Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 78 (2015) 29535-29539

ISSN: 2229-712X

AROMONcrypt: A technique to optimize the security by ensuring the confidentiality of outsourced data in cloud storage

L. Arockiam[1] and S. Monikandan[2]

[1]Department of Computer Science, St. Joseph's College (Autonomous), Trichy, India.

[2]Department of Computer Science, Manonmaniam Sundaranar University, Tirunelveli, India.

ARTICLE INFO

Article history:
Received: 6 November 2014;
Received in revised form:
21 December 2014;
Accepted: 1 January 2015;

Keywords
Cloud Security,
Confidentiality,
SEaaS, Encryption,
Obfuscation, Cloud Storage.

ABSTRACT

Cloud delivers computing resources as a service. Finest usage of cloud services is used to store information in the cloud servers. Pool of virtual servers is configured to store the users' data with low cost. Cloud offers more advantages to the cloud users. Apart from the advantages and benefits of cloud, it has more security issues and vulnerability on the data stored in the cloud. This paper proposes asecurity service named AROMONcrypt to address the security issues in cloud storage. This security service is used to secure the data in cloud storage. AROMONcrypt uses two types of techniques to ensure the confidentiality of data namely encryption and obfuscation. This paper also describes Security as a Service (SEaaS). SEaaS provides AROMONcrypt security service from the Cloud Service Provider (CSP). Simulation is conducted with AROcrypt and MONcrypt security services considering time and security level. AROMONcrypt provides optimum security, and MONcrypt has taken minimum time.

Introduction

Cloud evolvesas a computing infrastructure that provides quick delivery of computing resources as a utility services in dynamically scalable, virtualized manner [1] [2]. Cloud computing provides more sophisticated data storage [3]. Cloud delivers unimaginable Virtual Storage (VS)[4].Cloud resources are accessed at anytime, anywhere and any number of times. Users could store data in VS by using services like SaaS, PaaS and IaaS.

Cloud is a public environment. So, data stored in the cloud might mingle with other users' data. It leads to the data security issues [5]. Security is the main reason that enterprises are hesitating to move cloud computing. Once the security issue is rectified thoroughly then, the cloud will be the top business in IT industries. To address this security issues, this paper proposes a security service algorithm for cloud storage using encryption [6][7] and obfuscation[8][9] techniques. Encryption is the procedure of transforming the readable text into unreadable using an algorithm and a key. Obfuscation is same like encryption. Obfuscation is a method that masks illegitimate users by implementing a particular mathematical function or using programming methods.

Encryption procedures guarantee confidentiality. Encryption is not sufficient for public cloud storage environment. Encryption is integrated with obfuscation technique. Obfuscation alone is also not enough for complete protection of data in cloud storage because attackers could find actual data through reverse engineering or by using brute force method, which may compromise cloud data security[10].

This paper uses encryption and obfuscation techniques to safeguard the data from the attackers (insiders and outsiders). According to the proposed AROMON crypt, the data ready to upload to the cloud storage is analyzed to determine the type of the data like numeric and non-numeric. Once the type of the data is identified then based on the type, data are encrypted and obfuscated. Non-numerical data are encrypted, and numerical data are obfuscated. The cloud user keeps the key and other details used for AROMONcrypt.

Related Work

This section describes someof the obfuscation and encryption technique in the literature. Robertson summarizes some obfuscation techniques in [9] are listed below.

Base64 Encoding

Base64 is used when there is anecessity to convert binary data that need to be kept and moved over media that are designed to deal with textual data. It is to ensure that the data remain intact without modification during transport.

Base32 Encoding

The Base 32 encoding is considered to denote and omsequences of octets in a form that needs to be case insensitive, but that need not be human readable. The encoding process represents 40 bit groups of input bits as output strings of eight encoded characters. It arranges from left to right; a 40-bit input group is formed by concatenating five 8bit input groups. These 40 bits are then treated as eight concatenated 5-bit groups, each of which is transformed into a single character.Base32 encoding is ordered the most significant bit at the first level.

ASCII Encoding

ASCII is a character encoding scheme originally based on the English alphabet. ASCII encoding is the process of representing characters with base2 (binary) strings. These strings can then be further converted to other formats as needed (hexadecimal, base64, etc.).

