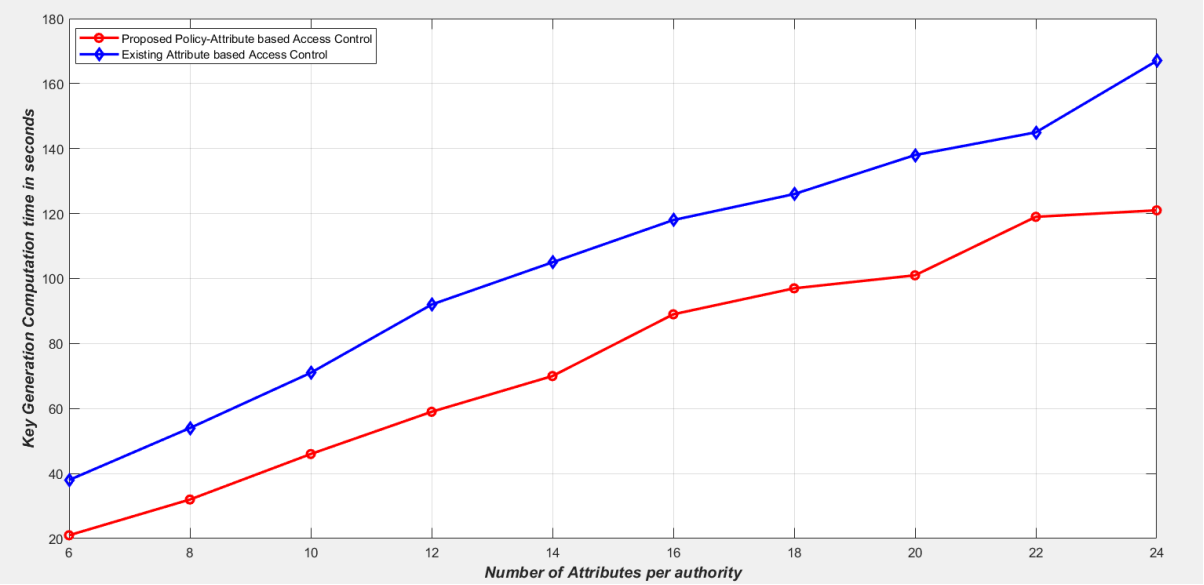

Figure 5: Graphical Representation of the key generation computation time in seconds with number of attributes per authority using proposed SA-BAC and existing A-BAC systems

Table 6 depicts the Decryption computation time in seconds using proposed SA-BAC and existing A-BAC systems for varying number of attributes per authority. Figure 6 represents the graphical representation of the decryption computation time in seconds with number of attributes per authorities using proposed SA-BAC and existing A-BAC systems. From the table 6 and figure 6, it is clear that the proposed SA-BAC performs the decryption in less time than the existing A-BAC system for the varying number of attributes per authority.

Table 6: Decryption Computation time in seconds by the proposed Strategy-Attribute based Access Control and existing Attribute based Access control system for varying number of Attributes per Authority

Number of attributes per authority	Decryption Computation time in seconds	
	Proposed Strategy-Attribute based Access Control	Existing Attribute-based Access control
6	18	26
8	28	39
10	39	56
12	48	72
14	64	89
16	75	98
18	88	110
20	97	128
22	105	139
24	116	156

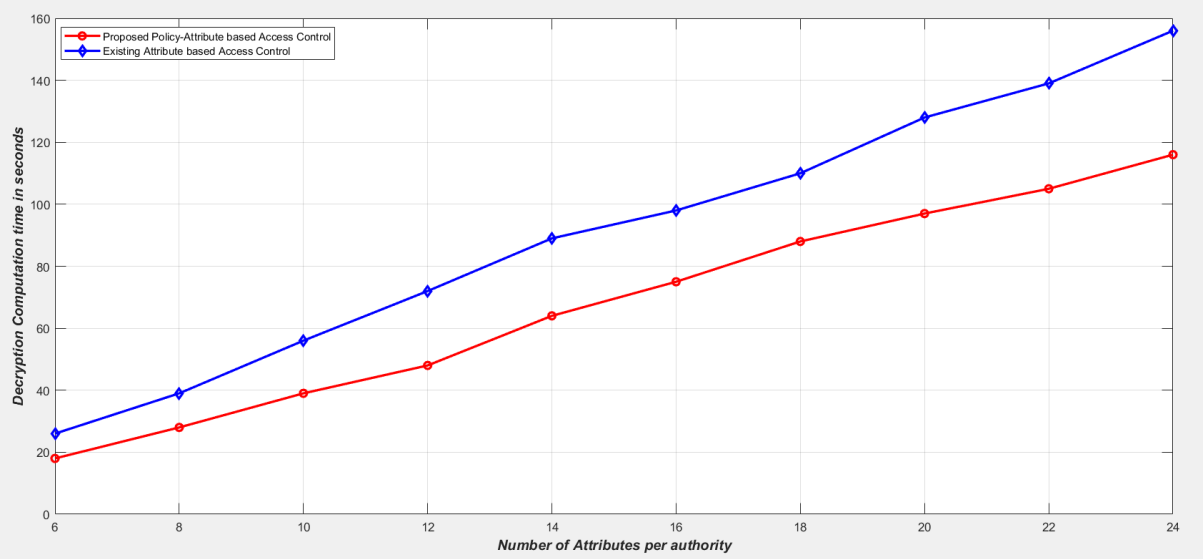

Figure 6: Graphical Representation of the decryption computation time in seconds with number of attributes per authority using proposed SA-BAC and existing A-BAC systems

