International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34103-34108

DMUCE- A Confidentiality Enabled Technique To Improve Cloud User Security

[1]Dr.D.I.George Amalarethinam and [2]B.FathimaMary

[1]Associate Professor, Director-MCA, Jamal Mohamed College, Trichy, Tamilnadu, India
[2]Research Scholar, Bharathiar University, Coimbatore, Tamilnadu, India
[1]di_george@ymail.com, [2]fathimamary02@gmail.com

Abstract
Cloud Computing is a type of computing that is based on internet. It provides various hosting and delivering services over the internet. It provides the computational resources to user based on demand of user. The main service provided by the cloud is data storage. It provides enormous amount of space to store user data. Organizations and enterprises are adopting cloud platform which provides Storage as a Services (STaaS) to reduce their capital investment and maintenance of storage server. Organization and enterprises data are moved to the cloud which has to be kept in the public cloud storage. Data protection and data security are the important issues in the cloud storage. Whenever unencrypted data moves to the public cloud storage, there is a possibility to hack the data during data transmission. To overcome this issue, an efficient algorithm namely Dynamic Matrix Unique Characters Encryption (DMUCE) is proposed in this work. This paper proposes an algorithm named DMUCE to ensure the data security for cloud users before outsourcing data to cloud. The aim of the proposed work is to increase the data security and strengthening the cipher text. The proposed DMUCE algorithm is implemented in java and it gives better performance and efficiency when compared with existing Attribute Based Cryptography (ABC) technique and also it increases complexity of cipher text.

Keywords: Cloud Computing**,** Cloud Storage, CSP, STaaS, Security, Cryptography.

I. Introduction

Cloud Computing is the use and delivery of computing services over a network. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. The "cloud" is composed of hardware, storage, networks, interfaces, and services that provide the means through which users can access the infrastructures, computing power, applications, and services on demand which are independent of locations[1]. The concept of cloud computing provides three kinds of services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) all of which means a service oriented architecture. In that Infrastructure as a Service, *Server, Storage and Network* services are provided by the Cloud. The primary usage of cloud computing services is data storage. Cloud Storage provides environment to store their use data via Storage as a Services for cloud users. Because of these benefits each and every organization moves their data to the cloud.

Besides the cloud benefits, it has number of issues related to security. Security is the highest concern in the cloud environment. Once the user data sent to the cloud, cloud storage provider(CSP)only responsible for that user data. Because, cloud storage are maintained and controlled by CSP, Users do not have the rights to control and monitor the data and does not know about where the data is kept[2]. Cloud is a public environment, where there is lot of possibilities to attack and hack the user data. So there is a need to protect that data against unauthorized access. To protect the data, security of data in the cloud is accomplished by cryptography. Cryptography, in modern days is considered as combination of three types of algorithms. They are (1) Symmetric-key algorithms (2) Asymmetric-key algorithms (3) Hash function algorithms.

Cryptography is a technique used for encryption and decryption. There are several cryptography techniques available for encryption and decryption. Cryptographic techniques are classified into Conventional and Public key cryptography[3]. Conventional cryptography is also referred as symmetric key cryptography. The same key is used for encryption and decryption in symmetric key cryptography. Public key cryptography is called as asymmetric key cryptography. Public key and private key are used for encryption and decryption respectively. According to Tim Mather[4], symmetric encryption is more suitable to handle encryption at minimum time and efficient for large volumes of data in cloud storage.

Data Security is an important issue in Cloud Storage Environment. Data can be stored anywhere in any data centre and users don't have any rights to control and monitor the data. When a user moves their data to public cloud storage, data attack may also appear during transmission. Hence it is needed to propose and implement user side encryption to improve the data security. This paper proposes a confidentiality enabled technique to encrypt the data before being forwarded to cloud storage.

The content of this paper is organized as follows. The section II reviews with related works concerning the ways to improve the data security of public cloud storage. Section III describes the proposed methodology and section IV explains the proposed DMUCE algorithm for encryption and decryption process. Section V represents the illustration of the proposed algorithm and section VI presents an implementation and experimental results. It also discusses about the features of the

proposed algorithm. Section VII presents the conclusion of the proposed algorithm.

