World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 5.924

UNCRACKABLE CIPHER DYNAMIC DOUBLE ENCRYPTION STANDARD IN CLOUD FOR DATA ACCESS CONTROL AND PRIVACY PRESERVING MECHANISM

Dr. K. Sai Manoj*

CEO, Amrita Sai Institute of Science and Technology.

Article Received on 14/10/2019 **Article Revised on 04/11/2019** **Article Accepted on 25/11/2019**

***Corresponding Author**
Dr. K. Sai Manoj
CEO, Amrita Sai Institute of Science and Technology

ABSTRACT

Now-a-days, the excessive use of internet cloud has received much attention. Cloud computing is the evolving paradigm that provides the services in which cloud consumers can remotely store their data into the cloud and access the on-demand high-quality applications. Cloud computing is mainly used for resource sharing and with very low-maintenance. In the existing Extendable Access Control System procedure, the authority is the trusted party, but in many cases, they may perform an illegal action which leads to data loss. In the proposed work encryption of data is done through Uncrackable Cipher Dynamic Double Encryption Standard (UCDDES). UCDDES performs with the key length of 32, 40 and 48. After dynamically selecting the key length the data governor sent the key request to the authority. Then the data governor generates the partial secret key based on the obtained key length. It is further used to decrypt the data and store in the cloud. As a result, the security of cloud and access control are improved and the problems faced by the unauthorized user/ hackers accessing data are reduced. It also increased the cloud security and prevented from dictionary attacks, brute force attacks, collision attacks, and so on.

INDEX TERMS: Cloud computing, Cloud network security, Dictionary attacks, Data security issues, UCDDES based data encryption.

I. INTRODUCTION

Cloud computing is mainly used for resource sharing and with very low-maintenance. Even though there is no unique description for cloud computing, however, one common meaning given by many researchers originates from the National Institute of Standards (NIST): a model for network access for sharing resources like application, storage, network, services and servers that would be released with less effort.[1]

In cloud computing, a user can store their information at remote storage servers. Cloud storage model consists of three entities, such as a cloud storage server, Third Party Checker (TPC) and user's group. The user's group can consist of data owner and the user who ratified to access the data and can be altered by the data owner. The group of user can have the data storage services provided by the cloud storage server. The TPC can provide data integrity of the shared data store in the cloud server. In the remote storage cloud server, the data owner could encrypt and upload its data. Sometimes, the cloud server gives invalid results such as hardware/software failure, malicious attack, and human maintenance.

These remote storage servers are coped by a Cloud Service Provider (CSP) frequently as a third party.[2], [3] Moreover, computer hardware like memory, disk space, and processor are virtualized and provided to the end users as a facility via the public Internet.[4], [5] Several virtual machines are distributed across a set of powerful data centers with different topographical points that serves as a cloud facility, which is interrelated using the telecommunication links. Furthermore, the cloud users have to pay based on the actual amount of service they have utilized as corresponding to water or electricity bill.[6], [7] The Cloud Computing model provides some benefits to both users and service providers.

The benefits for an end user are the rapid elasticity, low cost for maintenance, negligible upfront investment, measured service and global access to cloud services.[8], [9] In cloud computing, the virtualization technology utilization consequences in high utilization of resource and thus carry out less costs for electrical energy to service providers. Although clouds are more dependable and have more powerful infrastructure compared to personal computers, there are still security worries that prevent users to deploy their businesses in the cloud and therefore decreases the growth of cloud computing. The cloud provider lose their control over the data and this is the reason why the people do not trust them.[10], [11] Sensitive information in cloud storage should also be protected from unauthorized access.

As a result, the people who are generating data are wished to know about the confidentiality of the information by using cryptographic Access Control Systems (ACS). In recent times, investigators have suggested numerous data ACS safeguard the stored information in the cloud. Such schemes authorize the data governor to handle authorized users securely and repeal their authorization rights. Attribute-based encryption (ABE) is a significant technique interpreting the unique characteristics of the user, data governor, or cloud environment to control the data access implementation.[12]-[15]

Our Contribution

The contributions of the present research work include the following:

- To implement encryption technique and access control mechanism.
- To provide the security of the network.
- To reduce the unauthorized data.
- To reduce the issues related to data and key leakage and also to secure the data with low cost.

