I.J. Information Technology and Computer Science, **2016, 5, 28-36**
Published Online May 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.05.03

Security in Fog Computing through Encryption

Akhilesh Vishwanath
Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA
E-mail: avishwan@students.kennesaw.edu

Ramya Peruri
Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA
E-mail: rperuri@students.kennesaw.edu

Jing (Selena) He
Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA
E-mail: jhe4@.kennesaw.edu

***Abstract*—**Cloud computing is considered as one of the most exciting technology because of its flexibility and scalability. The main problem that occurs in cloud is security. To overcome the problems or issues of security, a new technique called fog-computing is evolved. As there are security issues in fog even after getting the encrypted data from cloud, we implemented the process of encryption using AES algorithm to check how it works for the fog. So far, to our analysis AES algorithm is the most secured process of encryption for security. Three datasets of different types are considered and applied the analysed encryption technique over those datasets. On validation, entire data over datasets is being accurately encrypted and decrypted back as well. We took android mobile as an edge device and deployed the encryption over datasets into it. Further, performance of encryption is evaluated over selected datasets for accuracy if the entire data is correctly encrypted and decrypted along with the time, User load, Response time, Memory Utilization over file size. Further best and worst cases among the datasets are analysed thereby evaluating the suitability of AES in fog.

***Index Terms*—**Cloud Computing, Fog Computing, DES Algorithm, 3DES Algorithm, AES Algorithm, Encryption.

I. Introduction

In the present world each and every organization from large scale to small scale industries are been relying on the cloud computing technology [2] to store their data as well as to use the resources as per their requirement. Cloud provides pay per use concept. The number of devices connected to internet has exceeded the world's population in the year 2010 and at present it is double the world's population and in the next five years from now it would be about 50 billion of devices connected to the internet.

At present it seems to be good enough to store and retrieve the data but as the number of devices connected to internet increases there would be definitely a problem in storage as well as information retrieval process. Hence, to overcome the above problem the fog computing [Fig. 1] concept has been introduced. In cloud computing concept all the data produced from the users will be directly stored into the cloud and then it is analysed with massive warehouses with analytics going on it and then decisions are made to act on data and eventually notifications are pushed to act on those decisions.

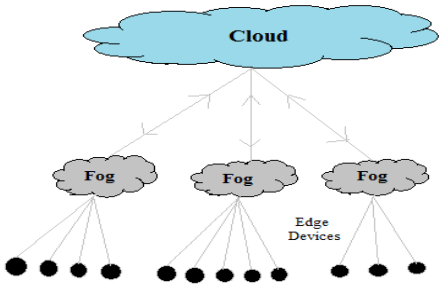


Fig.1. Fog Computing Architecture

In fog computing [6] the users will be notified what are the actions that are needed to be taken on the data and then analytics are applied on the received data and stored it into the cloud. In the fog computing process, application comes to the data not the data to the applications. Fog computing is said to be an extension of the cloud but not a replacement of it. As the number of the devices connected to the internet has been increasing at rapid speed and even advancement in the Internet of Things has led this number to increase drastically. In the future, the world would be full of sensors and there would be huge amount of data produced by these devices and storing data into cloud and retrieving is highly difficult. Hence fog has been introduced. For example, at present each of the airplane produces around 20 terabytes of data produced in an hour and they need to be stored into cloud space. This is done to all the planes around the globe where retrieving necessary data from cloud will not be possible leading for the need of fog computing. Cisco predicts that in the next decade, Internet of Things will be at 14.4$ trillion value of stake for companies and

Copyright © 2016 MECS

industries. Internet of Things had led evolution to the fog computing because of the increased number of devices producing massive amount of data.

In cloud computing there are many security issues as of man in the middle attack and even the encryption of data is not safe method for cloud. It does not identify the difference between user and attacker. It does not concentrate on the security of the data. Twitter was one of the example for the data theft in the cloud. Cloud provides various services for storing and accessing of the data in which the major problem is that, failure to provide security for the data against attackers. It is not providing any level of assurance to the user about the security of the data. Hence developing a more secured cloud is not enough because there would be continuous attacks happening on the cloud and there are chances that the data would be leaked or it might be lost forever. Hence, fog computing came into existence which is considered to be the most secured form of data storage.

