Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT FILE ACCESS CONTROL TECHNIQUE FOR SHARED CLOUD DATA SECURITY THROUGH KEY-SIGNATURES SEARCH SCHEME

[1] TARASVI LAKUM, [1] PROF.B. TIRAPATHI REDDY*

[1]Research Scholar, [1]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P, India,

E-mail: tarasiru1@gmail.com, tirapathireddyb@kluniversity.in

ABSTRACT

Through cloud services, the cloud based users and organizations are storing and sharing the data in the advanced cloud computing environment. However, the recent cloud data breaches have raised privacy and secure concerns in the cloud managed data, due to vulnerability in untrusted cloud access control system . However, designing an efficient trusted access control system in cloud through enabling a cryptographically file access control technique is still challenging. In this research contribution, a cloud data security system for an efficient file access control technique that provide practical trusted security for shared cloud data is proposed. Proposed file access control technique revokes Key-Signatures Search Scheme which provides confidential hosted-file data access, by delegating role-based public key-revoking in cloud hosted environment to update encrypted data. In Key-Signatures Search Scheme, the cloud data security is made by encrypting the file by a hosted-file key management, which records hosted-file and its revocation keys simultaneously for key-access enforcement and file-access revocation. In each hosted-file revocations, cloud administrator checks for any in-secure data breach, if found, for that particular hosted-file a new revocation key is updated and request for a new encrypted hosted-file with updated file-access permissions. In Key-Signatures Search Scheme, three stages of key-signatures are monitored and updated based on how hosted-file access if enforced, after enforcing how file is granted for access and finally how revocation of grant hosted-file access is made. By monitoring through these three ways, Key-Signatures Search Scheme, enforces a dynamic hosted-file access control technique at cloud data user side which improves the file access control providing efficiency, which does not require re-submission of revocation keys and repeated file access granting checks, and providing security for a large hosted-files at the cloud data owner side by instant hosted-file key-revoking for access control. Cloud data security framework and system implementation is made in the proposed work to demonstrate the Cloud Data Security and Efficiency of the proposed technique. Proposed Key-Signatures Search Scheme uses formalized shared cloud date framework and cloud owner with user system implementation to establish the cloud data hosted-file security and show efficiency of proposed file access control technique through hosted cloud system design.

Keywords: *Cloud computing, Role-Based Access Control, Key Encryption Schemes, Cloud Data Security, Revocation*

1. INTRODUCTION

With the significant advancements in distributed cloud computing [1] [2] [3], users and organizations are finding it progressively engaging to store and share information through cloud administrations. Cloud administration providers give abundant cloud based administrations, going from limited scope individual administrations to huge scope industrial administrations. However,

recent information breaches , like, releases of private photographs, have raised concerns with respect to the protection of cloud-managed information.

All things considered [4] [5] [6], a cloud specialist provider is generally not secure because of plan drawbacks of software and framework vulnerability . As such, a basic issue is the manner by which to enforce information access control on the possibly untrusted cloud [8] [9].

To overcome these issues [11] [12] [13], this paper present file Access control technique through Key-Signatures Search Scheme (Access-KSS), a cryptographically implemented unique access control system on untrusted cloud[15] [16]. Access-KSS delegates the cloud to refresh encrypted files in permission revocation. In Access-KSS, a file or document is encrypted by a symmetric key list which records a file key and a succession of revocation keys. In a revocation, the executive transfers another revocation key to the cloud, which encrypts the file or document with an another layer of encryption and updates the encrypted key list likewise. Same as previous works[19][20][21], having a honest-but-curious cloud, i.e., the cloud is honest to perform the re-encryption of files or documents and appropriately update previous encrypted files or documents) however is interested to passively assembling sensitive data or information. Although the essential thought of layered encryption is straightforward, it involves tremendous specialized challenges. For example, the size of key list and encryption layers would increment as the quantity of revocation tasks, which causes extra decryption overhead for clients to get to documents. To overcome such an issue, Access-KSS is proposed.