Hexadecimal Encoding

Hexadecimal is a positional notation scheme with a base of 16. It uses sixteen discrete symbols that are 0 to 9 and A, B, C, D, E, F to denote values from ten to fifteen. For example, the number 2AF3 is equal to 10995.

Tele:
E-mail addresses: moni.tamil@gmail.com

XOR Encoding

XOR is one type of the bitwise operation used to (among other things) obfuscate data. A bitwise operation workson one or more bit arrangements or binary numerals at the level of individual bits. A bitwise XOR considers two bits with the same length. The result is 1 if anyone bit is 1 or theresult is zero if both are 0, or both are 1.

Recently, obfuscation is focused to maintaining the security of data in the cloud storage. Obfuscation alone could not provide maximum protection to data in the cloud storage. Hence, encryption is also integrated with obfuscation to providea real security to cloud data.

Sunitharani et al. [11] proposed an encryption methodcall hybrid algorithm in order to providesecurity in the cloud. The proposed methodology is used three algorithms sequentially to encrypt a message. First, plaintext is encrypted by the ceaser cipher then the encrypted result is once again encrypts using RSA substitution process and finally the result is again encrypted by the monoalphabetic substitution method**.** This technique has taken more time to encrypt the text by three algorithms one by one.

Subhasri P. et al. [12] proposed a Multi-level Encryption algorithm to secure the data in the cloud. The proposed algorithm uses rail fence and ceaser cipher algorithm. Initially, plaintext is encrypted using rail fence method. Assign the position value i to each letter in the encrypted text. Generate the ASCII values of each character. Assign a key and apply it on the text using the formula: E = (p + k + i) % 256, where p denotes Plaintext, k denotes key and i denotes Position. Algorithm produces the ASCII character of the equivalent decimal value. Key used for encryption is not generated. Maintain the position of each character in the text requiring additional storage. Here, Author has not mentioned where the characters position details are maintained.

Motivation and Objective

From the literature study, it is come to know that, there are vast numbers of works undergoing to secure the data in thecloud.This effort is motivated by the following reasons,

≈ To provide security as a service to theuser from the CSP

≈ To ensure the data in the cloud is only accessed by the data owner.

≈ To obfuscate important numerical data.

≈ To reduce thetime taken for AROMONcrypt by executes the algorithms in aparallel manner.

The objective of this paper is to propose a security service algorithm to optimize the security by ensuring the confidentiality of data in cloud storage. To maximize the performance, security level and to reduce data size uploaded to the cloud storage.

Security as a Service

Cloud computing provides Anything as a Service (XaaS). SEaaS is one of the services provided by the Cloud Service Provider (CSP). SEaaS provides different cryptography algorithm and obfuscation based technique for securing the data in the cloud. Figure 1 shows the cloud environment with CSP, which has SEaaS as one of its services. CSPs provide many services like SaaS, PaaS and IaaS. Security is also provided as a service. In figure 1, CSP_1 provides SEaaS and also provides other services. Here, CSP_1 is used only for security service, and not for storing the data. Users could store data with other CSPs who provide storage as a service.

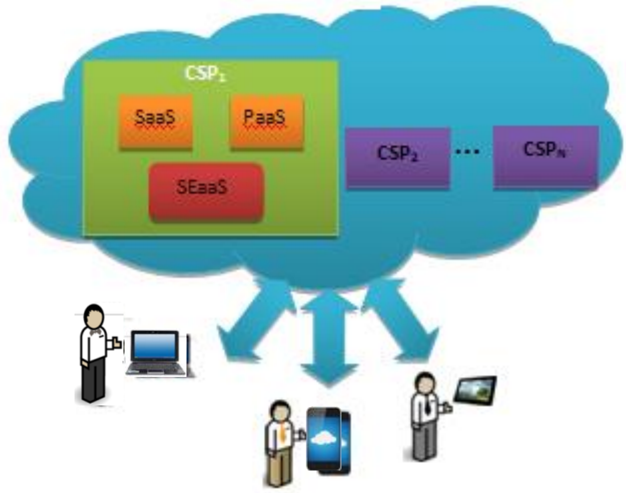

Figure 1. CSP with Security as a Service (SEaaS)