The rest of this paper is structured as follows: Section 2 examines the literature review in the area of WSNs. Section 3 briefly discussed the methodology part of the study. Section 4 presents the proposed methodology, i.e., Uncrackable Cipher Dynamic Double Encryption Standard. Section 5 provides evaluation results of UCDDES and compares them against two of its best competitors in the literature. Section 6 discusses the conclusion.

II. LITERATURE REVIEW

A. Attributed-Based W Access Control (ABAC) scheme was presented by Qiu et al., (2018) to protect the financial customers' privacy data. Assurance of data privacy in semantic approach to user access control is made. Higher-level secure sustainability is obtained as it could contract with dynamic threats, together with the developing and future threats. Oblivious Random Access Memory (ORAM) for high security and data sharing is proposed by Yuan et al., (2018). The data block can be avoided from modification by shuffling. The IND-CPA security is provided for the system with an ID-Based signature and the Path-ORAM security properties. As a result, the system showed the best computation complexity.

The explanation regarding protection by Srinivasan et al., (2018) includes the cloud attacks, integrity, privacy, vulnerability in resource sharing and leakages. The services on quality of service, data transmission and the significant information omitting are assured. This proficient technique preserves the environment of cloud. Analysis of security in addition to privacy

identified the competence of the suggested procedure then extended productive efficiency with the safe cloud environment.

Iyapparaja et al., (2017) suggested a different encryption technology and signature key on cloudlets also and those signature key send to a register email id. Every cloud was divided in cloudlet, for specific cloudlet needed to access user must register on them. In this user can use other cloud information and remove other data with the appropriate verification in user side and main cloud server side. The user may store any information such as pdf, image, text, etc. Once the customer can store data on cloudlet, another user can use those data.

Access control systems based on attributes and associated every scheme's functionality in was given by Sookhak et al., (2017) addition to the characteristic. The attribute-based approaches are established on the design; access control mode, annulment method, annulment mode, annulment issue, and controller are presented. The pros and cons of the ABE technique was identified by this method.

A dynamic three-layer encryption system was proposed by Tang et al., (2018) on DES and network coding. The scheme benefited by attaining a dynamic transition amongst efficiency and security. It increases its adaptability to numerous cyber conditions. The running ratio of the innovative scheme was somewhat lower or equivalent to triple DES.

III. PROPOSED METHODOLOGY

Cryptographic algorithms are being used for the protection of data, the security objective called confidentiality[1]-[3] which is the parameter that is considered and integrated with the system.[2]-[4] However,still, the attackers, electronic malfunctions, the virus and the electronic eavesdropper try to attack the information and their transmission. Advanced Encryption Standard (AES) comprises ten rounds of difficult arithmetic and matrix operation that leads to suspension in a conversion process. In an AES procedure, there are also some security concerns. For overcoming those concerns, a novel algorithm called Uncrackable Cipher Dynamic Double Encryption Standard (UCDDES**)** is proposed.

The proposed methodology is clearly given in below Fig.1. In the encryption process, firstly the plaintext is converted to cipher text 1. Then, cipher text1 is further converted to cipher text 2 and it is stored in a cloud. At another end, the cipher text2 is downloaded. Finally, decryption of cipher text 1 takes place. Cipher text 1 will be deciphered to obtain the plaintext.

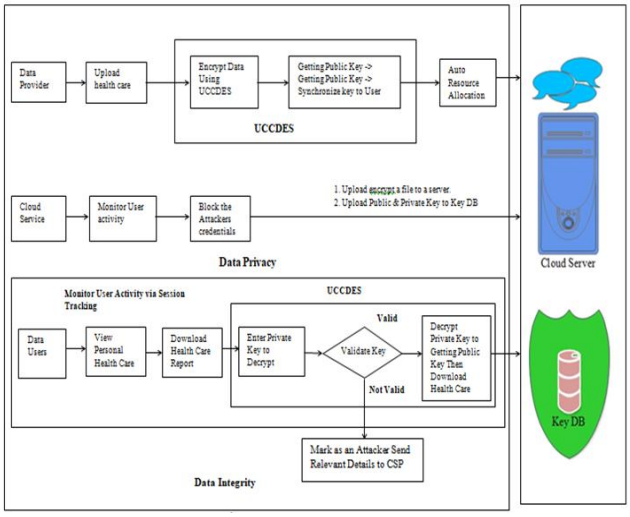

Fig. 1. Proposed methodology.