II. Literature Review

Sudhansu et al. [5] proposed two algorithms for Data Security. In the proposed model, RSA algorithm is used for both Encryption and secure Communication and MD5 hashing is used for digital signature and hiding key information. This paper does not describe how the key is shared between Data owner and Cloud Service Provider.

Asif et al. [6] proposed new encryption algorithm for Data security in the Cloud. The proposed hybrid approach uses a data compression method to reduce the size of original data and then encrypt the data using ASIF Encryption Algorithm. It reduces the size of data and requires less storage space because of Data Compression method. This Algorithm performs multiple rounds based on the length of the key. It generates a random key in each round and also selects the key randomly in each round to encrypt the data. Matrix techniques are used for encryption. But, this paper does not describe the key sharing technique.

Santosh et al. [7] proposed Partitioning technique for Cloud Storage Security. Third Party Auditor is responsible for partitioning the data, secret key generation for each partition, encrypt each partition using respective keys, sending partition at appropriate cloud server. RSA algorithm is used for encryption and decryption. This proposed technique has taken more time for encryption and decryption.

Manikondan et al. [8] proposed Arocrypt Symmetric encryption algorithm to make the cloud data secure. Plaintext is converted into ASCII values. A square matrix is formed based on the number of characters in the plaintext. The square matrix is divided into three matrices called upper (*UMAT*), diagonal (*DMAT*) and lower (*UMAT*) matrix. The matrices are encrypted by the different keys. Another square matrix is constructed based on encrypted value. But, the encryption is performed only at Cloud Service Provide side.

Sunitha rani et al. [9] proposed methodology for Data Security. This Methodology used three encryption algorithms to encrypt a Data. First, plaintext is encrypted by the ceaser cipher. Then the encrypted result is again encrypted by RSA substitution algorithm and finally encrypted result is again encrypted by the monoalphabetic substitution method. This technique has taken more time to encrypt the data by three algorithms one by one. Priya et al.[10] combined Ceaser cipher and Attribute Based Cryptography(ABC) and proposed symmetric algorithm to improve the data security at cloud data storage end. This methodology contains three level encryption to encrypt the data. So, this methodology takes more time to encrypt the data and concentrate only on cloud storage data.

In order to secure the data, the cryptography techniques are used. The user encounters lot of issues occurred in the cloud storage[11]. So the user needs to be aware of issues before going into storing the data in cloud storage. From the Review of literature, existing encryption is based on substitution, transposition and ASCII values. So it reduces the complexity of cipher text. Various cryptography algorithms are proposed for CSP encryption only[12][13]. But it needs to ensure the confidentiality of transmitting data over untrusted network. When a data owner transferring their unencrypted data to public cloud storage, there is possibility to hack and attack the data during transmission [14][15]. So, the data must be in encrypted form before moving to public cloud Storage. The literature review reveals that, sophisticated symmetric encryption algorithm is needed for cloud users.

III. Proposed Methodology

Outsourcing the data to the cloud provides lot of benefits to the organization and enterprises. Outsourced data should be in encrypted format in cloud storage. However, the data also should be in encrypted form before moving to cloud to protect the data during data transit. Normally, there are two ways to protect the data from unauthorized access. One is cloud user side encryption and another one is CSP's side encryption. This proposed methodology is based on cloud user side encryption as shown in figure 1. It ensures the confidentiality, and protects the data from unauthorized user during transit over unreliable network. DMUCE algorithm is proposed for users before outsourcing their data to cloud storage.

