Obfuscation with Fuzzy Based Data Security Algorithm for Improving the Security in Cloud (OFDSA)

A.Ahadha Parveen[1†] and P.S.S Akilashri[2††],

Department of Computer Science, National College (Affiliated to Bharathidasan University), Trichy, Tamil Nadu, India,620020

Abstract

Cloud computing refers to the way of storing, processing and managing data over the internet instead of a local computer or server. The most significant problem associated with cloud computing is security. Brute-force Attack, Cipher texts Only Attack, Known Plaintext Attack is one of the challenging security threats while sharing data and resources over the cloud. The cloud provider should make sure of its user's data storage confidentiality. In addition, the conventional "Advanced Encryption Standard" (AES) algorithm requires to be improved to deal with the rising security risks in the cloud setting. To handle these issues, this research proposes a new obfuscation with fuzzy based data security algorithm called *OFDSA* (Obfuscation with fuzzy based Data Security Algorithm) to increase the security level and protect the data stored in the cloud environment. Obfuscation is the most efficient technique to protect data from various cloud security threats. In spite of fuzzy incorporation, one could get secure and optimal communication done in the transmission of data among the systems over inter and intranets.

Keywords:

Obfuscation; AES; Encryption; Fuzzy Logic.

1. Introduction

Cloud computing emerges as a new web-based computing technology. It offers different computing services such as data storage, memory, networking, software, and databases. It basically consists of three service models namely "Infrastructure as a Service" (IaaS), "Platform as a Service" (PaaS), and "Software as a Service" (SaaS). Similarly, it basically contains three development models namely Private Cloud, Public Cloud, and Hybrid Cloud. Some other types of service models are storage as a service, Application as a Service, Network as a Service, etc. Correspondingly, other type of cloud development model is community model [1].

Though cloud is advent technology, it still faces some significant challenges such as security and privacy for data storage compared to other technologies. An efficient Obfuscation and encryption mechanism is required to protect the data from various cloud security and privacy issues. The data security comprises confidentiality, integrity, and access controllability [2].

A. Obfuscation in Data Security

"Obfuscation in data security" is to convert the source data in to an unintelligible form using either mathematical computations or programming functions or the combination of both. Hence, the result is difficult to understand by intruders. In recent times, this mechanism plays a major role for securing data storage in cloud.

B. Encryption and Ciphertext

Encryption is the conversion of original data into unreadable form called Ciphertext by using certain keys or methods. An authorized party can only access the data and others cannot.

C. Obfuscation using Encryption

Obfuscation Using Encryption is the process of converting the source code into an unreadable formatting syntax when using an encryption key. A major difference between the term's obfuscation, encryption and obfuscation using encryption is exposed in TABLE I.

Manuscript received November 5, 2022
Manuscript revised November 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.11.69

Parameter	Obfuscation	Encryption	Obfuscation Using Encryption
Definition	Change the syntax of any form of data to another form. It is used in the context of program code. The program code will produce the same output between the plain text and the obfuscated code	Encryption is the conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties.	Change the syntax of the program code to an unreadable formatting syntax when using an encryption key. The program code will produce the same output between plain text and the obfuscated code.
Key requirement	It does not require a key to decode data to its original form	Requires a key	Requires a key
Changes in data syntax	In an elusive form	In unreadable form	In unreadable form

TABLE I. DIFFERENCE BETWEEN THE TERMS OBFUSCATION, ENCRYPTION AND OBFUSCATION USING ENCRYPTION

D. Fuzzy set theory

Fuzzy set theory derived from classical set theory where elements contain varying degrees of membership. A fuzzy set is any set that permits its elements to contain different degree of membership in the specified range [0, 1]. Due to its extensive series of significant advantages such as handling uncertainty and logical reasoning, it is widely used in cloud computing. In cloud, so many security issues can solve with the fuzzy logic. In theoretical view, fuzzy set $\tilde{A}$ is specified as a set of ordered pair $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))/x \in X, \mu_{\tilde{A}}(x) \in [0,1]\}$, where $\mu_{\tilde{A}}(x)$ is the membership function of $x \in X$ *and* X represents the universal set.

E. Brute force attack

A brute force attack, also called as exhaustive search. In this cryptographic hack, the attacker tries to decrypt the password by assuming probable combinations of a determined password.

