# A Combination of EOBE to Enhance Data Security in Cloud using SKM Algorithm

**Shaik khaja mohiddin, Y Suresh Babu**

***Abstract— Protection of client's data in cloud is an important aspect which a cloud service provider assures to a cloud user. Though using high end sophisticated methods security is provided to client's data in cloud. Besides this theft of data has emerged as one of the common and major issue and which has grown to a big challenge even today also. In this paper we have proposed a unique algorithm which provides security for the data in different forms stored in cloud. In this paper we used a unique algorithm "SKM" by combining EOBE methods where encryption, obfuscation followed by binary encryption is effectively used to reach out the required target task. The data may be either, numeric, non-numeric or binary data related to image files. Depending on kind of data the required process is initiated in order to convert the data in a safe form. From the suggested algorithm a reduced timing with respect to encryption, decryption was achieved. Though the security concept in cloud is a bit difficult to deal, our proposed algorithm has achieved maximum security level among the existing ones. Using industrial tool the observed results are compared with the existing and popular methods of security algorithms. This creates a confidence among several cloud users and small scale business organization to move and enhance their belief on security of data in cloud.***

***Index Terms— KGMaaS, STaaS, SEaaS, Obfuscation, Cloud Forensics.***

## I. INTRODUCTION

Confidentiality is one of the major factors to be considered under security aspect of data stored in the cloud. Confidentiality assures that data can be accessed by the authorized persons only. Sometime it may be compromised by the unauthorized access of the stored data in the cloud. In general when we store the data in the cloud the keys used for the access of stored data in cloud, storage of data in cloud and security for the client's data in cloud is provided by the same cloud service provider. In this situation there is a possibility for the cloud service provider to steal client's credentials and to manipulate or to compromise client's data. Out of many challenges which are tough to deal in the area of cloud forensic concepts are one among such, cloud security is also one of the major factors. If a proper cloud security is not provided to client's data stored in cloud then it slowly leads to the failure of building faith from customer side towards the cloud. But when the same concept is made by hiring different cloud service providers i.e. accessing and generation of keys service from one cloud, security service to be hired from another cloud, and storages of data service from after cloud service provider. Due to this the probability for client's data modification by intruders becomes very minute. Making data in an unreadable format is referred as encryption [1][2].

The way in which more and more volume of data being stored in cloud, the way in which more diversity of data is being stored, the different ways of variations from data to data which are stored and then accessed by the client with in cloud. Due to huge data loss resulting from cyber attacks. Which are carried out with respect to the cloud, though security measure is taken to counter these attacks but every time intruders gain an upper hand to perform attacks for gaining access to client's data in cloud. Beside huge deployment and migration of IT industries to Cloud. There is always a slant on security measure. After the attacks one has to find the exact reasons which gave a scope for the attacks if exact reasons are known then counter measures can be taken to fulfill this loss. High level scientific research has to be performed on cloud forensic concept which is still in its mid path and there has to be a lot to achieve its peak state where cloud gains an upper hand to the intruders.

## II. MOTIVATION FOR THE WORK

Though there exist many challenges related to cloud forensics with respect to data collection, architectures, analysis, anti-forensics, a lot of legal issues, regarding marinating of standards. But as per the security aspect among the existing security concept bit challenging to be addressed. It's a kind of challenge where we think that we found the solution of the existing problem than many other challenges which are synonymous come from other different corners which we did not expect. To protect clients data in cloud various algorithms were also used such as [3][4][5][6]. Existing obfuscation methods were having certain constrains [7]. This area needs an update daily and to say clearly we need to be alert every minute of minute. Then only one can gain a small knowledge about the concepts of cloud forensics which need to be addressed instantly.