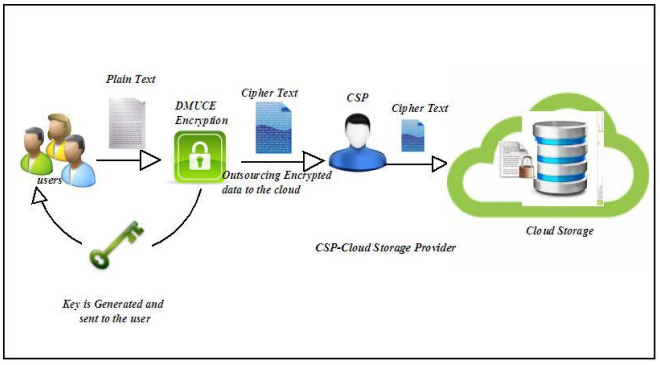

Figure 1. Proposed Methodology

4. DMUCE Algorithm

The proposed DMUCE algorithm improves classical symmetric encryption by integrating substitution cipher, transposition cipher and ASCII values. The existing encryption method uses substitution, transposition and ASCII values for corresponding alphabets. But the proposed algorithm uses neither ASCII values nor substitution and transposition of characters. DMUCE algorithm performs encryption in two levels. Initially the unique characters (UCS) are collected from the plain text. From the UCS Actual Character Set (ACS) is created. In first level, unique character positions are encrypted and in second level Actual Character Set (ACS) are encrypted. In first level encryption, Encrypted Base Matrix (EBM) of size mxn is created where m is the number of unique characters and n is the maximum number of appearances of unique characters. The size of the row is set to the number of characters in ACS and the size of the column is set to the maximum number of occurrences of any character in ACS. This Matrix is called as EBM. The remaining columns and rows values are set to 0.Tthe random integer is generated is called as Random Matrix (RM). EBM is encrypted by performing the XOR operation with RM. XOR operation is

performed only for non zero elements in the corresponding EBM. The resultant matrix Encrypted Matrix(EMAT) is more complicated by doing matrix transpose technique. The transpose of the EMAT is called as $EMat^{T.}$ Finally to improve the efficiency of DMUCE, sparse matrix concept is used. Sparse matrices are specialized data structure, which allows to store only non zero elements and save a lot of memory and CPU time when working with such matrices. There are several techniques used in sparse matrix. In that Compressed Row Storage (CRS) techniques are used in DMUCE. CRS representation is good for numerical work. In DMUCE, non zero elements, column indices and row pointer are maintained according to the CRS algorithm. Finally, the non zero encrypted elements are maintained. This CRS matrix is called as $EMat^{S}$.

In second level encryption, another random integer is generated and called as Random Set (RS). ACS is encrypted by performing the XOR operation with RS. The resultant character set is called Encrypted Character Set (ECS). It will make the ACS more complicated and is very much useful for hiding the original character involved in the plain text. Finally, content of ECS and $EMat^{S}$ are considered as a cipher text. The secret key is having a main role in encryption and decryption. In DMCE, random value is generated for both level of encryption. The random values are merged with the text to keep the key secret well. Algorithm 1 shows the DMUCE Encryption steps and the Algorithm 2 shows the DMUCE Decryption steps. Figure 2 and Figure 3 show the architecture of encryption and decryption algorithm. The accuracy of proposed algorithm is evaluated with sample text as explained in Section 5.

ALGORITHM 1 DMUCE ENCRYPTION ALGORITHM

1. Create UCS from file
2. Create EBM for size m x n where m is Number of Unique Characters and n is Maximum Number of appearance of any character.
3. EBMij=Position(c) $\forall$ c $\in$ UCS
4. Create RM using random Integer
5. EMat=EBM XOR RM except 0 values in EBM
6. $EMat^{T}$=Transpose(EMat)
7. $EMat^{S}$=SparseMatrix($EMat^{T}$)
8. Create ACS from UCS
9. Create RS using random character
10. ECS=ACS XOR RS
11. SK Generated with RM base and RS base with random string

Where
UCS stands Unique Character Set
EBM stands Encryption Base Matrix
RM stands Random Matrix
EMat stands Encrypted Matrix
ACS stands Actual Character Set
RS stands Random Set
ECS stands Encrypted Character Set
SK stands Secret Key

ALGORITHM 2 DMUCE DECRYPTION ALGORITHM

1. RSBV<=Extract from SK
2. RMBV<=Extract from SK
3. Create RS from RSBV
4. Create RM from RMBV
5. ECS<=Read from File
6. $EMat^{S}$<= Read from File
7. $EMat^{T}$=Generate($EMat^{S}$)
8. ACS=ECS XOR RS
9. EMat=Transpose($EMat^{T}$)
10. EBM =EMat xor RM except 0 values in EMat
11. Create Heap DH for size m x n
12. DH_{ij}<= ACS_i using EBM_{ij}

Where
RSBV stands Random Set Base Value
RMBV stands Random Matrix Base Value
DH stands Decrypted Heap

