Universe International Journal of Interdisciplinary Research
(Peer Reviewed Refereed Journal)

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

DESIGN AN ALGORITHM USING MULTI KEY ENCRYPTION SCHEME FOR SECURING DATA IN CLOUD COMPUTING ENVIRONMENT

Author's Name: Miss Roshni Nigwal

Affiliation: *Student, Computer Science and Engineering Colleges, [1]Malwa Institute of Technology, Indore, India*

E-Mail: roshnigwal100@gmail.com

DOI No. – 08.2020-25662434

Abstract

We need to use a new approach to improve cloud storage protection because of problems with transparency and security. The introduction of cloud storage will solve many business-related security issues and risks. By encrypting, decrypting, compressing, and sharing data, we will boost the security of cloud storage. Cloud storage has been linked to security issues in the past. We provide high-quality services to our customers. To ensure high data protection in cloud-based environments, steganography, encryption, and decryption are used. (Multi Key Encryption Scheme). Reduces the time it takes to retrieve data and store it in the cloud. It discusses the various benefits of cloud computing, as well as the main security issues and various cryptographic encryption algorithms. This paper will also provide an algorithm for using a multi-key encryption scheme in cloud computing.

Keywords*: Cloud computing; Cloud security; Public cloud; Data protection in public cloud; Data Protection;*

INTRODUCTION

The term "cloud computing" refers to the distribution of computer resources over the Internet. Cloud computing with high performance allows customers to access a variety of customer services. Customers benefit from cloud storage in a number of ways, including the opportunity to pay for on-demand services and services. In today's world of IT, cloud computing is a common trend. The data must be sent in vast amounts to massive cloud storage. In an insecure cloud storage network, sensitive data encryption is needed for safe and secure data transfer. Our data may be encrypted during transmission to protect it from unauthorized access. As the private cloud is secured by a single organization, the other data is vulnerable to attacks and privacy issues. Finally, a cloud provider stores data in a massive data center [1]. Data Information Content is managed by a third-party individual who is legally responsible for providing ongoing information and multimedia services. There is now a feeling of safety.

If the information is intentionally shared with third parties, a query and a clear example are given. After putting a malicious VM next to your target the channel intruder removes the information contents. This paper provides a data protection algorithm for cloud storage using a multi-key encryption system in order to protect data in the public cloud. This encryption algorithm can be implemented with three distinct randomly generated keys and bit conversion three coding keys are used in the proposed Multi-key Encryption Scheme (MKES) to decrypt the data. The information is encrypted and decrypted using the same keys. Symmetrical encryption is what this is. The remainder of the paper is formatted as follows: In Section II, the literary review is discussed. Section III discusses cloud computing issues, challenges, and potential methodology algorithms. The end of Section IV is known.

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

LITERATURE REVIEW

RELATED WORK

One of the major flaws of current safety models was the inability to maintain data confidentiality and integrity as access rules changed in practice, if a user's access rights are revoked, these solutions fail to offer reliable or safe replies, making it impossible for the revoked user to access encrypted data. As a result, researchers are urged to encrypt algorithms in order to fix the flaw. Raipurkar, K. V., et al[1] It becomes aware in the early stages of user identification and blocks malicious users. Cloud users are also protected by the LDAP authentication method and IP detection. Have a compression algorithm to reduce the usage of cloud storage. More consistency is provided by the two-way encryption algorithm. Nagesh, S. H., et al[2] presented various cloud deployment models and services offered by cloud. Most businesses are hesitant to use cloud services due to concerns about credibility, affordability, and privacy. Rani, K., et al[3] In addition, a solution was launched to improve cloud-based security by means of kludging, encryption decryption, compression, and other solutions that remedy defects with standard data-protection algorithms. and cloud slicing techniques. George, L. P., et al[4] Presents a public cloud application security algorithm as a service. The encryption algorithm can be enforced with three separate randomly generated keys and bit conversion. Three code keys have been included in the GLEnc proposal to decode the results. The information is encrypted and decrypted using the same keys. This is Symmetrical encryption Moghaddam, F. F., et al [5] four key parameters are used to re-encrypt protected rings: authentication in time, unauthorized authentication, user revocation, and data owner order. Gu, X., et al[6] The topic of collusion attacks is investigated by examining re encryption and traditional encryption methods, as well as how to select a collusion attack encryption algorithm in the encryption model. Reddy, Y. et al [7] a game model uses four parameters to recognize social media messages: a person, a time, a post, and a network node. The parameters are used to decide how the content of the communication should be interpreted. This messages are sent by one or more users that create keyword frequency events. Xiong, J., et al[8] KP-TSABE is able to deal with some big safety issues by supporting a fixed permit period and providing sophisticated access control during period control. Xia, Z. et al [9] Create a searchable encryption framework that can handle both accurate and flexible dynamic document collection search rankings using multiple keywords.