The rest of the paper is coordinated as follows. In Section II, proposed framework model related works is presents, Section III recognizes a few basic issues for cryptographically authorized access control, from which previous standards of Access-KSS and proposed Access-KSS are discussed. Section IV depicts the implementation, simulation and analysis of Access-KSS. In Section V, talk about conclusions of proposed Access-KSS is made. In Section VI, future direction of Key Encryption Schemes through proposed Access-KSS is suggested.

2. RELATED WORK

In response of these security issues, various research contribution works have been proposed to help access control on untrusted cloud administrations by utilizing cryptographic primitives. Progressed cryptographic primitives are applied for authorizing many access control standards. For instance[22][23], attribute-based encryption (ABE) is a cryptographic partner of attribute-based access control (ABAC) model. Be that as it may, previous works primarily think about static situations in which access control strategies rarely change. The previous works cause high overhead when access control approaches should be changed in practice. At a first look, the revocation of a client's permission should be possible by revoking his access to the keys with which the files or documents are encrypted. This arrangement, however, isn't secure as the client can keep a local copy of the keys prior to the revocation. To prevent such an issue, files or documents have to be re-encrypted with new keys. This requires the files or documents owner to download the document, re-encrypt the files or documents, and upload it back for the cloud to update the previous encrypted file, incurring prohibitive communication overhead at the files or documents owner side.

Right now, contributed research work literature survey hypothesis is outlined below: through research examining the issue of dynamic data or information access control.

Garrison et al. [24] [25] proposed two revocation schemes. The primary plan requires a manager to re-encrypt files or documents with new keys as examined previously. This scheme causes a significant communication overhead. Instead, the second scheme delegates clients to re-encrypt the files or documents at the point when they need to modify the files or documents, easing the administrator from re-encrypting files or documents by itself. This scheme, has come with a security penalty as the revocation activity is delayed to the next user's change to the files or documents. As a

result, a newly revoked client can in any case get to the record access before the next writing activity.

Wang et al. [7] proposed another revocation scheme, in which the symmetric homomorphic encryption scheme is used to encrypt the file. Such a design enables the cloud to straightforwardly re-encrypt files or documents without decryption. However, this scheme brings expensive files or documents read/write overhead as the encryption/decryption activity includes equivalent overhead with the public key encryption plans.

Gudes et al. [5] investigate cryptography to enforce progressive system access control without considering about dynamic arrangement situations.

Akl et al.[22] propose a key task plan to simplify key administration in hierarchical access control strategy. Likewise, this work doesn't consider strategy update issues. Afterward, Atallah et al. propose a strategy that permits strategy updates, however on account of revocation, all descendants of the affected node in the access hierarchy of importance should be updated, which includes high calculation and communication overhead.

Ibraimi et al.[14] cryptographically support role based access control structure using mediated public encryption. However, their revocation activity relies on extra confided infrastructure and an functioning entity to re-encrypt all influenced files or documents under the new arrangement. Similarly,

Nali et al.[4] enforce role based access control structure utilizing public-key cryptography, however requires a progression of active security mediators.

Ferrara et al.[13] characterize a secure model to formally demonstrate the security of a cryptographically enforced ABAC framework. They further show that an ABE-based development is secure under such model. Notwithstanding, their work focus around hypothetical analysis.

Pirretti et al.[18] propose an enhanced ABE-based access control for distributed file systems and social organizations, however their development doesn't expressly address the dynamic revocation. Siever is a quality based access control framework that allows users to selectively expose their private data or information to third web administrations. Sieve utilizes ABE to implement attribute based access policies and homomorphic symmetric encryption to encrypt data or information. With homomorphic symmetric encryption, an information owner can delegate revocation tasks to the cloud assured that the privacy of the data or information is preserved. This work however brings prohibitive computation overhead since it adopts the homomorphic symmetric encryption to encrypt documents.