Currently in Fog Computing, Decoy system [7] is being used as current security system for authorization of data. Decoy system is said to be a system of deception in which phony components are setup for enticing of the unauthorized users by giving vulnerabilities of a system thereby restricting unauthorized access to the network. It is a process where the files are full of traps and are included by the service provider. These decoy system consists bogus files in it with the sensitive names such as social security number, credit cards details as file names on it. These are most likely deceivable part for the attackers and there are chances that they might click on it and try to download it. Once they download the file, an alert will be generated and system will be notified with the attack. This decoy system method has been incorporated with the user behavior profiling where any unauthorized access will be notified to the system.

There are still some problems with the existing method leading to hacking and accessing of data in fog. This motivated us to think for encryption of data in the level of fog in cloud system. Hence, in this we are trying to achieve more security at the level of fog by introducing encryption to the data by using the Advanced Encryption Standard algorithm technique. The paper introduces AES algorithm in the fog environment, so whenever user sends data to fog for storing in the cloud, the fog will encrypt the data and send it to the cloud. And whenever user requests for the data, the encrypted data travels from cloud to fog and fog to end user and the data will be decrypted at end user. The Algorithm will be AES which is said to be most advanced and secured encryption algorithm. This paper majorly focuses on implementing this algorithm in mobile device as end user and even to show which type of dataset will be suitable for this kind of encryption technique making use of different types of datasets for evaluating their performance over encryption. It includes analysing of best and worst possible cases for each of the dataset so that suitability of AES in environment of fog can be evaluated. This paper contributes to security of data by introducing AES algorithm in fog computing, which makes the data of the end user more secured while the data is travelling from cloud to fog or fog to cloud.

The paper is organized as follows, Section II presents the literature review where the paper discusses about various types of encryption standard that are available now and why to use AES over other encryption standards. Section III provides the problem definition in which the paper define the existing problem in the fog computing and defining what our proposed methodology to overcome the problem. Section IV gives the solution of how the algorithm is used and the datasets that are used for encryption and decryption. Whereas Section V describes about the performance evaluation where it shows the metrics that are used to calculate the performance. Conclusion of the research paper and also the future work that will be performed are mentioned in Section VI.

II. Literature Review

Data Encryption standard (DES) was once most widely used encryption standard, which uses symmetric key algorithm for encryption of data. This was considered to be basic building block for the advancement in the modern cryptography in present world. DES [12] has 56 bits of key size and whereas the block size is 64 bit. For many applications when considered DES is said to be the most insecure technique for many applications. This is because of its key size which is 56 bits and this could be brute forced. Two companies together had break the DES algorithm key in 22 hours and 12 minutes. This shows how weak the algorithm is. Some of the attacks that could break the key faster than the Brute force are Differential Cryptanalysis, Linear Cryptanalysis and Improved Davies Attack.

The predecessor of the DES algorithm is 3DES [14] which is named as Triple Data Encryption Standard. Where 3 instances of DES are cascaded. The initial 56 bit key was sufficient, but the increase in computational power made brute force easy. Triple DES has made no changes to the previous DES algorithm except the increase in the key size, where it can have 56 or 112 or 168 bits of key size and whereas the block size remains same as 64 bits as DES. Triple DES was said to be 2½ time more secured than the DES algorithm. Even in Triple DES is vulnerable to security attacks meet in the middle attack. As DES algorithm was designed for hardware implementation, it is not reliable in hardware in the same way Triple DES do not function properly in software applications.

To overcome the above problem mentioned Advanced Encryption Standard (AES) [17] is considered as more effective. Which is considered to be the most advanced and secured standard for encryption of electronic data. AES is considered to be successor of the DES which uses standard symmetric key encryption for many of the US federal organizations. AES accepts of the key size of 128, 192, 256 bits of size. Whereas 128 is already considered to be unbreakable and there were many open competition held by many organization to break the key but it was

Copyright © 2016 MECS

never done. On comparing all the available encryption algorithms, AES would be the better and most secured type of algorithm that could be implemented in the fog. So far encryption technique has not been proposed for security in the fog computing. As a conclusion over all the different type of encryption techniques, AES can be considered more suitable and adaptable for the environment of fog. Hence this paper includes applying of AES algorithm for security of the data in fog computing through an edge device of mobile.