3.1 UCDDES based data encryption with the user revocation and partial key parameters

For cloud storage, encryption is an important one. In the existing methodology, Extendable Access Control System (EACSIP) algorithm is built on top of an ideal cryptographic primeval, i.e., Functional Key Encapsulation with Equality Testing (FKE-ET). It assumes the authorities as the trusted party and get the security parameters from the authority. Then the information from the data owners is encrypted through the authority. Further, they are sent to the cloud storage. Then the user can download the information from the cloud through private keys and access policies. The existing system does not offer the efficient encryption technique; the classical symmetric encryption technique was used which is not efficient for the real-time scenario.

So, the proposed work encrypted the data through UCDDES. Generally, the UCDDES contains the key with the length 32, 40 and 48. Because of the three-key length, the intruder cannot know the combinations for the entire three key which is the huge task. Further, in the suggested system, even the security user cannot crack the codes once it is generated.To make the plain text into more secured ciphertext, the proposed framework is separated into two phases, they are

1) **Phase I –Cipher Text1:** The initial phase which gets the plaintext as input. The input text is then converted into the first form of ciphertext with Advanced Substitution method with 128 bit and a length of a key of 16.

2) Phase II – Cipher Text2: The ciphertext which is generatedin phase I is given as input to the phase II. The ciphertext is further encoded with a symmetric key having a length of 16 again (Private Key). This key is similar to a one-time pad key which can be utilized only one time for encryption and decryption.

UCDDES offers advantages such as

- Protection from the collision attack, the SQL injection, the dictionary attack and the brute force attack.
- Achieving data integrity, data confidentiality, and data privacy.

3.2 Algorithm 1: UCDDES Algorithm for Encrytion – Phase 1

In the encryption process of UCDDES phase 1, a plain text (t1) is given as input. The aim of this algorithm is to get a cipher text (C1). The block size considered includes 128, 192, 256 bits. The cipher text contains the secret key with 32, 40, 48 bits lengths. Key expansion is done by Add Round Key to attain the cipher text (C1).

Input: Plain Text (t1), R: Random, Nr: Total_Rounds, Nb: Constant value
Output: Cipher Text (C1)
Compute: Block Size R[128, 192, 256-bit Plain_Text]
For Cipher (t1, Secret_Key) **then // UCDDES contains the Secret_key length of 32, 40 and 48**
Byte_State [4, Nb]
for (State = t1)
Add_Round_Key (State, w[0, Nb-1]) **//Key Expansion**
for round ==1 : Nr-1 **then**
    Sub_Byte(State)
    Shift_Rows(State)
    Mix_Columns(State)
    Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
end for
  Sub_Byte(State)
  Shift_Bytes(State)
  Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1)
  C1 = State

3.3 Algorithm 2: UCDDES Algorithm for Encryption-Phase 2

The input for encryption in UCDDES Phase 2 is cipher text 1. In order to find the cipher text (C2), the same steps are followed as in phase 1. For key expansion, in the initial round, Add Round Key procedure is done. In this operation, the input to the round is exclusive-ORed with the round key. The SubBytes phase of AES involves splitting the input into bytes and passing each through a Substitution Box or S-Box. In ShiftRows, the internal state of the cipher text is shifted. MixColumns performs operations splitting the matrix by columns instead of rows.In the main rounds, SubBytes, ShiftRows, MixColumns, AddRoundKey processes are done. In the final round, SubBytes, ShiftRows, and AddRoundKey are done.

Input: Cipher Text (C1), Nr: Total_Rounds, Nb: Constant value
Output: Cipher Text (C2)
if (C1 != NULL) **then**
For Cipher2 (C1, Secret_Key) **then** // **UCDDES contains the Secret_key length of 32, 40 and 48**
Byte_State [4, Nb]
for (State = C1)
Add_Round_Key (State, w[0, Nb-1]) //**Key Expansion**
for round ==1 : Nr-1 **then**
Sub_Byte(State)
Shift_Rows(State)
Mix_Columns(State)
Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
end for
Sub_Byte(State)
Shift_Bytes(State)
Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1)
C2 = State

3.4 Algorithm 3: UCDDES Algorithm for Decryption – Phase 1

In the phase1 of decryption process, C2 is given as input. The output to be obtained is C1. The length of block size is 128, 192, and 256 bits. Three stages of decryption take place. AddRoundKey procedure is done in Inverse Initial Round. In the inverse Main Round, AddRoundKey, MixColumns, ShiftRows, and SubBytes are done whereas in the inverse Final

Round AddRoundKey, ShiftRows, SubBytes are performed. The Decipher1 contains the secret key of length 32, 40 and 48 bits. Key is expanded during AddRoundKey.