## III. PROPOSED WORK

In our proposed model we took three services from three different cloud services in order not to compromise the security of issues of data i.e. Key generation and management services hired from one cloud, Security service hired from

**Revised Manuscript Received on October 15, 2019**

**Shaik khaja mohiddin,** Research Scholar, Dept. of CSE, Acharya Nagarjuna University., Asso.Prof.,Dept. Of CSE. VVIT, Guntur, Andhra Pradesh, India

**Dr. Y Suresh Babu,** Dept. Of CSE,JKC College, Guntur,Andhra Pradesh, India

another at the same time storage service is hired from another cloud service provider. The need for a dedicated framework for cloud forensic investigations as mentioned by [8]. Even some times IDS also plays an important role regarding sniffing the greedy steps of intruders [9]. These services are including Key generation and management services which are helpful for the generation of keys and from this the generated keys are then managed. Security service where we are running our proposed algorithm of security. Storage service with the help of which the encrypted data is stored in the cloud. In our proposed algorithm we have designed three algorithms which are used to encrypt, obfuscate and to perform binary encryption with three different kinds of user data. This may be ranging from numeric, non-numeric and binary data. As any kind of data which has to be stored in the cloud falls under one of these categories. With respect to these three kinds of data obfuscation, encryption and binary encryption is carried out.

The results which are generated from our work are then compared with the industrial security level checker " ABC" All Block Cipher Tool". It is an industrial based tool which is very strong technically in order to check the security levels of existing as well as any kind of the proposed algorithms. The architecture of our proposed work is as shown in the below figure.

The below figure shows the followed architecture in order to get the desired results. Obfuscation methods are used in order to protect data in cloud it is also considered as one way of providing security to the cloud. Obfuscation can be of several types such as source code obfuscation, location obfuscation, obfuscation by encryption, diversification, securing the browsers, noise obfuscation and data obfuscation. Out of these many obfuscation depending on the concept and need we use the required kind of obfuscation in order to provide security for clients data stored in cloud. Though out of all these various kinds of obfuscation the choice of correct method of obfuscation gives eminent results. Which is also helpful in order to enhance the level of security to client's data in cloud.

Here we use the concept of obfuscation by encryption. This also plays an important significance role for creating security for the numeric data stored in the cloud. For this we have framed obfuscation by encryption algorithm and also shown one example that how the process is carried out. This concept is a part of our three methods along with encryption used for plaint text, binary encryption used on binary files and along with our obfuscation used in the proposed SKM algorithm.

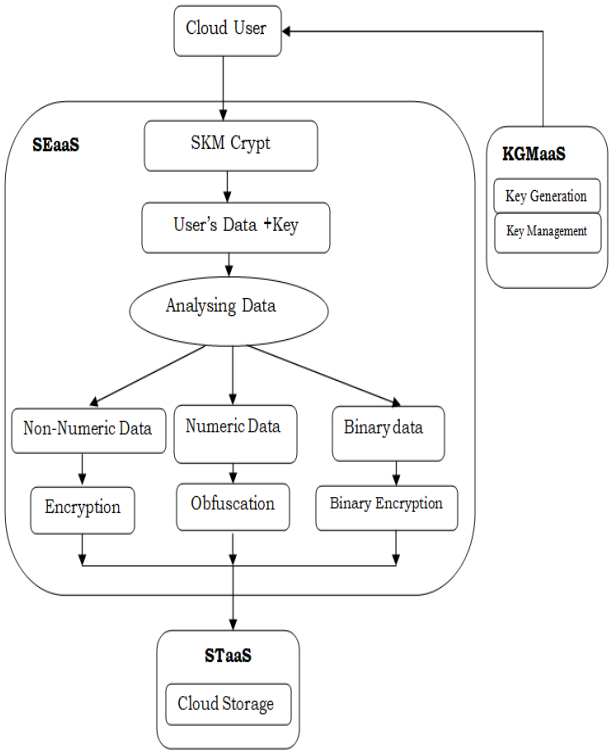



**Fig1: Architecture to perform SKM Algorithm**

## IV. METHODOLOGY OF THE PROPOSED SYSTEM

Here simulation results are carried out in cloud environment. User gets a key which is generated as Key Generation. Where KGMaaS, SEaaS, STaaS are the three different services which are being accessed from three different services with respect to three different cloud service providers. This method is followed in order to enhance the security levels. There by reducing the probability of a cloud service provider in order to make any sort of loss of integrity with respect to client's data inside the cloud. Here KGMaaS stands for the Key management and key generation, when are where necessary the required keys are generated and accessed. SEaaS security as a service is hired in order to carryout protection to our entire process which carried to distinguish the data and to follow the appropriate method in order to achieve the required and provide the required security

**Pseudo code for the key Generation**

Step1: Seedrandom_numgen ()
Step 2: Start
Step 3: rand ⟵ seedrandom_numgen ()
Step 4: return rand
Step 5: end

**Fig: Algorithm to generate Random keys**

From the above code a seed based random number is generated which is used during the generation of UUMAT. These generated random numbers through key management as services are utilized during the binary encryption process.