M. Maffei et al.[17] permits an information owner to enforce an access network for a list of approved users and provides strong data protection in two folds. First, user access patterns are stored from the cloud by utilizing

J. R. Lorch et al.[10] strategies. Second, arrangement attributes are stored from the cloud by utilizing attribute-hiding predicate encryption. The cryptographic algorithms, however, cause extra execution overhead in information communication, encryption and decryption.

Additionally, [17] doesn't support dynamic approach update. Over encryption is a crypto-graphical method to authorize an access matrix on outsourced information. Over-encryption utilizes twofold encryption to implement the entire access matrix. Subsequently, the administrator needs to depend on the cloud to run complex algorithms over the network to update access strategy, expecting an high level of trust on the cloud.

Altogether, objectives of proposed work through Access-KSS are :

a) to achieve proficient revocation,

b) effective file or document access and

c) quick revocation simultaneously.

For revocation proficiency, Access-KSS brings about lightweight communication overhead at the administrator side as it doesn't need to download and re-upload file or document information.

For quick revocation, the authorizations of users are immediately revoked as the files or documents are re-encrypted[26][27][28][29]. For file or document access efficiency, the files or records are as yet encrypted by symmetric keys.

Real examinations of Access-KSS suggest three significant orders more efficient in correspondence in access revocation compared with the first scheme, and is almost two significant orders more effective in computation in file or document access compared with the scheme. Finally, Access-KSS can quickly revoke access authorizations compared with the other schemes[30][31][32][33][34][35].

3. PROPOSED WORK

Access-KSS aim to provide confidentiality and access control for the cloud-hosted file data.

A. Previous designs

In response, [26][36][37proposed two revocation schemes. The first scheme requires the administrator to re-encrypt file data by itself in a revocation. This scheme completes the revocation immediately with a potentially high communication overhead. Differently, the second scheme relies on next users writing to the F tuples to re-encrypt the F tuples. This scheme [38][39], however, comes with security penalty as it delays the revocation to the next writing, creating a vulnerability window in which revoked users can continuously access the F tuples which they have accessed previously and cached the file keys.

To alleviate the overhead of file data re-encryption, in [7] another revocation schemes [40][41], in which the symmetric homomorphic encryption scheme is used to encrypt the file data. Instead by re-encrypting the file data by itself[42], the scheme enables the administrator to delegate the cloud to update F tuples from old file keys to new file keys without decryption. The problems in [43][44], however, is that the cost of homomorphic symmetric encryption is comparable with public key encryption schemes [45][46][47], incurring prohibitive computation overhead during file reading/writing.

B. Proposed Design

To overcome these limitations, Access-KSS develops new techniques using lightweight symmetric encryption scheme with three key methods as follows, which achieves objectives when compared to other research contributions :

First, Access-KSS proposes Key-Signatures encryption technique to assign the cloud to refresh strategy information. For a file or document, the administrator attaches another revocation key toward the finish of its key list and requests the cloud to refresh this key list in the policy information.

The size of the key list anyway increments with the revocation activities, and a client needs to download and decrypt an enormous key list in each file access.

To overcome this issue, Access-KSS utilizing the key rotation method to compactly encrypt the key list in the policy information. Therefore, the size of the key list keeps the same regardless of revocation activities.

Second, Access-KSS proposes flexible role-based encryption procedure to designate the cloud to update file or document information.

For a file or document, the administrator request the cloud to encrypt the file or record with another layer of encryption. Additionally, the size of the encryption layers increments with the revocation tasks, and a client needs to decrypt on multiple times in each file or document access.

To overcome this problem, Access-KSS enables the administrator to characterize a tolerable bound for the file or document. When the size of encryption layers arrives at the bound, it tends to be made not to increment anymore by assigning encryption tasks to the cloud. Subsequently, the administrator can flexibly change a tolerable bound for each file or document to accomplish a balance among effectiveness and security.