III. Problem Definition

Security is one of the major concern because there are lots of sensitive data around us. It could be any company pricing details, or even it could be national secret. All the data must be secured and should make sure that it has all necessary methods in it, which makes an attacker difficult to crack the key. The paper primarily concentrates on security and privacy of data as key part. In the system of cloud, though secured data is sent to the fog from cloud, predicting the security threat in the fog such as man in the middle attacks, we would like to add a second layer of security within the level of fog.

A. Existing Security system in fog:

In fog at present Decoy system [8] is considered as a security model, where in Decoy system user has to firstly signup and then give the login details and once he had logged in, they needs to answer the security question which was given while creating of the account. Decoy system is another method for trapping the attackers with the bogus files deceiving the attackers by showing the file with bogus names on it and only user who knows his data is able to know that this is bogus file and attacker will not be knowing about the difference between the bogus file and original file and once he clicks on the file and try to download it, the system will be notified about the attacker and thus the information could be secured from attackers. But method of security is not suitable because while answering there is risk that the attacker might guess the security question or any of the person who know the user very well might answer the question and he could hack the data, which is major security breach in the present architecture.

To avoid the above problems we are introducing a solution by Advanced Encryption Standard(AES) algorithm where data will be encrypted so that even if the attacker wants to access the data from present architecture of decoy system this makes him difficult to access the data. There are chances of leakage of data from cloud to fog or vice versa, hence by adding encryption to already encrypted data makes nearly impossible to access the data. By this method the data could overcome the man in the middle attack where an attacker will be continiously trying to enter into the data and this might be a threat to the sensitive data.

B. Proposed System Model

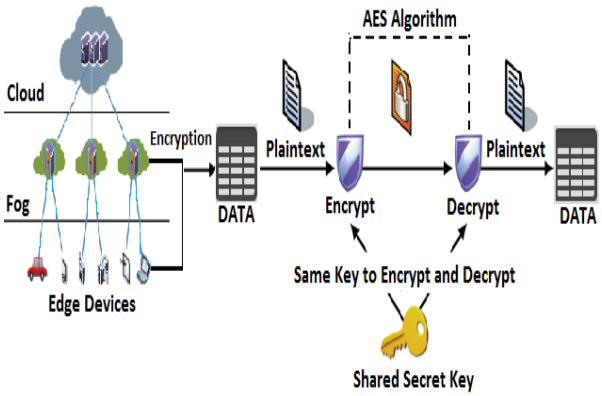


Fig.2. System Architecture

As in fig 2 technique of encryption has been chosen to apply over the data in fog for more security. From the study of related work, the formulations for AES encryption algorithm is taken for obtaining security in fog. The research aim is to achieve security in fog which is the second level of cloud system by making use of AES encryption algorithm and applying it over the selected datasets through deploying it in a mobile edge device and thereby collect the performance metrics over three datasets and evaluate best and worst cases in all the aspects of datasets. For analysing it in the fog environment, edge device of mobile is considered and application is designed so as to encrypt and decrypt the datasets chosen using the encryption technique of AES which is a symmetric key encryption making use of one common secret key for encryption and decryption. Different datasets having different sizes of data and of text, strings, and images are selected for testing the encryption method. Performance is evaluated for these datasets that are tested deploying in the edge device-mobile. On evaluating various factors like encryption, decryption time, utilization of memory, response time for each data set, best and worst possible cases are monitored.

In other words, the main aim of research is to provide security in the second layer of the cloud system-fog using the technique of AES encryption. Edge device mobile with 3 datasets are chosen for testing the case of security in fog. Finally performance for each aspect of dataset with respect to metrics like time, memory utilization, response time is evaluated and best, worst cases are monitored.

IV. Solution

AES encryption algorithm has been analysed. We have formulated the code using Java language objects and classes as main constructs. The code is been simplified by making use of static constructs to yield good space and time complexity.