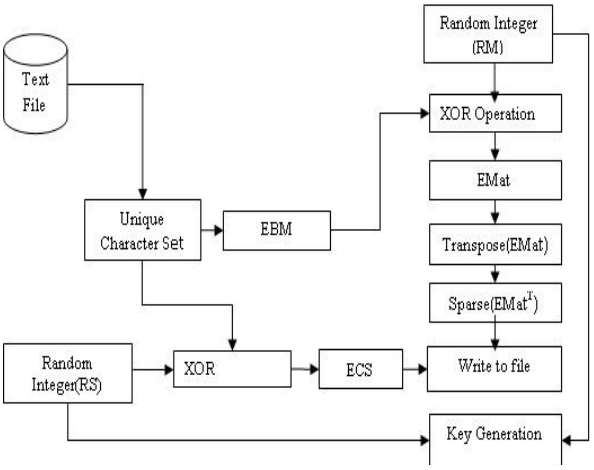

Figure 2. Architecture of proposed DMUCE Encryption Algorithm

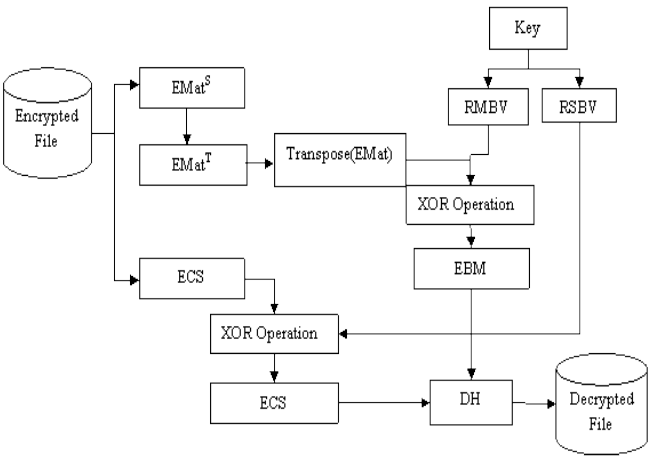

Figure 3. Architecture of proposed DMUCE Decryption Algorithm

5. Illustration of working of the algorithm

The proposed algorithm is tested with the sample text. The encryption and decryption process is given below.

Encryption process

Figure 4 shows the over all process of encryption. The Plain Text is "**coffee**". The unique characters are "**cofe**". The UCS and Positions are taken from the plain text shows in figure a. Based on UCS and its position, Dynamic matrix is constructed in 4X2 size as shown in figure b. Random number is generated as 6319. It performs XOR operation with EBM and the resultant matrix is shown in figure c. The transpose of the matrix is given in figure d. CRS technique is applied on the matrix to store the non-zero elements. Compressing the matrix by using CRS, $EMat^S$ are shown in figure e. To encrypt the unique characters, another random number is generated as 962. It performs XOR operation with unique characters and encrypted character set is shown in figure f. Finally, $EMat^S$ and ECS are considered as cipher text is shown in figure g. Based on the random number generation, the secret key is generated as "7858-6142-1359-8565-6896-12461-2858-2854-1000-5000".

Decryption Process

Figure 5 shows the overall process of decryption. The decryption process is reverse of the encryption process. The secret key will be given as input to the decryption process. From the cipher text, matrix non zero values $EMat^S$ and ECS are extracted is shown in figure b. Based on the non zero values appeared in $EMat^S$ and their column indices, $(EMat)^T$ is extracted as given in figure c. The transpose of the $(EMat)^T$ is shown in figure d. It performs XOR operation with already generated random number in encryption process and the generated character position matrix is shown in figure e. ECS performs XOR operation with already generated another random number and produces the unique characters as shown in figure f. Based on the unique characters and its position matrix, plain text is generated as "coffee".