During the activity of a file or document, its encryption layers constantly increment until a pre-defined bound is reached.

Access-KSS proposes role-based access revocation and hosted-file key management encryption methodology to occasionally refresh the symmetric key list of the file or document and eliminate the bounded encryption layers over it through wiring activities.

In explicit, the following user to write to the file or document encrypts the writing content by

another symmetric key list containing another record key, and updates the key list in the policy information.

With this technique, Access-KSS occasionally eliminates the limited encryption layers of file or documents while amortizing the weight to a huge number of writing clients.

Proposed design uses the following notation:

- u is a user,
- r is a role,
- p is a permission,
- f_n is a file name of a file f,
- c is a ciphertext (either symmetric or public encryption), and
- v is a version number.

The proposed scheme shown in figure 1, follows the following Implementation Method:

o Design and analyze Access-KSS based on the role-based access control (RBAC) model named, which is widely used in practical applications.

o RBAC model describes permission management through the use of abstraction: roles describe the access permissions associated with a particular (class of) job function, users are assigned to the set of roles entailed by their job responsibilities, and a user is granted access to an object if they are assigned to a role that is permitted to access that object.

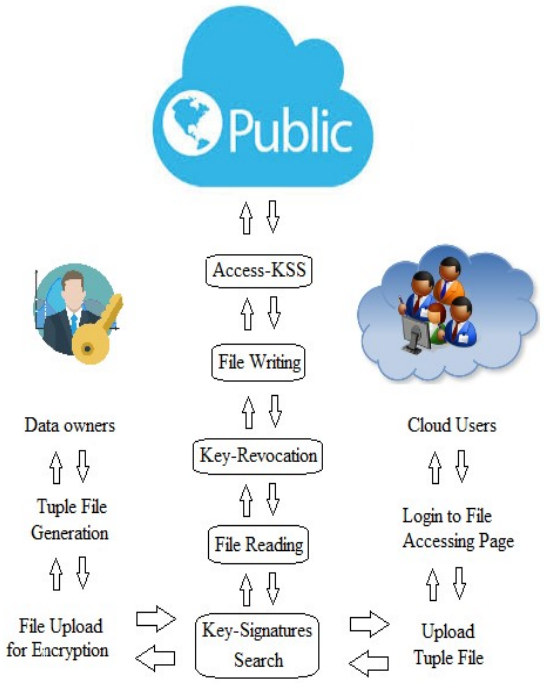

Figure 1: Proposed Access-KSS Scheme Flow Diagram

C. File management

Access-KSS use three types of files to store metadata for file management. Access-KSS introduce them as follows.

1. USERS: A record (u, ek_u) contains a user identity u and the encryption key ek_u of u.

2. ROLES: A record (r, ek_v) contains a role identity r and the encryption key ek_{vr} of r.

3. FILES: A record (fn) contains the file name fn of a file.

D. Key-Signatures Search management

In proposed system, the administrator, roles and users are associated with cryptographic keys. Access-KSS introduce them as follows.

1. Administrator keys: The administrator plays a role of super user in the system. It has an encryption key pair (ek_{SU} , dk_{SU}) of a public key encryption scheme and a signature key pair (sk_{SU} , vk_{SU}) of a digital signature scheme.

The encryption key pair is also used by a user to create a special FK tuple when adding a new file into the system.

2. User keys: A user read key of u is an encryption key pair (ek_u, dk_u) of a public key encryption scheme. This key is used to encrypt/decrypt role key (RK) tuples for u.

3. Role keys: A role key of r is an encryption key pair (ek_r, dk_r) of a public key encryption scheme. This key pair is used to encrypt/decrypt file key (FK) tuples for r.