Copyright © 2016 MECS

A. Proposed methodology:

Advanced Encryption Standard which is also known as Rijindeal is an encryption technique used by US government. AES is known for design based principle which has substitution and permutations and is said to be fast in both software as well as hardware. It has fixed block size of 128 bits and key size of 128, 192 or 256 bits. AES operate on the 4x4 column major order. AES will be performing many rounds of transformation to convert the plaintext to cipher. Below are the number of repetition for each of the size of bit key.

10 cycles of repetition for 128 bit key
12 cycles of repetition for 192 bit key
14 cycles of repetition for 256 bit key

There are four rounds of steps performed on the dataset they are

1) **The SubBytes step:** The substitution byte of each byte could be found in lookup table. The size of lookup table is 16×16. Substitute byte for given input could be found by dividing the byte into two 4-bit pattern, resulting an integer value from 0 to 15. These could be represented by Hexadecimal values from 0 to F. Where one of it is used to find the row index and another is used for column index to get into the 16×16 Lookup table. In fig 2 each of the SubBytes step of dataset is replaced with the 8-bit lookup table. The Substitution step concentrates on reducing the correlation between input and output bits at byte level.

Algorithm 1:

```
Void SubByte(byte[][] state) {
        for (int rw=0; rw<4: rw++)
        for (int cl=0; cl<N; cl++)

state[rw][cl]=SBox[state[rw][cl]];
}
```

- *Step 1:* As in algorithm 1, initially the dataset are stored in the block.
- *Step 2:* Next the each of the block will be considered which has size of 256 bit.
- *Step 3:* Now each block is divided into two and considered as row and column value of S box.
- *Step 4:* Now the value is taken from the S box and the data is replaced by hexadecimal value.
- *Step 5:* Now the 1-4 steps are continued for all of the blocks in the same way.

2) **The ShiftRows step:** The most important matrix representation of the state array happens here as in Fig.3.The ShiftRow transformation behaves like. 1) It won't shift the state array at all in the first row. 2) Circularly second row will be shifted by one byte to the left. 3) In the third row circularly shifting two bytes to the left. 4) In the fourth row it will circularly shift three bytes to left. In Shift Row step each of row will be swapped to its left depending on the index of row. In the same way for decryption, the corresponding rows will be shifted to opposite direction. The first row remains unchanged, in the second row the row will be shifted to right by one byte. Third row will be shifted to right by 2 bytes and in fourth row they are sifted to 3 bytes to right.

Algorithm 2:

```
Void ShiftRow(byte[ ][ ] state) {
byte[ ] s= new byte[4];
for (int t=1; t<4; t++)
        for (int d=0; d<N; d++)
                s[d]=state[t][(d+t)%N];
for (int d=0; d<N; d++)
        state[t][d]=s[d];
}
}
```

- *Step 1:* In algorithm 2, the hexadecimal values will be shifted to left, the row 1 will not be shifted.
- *Step 2:* In the row 2 it will be shifted to 1 byte to left, the loop will be continued until all of the blocks in the row are shifted to left.
- *Step 3:* In row 3 the block will be shifted to 2 byte left and continued for all of the bytes in the row.
- *Step 4:* In row 4 the block will be transferred to left by 3 bytes and the same process is continued.

3) **The MixColumns step**: In Mix Column each byte of the column in dataset is replaced with function of all bytes in the existing column as in Fig 4. And more importantly, each byte in the column will be replaced by the two times of that byte, plus three times of next byte, plus the byte that comes next, plus the byte the follows.

Algorithm 3:

```
Void MixColumn(byte[ ][ ] st) {
byte [ ] p= new byte[4];
for (int cl=0; cl<4; cl++) {
        p[0]=(0x02 # st[0][c]) ^ (0x03 #
st[1][c]) ^ st[2][c] ^ st[3][c];
        p[1]= st[0][c] ^ (0x02 # st[1][c]) ^
(0x03 # st[2][c]) ^ st[3][c];
        p[2]= st[0][c] ^ st[1][c] ^ (0x02 #
st[2][c]) ^ (0x03 # st[1][c]);
        p[3]=(0x03 # st[0][c]) ^ st[1][c] ^
st[2][c]  (0x02 # st[3][c]);
        for( int j=0; j<4; j++)
                st[i][cl]=p[j];
}  }  }
```

- *Step 1:* As in algorithm 3 we take one column at a time and start applying multiplication on it.
- *Step 2:* Each of the column is multiplied against the value of the matrix.
- *Step 3:* Now the results will be XORed and generates four result bytes for next state.
- *Step 4:* Now the multiplication will be applied to one matrix row to one state column.