PROPOSED METHODOLOGY

Since it improves efficiency and trustworthiness, online data sharing is now one of the most important requirements for any enterprise. Cloud computing has expanded beyond shared boundaries, allowing multiple users to connect to shared data and collaborate. However, online data security is critical to the cloud's success, necessitating the use of dependable, secure encryption systems. Data owners can opt to encrypt their data/files online so that specific users may access decryption rights while also having the ability to revoke access. The server is a popular method for implementing access control protocols while ensuring privacy. [2]. this method is vulnerable to increased privilege attacks in shared data environments like clouds, where multiple users can share a single server. A third-party auditor trusts [3] current online data exchange technology, or it uses a user's key to encrypt their data while retaining privacy [4]. In each case, a consumer wants a stable and cost-effective encryption system with standardized security guarantees, high scalability, and user-friendliness. The most difficult aspect of such an encryption scheme is essentially sharing encrypted data. A cloud sharing

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

system can only function if data owners can easily grant multiple users access to their data through cloud servers. Figure 1 depicts an online cloud data sharing system in action. Assume you're a data owner. Alice uses a data sharing service like One Drive[5] to keep all kinds of records (here class may refer to any data structure such as a file, folder or any collection of these). It wants to add an extra layer of protection

By encrypting the files. You now want to give a community of SS data user's access to a specific S subset of those documents. Any customer who can access those types of data in this link must have decryption rights. The task at hand is to create an online part-data sharing scheme that will enable Alice to carry out its tasks effectively and safely. A naive (and inefficient) solution would be to include a decryption key for each type of message, which would then be exchanged for safe channels with the designated recipients. This strategy is unworkable for two reasons. First, as the number of data types grows, so does the number of secret keys. Second, Alice will encrypt the corresponding database, and the new set of keys will be distributed to all current users.