F. Ciphertext-only attack

Ciphertext-only attack is also known as ciphertext attack. In this attack, the adversaries access the set of cipher text, but not the input plaintext. Frequency analysis is the most important traditional method used to break the cipher text.

G. Known-Plaintext Attack

In this attack, the adversary accesses both the crib (plaintext), and its ciphertext (encrypted data).

Moreover, Section 2 describes various obfuscation and encryption algorithms related to the data security in cloud. Section 3 explains different security algorithms that are used to evaluate the proposed OFDSA methodology. Furthermore, Section 4 describes the proposed OFDSA methodology. Simulation results and analysis of the proposed OFDSA is exposed in Section 5. At last, conclusion is presented in Section 7.

2. Related Work

This section describes some important obfuscation and encryption algorithms related to the data security in cloud. The purpose of this study is to make the system more secure and privacy of cloud computing using obfuscation with fuzzy based data security algorithm called *OFDSA* (Obfuscation with fuzzy based Data Security Algorithm) which secures the data from harmful malware attacks in the cloud environment.

Barack et al. [3] specifies that the obfuscation code cannot be easily understood by others, but executing its functions as the original code is impractical. At the same time, Schneider and Locher [4] proved that the obfuscation using encryption can be practical and maintain the confidentiality of original code using this technique. Moreover, obfuscation without encryption is still considered as a susceptible mechanism in the security aspects. So that, this research proposes an enhanced obfuscation using encryption mechanism to increase the data security level over the cloud.

In addition, fuzzy incorporation also provides enhanced data security over the cloud. Hence, this research work proposes obfuscation with fuzzy based data security algorithm called *OFDSA* (Obfuscation with fuzzy based Data Security Algorithm). Ramalingam and Arul Marie Joycee [5] presented an innovative obfuscation mechanism to protect the data storage in cloud. This technique dynamically secures data from the interrupters. Abid Murtaza et al. [6] proposed an efficient algorithm by eradicating the difficult functions in the symmetric key algorithms. Arul Oli and Arockiam [7] proposed a novel data encryption mechanism with an obfuscation technique that encrypts numeric type of data stored in the cloud. In these methods, processing time is high compared to the proposed system.

Chopra and Lata [8] presented a new encryption method called “128-bit AES algorithm”. The experimental outputs demonstrate that the “128-bit AES algorithm with fuzzy” concepts provides higher security and reliability. Ryndel and Ariel [9] presented an improved DES symmetric algorithm integrating the f-function and eradicating X-OR operations. But both of these methods offer lesser security compared to the proposed *OFDSA.*

Kashmar et al. [10] developed a fuzzy based AES algorithm to encrypt sensitive data. This work proposes three encryption techniques to secure data against cryptographic attacks. Ismanto and Salman [11] presented a technique to enhance security level of the data through obfuscation mechanism with AES algorithm. But these techniques didn’t compare with any existing methods.

3. Security Algorithms

This section explains some important security algorithms that are used to estimate the proposed OFDSA methodology.

A. Advanced Encryption Standard (AES)

Rijndael is a nickname for the AES algorithm, which is more widely known as AES. It's a technique that uses symmetric keys to convert plain text in 128-bit blocks into encrypted text using keys of 128-bit, 192-bit, or 256-bit length. The Advanced Encryption Standard (AES) has replaced DES and 3DES as the industry standard and is considered to be more secure. There are four main types of transformation in it. The key addition, substitution, permutation, and mixing are the four operations.

B. Obfuscating Conjunctions (OBCO)

The OBCO [12] mechanism is used to obfuscate a class of conjunction operations. Basically the conjunction function is more protected for all forms of distributions. This OBCO technique declares that the proposed obfuscator gives promising protection for all conjunctions against general intruders. In addition, this technique is associated with multilinear maps.

C. Structure vs. Hardness through the Obfuscation Lens (SHOLENS)

The SHOLEN [13] is a novel encryption system depends upon Collision-resistant hashing that employs indiscriminability obfuscation and totally black-box structures.

D. ODSA (Obfuscation Data Security Algorithm)

The ODSA [14] is a symmetric key encryption technique that produces ciphertext using a new obfuscation mechanism. As part of the algorithm, this technique is offered. In order to make the ciphertext more difficult to break, this ODSA has a key size of 256 bits. In addition, Mod 256 mathematical logic is applied. This ODSA algorithm examines the input data one character at a time and converts it to a 2-bit ciphertext for each of those characters as it goes.