Input: Cipher Text (C2), R: Random, Nr: Total_Rounds, Nb: Constant value
Output: Cipher Text (C1)
Compute: Block Size R[128, 192, 256-bit Plain_Text]
For DeCipher1 (C2, Secret_Key) **then // UCDDES contains the Secret_key length of 32, 40 and 48**
Byte_State [4, Nb]
for (State = C2)
Add_Round_Key (State, w[0, Nb-1]) **//Key Expansion**
for round ==1 : Nr-1 **then**
Inv_Sub_Byte(State)
Inv_Shift_Rows(State)

Inv_Mix_Columns(State)
Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
end for
Inv_Sub_Byte(State)
Inv_Shift_Bytes(State)
Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1)
C1 = State
call(Decipher2)

3.5 Algorithm 4: UCDDES Algorithm for Decrytion – Phase 2

In phase 2 of the decryption process, C1 is taken as input. Plain text (t1) is the expected output. Decipher 2 contains the secret key length of 32, 40, and 48 bits. The same process is carried out as in phase 1 to get t1.

Input: Cipher Text (C1), R: Random, Nr: Total_Rounds, Nb: Constant value
Output: Plain Text (t1)
Compute: Block Size R[128, 192, 256-bit Plain_Text]
For DeCipher2 (C1, Secret_Key) **then // UCDDES contains the Secret_key length of 32, 40 and 48**
Byte_State [4, Nb]
for (State = C1)

Add_Round_Key (State, w[0, Nb-1]) **//Key Expansion**

for round ==1 : Nr-1 **then**

Inv_Sub_Byte(State)

Inv_Shift_Rows(State)

Inv_Mix_Columns(State)

Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])

end for

Inv_Sub_Byte(State)

Inv_Shift_Bytes(State)

Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1)

t1 = State

end

3.6 Explanation of the algorithms

1) Consider a block size as given below:

$$\text{Block size} = R\begin{bmatrix}128\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\\192\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\\256\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\end{bmatrix}+$$

$$\begin{bmatrix}128\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\\192\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\\256\ bit\ Plaintext(4\ words\ /\ 16\ bytes)\end{bmatrix}$$

Where R denotes Random bit generation.

2) The overall number of rounds considered is 10, 12 and 14.

$$\text{Number of rounds} = \begin{bmatrix}10\ Rounds\\12\ Rounds\\14\ Rounds\end{bmatrix}+\begin{bmatrix}10\ Rounds\\12\ Rounds\\14\ Rounds\end{bmatrix}$$

3) The master key size considered is as mentioned below.

4) Representations

a) Input array 1 -> State Array 1 -> Input array 2 -> State Array 2 -> Output Array. Input plain text are stored in the array table of size 4*4 + 4*4

$$\begin{bmatrix}0 & 4 & 8 & 12\\1 & 5 & 9 & 13\\2 & 6 & 10 & 14\end{bmatrix}\begin{bmatrix}0 & 4 & 8 & 12\\1 & 5 & 9 & 13\\2 & 6 & 10 & 14\end{bmatrix}$$

0 Key table -> Key Expansion

b) Substitution of Bytes:

S Box input Array 1 -> State Array 1 -> S Box input Array 2 -> State Array 2 ->S'Box output Array.

c) Shift rows: S'Box output -> Input for Shift Box

Shift Input array 1 -> Shift output Array 1 -> Input Array 2 -> Output Array 2

d)Transformation of Mix Column: Shift Box Output -> Input for mix columns

e)Add round keys: It represents XOR operation with state array.

IV. RESULTS AND DISCUSSION

Proper cryptography algorithm is chosen for an application in order to avoid the consequences of encryption. Therefore, the parameters like performance, weakness, andstrength of the procedures must be discussed in detail to prove the performance of the chosen algorithm.[26] The proposed encryption procedure is analyzed based upon several metrics, andcompared.

Table 1. Comparison of results.