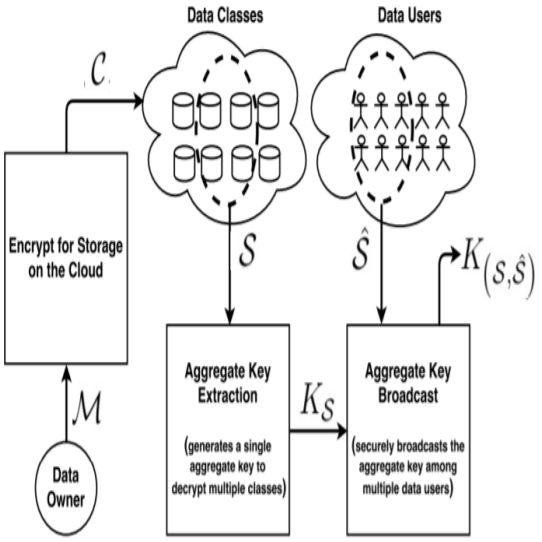

Figure 1: Proposed data Encryption technique

For cryptography, several keys are suggested (Row Based Encryption and Colom based Encryption) AES and RSA are two public coding schemes. An MKES [5][6] A trusted private key generator maintains a master-secret key and generates a secret key for each user depending on the user's identity Each user will receive an encrypted message containing his ID and some public parameters, which will be decrypted by a trusted party using a secret password. Key aggregation is possible for these two schemes, but each key must be separated into its own identity. MKES allows the decryption of several cipher texts with a single, compact key, but it must not be encrypted under closed identities. In practice, MKES does not deal with arbitrary identities. Attribute of Basic Encryption With attribute-based encryption, each user can be classified based on a variety of characteristics (ABE). An encrypted file stored in the cloud can only be decrypted by users who have access to the right secret key. The secret key must be securely forwarded to the user who complies with the data owner's Access Control Policy. One significant disadvantage of this scheme is that if a user's right of access is revoked, the entire cipher text in the cloud must be encrypted.

The technique is ineffective and difficult to assess. Since the decryption key is usually

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

transmitted over a secure channel, public key cryptosystems require smaller key sizes. Furthermore, with limited resources, such as wireless sensor nodes and smartphones, it is difficult to store large-scale decryption keys. Allowing users to use a constant key to decrypt multiple data classes that can be quickly spread to multiple users would be an effective solution. We focus on integrating standalone MKES with broadcast encryption to support m data users and m0 data owners while lowering

Secure channel requirements from O(mm0) to O(m+m0).

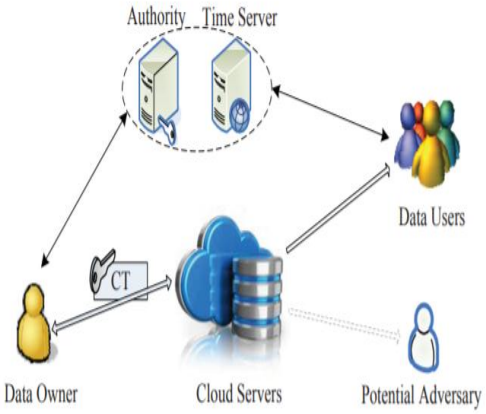

Figure. 2. Proposed approach working model

(1) Data Owner. The data owner may share sensitive information or files with friends (data users). Both shared data is saved on Cloud servers that are outsourced. (2) Authority. It is a significant organization that generates, distributes, and manages all private keys while relying on other system operators. (3) Time Server. It's a time reference server that doesn't communicate with other machines. He is in charge of ensuring that the appropriate release period is met. (4) Data Users. Data users are individuals who have passed the ID and have access to data that has been externalized by the data owner. Consumers of data. It should be remembered that only authorized users have access to the shared data during the authorization process. (5) Cloud Servers. It provides nearly infinite storage and management space for all computer data and information. Data can be stored on cloud servers by additional organizations with limited storage space. (6) Potential Adversary. It is a foe of polynomial times, as described in Section 3.4 of the KP-TSABE safety model. a) The database owner sends the information to the service provider. The service provider's approach visible in the algorithm after obtaining the user database.

RevoList DKPRSP (DKPUDO (ResultSet0));
Message with a private key and public key of the database owner and then stores encrypted data tables and revocation list. Because databases are encrypted using row encoding technology and data are encrypted by service providers, does not have access to the keys used to encrypt rows, the service provider will be unable to access the individual data. A UID of the revoked customers can be found on the service provider's termination list. If a user's application is approved, his UID is reviewed against the withdrawals list. The application is rejected if the User's UID is on the list.

b) KEY ACQUISITION PHASE: Calculation demonstrates how to keep the keys secure. 2. The

customer requests the key from the database's owner in order to communicate more thoroughly with the database's owner and service provider. The Database Owner then sends open/private keys as well as a symmetric key, which scratches the customer point's line.
Calculation 2: Key procurement stage.
Step 1: User asks for the DO for the keys.
Send (EKPUDO (EKKS (N1jjreq) jjKS));
Step 2: DO reacts with open/private keys and a symmetric key.
Receive (EKKS (EKPRDO (N1jjreqjjKjjPRUSRjjPUUSR)));.