4. Methodology

Fig. 1 describes the proposed obfuscation with fuzzy based data security algorithm called *OFDSA*. The proposed security framework enhances the security level and protects the data stored in the cloud environment.

At first, the input of the source code is obfuscated using data obfuscation technique. After that, the obfuscated output is encrypted using AES algorithm and secret key is created. Then fuzzification takes place on the encrypted data and the cipher text is produced. Here after, cipher text is defuzzified and decrypted using the generated secret key. And then the decrypted data is changed into obfuscated data. At last, obfuscated data is converted in to the plain input text.

A. Key Generation

The key is a very important component in encrypting data saved in the cloud. The data encryption operation employs a key, which is why it is so vital for data storage security. It will be more difficult for the opponent to find the key if the length of the key is extended. A secret key with a length of 256 bits is generated using developed java code and utilised in the suggested OFDSA. A mix of alphabetic letters, arithmetic numbers, and other symbols make up the concealed code.

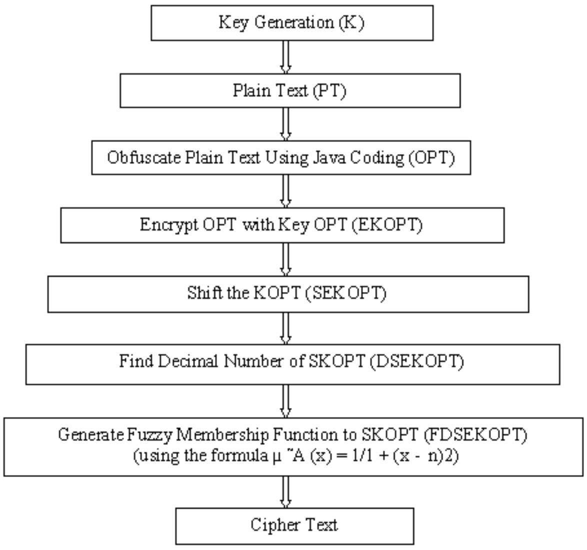

Fig. 1 Block diagram of the proposed OFDSA methodology

B. Algorithm

Step 1. Take Inputs: Source code input is taken.

Step 2. Obfuscate Original Source Code: Original source code is obfuscated using java coding. This obfuscation technique makes the data difficult to understand by an adversary. And also decrease the coding size as well as increases the processing speed.

Step 3. Encrypt Obfuscated Code Using AES: obfuscated data is encrypted with the key using AES algorithm. Then, the encrypted file in the form of ASCII character is stored in the memory for evaluating later procedure. AES algorithm protects data storage and provides data confidentiality in the cloud.

Step 4. Shift Encrypted Code: Shift operation is performed on the ASCII character of the encrypted file. Then the shifted ASCII character is converted into the decimal value and put in to the fuzzy formation process.

Step 5. Applying Fuzzy Logic: Fuzzy membership function is applied on the decimal value of the shifted ASCII character and the fuzzy membership based output is obtained using the formula

$$\mu \tilde{A}(x) = 1/1 + (x - n)^2 \quad (1)$$

where n is the real numbers

At last, the cipher text is generated. For the defuzzification of the cipher text, the inverse process of fuzzification is executed. MATLAB function is used for the computation the membership values of fuzzy set $\tilde{A}$.

Step 6. Decryption of Cipher-Text Using AES: Decryption procedure is applied on the cipher text to obtain encrypted text again.

Step 7. De-Obfuscation of Encrypted Code Using Obfuscator: De-obfuscation id performed on the decrypted data to acquire original source code.

The proposed OFDSA methodology is illustrated step by step by taking simple plain text “hai” as an input. At first, the input “hai” is obfuscated using data obfuscation technique [14]. After that, the obfuscated output is encrypted using AES algorithm. The encryption key generated for this example is &56780)(><^%$#@!^&*()&^%%$#@1234. Then the residual computational operations are executed and cipher text is generated. They are revealed in TABLE II.