Table 7 depicts the computation overhead by Proposed Strategy-Attribute based Access Control and Existing Attribute based Access Control. Figure 7 represents the graphical representation of the Computational Overhead in (ms) for the Proposed SA-BAC and existing A-BAC method for given number of requests. From the table 7 and figure 7, it is clear that the proposed P-ABC method takes less computational time than the existing ABC.

Table 7: Computation overhead in (milliseconds) using Proposed Strategy-Attribute based Access Control and Existing Attribute based Access Control for varying number of requests

Number of Requests	Computation Overhead in (ms)	
	Proposed Strategy-Attribute based Access Control (P-ABC)	Existing Attribute based Access Control (ABC)
1000	985	1021
2000	1041	1125
3000	1174	1257
4000	1214	1384
5000	1374	1498
6000	1414	1532
7000	1574	1684
8000	1698	1725
9000	1702	1824

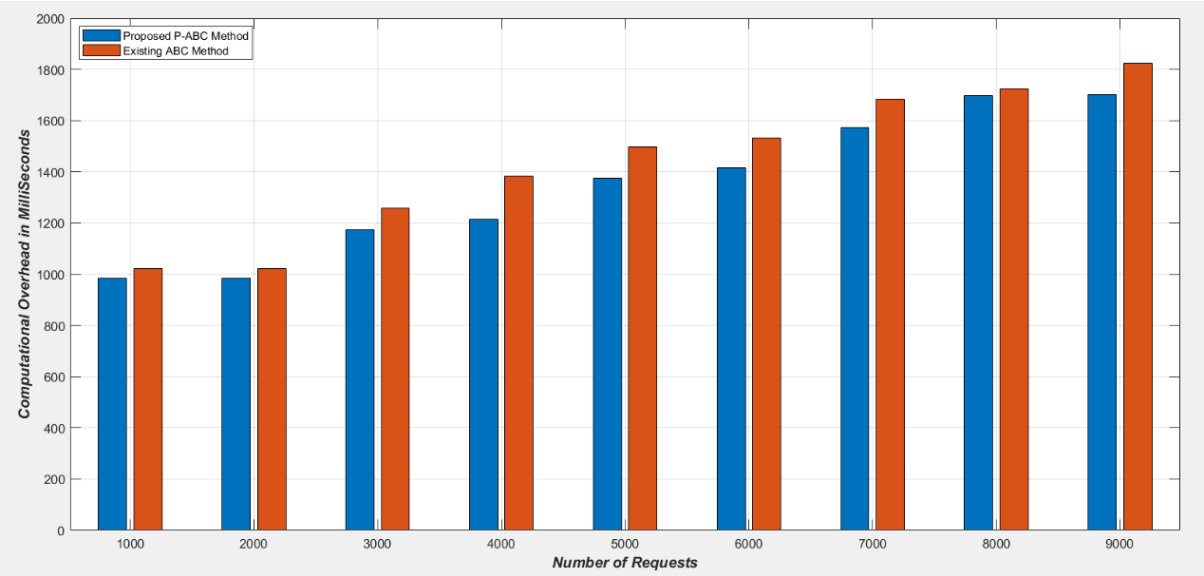

Figure 7: Graphical representation of the Computational Overhead in (ms) for the Proposed SA-BAC and existing A-BAC method for given number of requests

5. CONCLUSION

In this research work, Strategy-Attribute based Access Control scheme is presented for the cloud storage systems, which is secure and efficient. Moreover, the proposed system does not require any central authority and coordination among multiple authorities,thus eliminating the burden of heavy communication andthe delay of collaborative computation. The proposed system performed in less computation time for the encryption, key generation and decryption with varying number of authorities and varying number of attributes per authorities using DFO based key generation method. The proposed system is more suitable for practical access control since it supports dynamic operations. Moreover, it supports large universe of attributes.

REFERENCE

[1] Tankard, Colin. "Big data security." *Network security* 2012.7 (2012): 5-8.

[2] Demchenko, Yuri, et al. "Addressing big data challenges for scientific data infrastructure." *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, 2012.

[3] Islam, Md Rafiqul, and Md Ezazul Islam. "An approach to provide security to unstructured Big Data." *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*. IEEE, 2014.

[4] Lee, Myungcheol, et al. "Load adaptive and fault tolerant distributed stream processing system for explosive stream data." *2016 18th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2016.

[5] Demchenko, Yuri, et al. "Big security for big data: Addressing security challenges for the big data infrastructure." *Workshop on Secure Data Management*. Springer, Cham, 2013.

[6] Moustafa, Nour, et al. "Collaborative anomaly detection framework for handling big data of cloud computing." *2017 Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2017.