4) The AddRoundKey step: In the AddRoundkey step each of the byte is combined with bytes of Roundkey using XOR operation**.**

Algorithm 4:

```
Void AddRoundKey(byte[ ][ ] sta)
 {
for (int cl=0; cl<N; cl++)
        for (int rw=0; rw<4; rw++)
                sta[r][c] = sta[r][c] ^
n[nCount++]
}
```

- *Step 1:* In algorithm 4, each of the 16 bytes in the state will be XORed with the 16 bytes of the expanded key for present round.
- *Step 2:* This will be continued for all of the rows in the state.
- *Step 3:* In the next round of AddRoundkey operation we will not call the first 16 bytes of expanded key but instead we use the bytes from 17 to 32.
- *Step 4:* In the same way we go on to other rounds in the state.

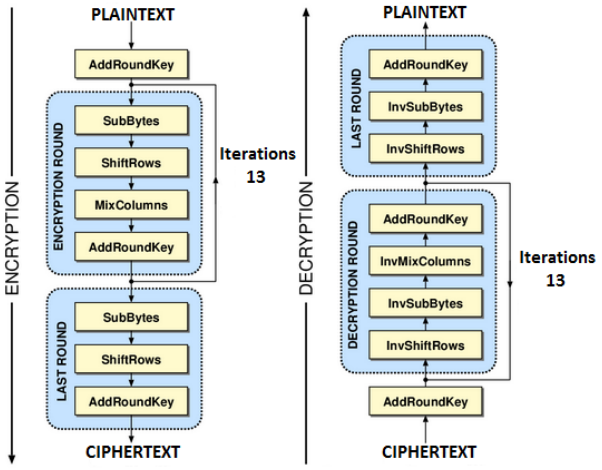


Fig.3. Architecture of AES Algorithm

In this way AES algorithm performs these four steps of operations in each of the iteration and generate a cipher text as fig. 3. The reverse process of the above four steps of the encryption will be the decryption where we perform all the above four steps and also the iterations are also performed, so as to generate the plain text.

B. Implementation:

To demonstrate AES we have implemented it on end device with android as operating system. In which the files will be selected and the user needs to give path of the file and provide a password, which will act as key and the same key needs to be entered while decrypting the data.

Fig.4. General UI Designed

The general UI is designed with options as in Fig 4 to upload file from the memory card of the device and then password have to be given for encrypting the selected dataset and then encrypt button allows the dataset to be encrypted.

To encrypt the data it will be using encrypt (byte [] in, byte [] key) method which will convert the plain text into the cipher text.

- generatekey(byte[] key) method is used for generation key.
- encryptblk(byte[] blk) method is used to encrypt the whole block of data.
- The above method also uses SubByte(), ShiftRow(), MixcColumn(), and AddRoundKey methods, for each of the iteration to produce a cipher text.
- In the overall operation the generated output will be in the cipher text in the format of byte [].

Fig.5. Encrypted dataset of amazon-cococola

The fig 5 shows the encrypted file that is generated on encrypting the file of amazon and coco cola dataset. It shows how the pattern is used and each and every line of

the data is encrypted.

In the same way for decryption it use decypt(byte[] in, byte[] key) for converting the cipher text to original plain text.

- generatekey(byte[] key) method is for generating the key.
- decryptblk(byte[] blk) method is used to decrypt the whole cipher text.
- The above method uses RevSubBytes(), RevShiftRows(), RevMixcolumns(), and RevAddRoundkey() for decrypting the data file.
- The generated output will be in the format of byte[].