Pseudo code for **SKM Algorithm**

Step1: Take the input file

Step 2: Take the key generated from KGMaas.

Step 3: Provide the security to client's data in SEaaS

Step 4: Combine user data with generated key.

Step 5: Analyze the type of input file.

Step 6: For numeric data perform Encryption

For non numeric data perform Obfuscation

For binary data perform Binary Encryption

Step 7: Store the data in STaaS.

**Fig2: Showing the steps in SKM Algorithm**

Using the proposed algorithm after considering the input file. The data inside the input file is categorized into numeric, non-numeric and binary encryption. If the data

Pseudo code for **Encryption**

Step1: Take the input file.

Step 2: Count the text characters in the input file (N).

Step 3: Text converted to ASCII code.

Step 4: Convert ASCII to Binary.

Step 5: Obtain 1's complement and then 2's complement for the binary data.

Step 6: Convert obtained binary to base64.

Step 7: Convert obtained Base64 to ASCII and ASCII to Decimal.

Step 7: Count the number of decimal numbers (DN) and form a nearest square with that number.

Step 8: Generate a square matrix.

//Form a square matrix with that and fill the matrix with the decimal number, if any empty space is left then again start filling from the starting number.

Step 9: Divide the entire matrix into three parts Upper, Lower and Diagonal elements.

Step 10: Exchange upper to lower matrix and vice-versa and interchange the diagonal elements in reverse order.

Step 11: Generate ASCII value for the final matrix which is obtained

For Decryption for the same steps in a reverse order.

**Fig3: Algorithm to perform Encryption**

The above algorithm is used to perform encryption of plain text of non-numeric type.

**Pseudo code for Obfuscation:**

Step 1: Take the input.

Step 2: Convert Binary number to the input.

Step 3: Find the total number of bits (N)

Step 3: Find Square of N

Step 4: Generate Prime number up to N.

Step 5: Choose any prime number and convert it to binary

Step 6: Perform XOR with generated binary to the binary of the prime number

Step 7: Find the 1's and 2's complement of the same

Step 8: Convert the obtained results to Hexadecimal to UTF.

Step 9: Convert the obtained UTF to ASCII.

For De-Obfuscation perform the same in reverse order.

**Fig4: showing the algorithm for obfuscation**

The following is the output obtained in obfuscation for the given input.

Step1: Input: 1223455564894596486259312153156132 12

Step 2: 10111100100000001101000000101111 10100000010 11110011111110100001010111011000011010010100100010110110111001101100001110 0

Step 3: N (117), 117*117 =13689

Step 4: let one chosen prime number is 1901 (11101101101)

Step 6: (XOR) 10111100100000001101000000101111010 00000101111001111111010000101011101100001101001010010001011011011100110001011100001

Step7: **1's complement:** 01000011011111110010111 1110 10000101111110100001100000001011110101000100111 10010110101101110100100100011 00 11101 0001110

**2's complement:** 01000011011111110010111111010000 10111111010000110000000101111010100010011110010 1 1 01011011101001001000 11 00 11101 0001111

Step 8: 437f2fd0bf43 (hexadecimal)