4. File keys: A file key of f_n is a symmetric key list (k^0, k^1,... k^t) of a symmetric key encryption scheme and a rotation key pair (rsk_{fn} , rpk_{fn}) of a key rotation scheme. (k^0, k^1,... k^t) is used by users to encrypt the F tuple of f_n, and (rsk_{fn} , rpk_{fn}) is used by the administrator to compactly store (k^0, k^1,... k^t) in the FK tuple of f_n.

E. Access-KSS Scheme Methodology

In response to data breaches security issues, this paper presents Access-KSS scheme. The proposed scheme methodology is outlined below:

a) In Access-KSS scheme, a revocation file is accesses through encrypted by a defined symmetric key list through which it records a encrypted file key and a revocation layer sequences of revocation keys.

b) In layers of key revocations, the authorized administrator provides a new revocation key to the cloud location, which encrypts the relocated file with a new layer of file encryption and updates the layers of encrypted key list with the file revocations.

c) An append-aware file encryption strategy is presented to keep the size of the key constant with number of repetition times of revocation operations.

d) A time bound update file encryption data is presented to adjust the encryption key operation request the cloud can encrypt the file with layers of encryption.

e) A repeated encryption is presented to refresh the symmetric key list of the revocation file and remove the time bounded encryption layers over it through overwrite operations.

f) Altogether, Access-KSS scheme achieves efficient key revocation, efficient file access mechanism and immediate key and file revocation simultaneously. For key and file revocation efficiency, Access-KSS scheme uses lightweight symmetric encryption scheme at the authorized administrator side as it does not need to re-encrypt and re-upload file data. For time bound revocations, the key permissions of users are revoked in access time as the files are re-encrypted. To improve the file access efficiency, the files are still encrypted by lighten symmetric keys.

4. SYSTEM EVALUATION

To compare proposed work, considered two revocation schemes as deferred re-encryption (DRe) and the revocation scheme as homomorphic reencryption (HRe).

In proposed implementation, Access-KSS deploy the simulation on a PC which is equipped with a 4-core Intel MP 2.0 GHz processor and 16 GB RAM. Access-KSS compare the performance of the four systems in access revocation and file reading/writing.

A. Key-revoking scheme end-to-end experiments

Access-KSS only uses lightweight symmetric encryptions to encrypt file data and for access revocation, Access-KSS uses Key-Signatures encryption strategy to delegate the cloud provider to update RK/FK tuples.

Access-KSS also uses adjustable role-based encryption strategy to delegate the cloud to update F tuples. As the administrator only sends symmetric keys for the cloud provider to encrypt files, it costs far less overhead to update F tuples than previous works.

For file read/write, Access-KSS constrains encryption layers over files to improve the efficiency of file read/write operations. In specific, Access-KSS uses the adjustable role-based encryption strategy to constrain the encryption layers in revocation operations and role-based access revocation encryption strategy to remove them periodically. The combination of the two strategies ensure that the encryption layer of each file is under an upper bound all the time. More interestingly, the administrator can adjust this upper bound to suit specific application requirements by combining the two strategies in a flexible way.

Table 1 compares the execution time of HRe and Access-KSS in reading operations. Simulation is seen that HRe takes 80 computation times higher than Access-KSS (k = 5) and 6.7 total times higher than Access-KSS (k = 5). Also compares the execution time of writing operations. Simulation is seen that DRe and Access-KSS takes same time.

The reason is that all the schemes use symmetric key encryption scheme to encrypt files and the role-based access revocation encryption adopted in Access-KSS incurs no additional cost at user side.