Data Size in MB	Encryption time (s)		Decryption time (s)		Upload time (s)		Download time (s)	
	Offline method [27]	UCDDES	Offline method [27]	UCDDES	Offline method [27]	UCD DES	Offline method [27]	UCD DES
5	1.7	1.2	1.5	1.1	1.5	1.3	1.2	0.97
10	5.1	4.9	4.9	4.2	3.7	3.1	3.2	2.8
20	11.7	11.2	11	10.3	6	5.3	4	3.7
50	17.7	17.1	16	15.3	15	14.3	11	10.3
100	20.2	19.6	21	20.4	32	31.7	17	16.4
200	27.1	16.3	27.1	26.3	54	53.1	26	25.3
512	34.2	33.5	37	36.4	70	69.3	40	39.3
1024	43.1	42.2	40	39.2	80	79.1	70	69.1

4.1. Encryption time –It is recognized as the conversion time from plain text to the ciphertext which is built including the key length, inputblock length, and mode. Encryption time is considered to be important since it plays a major role in the performance of an algorithm.

4.2. Decryption time - It is known as the conversion time from ciphertext to the plaintext which includes the key length, inputblock length, and mode. The system works fast if the decryption time is less.

4.3. Upload time –It is recognized as the time taken to store the data into the cloud system.

4.4. Download time – It is acknowledged as the time taken to retrieve the information from the cloud system.

The encryption, decryption, upload and download time are taken in milliseconds. The performance of a system depends entirely on the above mentioned metrics. If the time taken for encrypting, decrypting, uploading and downloading is less, then the system performs better than usual.

In Fig.2, the encode time is compared with offline method. The Offline technique consumes high encode time, and our UCDDES technique takes less encode time. When the input file size increases the encoding time also increases. For example, for a 5 Mb file, the encryption time is 1.2 milliseconds whereas for 1024 Mb file, it is 42.2 milliseconds. The UCDDES encryption technique performs well compared to offline technique.

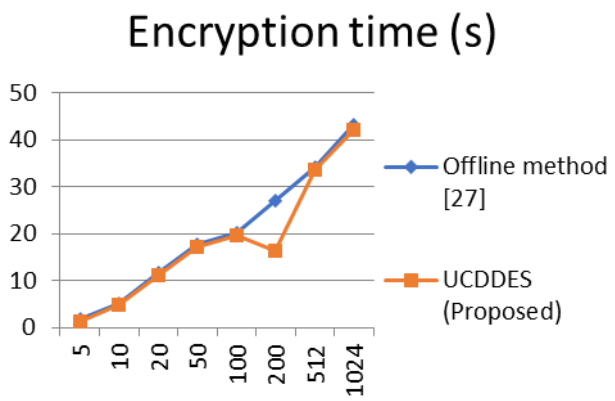

Fig 2. Evaluation of time of Encryption among various algorithms.

In below Fig.3, the decode time of UCDDES is less when associated with the offline technique. The offline technique takes high decode time, and our UCDDES technique takes low decryption time. When the input file length increases the decode time also increases. For example, for a 5 Mb file, the encryption time is 1.1 milliseconds whereas for 1024 Mb file, it is 39.2 milliseconds.

The upload text time is compared with offline technique. The Fig.4 demonstrates that the upload time is high for Offline technique and low for UCDDES. The time increases together

with the file length. For example, for a 5 Mb file, the encryption time is 1.3 milliseconds whereas for 1024 Mb file, it is 79.1 milliseconds.

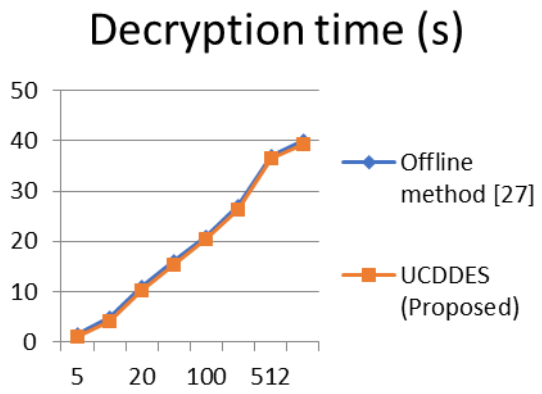

Fig 3. Comparison of Decryption time among various algorithms.