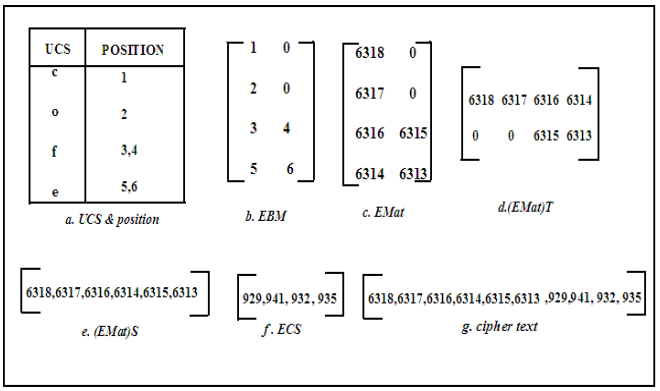

Figure 4. Work Flow of Encryption Process

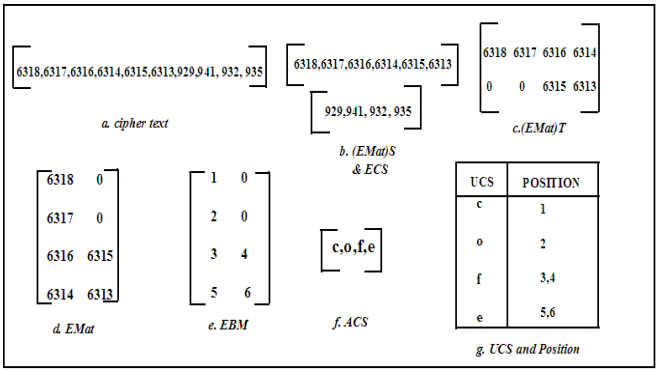

Figure 5. Work Flow of Decryption Process

6. Experimental Results and Discussion

The proposed DMUCE algorithm is implemented in java. The screen shot of this encryption and decryption process are shown in figure 6 and figure 7. The time taken for encryption and decryption of various size of file are measured. Encryption and decryption time of the proposed DMUCE algorithm is given in table 2.

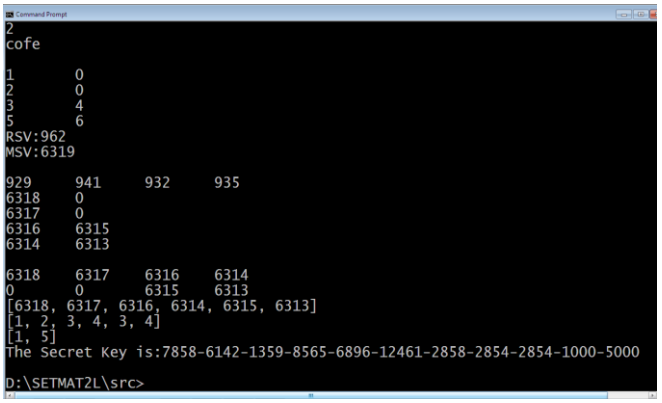

Figure 6. Encryption using DMUCE algorithm

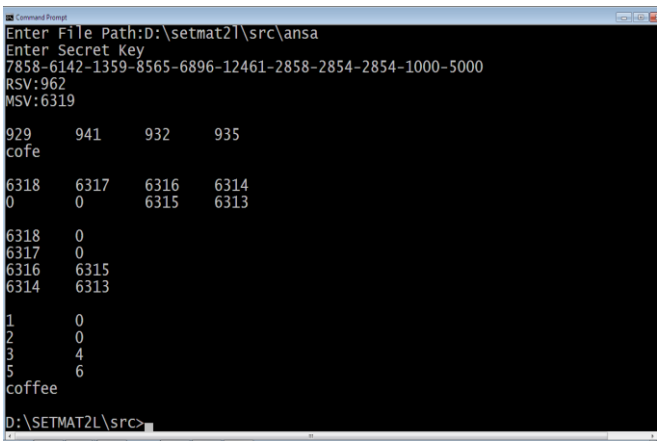

Figure 7. Decryption using DMUCE algorithm

The performance and security level of proposed DMUCE algorithm is compared with the existing Attribute Based Cryptography (ABC) technique [10]. The time taken for

encryption and decryption are calculated in nanoseconds (ns) and shown in Table 2 and Table 3 respectively. In Table 2, when the file size is 32 bytes, ABC technique produces the encryption time as 6246751 ns whereas the proposed DMUCE algorithm produces in 4300925 ns. Similarly when the file size is 160 bytes, ABC technique gives the encryption time as 22139443 ns and the proposed DMUCE algorithm takes 5177805 ns. In Table 3, when the file size is 32 bytes, ABC technique produces the decryption time as 25537781 ns whereas the proposed DMUCE algorithm produces in 9174416 ns. Similarly when the file size is 160 bytes, ABC technique gives the decryption time as 63898766 ns whereas the proposed DMUCE algorithm takes 18858995 ns. Figure 8 and Figure 9 shows the graphical representation of ABC and proposed DMUCE. Time taken for ABC and proposed DMUCE algorithm are calculated for different sizes of data. The result shows that the proposed DMUCE algorithm has taken minimum time duration for encrypting and decrypting the data of different sizes when compared to the ABC technique. From this analysis, proposed DMUCE algorithm gets minimum time for encryption and decryption than ABC technique.