Table 1: Performance In File Reading And Writing

	File Reading Time (Sec)		File Writing Time (Sec)	
File Size	HRe	Proposed Access-KSS	DRe	Proposed Access-KSS
10 M	17.7	15.8	17	16.8
100 M	176.8	164.1	175.1	168

Identifications from the proposed work:
The following outline covers the identifications of proposed research work :

a) Through an identified update key feature in the proposed work, the revocation key does not require to increase its size and causing the size of the key remains constant.
b) The updated key in-turn increases the number of encryption layers, making the user to decrypt the file key multiple times, by providing a bounding file size, increase of encryption layer is limited.
c) A malicious cloud provider may modify tuples and generate fake tuples, to avoid this, a signature based data integrity scheme is identified and proposed in this work, where the administrator assigns data integrity scheme to authenticated cloud provider to modify the tuples, making a direct update through validating the attached signatures, making a secure modifications.
d) During cloud provider access control, role based access controls are defined, which stores the traces of number of instances of each operation, causing the user revocation to trigger multiple files in less than a second, causing the file access time to increase. Through the defined role-specific continuous cryptographic time Markov chains, the generated traces are time-bounded and time-limited, if the defined time reaches to limit, the administrator revokes user to re-encrypt in the average time and reduces the number of encryption layers of a file, which in-turn causes file access time to reduce.

Limitations:
The common limitation of the proposed research work is:
a) It uses only lightweight symmetric encryptions to encrypt the file data.
b) It uses a key specified files list, which narrows the access-key features only to cloud providers.
The specified limitation of the proposed research work is:
a) The dynamic revocation construction is made based on the distributed file systems only, but limited with the social networks.

5. CONCLUSIONS

Proposed Work introduced Access-KSS, a framework that gives useful cryptographic

implementation of dynamic access control in the potentially untrusted cloud provider. Access-KSS meets its objectives utilizing three strategies.

Specifically, Access-KSS propose to designate the cloud to refresh the policy data in a privacy-preserving way utilizing an Key-Signatures encryption methodology.

Furthermore, Access-KSS propose a role-based access revocation encryption technique to avoid the file or document understanding overhead.

The hypothetical analysis and the performance evaluation show that Access-KSS accomplishes significant higher efficiency in access revocations.

6. FUTURE SCOPE

Access-KSS can enable efficient data traverse paths for both Cloud user and server, such an interference causes extra energy consumption, as well. Access-KSS, based on our real-world deployment on Cloud Machines, can significantly improve computation and input/output performance for hybrid clouds. Moreover, this design also improves the existing virtualization overhead and naturally optimizes the overall energy efficiency.

REFERENCES

[1]. C. Jin and M. van Dijk, "Secure and efficient initialization and authentication protocols for shield," IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 1, pp. 156{173, 2019.

[2]. D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, SIAM Journal on Computing, vol. 32, no. 3, 2003.

[3]. D. Boneh, K. Lewi, H. Montgomery, and A. Raghu Raghunathan, Key homomorphic PRFs and their applications, in CRYPTO, 2013.

[4]. D. Nali, C. M. Adams, and A. Miri, Using mediated identity-based cryptography to support role-based access control, in ISC 2004, 2004.

[5]. E. Gudes, The Design of a Cryptography Based Secure File System,IEEE Transactions on Software Engineering, vol. 6, no. 5, 1980.

[6]. E. Shen, E. Shi, and B. Waters, Predicate privacy in encryption systems, in TCC, 2009.

[7]. F. Wang, J. Mickens, N. Zeldovich, and V. Vaikuntanathan, Sieve: Cryptographically Enforced Access Control for User Data in Untrusted Clouds, in NSDI, 2016.

[8]. G. Ateniese, D. H. Chou, B. Medeiros, and G. Tsudik, Sanitizable Signatures, in proceedings of ESORICS, 2005.

[9]. J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-policy attribute based encryption, in IEEE S&P, 2007.

[10]. J. R. Lorch, B. Parno, J. W. Mickens, M. Raykova, and J. Schiffman, Shroud: ensuring private access to large-scale data in the data center, in FAST, 2013.

[11]. J. Wang, X. Chen, J. Li, J. Zhao, and J. Shen, Towards achieving flexible and verifiable search for outsourced database in cloud computing, Future Generation Computer Systems, vol. 67, 2017.