SEaaS contains three algorithms namely AROcrypt, MONcrypt and AROMONcrypt. These algorithms are particularly used for a particular type of data. If users want to upload sensitive data to the cloud, they may use any one of the algorithms from SEaaS. Figure 2 represents the security service algorithms provided by SEaaS in a CSP. AROcrypt algorithm is used to encrypt non-numerical data only. MONcrypt is used to obfuscate numerical data only. AROMONcrypt is usedto encrypt and obfuscate the numerical and non-numerical data. This paper only describes AROMONcrypt security service algorithm in SEaaS.

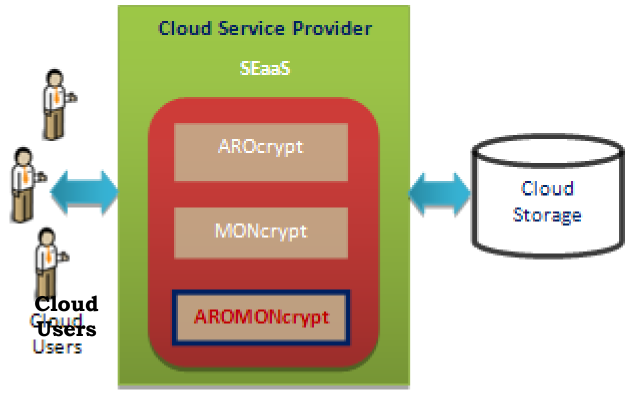

Figure 2. Security service algorithms provided by SEaaS

Methodology

AROMONcrypt is one of the security service algorithms in SEaaS. SEaaS publishes the list of security service algorithms to the users when a storage request is made by the users. Users chooseAROMONcrypt security service algorithm in SEaaS. SEaaS sends an executable file for the selected security algorithm AROMONcrypt. SEaaS instructs the Key Generation and Management as a Service (KGMaaS) to provide keys to the users who choose AROMONcrypt in SEaaS. SEaaS sends the details of selected security algorithm and user related information to KGMaaS.

KGMaaS generates keys suitable for the AROMONcrypt security service algorithm by the users. Keys are directly communicated to the users, not through CSP of SEaaS.Users submit the data along with keys to the AROMONcrypt security service algorithm to encrypt or obfuscate the data. Once the data are encrypted or obfuscated, they are uploaded to the cloud storage in another CSP. The key values used in encryption and obfuscation are maintained at the users' side. To retrieve the data from the cloud storage, decryption and deobfuscation procedures are used. The cipher text is converted into original plain text at users' side.

Proposed Work

AROMONcrypt is proposed to encrypt and obfuscate the non-numerical and numerical data. AROMONcrypt security service algorithm is based on encryption and obfuscation. Both encryption and obfuscation are applied on the data simultaneously. It is one of the security service algorithms in SEaaS. Once the users submit the data to the AROMONcrypt algorithm, it analyses type of the user's data.

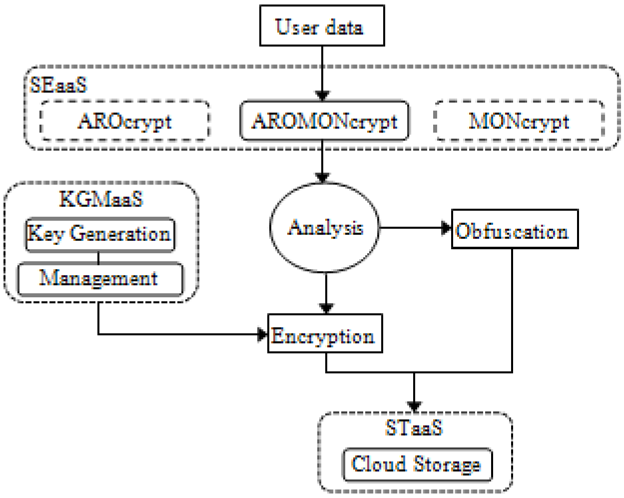

Figure 3. Data stored in the cloud using AROMONcrypt Security Service

AROMONcrypt starts encrypting the non-numerical data and obfuscating the numerical data. Keys for encryption are received from KGMaaS. The users keep the keys for decryption and deobfuscation. CSPs of SEaaS or STaaS have not aware of the keys used for security service algorithm. AROMONcrypt security service algorithm protects the cloud user's data from inside and outside attacks. The Figure 3 shows that the data storage procedure of AROMONcrypt in SEaaS.