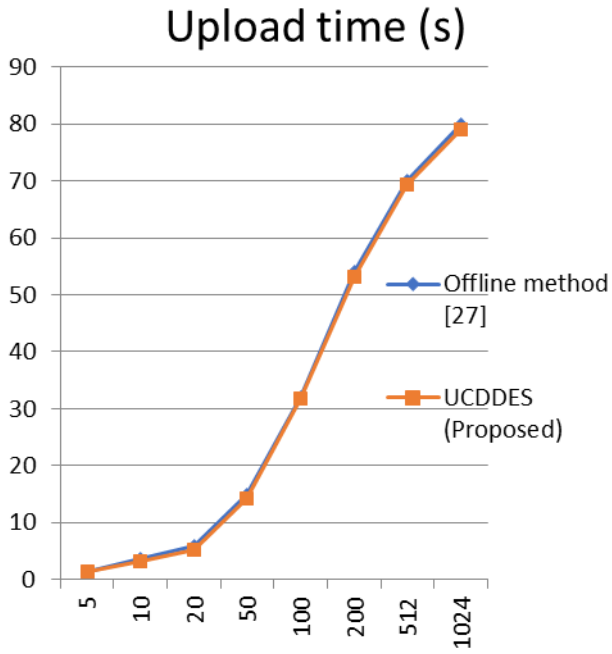

Fig 4. Comparison of Upload time among various algorithms.

The text download time is less in UCDDES compared to offline technique. The download time is lower than the upload time. Below Fig.5 displays that the upload time is high for Offline technique and low for UCDDES. The download time increases together with the file length. For example, for a 5 Kb file, the encryption time is 0.97 milliseconds whereas for 1024 Mb file, it is 69.1 milliseconds

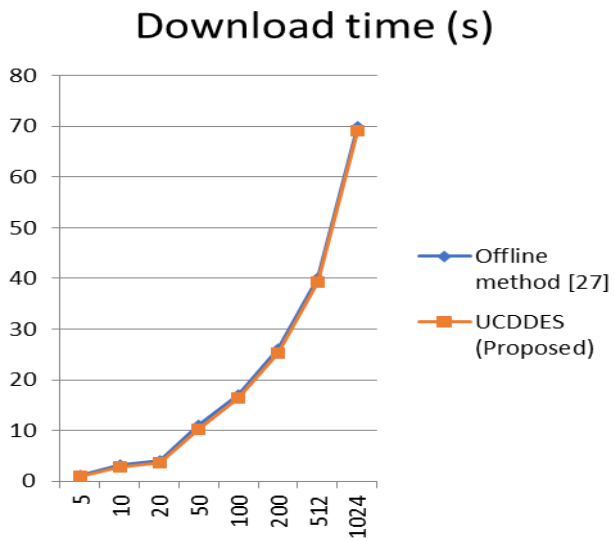

Fig 5. Comparison of Download time among various algorithms.

IV. CONCLUSION

The UCDDES –Uncrackable Cipher Dynamic Double Encryption Standard has been used to encode the input data. The encoding procedure is carried out in two stages: the first stage encrypts the plaintext with the key. The output of the first stage is given as the input to the second stage where again the double encoding takes place with another key. Then the text is saved in the cloud. Then the text can be downloaded from the cloud. Decryption of the text is done with the keys given. By using this encoding technique, the data integrity, data confidentiality, and data privacy are improved. The proposed technique is utilized to prevent SQL injection, Brute force, Collision attack and dictionary attack. The encoding time and decryption time will further be reduced in the future work to achieve better performance.

REFERENCES

1. Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).
2. Zhou, Z., & Huang, D. (2012, October). Efficient and secure data storage operations for mobile cloud computing. In Proceedings of the 8th International Conference on Network and Service Management (pp. 37-45). International Federation for Information Processing.
3. Sookhak, M., Talebian, H., Ahmed, E., Gani, A., & Khan, M. K. (2014). A review on remote data auditing in single cloud server: Taxonomy and open issues. Journal of Network and Computer Applications, 43: 121-141.
4. Kumar, K., & Lu, Y. H. (2010). Cloud computing for mobile users: Can offloading computation save energy?. Computer, 43(4): 51-56.