Table2.Encryption time calculation with different File Size

File Size(Bytes)	ABC	Proposed DMUCE
32	6246751	4300925
64	14248046	4493401
96	19379122	4503126
128	20127895	4973983
160	22139443	5177805

Table3.Decryption time calculation with different File Size

File Size(Bytes)	ABC	Proposed DMUCE
32	25537781	9174416
64	36296830	9651351
96	36991265	16569139
128	62437562	18574130
160	63898766	18858995

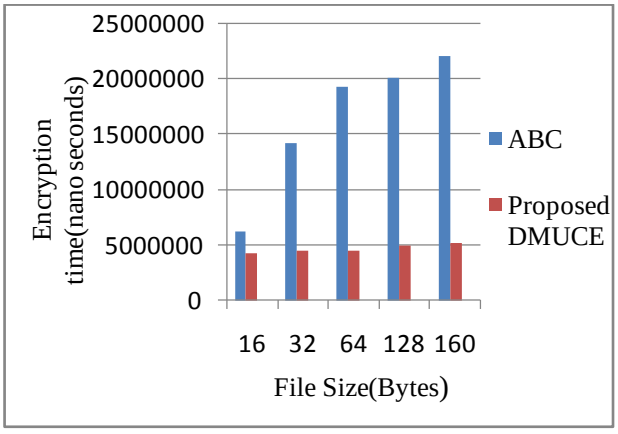

Figure 8. Comparison of Encryption time

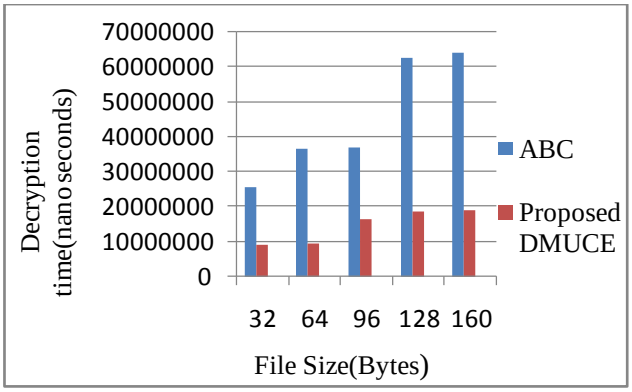

Figure 9. Comparison of Decryption time

It takes minimum time to generate dynamic matrix and unique character position values are used instead of using ASCII values. Encryption is only for the unique characters and their positions. It enhances the security of the cipher text and makes more complication. DMUCE algorithm also increases the security by using random values for encryption. The proposed DMUCE algorithm encrypts unique characters only, so the value of the cipher text cannot be repeated even if the character is repeated more than once.

The features of the DMUCE algorithm are

1. Matrix is generated dynamically at run time based on the unique characters.
2. Increases the complexity of cipher text such as character to number conversion using position value instead of using ASCII.
3. Performs encryption in two levels.
4. Encryption is only for unique characters.
5. Matrix size is changed based on unique characters appeared in the plain text.
6. Random number generation increases the security of the cipher text.
7. It improves the efficiency by using CRS technique.

The proposed DMUCE algorithm is applied on both numeric and non numeric data. It can protect the data from unauthorized access during data transmission to the cloud. It ensures the confidentiality of cloud user data. Instead of trusting unreliable CSP, cloud users can encrypt their own data by using this proposed algorithm and send their encrypted data to the cloud without any issue. The appropriate key is also maintained by cloud user itself. So, no need to worry about the encrypted data when it moves to the cloud.