[12]. J. Wang, X. Chen, X. Huang, I. You, and Y. Xiang, Verifiable Auditing for Outsourced Database in Cloud Computing, IEEE Transactions on Computers, vol. 64, no. 11, 2015.

[13]. A.L. Ferrara, G. Fuchsbauer, and B. Warinschi, Cryptographically enforced RBAC, in CSF, 2013.

[14]. L. Ibraimi, Cryptographically enforced distributed data access control, Ph.D. dissertation, University of Twente, 2011.

[15]. M. Barhamgi et al. Privacy in data service composition. IEEE Transactions on Services Computing, 2019.

[16]. M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, Dynamic and efficient key management for access hierarchies, ACM TISSEC, vol. 12, no. 3, 2009.

[17]. M. Maffei, G. Malavolta, M. Reinert, and D. Schroder, Privacy and access control for outsourced personal records, in IEEE S&P,

2015.
[18]. M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, Secure attributebased systems, in ACM CCS, 2006.
[19]. R. S. Sandhu, Rationale for the RBAC96 family of access control models, in proceedings of ACM Workshop on RBAC, 1995.
[20]. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, Over-encryption: Management of access control evolution on outsourced data, in VLDB, 2007.
[21]. S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, Encryption policies for regulating access to outsourced data, TODS, vol. 35, no. 2, 2010.
[22]. S. G. Akl and P. D. Taylor, Cryptographic solution to a problem of access control in a hierarchy, IEEE TOCS, vol. 1, no. 3, 1983.
[23]. T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, Secure and Efficient Cloud Data Deduplication With Randomized Tag, IEEE Trasactions on Information Forensics and Security, vol. 12, no. 3, 2017.
[24]. T. L. Hinrichs, D. Martinoia, W. C. Garrison III, A. J. Lee, A. Panebianco, and L. Zuck, Application-sensitive access control evaluation using parameterized expressiveness, in CSF, 2013.
[25]. W. C. Garrison III, A. J. Lee, and T. L. Hinrichs, An actor-based, application-aware access control evaluation framework, in SACMAT, 2014.
[26]. W. C. Garrison III, A. Shull, S. Myers, and, A. J. Lee, On the Practicality of Cryptographically Enforcing Dynamic Access Control Policies in the Cloud, in IEEE S&P, 2016.
[27]. X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, New Publicly Verifiable Databases with Efficient Updates, IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 5, 2015.
[28]. Z. Yang, J. Lai, Y. Sun, and J. Zhou, \A novel authenticated key agreement protocol with dynamic credential for wsns," ACM Trans. Sen. Netw., vol. 15, no. 2, March 2019. [Online]. Available: https://doi.org/10.1145/3303704.
[29]. X. Wang, Y. Qi, and Z. Wang, Design and Implementation of SecPod: A Framework for Virtualization-based Security Systems, IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 1, 2019.
[30]. Zhiru Li, Wei Xu, Huibin Shi, Yuanyuan Zhang, and Yan Yan1, "Security and Privacy Risk Assessment of Energy Big Data in Cloud Environment", Hindawi, Computational Intelligence and Neuroscience, Volume 2021, October 2021, pp 1-11.
[31]. Yu Zhang, Wei He, and Yin Li, "Efficient Boolean Keywords Search over Encrypted Cloud Data in Public Key Setting", Hindawi, Mobile Information Systems, Volume 2020, August 2020, pp 1-15.
[32]. Jianhong Zhang and Menglong Wu, "Efficient Cloud-Based Private Set Intersection Protocol with Hidden Access Attribute and Integrity Verification", Hindawi, Security and Communication Networks, Volume 2021, September 2021, pp 1-13.
[33]. Sun Pan Jun, "A Trust-Game-Based Access Control Model for Cloud Service", Hindawi, Mobile Information Systems, Volume 2020, July 2020, pp 1-14.
[34]. S. Mary Virgil Nithya and V. Rhymend Uthariaraj, "Identity-Based Public Auditing Scheme for Cloud Storage with Strong Key-Exposure Resilience", Hindawi, Security and Communication Networks, Volume 2020, January 2020, pp 1-13.
[35]. Xiaofeng Lu, Songbing Fu, Cheng Jiang, and Pietro Lio, "A Fine-Grained IoT Data Access Control Scheme Combining Attribute-Based Encryption and Blockchain", Hindawi, Security and Communication Networks, Volume 2021, September 2021, pp 1-13.
[36]. Mahsa Beigrezaei, Abolfazel Toroghi Haghighat, and Seyedeh Leili Mirtaheri, "Improve Performance by a Fuzzy-Based Dynamic Replication Algorithm in Grid, Cloud, and Fog", Hindawi, Mathematical Problems in Engineering, Volume 2021,

