An Enhanced Data Access Control and Privacy Preserving Mechanism in Cloud Using Uncrackable Cipher Dynamic Double Encryption Standard

P. Jhansi Rani, Dr. M. Akkalakshmi

1Department of Information Technology, Gandhi Institute of Technology and Management, Hyderabad, Telangana, India

2Department of Information Technology, Gandhi Institute of Technology and Management

E-mail: 1jhansi.infoarea@gmail.com, 2akkalakshmi.muddana@gitam.edu

Corresponding Author: P. Jhansi Rani
Email: jhansi.infoarea@gmail.com

https://doi.org/10.26782/jmcms.2019.02.00009

Abstract:

Cloud computing is the evolving paradigm that provides the services in which cloud consumers can remotely save their data into the cloud and access the on-demand high-quality applications. In the existing technique explained an Extendable Access Control System procedure supposed that the authority is the trusted party, but in many cases, they may perform an illegal action which causes the data loss. The proposed work encrypted the data through Uncrackable Cipher Dynamic Double Encryption Standard (UCDDES). Generally, the UCDDES contains the key length of 32, 40 and 48. To randomly select the key length reduced the data security issues. After dynamically selecting the key length the data governor sent the key request to the authority. Then based on the obtained key length the data governor generated the partial secret key. It is further used to decrypt the data and stored in the cloud storage. The results improve the security of cloud and access control. It reduces the issue of unauthorized user/hackers accessing data. It increases the cloud security and prevents from dictionary attacks, brute force attacks, collision attacks, and so on.

Keywords: Cloud computing, data security issues, UCDDES based data encryption, cloud network security.
I. Introduction

Cloud computing as the emerging technology has become popular these days and established substantial interest from both academia and business. 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 [VIII].

In cloud computing, a user can store their information at remote storage servers. These remote storage servers are coped by a Cloud Service Provider (CSP) frequently as a third party [XVII, XXIV]. Moreover, computer hardware like memory, disk space, and processor are virtualized and provided to the end users as a facility via the public Internet [VII, XIV]. 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 [XV, XXII]. The Cloud Computing model provides some benefits to both users and service providers.

For an end user, the rapid elasticity, low cost for maintenance, negligible upfront investment, measured service and global access to cloud services are the benefits [XVI, XVIII]. 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 reason why the people do not trust the cloud provider to manage the data is that they lose their control over the data [XXI, XXIII]. Besides, sensitive information in cloud storage should 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[II, IV, V, XII].

Our contribution:
The main contributions of this research include the following:

- Encryption technique
- Access control mechanism
- Increase network security
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

Qiu et al. (2018) stated a system that protects the financial customers’ privacy data using Attributed-Based Access Control (ABAC) scheme. The proposed system aimed to assure the data privacy in which a semantic approach to user access control is made. Next, a user-centric approach that prevents the data from risky processes on the cloud side was suggested. In conclusion, the recommended scheme has higher-level secure sustainability as it could contract with dynamic threats, together with the developing and future threats.

Zuo et al. (2018) proposed a method that protects the information in the storage of cloud. Initially, the cryptographic key was protected by considering the two aspects. These aspects are utilized to maintain the key secrete. Next, the secret key can be revoked efficiently by incorporating the alternative re-encryption along with key separation methods. In conclusion, the data was secured in a fine-grained way by implementing the ABE technique. Additionally, the performance valuation and security analysis presented that the proposal was secure and effective, respectively.

Yuan et al. (2018) stated a method called Oblivious Random Access Memory (ORAM) used for high security and data sharing. The data block can be avoided from modification through shuffling. The IND-CPA security is provided for the system with an ID-Based signature and the Path-ORAM security properties. The system showed the best computation complexity.

Ning et al. (2018) dealt with the two categories of access exploitation they are, semi-trusted authority's illegal access, and cloud user's unauthorized access. These types of exploitation can be reduced by a cloud storage scheme called CP-ABE with auditing and white-box traceability known as Crypt Cloud+. The paper proved the security and utility of the scheme.

Srinivasan et al. (2018) explained the protection concern that includes the cloud attacks, integrity, privacy, vulnerability in resource sharing and leakages. The system involved with the information protection from loss and protection from hackers and attackers of a real-time environment. The system provided the authentication and authorization. The services on quality of service, data transmission and omit the significant information are assured. This proficient technique preserves the environment of cloud with enhanced performance evaluation. Additionally,
analysis of security in addition to privacy identified the competence of the suggested procedure then extended productive efficiency with the safe cloud environment.