5. Sookhak, M., Akhunzada, A., Gani, A., Khurram Khan, M., & Anuar, N. B. (2014). Towards dynamic remote data auditing in computational clouds. The Scientific World Journal, 2014.
6. Whaiduzzaman, M., Sookhak, M., Gani, A., & Buyya, R. (2014). A survey on vehicular cloud computing. Journal of Network and Computer Applications, 40: 325-344.
7. Sookhak, M., Gani, A., Khan, M. K., & Buyya, R. (2017). Dynamic remote data auditing for securing big data storage in cloud computing. Information Sciences, 380: 101-116.
8. Sookhak, M., Yu, F. R., Khan, M. K., Xiang, Y., & Buyya, R. (2017). Attribute-based data access control in mobile cloud computing: Taxonomy and open issues. Future Generation Computer Systems, 72: 273-287.
9. Sookhak, M., Gani, A., Talebian, H., Akhunzada, A., Khan, S. U., Buyya, R., & Zomaya, A. Y. (2015). Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues. ACM Computing Surveys (CSUR), 47(4): 65.
10. Wang, C., Ren, K., Lou, W., & Li, J. (2010). Toward publicly auditable secure cloud data storage services. IEEE Network, 24(4).
11. Shiraz, M., Sookhak, M., Gani, A., & Shah, S. A. A. (2015). A study on the critical analysis of computational offloading frameworks for mobile cloud computing. Journal of Network and Computer Applications, 47: 47-60.
12. Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2007, November). A data outsourcing architecture combining cryptography and access control. In Proceedings of the 2007 ACM workshop on Computer security architecture (pp. 63-69). ACM.
13. Sahai, A., & Waters, B. (2005, May). Fuzzy identity-based encryption. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 457-473). Springer, Berlin, Heidelberg.
14. Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 89-98). Acm.
15. Hur, J. (2013). Improving security and efficiency in attribute-based data sharing. IEEE transactions on knowledge and data engineering, 25(10): 2271-2282.
16. Qiu, M., Gai, K., Thuraisingham, B., Tao, L., & Zhao, H. (2018). Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry. Future Generation Computer Systems, 80: 421-429.

17. Zuo, C., Shao, J., Liu, J. K., Wei, G., & Ling, Y. (2018). Fine-Grained Two-Factor Protection Mechanism for Data Sharing in Cloud Storage. IEEE Transactions on Information Forensics and Security, 13(1): 186-196.
18. Yuan, D., Song, X., Xu, Q., Zhao, M., Wei, X., Wang, H., & Jiang, H. (2018). An ORAM-based privacy-preserving data sharing scheme for cloud storage. Journal of information security and applications, 39: 1-9.
19. Ning, J., Cao, Z., Dong, X., Liang, K., Wei, L., & Choo, K. K. R. (2018). CryptCloud+: Secure and Expressive Data Access Control for Cloud Storage. IEEE Transactions on Services Computing.
20. Srinivasan, S., & Raja, K. (2018). An Advanced Dynamic Authentic Security Method for Cloud Computing. In Cyber Security: Proceedings of CSI 2015 (pp. 143-152). Springer Singapore.
21. Divya, S. V., Shaji, R. S., & Venkadesh, P. (2018). An Efficient Data Storage and Forwarding Mechanism Using Fragmentation-Replication and DADR Protocol for Enhancing the Security in Cloud. Journal of Computational and Theoretical Nanoscience, 15(1): 111-120.
22. Cui, H., Deng, R. H., & Li, Y. (2018). Attribute-based cloud storage with secure provenance over encrypted data. Future Generation Computer Systems, 79: 461-472.
23. Iyapparaja, M., Navaneethan, C., Meenatchi, S., Kumar, P. J., & Suganya, P. (2017). A Privacy-Preserving Secure Access Control Mechanism in Cloud.
24. Sookhak, M., Yu, F. R., Khan, M. K., Xiang, Y., & Buyya, R. (2017). Attribute-based data access control in mobile cloud computing: Taxonomy and open issues. Future Generation Computer Systems, 72: 273-287.
25. Tang, H., Sun, Q. T., Yang, X., & Long, K. (2018). A Network Coding and DES Based Dynamic Encryption Scheme for Moving Target Defense. IEEE Access, 6: 26059-26068.
26. Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA, and Blowfish. Procedia Computer Science, 78: 617-624.
27. Sajjan, R. S., & Ghorpade, V. R. (2017, February). Secure online encryption with partial data identity outsourcing: An exemplar for cloud computing. In Wireless and Optical Communications Networks (WOCN), 2017 Fourteenth International Conference on (pp. 1-8). IEEE.