c) User registration phase in the calculation, the method for including another customer is defined. 3. The new user must send an application to the database owner. The database owner generates a customer section that includes subtle database elements as well as a key to measure. The database owner codes the majority of the code.
The customer's primary key and cryptographic key are then saved in the finished table of the data base holders (section is initially scrambled with its private key and afterward by open key of the Service Provider with the end goal of confirmation and secrecy between Service Provider and Database Owner). The client's UID, encryption key, and token are sent in a package by the Database Owner. The client then retrieves the data from the service provider's server in order to recover interest points. The nonce and timestamps for requests and responses are sufficient to enable people in the center to replay and dodge.
Calculation 3: Algorithm for enlistment of another client
Step 1: User sends a scrambled enlistment solicitation to Database Owner.
Send (EKPUDO(EKPRUSR(UserDetails)));
Step 2: Database Owner upgrades Key Constraints table at its end
KeyTable add(UID;EKUi);
Step 3: Database Owner encodes the column and sends it to the Service Provider.
Send(EKPUSP (EKPRDO(EKUi(rowUi))));
Step 4: Service Provider Updates its duplicate of the database
DBSP insert(DKPUSP (DKPRDO(EKUi(rowUi))));
Step 5: Now the client can straightforwardly contact Service Provider for recovering his subtle elements.

d) Communication between Service Provider and User: After receiving the keys and token, the customer will inquire about nearby points of interest with the service provider. The service provider initiates D-H if the Client UID is not included in the rejection round. It achieves the aim of not keeping the data base owner on the Site all of the time. Calculation 4 demonstrates how to ensure that the updated D-H trade calculation uses a standard KS session key to ensure secure communication between service providers and clients. The revised D-h key trading [16] is used in the strategy to prevent attacks in the middle by screwing the Diffie Hellman parameter and using nonce in each carrier.

The method (Di) employs a well-known KS session key created by a modified D-H business, and the query result is scratched by the administrator (ri). [16.] Since only the customer has access to the message, this form of encoding ensures confidentiality between the service provider and the customer. The session key is valid for a set period of time, ensuring that secure communication is maintained over time. The client decodes the message after an encoded

reaction. Until the data is decoded, it is securely and easily listed with the customer. Calculation 4 : Algorithm for secure correspondence between Service Provider and client
Step 1: User sends demand for information access to Service Provider.
Send(UID; Token; q;N1);
Step 2: Service Provider checks denial rundown if the client is not in the rundown then just inquiry is handled
StorageArray Receive(UID; Token; q);
If in the event that (StorageArray[1] == RevoList(UID)) then
Goto step 7 else
Goto step 3
end if
Step 3: Service Provider and client trades a session key Ks created with the assistance of changed D-H Key Exchange Algorithm.
Step 4: Service Provider scrambles the information utilizing shared session key

EOi EKs(EKUi(rowUi));
Step 5: Service Provider sends the scrambled information to the User
Send(EOi;N1 + 1);
Step 6: Exit(previous step going to terminated);
Step 7: Service Provider sends a dismissal message to the User
Send(request can't be allowed);

4. Add Round Key
4. Final Round (no Mix Columns)
1. Sub Bytes- In this step, the 8-bit Rijndael S-box substitution is used to replace each byte of the state matrix with a sub-byte.
2. Shift Rows- The shift line phase operates on the state rows, cyclically moving bytes in each row with a specific offset. For AES, the first row remains unchanged. Any byte in the second row is shifted to the left. The third and fourth lines are also updated with two and three offsets, respectively. Add a sub key to the status in the Add Round Key. Each turn is extracted using a sub key of Rijndael's main calendar's main key, with each sub key being the same size as the state. On a bit-by-bit basis, the sub key is replaced by the byte XOR of the sub key.
It's an encryption algorithm that gives you a lot of power when it comes to computing encrypted data (cutting text) and returning encrypted performance. This algorithm can be used to address a variety of security and privacy issues. Encryption and decryption are both done with this algorithm.
The client site and the provider site both work with encrypted data. When data is exchanged between the customer and the service provider, this will eliminate the danger and hide plaintext from the service provider.
Homographic encryption allows sophisticated, non-original data mathematical operations to be performed on encrypted data.