[7] Zhao, Jiaqi, et al. "A security framework in G-Hadoop for big data computing across distributed Cloud data centres." *Journal of Computer and System Sciences* 80.5 (2014): 994-1007.

[8] Xie, Xingxing, et al. "New ciphertext-strategy attribute-based access control with efficient revocation." *Information and Communication Technology-EurAsia Conference*. Springer, Berlin, Heidelberg, 2013.

[9] Ruj, Sushmita, and Amiya Nayak. "A decentralized security framework for data aggregation and access control in smart grids." *IEEE transactions on smart grid* 4.1 (2013): 196-205.

[10] Wang, Changji, and Jianfa Luo. "An efficient key-strategy attribute-based encryption scheme with constant ciphertext length." *Mathematical Problems in Engineering* 2013 (2013)..

[11] Hu, Vincent C., et al. "Guide to attribute-based access control (ABAC) definition and considerations (draft)." *NIST special publication* 800.162 (2013).

[12] Choi, Chang, Junho Choi, and Pankoo Kim. "Ontology-based access control model for security strategy reasoning in cloud computing." *The Journal of Supercomputing* 67.3 (2014): 711-722.

[13] Chen, Hongsong, Bharat Bhargava, and Fu Zhongchuan. "Multilabels-based scalable access control for big data applications." *IEEE Cloud Computing* 1.3 (2014): 65-71.

[14] Su, Jinshu, et al. "ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things." *Future Generation Computer Systems* 33 (2014): 11-18.

[15] Branch, Shahre Rey, and Shahre Rey. "Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing." *International Journal of Networked and Distributed Computing* 6.1 (2018): 35-42.

[16] Jothi, S., and A. Chandrasekar. "An efficient modified dragonfly optimization based mimo-ofdm for enhancing qos in wireless multimedia communication." *Wireless Personal Communications* 122.2 (2022): 1043-1065.

[17] Subhashini, M., & Gopinath, R., Mapreduce Methodology for Elliptical Curve Discrete Logarithmic Problems – Securing Telecom Networks, International Journal of Electrical Engineering and Technology, 11(9), 261-273 (2020).

[18] Upendran, V., & Gopinath, R., Feature Selection based on Multicriteria Decision Making for Intrusion Detection System, International Journal of Electrical Engineering and Technology, 11(5), 217-226 (2020).

[19] Upendran, V., & Gopinath, R., Optimization based Classification Technique for Intrusion Detection System, International Journal of Advanced Research in Engineering and Technology, 11(9), 1255-1262 (2020).

[20] Subhashini, M., & Gopinath, R., Employee Attrition Prediction in Industry using Machine Learning Techniques, International Journal of Advanced Research in Engineering and Technology, 11(12), 3329-3341 (2020).

[21] Rethinavalli, S., & Gopinath, R., Classification Approach based Sybil Node Detection in Mobile Ad Hoc Networks, International Journal of Advanced Research in Engineering and Technology, 11(12), 3348-3356 (2020).

[22] Rethinavalli, S., & Gopinath, R., Botnet Attack Detection in Internet of Things using Optimization Techniques, International Journal of Electrical Engineering and Technology, 11(10), 412-420 (2020).

[23] Priyadharshini, D., Poornappriya, T.S., & Gopinath, R., A fuzzy MCDM approach for measuring the business impact of employee selection, International Journal of Management (IJM), 11(7), 1769-1775 (2020).

[24] Poornappriya, T.S., Gopinath, R., Application of Machine Learning Techniques for Improving Learning Disabilities, International Journal of Electrical Engineering and Technology (IJEET), 11(10), 392-402 (2020).

[25] Karthikeyan, B., and Dr S. Hari Ganesh. "Encrypt-Security Improved Ad Hoc On Demand Distance Vector Routing Protocol (En-SIm AODV)." *ARPN Journal of Engineering and Applied Sciences (ISSN: 1819-6608)* 11.2 (2016): 1092-1096.

[26] Karthikeyan, B., N. Kanimozhi, and S. Hari Ganesh. "Analysis of reactive AODV routing protocol for MANET." *2014 World Congress on Computing and Communication Technologies*. IEEE, 2014.

[27] Karthikeyan, B., Dr S. Hari Ganesh, and Dr JGR Sathiaseelan. "High Level Security with Optimal Time Bound Ad-Hoc On-demand Distance Vector Routing Protocol(HiLeSec-OpTiB AODV)." *International Journal of Computer Science Engineering (E-ISSN: 2347-2693)* 4.4 (2016): 156-164.

[28] Karthikeyan, B., N. Kanimozhi, and Dr S. Hari Ganesh. "Performance and analysis of ad-hoc network routing protocols in manet." *NCAC* (2013): 65-71.

[29] B. Karthikeyan, Detection of Selective Forwarding Attacks in Wireless Sensor Networks, International Journal of Electrical Engineering and Technology (IJEET), 11(9), 2020, pp. 376-392.

[30] B. Karthikeyan, Cluster based Malicious Node Detection for Mobile Ad Hoc Networks, International Journal of Advanced Research in Engineering and Technology (IJARET), 11(12), 2020, pp. 3501-3510.