<table>
<tr><th>Algorithm</th><th>Step by step execution with example</th></tr>
<tr><td>Key (K)</td><td>&56780)(><^%$#@!^&*()&^%%$#@1234</td></tr>
<tr><td>Plain Text (PT)</td><td>hai</td></tr>
<tr><td>Obfuscated Plain Text (OPT)</td><td>eval(function(p,a,c,k,e,d){e=function(c){return
c};if(!''.replace(/^/,String)){while(c--){d[c]=k[c]||c}k=[function(e){return
d[e]}];e=function(){return'\\w+'};c=1};while(c--){if(k[c]){p=p.replace(new
RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return p}('0',1,1,'hai'.split('|'),0,{}))</td></tr>
<tr><td>Encrypt OPT using AES algorithm with key (EKOPT)</td><td>8AC98479618127243B8580A6E05B3A96DDEF409E71A2FEC29AFD73D89F
5E8D2E63BA6707E9A7917A4EFA77D5CD9F3CCE0D417BE93F53D918EC
D85F42577B51746D8F8DAA56F559669EC71717BC9FDA545FE723E1A337
C59943F89AD96BBED8E5CE18708A6C33A9C044D12613511044675A795B0
E0384CE10ABBD2FA86952EC8FB93D213FBE704A409188DC09788250311
E9EA7DD98F8DBF7154F351B1492FFF3721F9BE9942BA2489EB91FFD994
C9875F5E575F92207119895698 8E7DE380A1E7F89AB3539D3F2412373EE85
FC6ADEC227476B6DB0673C49A2B5A3CFA78B838479D45AE0C6FEB65B6
A9703B0EE7BC4B05DECA25FE371D583FCD4FEC505731BED345DB65DD
6854071A50F69611D363E6346F0104BCF24C35B1EF814B0F259F370</td></tr>
<tr><td>Shifted KOPT (SEKOPT)</td><td>J? D9!Agd{E@f? zV? /? ^1b>? Z=3? _? Mn#z'G)gQ:? :7? _|? M? ;)? ?
? X,? ? ? ? ;? 4OMj? 5? &^WW|_? ? ? 'c!cw? Y? 8Z? +~? %? X0J,si
? ? ? fS? P? '? 9? NCD? Pk}oh)? ,Oy}a? ~0\? QH? I8B? q^^g? X8? 7
U? u[TR?32_[)Tkb? ^y_=Y? X55%59bGQXU)H? xJ^?Ikuy? 2? c3.E<*? ?
g+p'|? buc? gKCD9? ? ? >v? *WCp.;? p? ,b? #1? C<? >? E3[t? v? ? E
? 1eO)!]v~#? 0P? ? d? ? ^8Tp2? 30</td></tr>
<tr><td>Decimal Number of SKOPT (DSEKOPT)</td><td>74 63 68 57 33 65 103 100 123 69 64 102 32 63 122 86 63 47 63 94 49 98 62 63
90 61 51 63 95 63 77 110 35 122 39 71 41 103 81 58 63 58 55 63 10 95 124 63 77
63 59 41 63 63 63 88 44 63 63 63 63 59 63 52 45 79 77 106 63 53 63 38 94 7 87
87 124 95 63 63 63 39 99 33 99 119 63 89 63 56 90 63 43 126 63 37 63 88 48 74
44 115 105 63 63 63 102 83 63 80 63 39 63 57 63 78 67 68 63 80 107 125 111 104
41 63 44 79 121 125 97 63 126 48 10 63 81 72 63 73 56 66 63 113 94 94 103 63
88 56 63 55 85 63 117 91 84 82 63 51 50 95 91 41 84 107 98 63 94 121 95 61 89
63 88 53 53 37 53 57 98 71 81 88 85 41 72 39 63 120 74 94 63 73 107 117 121 63
50 63 99 51 46 69 60 42 63 63 103 7 43 45 112 39 124 63 98 117 99 63 103 75 67
68 57 63 63 32 63 62 118 63 42 87 67 112 46 59 63 112 63 44 98 63 35 49 63 67
60 63 62 63 69 51 91 45 116 63 118 63 63 69 63 49 101 79 41 33 93 118 126 35
63 48 80 63 63 100 63 63 94 56 84 112 50 63 51 48</td></tr>
<tr><td>Cipher Text =
Fuzzy Membership Function of DSEKOPT (FDSKOPT)
[using the formula µ ˜A (x) = 1/1+ (x - n)2]</td><td>0.000000 0.089754 0.097543 0.087654 0.058824 0.015385 0.058824 1.000000
0.058824 0.200000 0.058824 1.000000 0.078952 0.089754 0.097543 0.087654
0.075432 0.054321 0.034567 0.012376 0.035678 0.076453 1.000000 0.000000
0.012387 1.000000 0.015385 0.058824 0.086544 0.076543 0.089456 0.065432
0.037777 0.056784 0.048765 0.065432 0.089754 0.097543 0.087654 0.089754
0.097543 0.087654 0.087553 0.075432 0.054321 0.034567 0.012376 0.075432
0.054321 0.045777 0.098765 0.065534 0.065432 0.076543 0.054322 0.064343
0.065432 0.097654 0.012376 0.023876 0.054372 0.076432 0.089654 0.074326
1.000000 0.056432 0.078654 0.054321 0.075432 0.095432 0.037657 0.000000
0.000000 0.065432 0.054321 0.054325 0.056785 0.065890 0.068808 0.054387
0.065565 0.000767 0.007665 0.065434 0.038654 0.065432 0.087654 0.056789
0.065444 0.054324 0.098767 0.054325 0.005300 0.0005353 0.00006 0.006565
0.007665 0.007665 0.543261 0.083675 0.066442 0.0163546 0.04566 0.065746
0.054653 0.087007 0.007654 0.076565 0.054327 0.0652134 0.06543 0.065545</td></tr>
</table>