Divya et al., (2018) focused on a data forwarding tactic named "Secure Cloud Data Forwarding Framework (SCDF)" which helpfully attained security and performance problems. Now, the file was divided into different fragments and stored in a cloud server and transmitted to the customers using DADR (Distributed Autonomous Depth First Routing) protocol when they requested. The multi-streaming feature of DADR makes the fragments to select an alternative path if any failure happened. Using elliptic curve cryptography and T-coloring techniques, the technique attained extra ordinary data security. Furthermore, the method also certified the users' confidentiality and implemented the collusion resistance. Additionally, the trial out comes presented that SCDF was more protected and effective.

Cui et al., (2018) presented an attribute-based cloud storage system with secure provenance. Initially, the storage system was without revocation, and later it extends to the revocation mechanism to thwart the data customers from retrieving the newly encrypted data. Then, the algorithm was implemented in the two both the approaches. The experimental outcome presented that the proposed systems are acceptable to be applied in practice.

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 able to use those data.

Sookhak et al., (2017) provided a complete investigation on access control systems which were based on attributes and associated every scheme’s functionality in 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. Centralized, Decentralized, and Hierarchal based classifications are made. This was performed to know the pros and cons of the ABE technique.

Tang et al., (2018) proposed a dynamic three-layer encryption system made on DES in addition to network coding. Established in the theoretical analysis, the novel scheme was exposed to have the benefit to attain a dynamic transition amongst efficiency and security. It increases its adaptability to numerous cyber conditions. The simulation outcomes also revealed that the running ratio of the innovative scheme was somewhat lower or equivalent to triple DES.

III. Methodology

That is why the cryptographic algorithms are being used for the protection of data, the security objective called confidentiality [VIII, XVII, XXIV] is the parameter that is considered and integrated with the system [VII, XVII, XXIV]. However, still, the hackers, electronic frauds, the virus and the electronic eavesdroppers are trying to
attack the information and their transmission. The procedure Advanced Encryption Standard (AES) has ten rounds of difficult arithmetic and matrix operation that causes suspension in a conversion process. In an AES procedure, there are also some security concerns. For overcoming those concerns, a novel algorithm is proposed which is recognized as UCDDES.

IV. Proposed methodology

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

The encryption system for the cloud storage is an important one. In the existing technique described the Extendable Access Control System (EACSIP) algorithm which is built on top of an ideal cryptographic primeval, i.e., Functional Key Encapsulation with Equality Testing (FKE-ET). It assumes that the authorities are trusted party and get the security parameters from the authority. Then the information from the data owners is encrypted through the authority. Further, they could send 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; it used the classical symmetric encryption technique which is not efficient for the real-time scenario.

So, the proposed work encrypted the data through UCDDES. Generally, the UCDDES contains the key length of 32, 40 and 48. Because of the three-key length, the intruder cannot attain 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 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 generated in 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.

The main advantages of UCDDES are

- Protecting from the collision attack, the SQL injection, the dictionary attack and from the brute force attack.
- Data integrity, data confidentiality, and data privacy are achieved through UCDDES.

Algorithm 1: UCDDES Algorithm for Encryption – Phase 1

Input: Plain Text (t1), R: Random, Nr: Total Rounds, Nb: Constant value
Output: Cipher Text (C1)
1 Compute: Block Size R[128, 192, 256-bit Plain_Text]

2 For Cipher (t1, Secret_Key) then // UCDDES contains the Secret_key length of 32, 40 and 48

Byte State [4, Nb]

4 for (State = t1)
5 Add_Round_Key (State, w[0, Nb-1]) //Key Expansion
6 for round ==1 : Nr-1 then
7 Sub_Byte(State)
8 Shift_Rows(State)
9 Mix_Columns(State)
10 Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
11 end for

12 Sub_Byte(State)
13 Shift_Bytes(State)
14 Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1])
15 C1 = State
16 call(Cipher2)

Algorithm 2: UCDDES Algorithm for Encryption-Phase 2

Input: Cipher Text (C1), Nr: Total_Rounds, Nb: Constant value

Output: Cipher Text (C2)