RESULTS ANALYSIS

Experiments in a current cloud have yielded some interesting results. Validation of the time and space complexity, network, and accessibility requirements of our proposed building.
This section examines the performance and efficacy of extended construction on three virtual

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

machines (VMs) in a public cloud environment: an Encrypt Data owner VM consumer, a Decrypt Database VM data user client, and a trusted Third Party Server VM for server operations. In the public cloud section, you'll find the following section. Since most computing operations have been completed, the two VM clients each have 1GB of RAM and the VM server has 4GB of RAM. Many good software applications for bilinear pairings have been published in the literature, but details about how they were implemented are not always available. We use the Encrypt Row open source library and a Columbus-based library in our implementation. We provide efficient APIs for asymmetrical prime order bilinear paired and elliptical curve operations for the entire phase of translating plain text into cypher text, often known as encrypting or decrypting text into a simple

Text .The time analyses for encryption and decryption for the proposed work are shown in Figure 3.

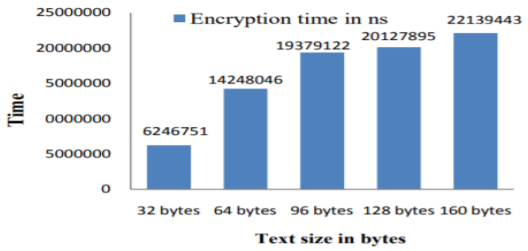

Figure 3 : Encryption time analysis

A small change in the input results in a large change in the output (for example, half the output bits rotate) The following conditions are met to achieve the avalanche effects situation: (e.g., flipping a single bit). • Any modification to a block cypher's key or plaintext will result in a significant change in the cypher's text. The avalanche effect, as described above, allows small changes to propagate quickly through algorithms, resulting in all output bits being subject to each entry bit before the algorithm ends. Avalanche Effect Formula is given below:

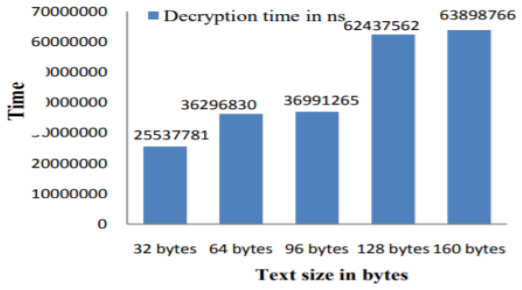

Figure: Decryption time analysis

TCP-based network communication between VMs was supported by the socket module's standard BSD sockets APIs. For any successful Inter-VM transfer from the server to the client, a single byte of customer recognition was given. For a case study with n=1000 documents and m=1000 users, an extended MKES scheme is available. First, we make setting up a configuration with just one instance B=1000 as simple as possible. There are two Random S and S subsets of sizes 500 and 500 in our case study. In theory, the device sends the aggregate key to 500 users for 500 objects.

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

CONCLUSION

Cloud computing is a modern, next-generation information technology that is gaining traction around the world. It has many advantages, but there are still some issues with this technology. The most difficult problem in this technology is security. To discuss the benefits and disadvantages of this security issue, we looked at several MKES encryption algorithms in this paper. In the cloud computing environment, we conclude that the MKES algorithm is the best algorithm for protecting useful data in an open network. When compared to other algorithms including RSA, DES, and AES, the MKES algorithm's ability to conduct encrypted data operations ensures high protection. The future task will be to use MKES algorithm hardware or software technologies to protect against some kind of cloud protection attack.