Algorithm#1 is the AROMONcrypt security service algorithm. Initially, it analyses to findsthe type of data (T), based on the data type, data is encryptedand obfuscated. If the data (T) are numeric, then the data are obfuscatedusing MONcrypt, if the data are non-numeric or special symbols, thenthe data are encrypted using AROcrypt.

Algorithm #1: AROMONcrypt Security Service Algorithm

1. start
2. T=plaintext
3. for i=1 to sizeof(T) then
4. if (isdigits(T(i))) then
num=num+T(i)
5. else
nonnum=nonnum+T(i)
6. end if
7. end for
8. Thread.encryption_text(nonnum)
9. Thread.obfuscation_digit(num)
10. End

In the algorithm, *T* represents the whole data uploaded to the cloud storage. AROMONcrypt is analysis the data. If it digit, data is concatenated into *num = num+T(i)*, otherwise the data is concatenated into *nonnum=nonnum+T(i)*. Once the data analysis is completed, the AROcrypt algorithm is called with the argument of *nonnum,Thread.encryption_text(nonnum)*. The MONcrypt algorithm is called with the argument of *num*, *Thread.obfuscation_digit(num)*. Both algorithms are called and executed simultaneously.

Algorithm #2: AROcrypt Security Service Algorithm

1. encryption_text(T)
2. start
3. Convert (T) into ASCII code
4. N= count (T)
// N- number of characters in T
//form the matrix for N character, maximum size of matrix is 25X25, if N>625 than divide the T into 625 character blocks and form matrix for each block.
5. matc =N/625
6. if matc>0
fori=1 to matc
Divide the T into 625 blocks, $N=n_1, n_2, n_3...n_n$ // n_1, n_2, n_3 are each individual matrix
end loop
end if
7. for p=1 to matc
8. Based on the value of N, form a square matrix MAT [MXM] > N,M=M
9. Apply T into the matrix from left to right
10. Divide the Matrix MAT into three matrices called UMAT, DMAT, LMAT
11. Read the Text T by UMAT(U), DMAT(D) and LMAT(L)matrix
12. Get three random integer number as KEYS K_1, K_2, K_3 for each matrix.
13. Apply the key K_1, K_2, K_3 for U, D, L to get first encrypted data
// [U-K_1, D-K_2, L-K_3]
14. Arrange the encrypted text into another matrix MAT_1 [MXM] based on number character in the key K_4
// Get random string value as a key K_4
15. Read the matrix MAT_1 column by column in order of key K_4
16. Resultant text from step 15 is converted to ASCII character code(C)
//C-Cipher Text
17. *end loop*
18. *end*

Algorithm#2 shows the AROcrypt security service algorithm. It is a symmetric encryption algorithm. It uses four keys for encryption and the same keys are used for decryption. The given plain text characters are converted into ASCII values. A square matrix is formed based on the number of characters in the plaintext. Maximum size of the matrix is 25_X25. The square matrix is divided into three matrices called upper (*UMAT*), diagonal (*DMAT*) and lower (*UMAT*) matrix. Apply the encryption to the matrices *UMAT*, *DMAT* and *UMAT* individually by using the keys K_1, K_2, K_3 respectively. Another square matrix is constructed with an encrypted value. Now the text is read column by column. Order of understanding the column is based on order of Key K_4. Finally, the ASCII code values are converted into character value. This value is ciphertext[13].