1 if(C1 != NULL) then
2 For Cipher2 (C1, Secret_Key) then // UCDDES contains the Secret_key length of 32, 40 and 48
3 Byte_State [4, Nb]
4 for (State = C1)
5 Add_Round_Key (State, w[0, Nb-1]) //Key Expansion
6 for round ==1 : Nr-1 then
7 Sub_Byte(State)
8 Shift_Rows(State)
9 Mix_Columns(State)
10 Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
11 end for
12 Sub_Byte(State)
13 Shift_Bytes(State)
14 Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1])
Algorithm 3: UCDDES Algorithm for Decryption – Phase 1

**Input:** Cipher Text (C2), R: Random, Nr: Total_Rounds, Nb: Constant value

**Output:** Cipher Text (C1)

1. Compute: Block Size R[128, 192, 256-bit Plain Text]
2. For Decipher1 (C2, Secret_Key) then // UCDDES contains the Secret_key length of 32, 40 and 48
3. Byte_State [4, Nb]
4. for (State = C2)
5.   Add_Round_Key (State, w[0, Nb-1]) //Key Expansion
6.   for round ==1 : Nr-1 then
7.     Inv_Sub_Byte(State)
8.     Inv_Shift_Rows(State)
9.     Inv_Mix_Columns(State)
10.    Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
11. end for
12. Inv_Sub_Byte(State)
13. Inv_Shift_Bytes(State)
14. Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1])
15. C1 = State
16. call (Decipher2)

Algorithm 4: UCDDES Algorithm for Decryption – Phase 2

**Input:** Cipher Text (C1), R: Random, Nr: Total_Rounds, Nb: Constant value

**Output:** Plain Text (t1)

1. Compute: Block Size R[128, 192, 256-bit Plain Text]
2. For Decipher2 (C1, Secret_Key) then // UCDDES contains the Secret_key length of 32, 40 and 48
3. Byte_State [4, Nb]
4. for (State = C1)
5.   Add_Round_Key (State, w[0, Nb-1]) //Key Expansion
6.   for round ==1 : Nr-1 then
7.     Inv_Sub_Byte(State)
8.     Inv_Shift_Rows(State)
9.     Inv_Mix_Columns(State)
10.    Add_Round_Key(State, w[round *Nb, (Round+1)*Nb-1])
11. end for
12 Inv_Sub_Byte(State)
13 Inv_Shift_Bytes(State)
14 Add_Round_Key(State, w[Nr*Nb, (Nr+1) * Nb-1])
15 t1 = State
16 end

EXPLANATION OF THE ALGORITHM

1) Consider a block size as given below:

\[
\begin{align*}
R &\quad 128 \text{ bit Plaintext (4 words / 16 bytes)} \\
&\quad 192 \text{ bit Plaintext (4 words / 16 bytes)} \\
&\quad 256 \text{ bit Plaintext (4 words / 16 bytes)} \\
&\quad 128 \text{ bit Plaintext (4 words / 16 bytes)} \\
&\quad 192 \text{ bit Plaintext (4 words / 16 bytes)} \\
&\quad 256 \text{ bit Plaintext (4 words / 16 bytes)} \\
\end{align*}
\]

Where \( R \) denotes Random bit generation.

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

\[
\text{Number of rounds} = \text{[10 Rounds]} + \text{[10 Rounds]} + \text{[12 Rounds]} + \text{[14 Rounds]}
\]

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

4) Representations:

1. 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

\[
\begin{bmatrix}
4 & 8 & 12 \\
0 & 5 & 9 & 13 \\
2 & 6 & 10 & 142 \\
3 & 7 & 11 & 15 \\
\end{bmatrix}
\begin{bmatrix}
0 & 4 & 8 & 12 \\
5 & 9 & 13 \\
10 & 14 \\
3 & 7 & 11 & 15 \\
\end{bmatrix}
= 128 \text{ bits}
\]

1 Key table -> Key Expansion
2 Substitution of Bytes:
S Box input Array 1 -> State Array 1 -> S Box input Array 2 -> State Array 2 -> S' Box output Array.

4. Shift rows: S' Box output -> Input for Shift Box
   Shift Input array 1 -> Shift output Array 1 -> Input Array 2 -> Output Array 2

5. Transformation of Mix Column
   Shift Box Output -> Input for mix columns

6. Add round keys:
   It represents XOR operation with state array.
In the suggested procedure in figure 1, first the plain text is converted to cipher text 1 through the encryption process, and then next, ciphertext 1 is further converted to cipher text 2. Finally, this cipher text 2 is stored in a cloud. At another end, the cipher text 2 will be downloaded and decrypted to cipher text 1, and then it will be deciphered to obtain the plaintext.