Figure 3 : Encryption time analysis

A small change in the input results in a large change in the output (for example, half the output bits rotate) The following conditions are met to achieve the avalanche effects situation: (e.g., flipping a single bit). • Any modification to a block cypher's key or plaintext will result in a significant change in the cypher's text. The avalanche effect, as described above, allows small changes to propagate quickly through algorithms, resulting in all output bits being subject to each entry bit before the algorithm ends. Avalanche Effect Formula is given below:

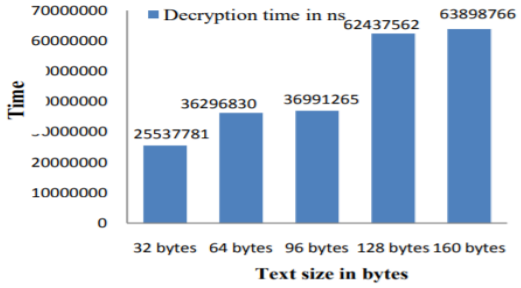

Figure: Decryption time analysis

TCP-based network communication between VMs was supported by the socket module's standard BSD sockets APIs. For any successful Inter-VM transfer from the server to the client, a single byte of customer recognition was given. For a case study with n=1000 documents and m=1000 users, an extended MKES scheme is available. First, we make setting up a configuration with just one instance B=1000 as simple as possible. There are two Random S and S subsets of sizes 500 and 500 in our case study. In theory, the device sends the aggregate key to 500 users for 500 objects.

CONCLUSION

Cloud computing is a modern, next-generation information technology that is gaining traction around the world. It has many advantages, but there are still some issues with this technology. The most difficult problem in this technology is security. To discuss the benefits and disadvantages of this security issue, we looked at several MKES encryption algorithms in this paper. In the cloud computing environment, we conclude that the MKES algorithm is the best algorithm for protecting useful data in an open network. When compared to other algorithms including RSA, DES, and AES, the MKES algorithm's ability to conduct encrypted data operations ensures high protection. The future task will be to use MKES algorithm hardware or software technologies to protect against some kind of cloud protection attack.

© UIJIR | ISSN (O) – 2582-6417
APRIL. 2021 | Vol. 1 Issue 11
www.uijir.com

REFERENCE

1. Raipurkar, K. V., & Deorankar, A. V. (2016). Improve data security in cloud environment by using LDAP and two way encryption algorithm. 2016 Symposium on Colossal Data Analysis and Networking (CDAN). doi:10.1109/cdan.2016.7570934.
2. Nagesh, S. H., Kumar, K. R. A., & Raj opal, K. T. (2017). Cloud architectures encountering data security and privacy concerns — A review. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). doi:10.1109/icecds.2017.8389745.
3. Rani, K., & Sagar, R. K. (2017). Enhanced data storage security in cloud environment using encryption, compression and splitting technique. 2017 2nd International Conference on Telecommunication and Networks (TEL-NET). doi:10.1109/tel-net.2017.8343557.
4. George, L. P., George Amalarethinam, D. I., & Chandran, A. S. (2018). GLEnc Algorithm to Secure Data in Public Cloud Environment. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). doi:10.1109/icacci.2018.8554451.
5. Moghaddam, F. F., Vala, M., Ahmadi, M., Khodadadi, T., & Madadipouya, K. (2015). A reliable data protection model based on re-encryption concepts in cloud environments. 2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC). doi:10.1109/icsgrc.2015.7412456.
6. Gu, X., Xu, Z., Xiong, L., & Feng, C. (2013). The Security Analysis of Data Re-Encryption Model in Cloud Services. 2013 International Conference on Computational and Information Sciences. doi:10.1109/iccis.2013.34.
7. Reddy, Y. (2018). Big Data Processing and Access Controls in Cloud Environment. 2018 IEEE's 4th Big data security conference, IEEE's International High Performance and Smart Computing Conference and IEEE's International Intelligent Data and Safety Conference (IIDSC) IEEE International Conference. IEEE (IDS). doi:10.1109/bds/hpsc/ids18.2018.00019.
8. Xiong, J., Liu, X., Yao, Z., Ma, J., Li, Q., Geng, K., & Chen, P. S. (2014). A Secure Data Self-Destructing Scheme in Cloud Computing. IEEE Transactions on Cloud Computing, 2(4), 448–458. doi:10.1109/tcc.2014.2372758.
9. Xia, Z., Wang, X., Sun, X., & Wang, Q. (2016). A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data. IEEE Transactions on Parallel and Distributed Systems, 27(2), 340–352. doi:10.1109/tpds.2015.2401003.
10. Patranabis, S., Shrivastava, Y., & Mukhopadhyay, D. (2017). Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud. IEEE Transactions on Computers, 66(5), 891–904. doi:10.1109/tc.2016.2629510.
11. Raipurkar, K. V., & Deorankar, A. V. (2016). Improve data security in cloud environment by using LDAP and two way encryption algorithm. 2016 Symposium on Colossal Data Analysis and Networking (CDAN). doi:10.1109/cdan.2016.7570934.
12. Nagesh, S. H., Kumar, K. R. A., & Rajgopal, K. T. (2017). Cloud architectures encountering data security and privacy concerns — A review. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). doi:10.1109/icecds.2017.8389745.
13. Rani, K., & Sagar, R. K. (2017). Enhanced data storage security in cloud environment using encryption, compression and splitting technique. 2017 2nd International