TABLE II. COMPUTATIONAL PROCESS OF THE PROPOSED OFDSA METHODOLOGY

C. Benefits of OFDSA

The following advantages may be inferred from this study thanks to its findings,

- The proposed OFDSA offers Because its coding cannot be read by anybody who is not specially authorised to view it, it has a very high level of security.
- The OFDSA result in the form of a fuzzy membership function offers an advantage, because it gives higher difficulty in breaking the cipher text.
- The proposed OFDSA offers a solid base that ensures the safety of data that is kept in the cloud. In spite of fuzzy incorporation, one could get the secure data storage over the cloud. The proposed OFDSA works well against the cyberattacks and the attackers those who break the cipher text.

5. Simulation Results and Analysis

The OPNET simulator is used in this research to evaluate the performance efficiency of OFDSA against AES, OBCO, SHOLENS and ODSA. OPNET Simulator is an open free software extensively used in cloud networking environment. It efficiently calculates various cloud computing metrics such as "security, encryption time, decryption time, throughput, latency, jitter, end-to-end delays, power consumption", and etc [15]. TBecause of its extensive support for cloud settings, network topologies, and Java programming, this research takes use of his simulator. The simulation is conducted on a PC with a 64-bit version of Windows 10, a 2.7-GHz Core i5-7200 CPU, and an 8-gigabyte RAM capacity. Some important simulation parameters are shown in TABLE III.

Parameter	**Value**
Coverage area	1000 x 1000 m
Number of nodes	50
Types of nodes	Heterogeneous
Node assignment	Random distribution
Traffic type	Typical real-world random traffic
Communication mode	Wireless
Data Size	1GB to 5GB

TABLE III SIMULATION PARAMETERS

In order to assess the performance of OFDSA, the following parameters are used: Encryption time, Decryption time and Security level.

A.Encryption Time

The encryption time is the total time taken to convert the source text in to the cipher by executing different functions specified in the respective encryption techniques. The acquired encryption time is exposed in TABLE III.

B. Decryption Time

The decryption process is identical to the encryption process, except it is carried out in the other way. It's the total length of time it took to convert the cypher text into the source text provided as input. The results are given in TABLE IV for your consideration.

C. Security Level

Security is an expected thing in any cloud communication process. Security is calculated in OPNET by activating predefined cloud security hazard models such as Brute Force attack, Cipher Text Only attack and Known Plain-Text attack [15]. These attacks are employed to represent the secure data transmission process throughout the simulation. The secure data transmission of the proposed OFDSA against AES, OBCO, SHOLENS and ODSA is estimated by using various security attacks such as Brute Force attack, Cipher text Only attack and Known Plain-Text attack. The security level is calculated using the following equation:

Security = (P_{cd} / P_{td}) x 100 (2)

Where P_{cd} indicates compromised data packets.