Step 9: UTF Code: C□/пC

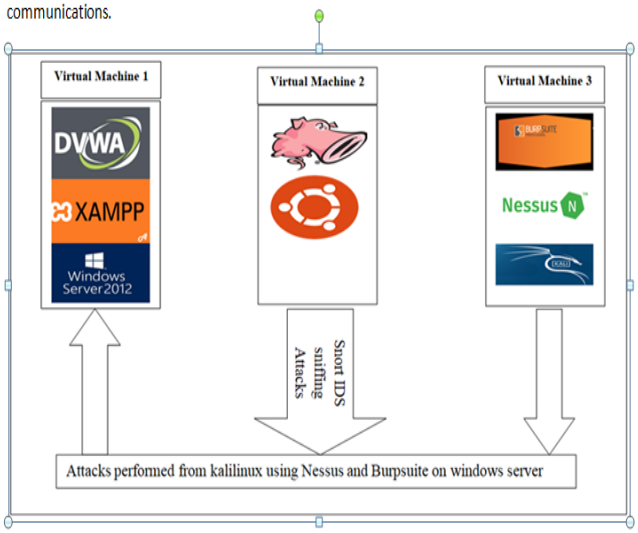
: Ascii code C□/Ð¿C

For De-obfuscation the reverse process is followed, where the ASCII code is taken as input and following the above steps in the reverse order we get the required plain text i.e. the number what we have gave as input for the obfuscation process.

Seed Random numbers: Generation of certain numbers which are not in a particular pattern or whose order are termed as random numbers. For example when we roll a die, when we flip a coin, when a rotate a wheel consists of certain number we are unaware that at what number the wheel will be stopping. A seed is a number which is taken from an unpredictable source. For example the noise which comes when we set the TV channel to a particular channel with no

signal we get a different noise. This noise is not stable it is changing continuously and we cannot predict it. Similar is the generation of a Seed random number.

Cryptography or cryptology (from Ancient Greek? pt??, Romanized: krypton "hidden, secret"; and ???fe?? Graphing, "to write", or -????a -logia, "study", respectively [1]) is the practice and study of techniques for secure communication in the presence of third parties called adversaries.[2] More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages;[3] various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation[4] are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, electrical engineering, 123346646879578545455 45555466555544556444 444 4444 444 44 44444 communication science, and physics. Applications of cryptography include electronic commerce, chip-based payment cards, digital currencies, computer passwords, and military communications.




Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shares the decoding technique only with intended recipients to preclude access from adversaries. The cryptography literature often uses the names Alice ("A") for the sender, Bob ("B") for the intended recipient, and Eve ("eavesdropper") for the adversary.[5] Since the development of rotor cipher machines in World War I and the advent of computers in World War II, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

**Fig 5: A sample input file given to the algorithm**

## V.SIMULATION RESULTS

For various sizes of the input files ranging from 1 to 5 Kb were considered for the task. The Encryption time, Decryption time and security levels of the proposed method were compared with that of the remaining popular methods providing the security of the same. As observed from the results the security levels of the proposed model is more compare with that of other popular algorithms which are shown in the below figure. Similar to UMAT i.e. Unicode matrix where 256 code ASCII codes are generated in the same way UUMAT, Unary Unicode Matrix which are 256*256 elements which are generated which are starting from All values from 0x0000 to 0xffff ( will be spread all over the UUMAT based on a seed. Even for Binary Decryption the same seed values is used which is generated for the binary encryption. The graphs were obtained by considering the variable quantities i.e. encryption and decryption times along y axis and the fixed size of the file of variable sizes along x-axis as shown in the below diagrams.