V. Results and discussion

Every encryption technique will have pros and cons, so we have to select the apt cryptography algorithm for an application. The parameters like performance, weakness, and strength of the procedures are to be learned [26]. The encryption procedure has to be analyzed based upon several metrics, and compared are described below.

5.1. Encryption time - It is recognized as the conversion time from plain text to the ciphertext that is built on the key length, input block length, and mode. This time is taken in the unit of milliseconds. This time is important for performance, and if the time is less, then the system works fast.

5.2. Decryption time - It is known as the conversion time from ciphertext to the plaintext that is based on the key length, input block length, and mode. This time is taken in the unit of milliseconds. This time is important for performance, and if the time is less, then the system works fast.

5.3. Upload time – It is recognized as the time taken to store the data into the cloud system. This time is taken in the unit of milliseconds. This time is important for performance, and if the time is less, then the system works fast.

5.4. Download time – It is acknowledged as the time taken to retrieve the information from the cloud system. This time is taken in the unit of milliseconds. This time is important for performance, and if the time is less, then the system works fast.

![Fig. 2 Evaluation of time of Encryption among various algorithms](image-url)
In figure 2, the encode time among various encryption algorithms is compared. The RSA technique takes high encode time, and our UCDDES technique takes low encode time. When the input file size increases the encoding time also increases. The UCDDES encryption technique performs well among others.

![Graph comparing encode time](image)

**Fig. 3 Comparison of Decryption time among various algorithms**

In figure 3, the decode time of all the techniques is low when associated with the encryption algorithms. The RSA technique takes high decode time, and our UCDDES technique takes low decryption time. When the input file length increases the decode time also increases.

![Graph comparing decode time](image)

**Fig. 4 Comparison of Upload time among various algorithms**
The upload text time is compared among various algorithms. The figure 4 demonstrates that the upload time is high for RSA and low for UCDDES. The time increases together with the file length.

![Graph showing compare upload time for various algorithms](image)

**Fig.5 Comparison of Download time among various algorithms**

The text download time is compared to various algorithms. The download time is lower than the upload time. The figure 5 displays that the upload time is high for RSA and low for UCDDES. The time increases together with the file length.

**VI. Conclusion**

We introduced the UCDDES –Uncrackable Cipher Dynamic Double Encryption Standard, which helps to encode the input data. The encoding is completed in two stages the first stage encrypts the plaintext with the key, and the output 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 and decrypts the text with the keys given. Through this encoding technique the data integrity, data confidentiality, and data privacy are improved. This novel technique is utilized to prevent SQL injection, Brute force, Collision attack and dictionary attack. The encoding time and decryption time will be further reduced in the future work to achieve better performance.
References:

I. 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.

II. 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.

III. 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.

IV. 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.

V. Hur, J. (2013). Improving security and efficiency in attribute-based data sharing. *IEEE transactions on knowledge and data engineering*, 25(10), 2271-2282.

VI. Iyapparaja, M., Navaneethan, C., Meenatchi, S., Kumar, P. J., & Suganya, P. (2017). A Privacy-Preserving Secure Access Control Mechanism in Cloud.

VII. Kumar, K., & Lu, Y. H. (2010). Cloud computing for mobile users: Can offloading computation save energy?. *Computer*, 43(4), 51-56.

VIII. Mell, Peter, and Tim Grance. "The NIST definition of cloud computing.” (2011).

IX. 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*.

X. 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.

XI. Qiu, M., Gai, K., Thuraaisingh, 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.

XII. 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.
XIII. 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.

XIV. 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*.

XV. 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.

XVI. 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.

XVII. 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.

XVIII. 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.

XIX. 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.

XX. 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.

XXI. Wang, C., Ren, K., Lou, W., & Li, J. (2010). Toward publicly auditable secure cloud data storage services. *IEEE Network, 24*(4).

XXII. Whaiduzzaman, M., Sookhak, M., Gani, A., & Buyya, R. (2014). A survey on vehicular cloud computing. *Journal of Network and Computer Applications, 40*, 325-344.

XXIII. 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.

XXIV. 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.

XXV. 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.