7. Conclusion

Cloud Storage as a service (STaas) provides cost-effective services to individual users as well as organization. It provides enormous amount of space with nominal cost. But, data security plays a vital role in Cloud. Whenever user moves their data to the cloud, there are many possibilities to attack the data during transit. Cryptographic techniques are providing security to the data. So encryption is needed for data before outsourcing their data to the cloud storage. This paper proposed a symmetric encryption algorithm to protect the data from unauthorized access during data transmission. So, this proposed DMUCE algorithm enhances the

confidentiality of data. This algorithm is used in order to encrypt the user data before moving to cloud. From the above sample working, it is clearly shown that the sample text having 6 characters but in the proposed algorithm only 4 characters are taken for encryption. In DMUCE algorithm, there is no repetition of values in cipher text. So, it improves the complexity of cipher text. The matrix size is generated dynamically and sparse matrix techniques are used to increase the efficiency of this proposed algorithm. By applying this proposed DMUCE algorithm, data owner can move their encrypted data to the cloud storage without affecting their original data during data transmission.

Reference

[1]. RashmiNigoti, ManojJhuria andDr.Shailendra Singh, "A Survey of Cryptographic Algorithms for Cloud Computing", International Journal of Emerging Technologies in Computational and Applied Sciences, Vol4, pp. 141-146, 2013.

[2]. Kelsey Rauber, "Cloud Cryptography", International Journal of Pure and Applied Mathematics, Vol. 85, pp. 1-11, 2013.

[3]. Ebtesam A. Alomari, Muhammad M. Monowar, " A Survey of Security Issues for Data Sharing over UntrustedCloud Providers", Journal of Emerging Trends in Computing and Information Sciences, Vol.5, pp.609-619, 2014.

[4]. Mather T., Kumaraswamy S. and Shahed, L., "Cloud security and privacy", Chapter 4, O'Reilly Media, Inc, pp. 61-71, 2009.

[5]. Sudhansu Ranjan Lenka, Biswaranjan Nayak, "Enhancing Data Security in Cloud using RSA Encryption and MD5 Algorithm", International Journal of Computer Science Trends and Technology, Vol.2, pp. 60-64, 2014.

[6]. Md Asif Mushtaque, Harsh Dhiman, Shahnawaz Hussain, " A Hybrid Approach and Implementation of a NewEncryption Algorithm for Data Security in CloudComputing", International Research Publication House, Vol.7, pp.669-675, 2014.

[7]. Santosh Jogade, Ravi Sharma, Rajani Kadam, " Partitioning Data and Domain Integrity Checking for Storage - Improving Cloud Storage Security Using Data Partitioning Technique", International Journal of Emerging Research in Management &Technology, Vol.3, pp.133-138, 2014.

[8]. S. Monikandan, Dr.L.Arokiam, "Arocrypt: A Confidentiality Technique For Securing Enterprise's Data In Cloud", International Journal of Engineering and Technology, Vol.7, pp.245-253, 2015.

[9]. Sunita Rani, AmbrishGangal "Cloud Security with Encryption using Hybrid Algorithm and Secured Endpoints", International Journal of Computer Science and Information Technologies, Vol.3, pp.4302–4304, 2012.

[10]. Aayushi Priya, Y.K. Rana, B.P. Patel, " Design and Implementation of an Algorithm to Enhance Cloud Security", International Journal of Computer Applications, Vol.113, pp.41-47, 2015.

[11]. Balajee Maram, K. Lakshmana Rao, Y. Ramesh Kumar, "Encryption and Decryption algorithm using 2-D Matrices", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 4, April 2013.

[12]. Dinesh P. Baviskar, Sidhhant N. Patil, Onkar K. Pawar, "Android Based Message Encyption/Decryption Using Matrix", IJRET, Volume 4, Issue 1, Jan 2015.

[13]. M. Yamuna, S. Ravi Rohith, Pramoth Mazumdar, Avani Gupta, "Text Encryption Using Matrices", IJAIEM, Volume 2, Issue 3, March 2013.

[14]. V.Masthanamma, G.Lakshmi Preya, " An Efficient Data Security in Cloud Computing Using the RSA Encryption ProcessAlgorithm", International Journal of Innovative Research in Science, Engineering and Technology, Vol.4, pp.1441-1445, 2015.

[15]. S. Arul Oli, Dr.L.Arokiam, " A Novel Approach for Ensuring Data Confidentiality in Public Cloud Storage", International Journal of Computer Applications, Vol.0, pp.1-5, 2014.