If the security level is higher, then the breakage of cipher is a difficult task to the adversaries. The power of the encryption mechanism is measured by the computation function that gives the unreadable cipher. The proposed OFDSA security level against AES, OBCO, SHOLENS and ODSA is shown in TABLE VI. The Graphical representation of the proposed OFDSA's Encryption time, Decryption time and Security level (TABLE IV, V and VI) against AES, OBCO, SHOLENS and ODSA are shown in Fig. 2,3 and 4.

Data (GB)	AES	OBCO	SHOLENS	ODSA	OFDSA
1	2137	2321	2562	2073	1983
2	3851	4299	4618	3800	3600
3	6090	6545	7318	5896	5525
4	8279	8869	9950	7930	7723
5	10435	11212	12313	9980	9710

TABLE IV ENCRYPTION TIME (mS)

Data (GB)	AES	OBCO	SHOLENS	ODSA	OFDSA
1	2039	2302	2476	2157	1976
2	4085	4293	4648	3885	3587
3	6052	6691	7625	5912	5612
4	8325	9029	9737	8040	7856
5	10410	11370	12354	10040	9870

TABLE V DECRYPTION TIME (mS)

Data (GB)	AES	OBCO	SHOLENS	ODSA	OFDSA
1	86.24	81.45	86.80	90.36	92.46
2	85.86	83.34	89.32	88.87	91.83
3	86.01	81.70	86.59	90.26	93.78
4	85.05	82.87	88.78	88.93	90.15
5	85.35	82.71	88.13	89.00	91.56

TABLE VI SECURITY LEVEL (%)

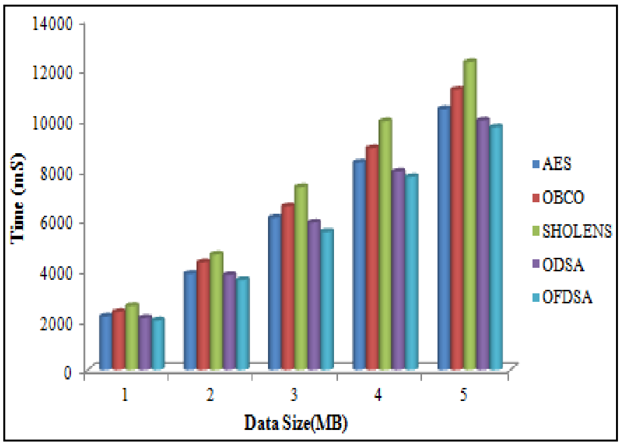

Fig.2. Encrptiom time (mS)

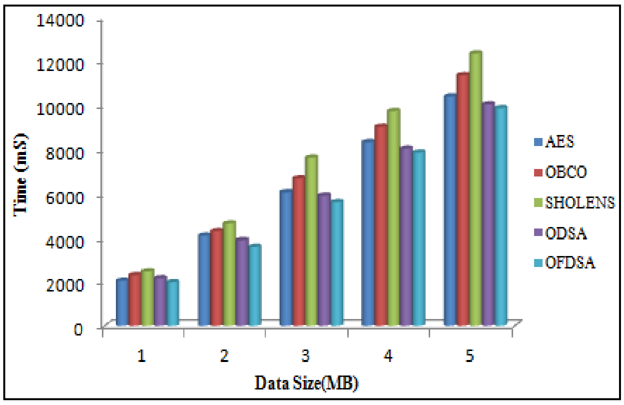

Fig.3. Decryption time (mS)

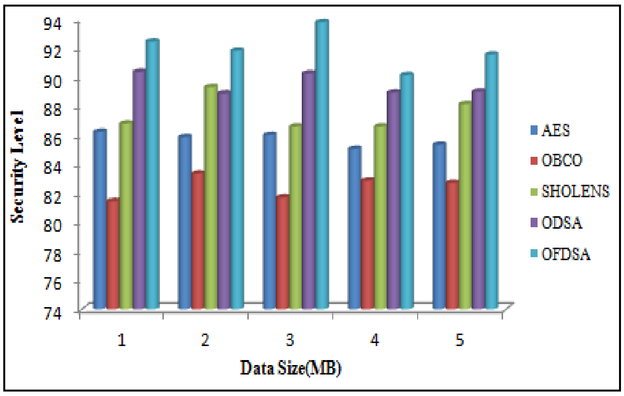

Fig.4. Security level

6. Conclusion

This research work proposed a novel obfuscation with fuzzy based data security algorithm namely OFDSA ("Obfuscation Data Security Algorithm") to enhance the security and protect data stored in the cloud. The processing speed comparison of the proposed OFDSA with AES, OBCO SHOLENS and ODSA proved that the OFDSA achieves significant improvement during both the encryption and decryption process. Experimental results also proved that the security level of OFDSA is superior to AES, OBCO, SHOLENS and ODSA. This evidently shows that the unintelligible cipher created by the OFDSA is difficult to understand and break by the intruders.