Algorithm #3: MONcrypt Security Service Algorithm

1. obfuscation_digits(T)
2. start
3. T=array of plaintext
4. n=sizeof (T)
5. for i=1 to n
RD(i)=pow(T(i),2)
count(i)=RD(i)/256
RD(i)=RD(i)%256
//count has no. of time RD divided by 256
6. end for
7. for i=1 to n
ab(i)=0
bw(i)=0

while RD(i)<32||RD(i)>126
if (RD(i)<32) then
bw(i)=bw(i)+1
//bw is count no. of time this if is executed
RD(i)=RD(i)+32
end if
i f (RD(i)>126)
ab(i)=ab(i)+1
//ab is count no. of time this if is executed
RD(i)=RD(i)-126
end if
end while
8. *end for*
9. *Convert the RD into ASCII code C*
10. *C = array of cipher Text*
11. *end*

Algorithm#3 represents the obfuscation process. It is invoked when the user uploads data to the cloud storage. Initially, the plain text values are arranged as an array of integer values T=T(1),t(2),T(3),…T(n). The mathematical function pow() is applied on the T(i). The pow() function is used to find square value T(i)(RD=pow(T(i),2)). RD(i) value is divided by 256 to find the count(i) value and also to get the reminder value in RD(i). The count(i) represents the number of times the RD(i) is divided by 256. It checks whether the RD(i) value is in the range of 32 to 126. If it does not, then calculate the RD(i) value to get range of value in between 32 to 126. Finally, get the obfuscated text by converting the RD(i) into ASCII character value. User's data are forwarded after completion of MONcrypt.

The count(i) value is maintained in the user side. This value is used as a key for deobfuscation. In MONcrypt security service algorithm, each plaintext value is converted into 8 bit character value. MONcrypt compresses the size of plaintext.

Simulation Results

The proposed AROMONcrypt security service and key generation is implemented in JAVA. Simulation is performed in the cloud environment (Amazon EC2). The cloud user machine connected to the cloud server has the configuration of Windows Operating System with core i3 Intel processor and 4GB RAM. The user data are encrypted before it is uploaded and decrypted when it is retrieved from the cloud. Thus, the encryption is done in the user machine connected to the cloud. Time taken for encryption and decryption is calculated in the user machine.

Amazon Elastic Compute Cloud (EC2) server is used for cloud storage. Key generation and AROMONcrypt security service are developed as web service and hosted in the Amazon server. The Amazon micro instance has the following configuration as Microsoft windows server 2008 Base 32 bit operating system, 2.5 GHz Intel Xeon processor, 613 MB RAM, 30GB of EBS (Elastic Block Storage). The users upload the data via user interface. Once the data are submitted, then they are encrypted and uploaded to the Amazon micro instance server. Security levels of the existing and proposed techniques are computed in Amazon server.

Security level is analyzed by using a security analysis tool called Hackman. This tool analyses the security level of three security service algorithm. This tool is installed in Amazon server for analyzing the security level. Hackman attacks the encrypted and obfuscated text in the Amazon server. It uses different attacks like dictionary attack, brute force attack, etc. to retrieve the original text. Map the plain text with retrieved text to find the percentage of original text retrieval. Based on percentage mapping, security level of the AROMONcrypt is estimated. In the same way, AROcrypt and MONcrypt security level are calculated.

Performance and security level of proposed AROMONcrypt security service algorithm is compared with AROcrypt and MONcrypt. Time taken for encryption and decryption is shown in Table 1 and Table 2 respectively. Security levels of AROMONcrypt, AROcrypt and MONcryptare compared, and results shown in Table 3. Simulation is conducted for different sizes of data. For each size of data, time taken for encryption and obfuscation, decryption and deobfuscation, and security level are noted and evaluated. Performance of proposed technique is measured by the time taken to complete encryption and obfuscation, decryption and deobfuscationprocess.

The Table 1 and Figure 4 represent the performance comparison of AROcrypt, MONcrypt and AROMONcrypt. The time taken by three security service algorithmsis calculated for different sizes of data.

The results show that compared to the AROcrypt and AROMONcrypt,MONcrypt has takenlesser time duration for different sizes of data. AROMONcrypt has takenlesser time duration than AROcrypt.