June 2021, pp 1-14.

[37]. Lina Ni, Xiaoting Sun, Xincheng Li, and Jinquan Zhang, "GCWOAS2: Multiobjective Task Scheduling Strategy Based on Gaussian Cloud-Whale Optimization in Cloud Computing", Hindawi, Computational Intelligence and Neuroscience Volume 2021, June 2021, pp 1-17.

[38]. Xianwei Zhu, ChaoWen Chang, Qin Xi, and ZhiBin Zuo, "Attribute-Guard: Attribute-Based Flow Access Control Framework in Software-Defined Networking", Hindawi, Security and Communication Networks, Volume 2020, January 2020, pp 1-18.

[39]. Ping Liu, Syed Hamad Shirazi, Wei Liu, and Yong Xie, "pKAS: A Secure Password-Based Key Agreement Scheme for the Edge Cloud", Hindawi, Security and Communication Networks, Volume 2021, October 2021, pp 1-10.

[40]. Yu Cui, Shunfu Jin, Wuyi Yue, and Yutaka Takahashi4, "Performance Optimization of Cloud Data Centers with a Dynamic Energy-Efficient Resource Management Scheme", Hindawi, Complexity, Volume 2021, February 2021, pp 1-18.

[41]. Sheng Hu, Shuanjun Song, and Wenhui Liu1, "A Framework of Cloud Model Similarity-Based Quality Control Method in Data-Driven Production Process", Hindawi, Mathematical Problems in Engineering, Volume 2020, May 2020, pp 1-10.

[42]. Hua Dai, Xuelong Dai, Xiao Li, Xun Yi, Fu Xiao, and Geng Yang, "A Multibranch Search Tree-Based Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data", Hindawi, Security and Communication Networks, Volume 2020, January 2020, pp 1-15

[43]. Javier Junquera-S´anchez, Carlos Cilleruelo, Luis De-Marcos, and Jos´e-Javier Martinez-Herr´aiz, "Access Control beyond Authentication", Hindawi, Security and Communication Networks, Volume 2021, October 2021, pp 1-11.

[44]. Qiang Lin, "Dynamic Resource Allocation Strategy in Mobile Edge Cloud Computing Environment", Hindawi, Mobile Information Systems, Volume 2021, August 2021, pp 1-10.

[45]. Quanrun Li, Chingfang Hsu, Debiao He, Kim-Kwang Raymond Choo, and Peng Gong, "An Identity-Based Blind Signature Scheme Using Lattice with Provable Security", Hindawi, Mathematical Problems in Engineering, Volume 2020, May 2020, pp 1-12.

[46]. Jingjing Guo and Jiacong Sun, "Order-Revealing Encryption Scheme with Comparison Token for Cloud Computing", Hindawi, Security and Communication Networks, Volume 2020, December 2020, pp 1-13.

[47]. Tsu-Yang Wu, Tao Wang, Yu-Qi Lee, Weimin Zheng, Saru Kumari, and Sachin Kumar, "Improved Authenticated Key Agreement Scheme for Fog-Driven IoT Healthcare System", Hindawi, Security and Communication Networks, Volume 2021, January 2021, pp 1-16.