References

[1] P. Mell, T. Grance, "The NIST definition of cloud computing", National Institute of Standards and Technology, 2011.

[2] L. Arockiam and S. Monikandan, "Data security and privacy in cloud storage using hybrid symmetric encryption algorithm", International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, issue. 8, August 2013.

[3] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S.P. Vadhan and K. Yang, "On the (im)possibility of obfuscating programs", Advances in Cryptology, pp. 1–18, 2001.

[4] J. Schneider and T. Locher, "Obfuscation using encryption", ABB Corporate Research, Switzerland, 2016.

[5] S. Ramalingam and K. Arul Marie Joycee, "FEDSACE: A Framework for enhanced user data security algorithms in cloud computing environment", International Journal on Future Revolution in Computer Science and Communication Engineering, vol. 4, issue. 3, pp. 49 – 52.

[6] A. Murtaza, S.J.H. Pirzada and L. Jianwei, "A new symmetric key encryption algorithm with higher performance", International Conference on Computing, Mathematics and Engineering Technologies, 2019.

[7] S. Arul Oli and L. Arockiam, "Confidentiality technique using data obfuscation to enhance security of stored data in public cloud storage", International Journal of Advanced Research in

Electronics and Communication Engineering, vol. 5, issue. 1, January 2016.
[8] S. Chhabra and K. Lata, "Enhancing data security using obfuscated 128-bit AES algorithm", An Active Hardware Obfuscation Approach at RTL Level, IEEE, 2018.
[9] V.A. Ryndel, M.S. Ariel and P.M. Ruji, "Enhanced data encryption standard algorithm based on filtering and striding techniques", ICISS, 2019.
[10] A.H. kashmar, A.I. Shihab, and Z.L Abood, "Develop AES algorithm based on fuzzy set theory", Second International Conference for Applied and Pure Mathematics, 2019.
[11] R.N. Ismanto and M. Salman. "Improving security level through obfuscation technique for source code protection using AES algorithm", In Proceedings of the 2017 the 7th International Conference on Communication and Network Security, pp. 18-22, 2017.
[12] Z. Brakerski, G.N. Rothblum, "Obfuscating conjunctions", Journal of Cryptology, Springer, pp. 289–320, 2017.
[13] N. Bitansky, A. Degwekar and V. Vaikuntanathan, "Structure vs. hardness through the obfuscation lens", Annual International Cryptology Conference, Advances in Cryptology, pp. 696-723, 2017.
[14] M. Kamal and G. Ravi, "Enhancing data security in public cloud storage using obfuscation techniques", International Journal of Advanced Science and Technology, vol. 29, no. 4, pp.7784 – 7796, 2020.
[15] R. Menaka, R. Ramesh, R. Dhanagopal, "Behavior based fuzzy security protocol for wireless networks", Journal of Ambient Intelligence and Humanized Computing, Springer, 2020.

A.Ahadha Parveen, completed her B.C.A from Jamal Mohamed College in the year 2005 and after which got placed in Cognizant Technology Solutions and worked there for a couple of years. She completed her M.C.A from Bharathidasan University in the year 2009.She cleared her UGC NET exams in Computer Science in the year 2019. She worked for Jamal Mohamed College, Trichy from 2015-2022 and at present is working as an Assistant Professor in Lady Doak College, Madurai. Her areas of interest include Cloud Computing, Cyber Security and IOT.

P.S.S Akilashri is the vice principal of National College and Head of the PG & Research Department of Computer Science, National College, Trichy. She is the coordinator of international conferences, seminars, and workshops. Her areas of interest include big data, digital image processing, cloud computing and software engineering. She has attained 2 patent rights for her research. She has won the best administration award, excellence teacher in higher education award for the year 2022 given by Sakya academy. She has authored a book named "Python Programming". She has published her research papers in 17 international journals.