UUMAT

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0x280A | 0x7842 | 0x3306 | 0x3323 | 0x4981 | 0x0A79 | 0x22C7 | 0x51E9 | 0x7F51 |
| x2EFA | 0x5739 | 0x3E31 | 0x4F1D | 0x5C10 | 0x027E | 0x420F | 0x5ABB | 0x7ACC |
| x0C28 | 0x007F | 0x10D4 | 0x3939 | 0x0F91 | 0x4AE5 | 0x1702 | 0x4CCE | 0x5FB7 |
| 0x461D | 0x1BC1 | 0x39F9 | 0x0A14 | 0x40EB | 0x70A1 | 0x47BD | 0x26E5 | 0x5668 |
| x37D4 | 0x2C47 | 0x3657 | 0x2E5D | 0x5E57 | 0x3637 | 0x275D | 0x190C | 0x264B |
| x6513 | 0x16A6 | 0x25BA | 0x0646 | 0x0FB4 | 0x34E9 | 0x78C2 | 0x6AFD | 0x1265 |
| 0x4C63 | 0x175D | 0x081C | 0x180A | 0x7C2E | 0x24AC | 0x3A63 | 0x61DE | 0x44AB |
| x59A4 | 0x0BA8 | 0x6D11 | 0x61A5 | 0x7BF0 | 0x3DAA | 0x1C97 | 0x51B4 | 0x4225 |
| x7A5C | 0x08CD | 0x6DBC | 0x355D | 0x459C | 0x402E | 0x6C7A | 0x79BC | 0x28F9 |
| 0x6FF2 | 0x2E33 | 0x1AA5 | 0x089D | 0x69F7 | 0x3495 | 0x7768 | 0x3ADB | 0x30EE |
| x611A | 0x6B1A | 0x0BAD | 0x33AD | 0x0E35 | 0x77F5 | 0x3BD9 | 0x6533 | 0x60C2 |
| x2710 | 0x4348 | 0x0105 | 0x25D2 | 0x1DCD | 0x416C | 0x37CC | 0x4F41 | 0x103D |
| 0x13AA | 0x6F6D | 0x4F7C | 0x4779 | 0x006F | 0x7676 | 0x6698 | 0x40C1 | 0x0F10 |
| x6CFD | 0x3048 | 0x2811 | 0x6770 | 0x3213 | 0x0119 | 0x6BAC | 0x37B4 | 0x3848 |
| x4400 | 0x16C0 | 0x70CA | 0x5CFC | 0x15E6 | 0x6D5B | 0x3052 | 0x4CAE | 0x7264 |
| 0x0A76 | 0x1AB1 | 0x491C | 0x693A | 0x6CAE | 0x451A | 0x28F1 | 0x35C7 | 0x2722 |
| x390D | 0x5B26 | 0x6910 | 0x1364 | 0x461E | 0x4EF9 | 0x6EC0 | 0x4818 | 0x1AF6 |
| x7ACF | 0x1AA2 | 0x7FE0 | 0x6474 | 0x127C | 0x6B07 | 0x17B2 | 0x1440 | 0x6694 |
| 0x55BE | 0x6E52 | 0x0DDE | 0x7BD2 | 0x170F | 0x6AC0 | 0x0A52 | 0x05AF | 0x1AF5 |
| x5968 | 0x7C38 | 0x369D | 0x32E2 | 0x3FE9 | 0x46C1 | 0x4E21 | 0x333A | 0x656E |
| x43C4 | 0x5405 | 0x40C8 | 0x1D1F | 0x6EA6 | 0x1B19 | 0x7498 | 0x45C2 | 0x0AC3 |
| 0x6A11 | 0x5830 | 0x0788 | 0x2593 | 0x3BAB | 0x0B69 | 0x7E10 | 0x78A8 | 0x68A3 |
| x636B | 0x617A | 0x6903 | 0x70F9 | 0x5257 | 0x03AF | 0x6BBA | 0x0A12 | 0x0B94 |
| x624B | 0x06AB | 0x3A7F | 0x51EF | 0x2587 | 0x04F5 | 0x203F | 0x0917 | 0x2819 |
| 0x2403 | 0x46FF | 0x4B53 | 0x772D | 0x070C | 0x235A | 0x181A | 0x148D | 0x4822 |
| x42BB | 0x7E25 | 0x23E8 | 0x0B85 | 0x0523 | 0x1F20 | 0x218A | 0x6559 | 0x4438 |
| x3C8C | 0x38DC | 0x4A75 | 0x6638 | 0x34B6 | 0x699B | 0x18C0 | 0x6982 | 0x1CDB |
| 0x2BA0 | 0x6422 | 0x7C27 | 0x1F67 | 0x23EE | 0x7AF8 | 0x79A7 | 0x1D20 | 0x1354 |
| x574D | 0x55F6 | 0x4A4C | 0x038B | 0x1F7D | 0x1546 | 0x34FF | 0x78C7 | 0x1202 |
| x5BAD | 0x197A | 0x7412 | 0x767F | 0x0E05 | 0x2AC1 | 0x2B99 | 0x6118 | 0x20F6 |
| 0x38BD | 0x4812 | 0x5E21 | 0x413D | 0x13C4 | 0x258F | 0x1D3A | 0x137F | 0x2B93 |
| x6333 | 0x10C2 | 0x5462 | 0x740B | 0x2725 | 0x2284 | 0x518D | 0x0371 | 0x097B |
| x5179 | 0x7168 | 0x63B4 | 0x5738 | 0x1B67 | 0x38D5 | 0x1007 | 0x73B2 | 0x6D0D |