Table 1. Performance Comparison based on Encryption and Obfuscation Time (Milliseconds)

Size	AROcrypt	MONcrypt	AROMONcrypt
1 MB	282	31	238
2 MB	468	62	412
3 MB	656	93	609
4 MB	889	125	821
5 MB	1102	156	1057
10 MB	2253	328	1197
15 MB	3388	484	3317

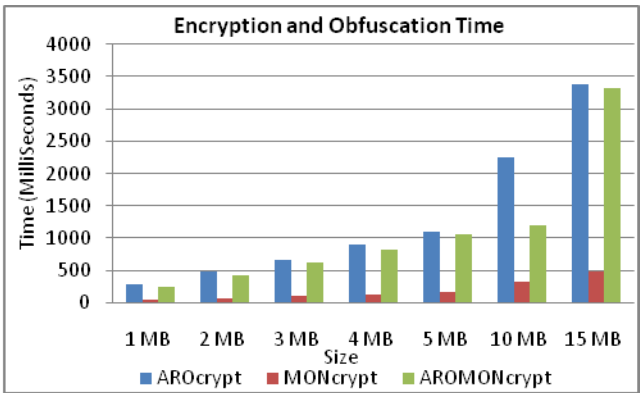

Figure 4. Performance Comparison based on Encryption and Obfuscation Time (Milliseconds)

The Table 2 and Figure 5 represent the performance comparison of decryption of AROcrypt, MONcrypt and AROMONcrypt. The time taken by the AROcrypt, MONcrypt and AROMONcrypt algorithms are calculated for different size of data.

The results show that compared to the AROcrypt and AROMONcrypt,MONcrypt has takenlesser time duration for different sizes of data. AROMONcrypt has takenlesser time duration than AROcrypt.

The Table 3 and Figure 6 represent the comparison of security level. The result shows that compared to the AROcrypt and MONcrypt algorithms, the proposed AROMONcrypt algorithm possess maximum level of security.

Table 2. Performance Comparison based on Decryption and Deobfuscation Time (Milliseconds)

Size	AROcrypt	MONcrypt	AROMONcrypt
1 MB	235	16	217
2 MB	438	31	398
3 MB	627	47	587
4 MB	843	51	789
5 MB	1069	63	996
10 MB	2216	109	2102
15 MB	3341	124	3217

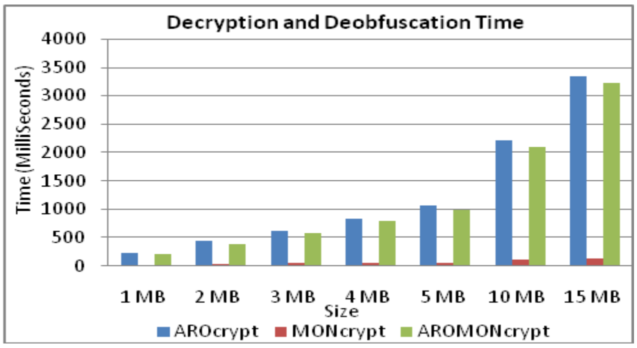

Figure 5. Performance Comparison based on Decryption And Deobfuscation Time

Table 3. Comparison of Security Level

S.No	Algorithm	Security Level(%)
1.	AROcrypt	89
2.	MONcrypt	87
3.	AROMONcrypt	90.5

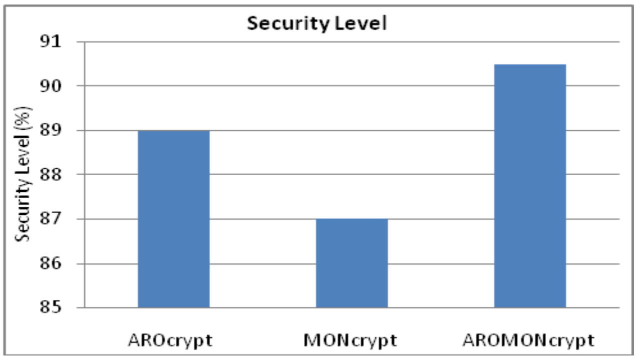

Table 6: Comparison of Security Level

By the observation of the above results, it is obvious that the integration of encryption and obfuscation gives more security to the data stored in the cloud. If the hackers try to find the logic, they apply that logicon the cloud data to get the complete original data. However, in case of proposed technique,the same logic is not applicable forcomplete data. So hackers could not be able to get the actual data in the cloud. It increases the confidentially of the data stored in the cloud.