Conference on Telecommunication and Networks (TEL-NET). doi:10.1109/tel-net.2017.8343557.

14. George, L. P., George Amalarethinam, D. I., & Chandran, A. S. (2018). GLEnc Algorithm to Secure Data in Public Cloud Environment. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). doi:10.1109/icacci.2018.8554451.
15. Moghaddam, F. F., Vala, M., Ahmadi, M., Khodadadi, T., & Madadipouya, K. (2015). A reliable data protection model based on re-encryption concepts in cloud environments. 2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC). doi:10.1109/icsgrc.2015.7412456.
16. Gu, X., Xu, Z., Xiong, L., & Feng, C. (2013). The Security Analysis of Data Re-Encryption Model in Cloud Services. 2013 International Conference on Computational and Information Sciences. doi:10.1109/iccis.2013.34.
17. Reddy, Y. (2018). Big Data Processing and Access Controls in Cloud Environment. 2018 IEEE's 4th Big data security conference, IEEE's International High Performance and Smart Computing Conference and IEEE's International Intelligent Data and Safety Conference (IIDSC) IEEE International Conference. IEEE (IDS). doi:10.1109/bds/hpsc/ids18.2018.00019.
18. Xiong, J., Liu, X., Yao, Z., Ma, J., Li, Q., Geng, K., & Chen, P. S. (2014). A Secure Data Self-Destructing Scheme in Cloud Computing. IEEE Transactions on Cloud Computing, 2(4), 448–458. doi:10.1109/tcc.2014.2372758.
19. Xia, Z., Wang, X., Sun, X., & Wang, Q. (2016). A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data. IEEE Transactions on Parallel and Distributed Systems, 27(2), 340–352. doi:10.1109/tpds.2015.2401003.
20. Patranabis, S., Shrivastava, Y., & Mukhopadhyay, D. (2017). Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud. IEEE Transactions on Computers, 66(5), 891–904. doi:10.1109/tc.2016.2629510.
21. Sherman SM Chow, Cheng-Kang Chu, Xinyi Huang, Jianying Zhou, and Robert H Deng. Dynamic secure cloud storage with provenance. In Cryptography and Security: From Theory to Applications, pages 442–464. Springer, 2012.
22. Erik C Shallman. Up in the air: Clarifying cloud storage protections. Intell. Prop. L. Bull., 19:49, 2014.
23. Cheng-Kang Chu, Sherman SM Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H Deng. Key-aggregate cryptosystem for scalable data sharing in cloud storage. Parallel and Distributed Systems, IEEE Transactions on, 25(2):468–477, 2014.
24. Lang, B., Wang, J., Li, M., & Liu, Y. (2018). Semantic-based Compound Keyword Search over Encrypted Cloud Data. IEEE Transactions on Services Computing, 1–1. doi:10.1109/tsc.2018.2847318.
25. M. Li, B. Lang, and J. Wang, “Compound concept semantic similarity calculation based on ontology and concept constitution features,” in Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on, 2015, pp. 226–233
26. S. Kamara, C. Papamanthou, and T. Roeder, “Dynamic searchable symmetric encryption,” in ACM Conference on Computer and Communications Security, 2012, pp. 965–976.