**Fig 6: A sample of generated UUMAT matrix**

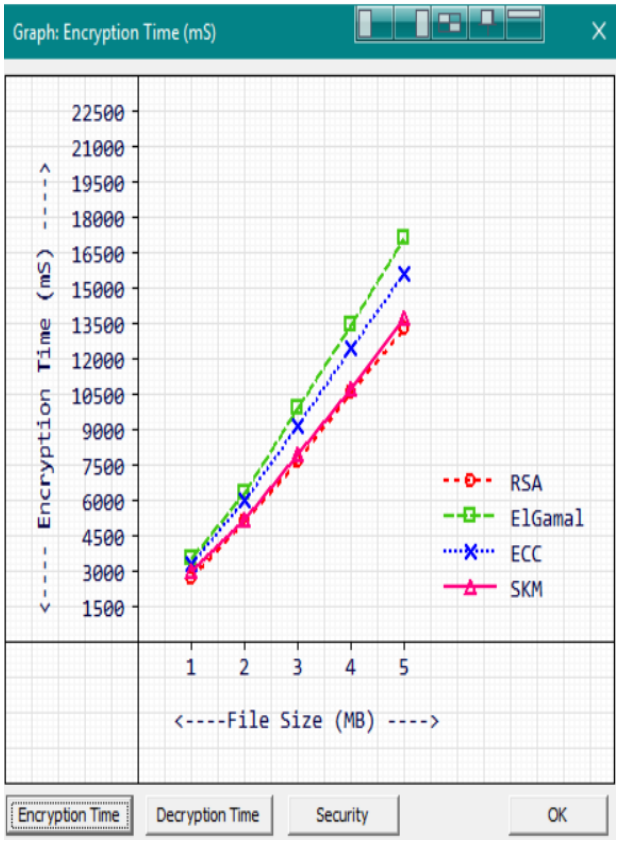



**Fig 7: Comparison of encryption time of proposed algorithm along with other algorithm**

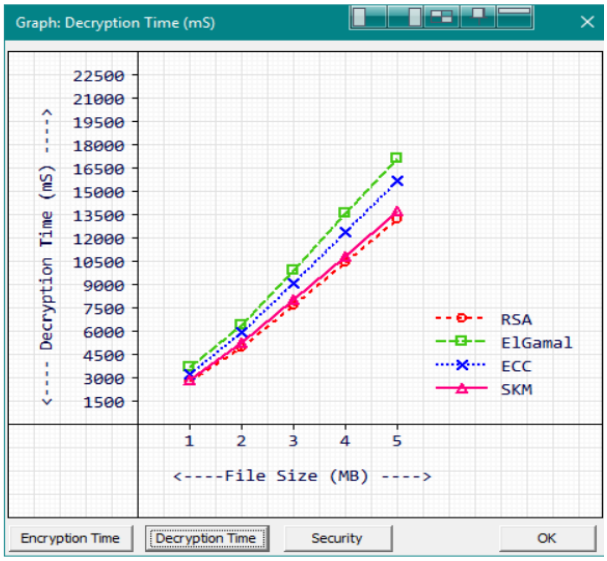



**Fig 8: Comparison of Decryption time of proposed algorithm along with other algorithm**

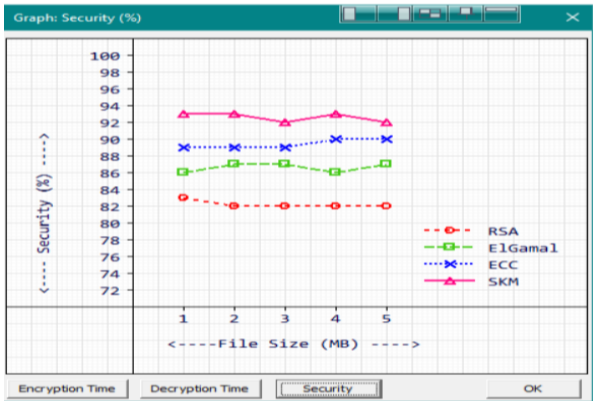



**Fig9: Security level of proposed algorithm compared with other algorithm**

```
This report is generated by SKM App on 22-07-2019 at 22:13:14

Parameter: Encryption Time (mS)
File Size (MB)      RSA     ElGamal     ECC     SKM
      1             2858      3622      3475    2882
      2             5008      6469      6009    5200
      3             7712     10080      9243    8004
      4            10550     13609     12492   10788
      5            13224     16980     15734   13786

Parameter: Decryption Time (mS)
File Size (MB)      RSA     ElGamal     ECC     SKM
      1             2743      3545      3440    2851
      2             5015      6456      5945    5286
      3             7781      9955      9292    7947
      4            10557     13510     12424   10884
      5            13294     17108     15699   13774

Parameter: Security (%)
File Size (MB)      RSA     ElGamal     ECC     SKM
      1               83        86        90      93
      2               82        87        89      92
      3               82        86        90      93
      4               82        86        89      92
      5               82        86        89      93

---- --- -- - end of report - -- --- ----
```

**Fig 10: Report generated for the encryption, decryption times along with security levels**

The above fig. shows the overall report generated for the encryption time, decryption time along with the security levels of the proposed along with the compared algorithms. From the generated reports it shows clearly that the security levels of our proposed SKM algorithm is more compared with that of RSA, ElGamal and ECC algorithm and also the encryption and decryption times are far better than ECC and ElGamal algorithms and narrow reachable to RSA.