Conclusion

Cloud provides tremendous amount of storage to the users. Cloud storage is virtually utilized in low cost. Cloud storage has many unique features. Apart from that, cloud has problem in security of data stored in the cloud. This paper proposed a security service namely AROMONcrypt to address the security issues in the cloud storage. AROMONcrypt security service is used to encrypt and obfuscate the data simultaneously. Simulation is conducted for AROcrypt, MONcrypt and AROMONcrypt by considering the time and security level of cloud data. The results show that the MONcrypt has taken minimum time duration compared with AROcrypt and AROMONcrypt. AROMONcrypt produces 91.5% of security level, compared with AROcrypt and MONcrypt; AROMONcrypt gives maximum security level and ensures the confidentiality of the numerical and non-numerical data stored in the public cloud environment.

Reference

[1] Wei-Tek Tsai, Xin Sun andJanakaBalasooriya, "Service-Oriented Cloud Computing Architecture", IEEE International Conference on Information Technology, 978-0-7695-3984-3/10,pp 684-689, 2010.

[2] Dr. L. Arockiam, S. Monikandan and G.Parthasarathy, "Cloud Computing: A Survey", International Journal of Internet Computing, ISSN No: 2231 – 6965, Volume 1, Issue 2, pp. 26-33, 2011.

[3] Yvette E. Gelogo and Sunguk Lee, "Database Management System as a Cloud Service", International Journal of Future Generation Communication and Networking Volume 5, Issue 2,pp 71-76, June 2012.

[4] Xiaojun Yu, Qiaoyan Wen, "A View about Cloud Data Security from Data Life Cycle", IEEE International Conference on Computational Intelligence and Software Engineering (CiSE), pp 1-4, Dec 2010.

[5] Tim Mather, Subra Kumaraswamy, and Shahed Latif, "Cloud Security and Privacy", O'Reilly Media, Inc., Chapter 4,pp 61-71, 2009.

[6] Manpreet Kaur and Rajbir Singh,"Implementing Encryption Algorithms to Enhance Data Security of Cloud in Cloud Computing", International Journal of Computer Applications, Vol. 70, No. 18, pp. 16-21, 2013.

[7] Rashmi Nigoti, Manoj Jhuria and Dr.Shailendra Singh, "A Survey of Cryptographic Algorithms for Cloud Computing", International Journal of Emerging Technologies in Computational and Applied Sciences, Vol. 4, No. 2, pp. 141-146, 2013.

[8] Siani Pearson, Yun Shen and Miranda Mowbray, "A Privacy Manager for Cloud Computing", Proceedings of the 1st International Conference on Cloud Computing, Springer-Verlag Berlin, Heidelberg, pp. 90 – 106, 2009.

[9] Chad Robertson, "PDF Obfuscation – A Primer", GIAC (GPEN) Gold Certification, TBA, The SANS Institute, pp. 1-40, 2012.

[10] Atiq, U.R. and M. Hussain, "Efficient cloud data confidentiality for DaaS", International Journal of Advanced Science and Technology,Vol. 35, pp. 1-10, 2011.

[11] Sunita Rani, Ambrish Gangal "Cloud Security with Encryption using Hybrid Algorithm and Secured Endpoints", International Journal of Computer Science and Information Technologies, Vol. Vol. 3, No. 3, pp. 4302 – 4304, 2012.

[12] Subhasri P., Padmapriya A., "Multilevel Encryption for Ensuring Security in Public Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 7, pp. 527- 532, 2013.

[13] Arockiam L., Monikandan S, and Sheba K Malarchelvi P. D., "Obfuscrypt: A Novel Confidentiality Technique for Cloud Storage", International Journal of Computer Applications, Vol. 88, No. 1, pp. 17-21, 2014.

[14] Anshu Parashar and Rachna Arora, "Secure User Data in Cloud Computing Using Encryption Algorithms", International Journal of Engineering Research and Applications (IJERA), Vol.3, No. 4, pp. 1922-1926, 2013.