## VI. CONCLUSION

The above simulation results obtained from the proposed work shows clearly that though the security levels of the proposed algorithm are high even with the existing one but depending on future it has to be raised further so that optimal security levels can be achieved this may even depend on the situation and keeping an eye on the future attacks it has to be maintained. The future attacks are going to be more challenge then the present one.

## REFERENCES

1. Yau, S.S. and An, H.G., 2010. Confidentiality protection in cloud computing systems. Int. J. Software and Informatics, 4(3), pp.351-365.
2. Rong, C., Nguyen, S. T., & Jaatun, M. G. (2013). Beyond lightning: A survey on security challenges in cloud computing. Computers & Electrical Engineering, 39(1), 47–54.
3. Veeraragavan, N., Arockiam, L. and Manikandasaran, S.S., 2017, February. Enhanced encryption algorithm (EEA) for protecting users' credentials in public cloud. In 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET) (pp. 1-6). IEEE
4. Oli, S.A. and Arockiam, D.L., 2016. 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 (IJARECE) Volume, 5.
5. Sheba Kezia Malarchelvi, P.D. & Manikandasaran, S.S. & Lawrence, Dr. L. Arockiam. (2019). MONCrypt: A Technique to Ensure the Confidentiality of Outsourced Data in Cloud Storage. International Journal of Information and Computer Security.
6. Maheshwari, V., Nourian, A., & Maheswaran, M. (2012). Character-based search with data confidentiality in the clouds. 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings.
7. Monikandan, S., & Arockiam, L. (2015). Confidentiality Technique to Enhance Security of Data in Public Cloud Storage using Data Obfuscation. Indian Journal of Science and Technology, 8(24).
8. Shaik Khaja Mohiddin & Dr. Yalavarthi Suresh Babu "A Technical Framework for Reducing and Analysing Vulnerable Cyber Attacks on the Cloud with the Inclusion of Snort based IDS Suitable to Carryout Forensic Analysis Further " JARDCS, Vol.10. Issue.10, PP: 323-333.
9. Mohiddin, Shaik & Suresh Babu, Y. (2019). A Relevance Technical Approach for Screening the Significance of IDS in Cloud Forensics. IJITEE vlo:4,Issue:2, PP-425-430.