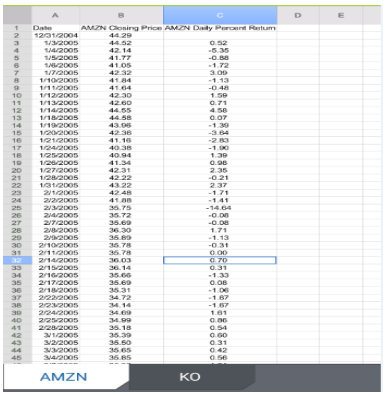


Fig.6. Decrypted file.

The Fig 6 shows the file that is decrypted after encryption of the dataset that is chosen. It reveals the original file of dataset of Amazon-coco cola on selecting the decrypt option and upload the encrypted file.

In the same way the algorithm was implemented on other two datasets and we have observed that the data is been encrypting as well as decrypting.

V. Performance Evaluation:

For implementing this encryption process over the data, we have chosen three datasets using some sources that contain different types of data of different sizes. The datasets which were considered for implementing the encryption over are:

- **Amazon.com & Coca-cola**-Daily returns, for ten years (2005 through 2014) for the stocks of two companies. [20]
  This dataset is chosen to indicate performance over small data as it is collection of information of two big companies amazon and Coca-cola which have same type of data with content of numbers only. The size of the dataset is 500kb and it contains structured data.
- **US Hospital Charge Data-** Contains information about inpatient and outpatient services.[21]
  This dataset is chosen to indicate performance over large data as it has many different types of fields with numbers, strings, special characters and is of huge size. The size of this dataset is about 5 Mb and it contains name, details of the person. This contains more than 150000 rows in it.
- **Crime Statistics-** This file contains informations about all the recent frauds and crimes in the recent times.[22]
  This dataset is chosen to indicate performance over data of bigger size and different types of content such as images, pictorial representations, strings, numbers. This is an unstructured dataset, which has a size of almost 10mb.

A) Simulation Settings

Now the implemention for AES encryption over these data sets are performed, where the entire data of datasets is encrypted and is decrypted as well without any loss of data. So as to perform such process over the layer of the cloud system Android device is considered and the algorithm is been deployed into the mobile.

Environment for the edge device:

- Development tool-Android SDK
- Programming language-JAVA
- IDE- Intellij
- Database-MySql.

For testing, apk is installed in One plus One mobile which is an cyanogen Android OS 12. This uses EMMC 5.0 to access and write 16GB or 64GB flash memory. 3GB of LP-DDR3 RAM. Qualcomm .801 Processor with 2.5GHz Quad-Core CPUs.

The metrics that are compared with the datasets over encryption are

1. User load vs CPU time

This considers how the CPU time varies when AES is being used on different sizes of datasets. So it gives the CPU time for each size of datasets which were selected.

2. File size vs Encryption time

This is used to calculate the time taken for encryption in case of each dataset of different sizes.

3. File size vs Decryption time.

This is used to calculate the time taken for decrypting each dataset of different sizes.

4. File size vs Memory utilization.

This gives results of how much memory will be utilized on using different size of datasets. So it gives the utilization of memory for each size of datasets which were selected.

Accuracy is also determined for each dataset to check if the complete dataset is encrypted and decrypted as well after decryption of the dataset without any loss of the data. For comparing and evaluating the performance of the datasets over encryption, a tool of Simulink is used by running it with the Linear analysis tool. In this the java code will be executed in the Simulink software tool

Copyright © 2016 MECS

and graph will be plotted accordingly, considering encryption and decryption while considering the given metrics.

The data for the metrics is collected using batch processing technique and functions of getMemoryinfo() for memory utilization and getCpuStat() for time taken by cpu and timestamp() for encryption and decryption time are used in the java scripts and graphs are plotted accordingly in tool of Simulink.

B) Simulation Results

1. User load vs CPU time

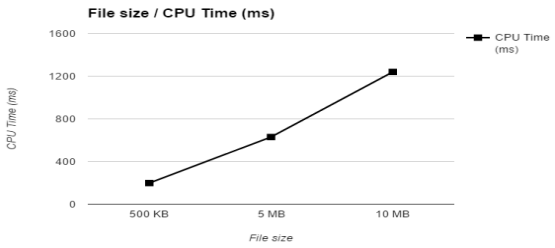


Fig.7. Comparing time taken by CPU for each size of dataset chosen.

Fig 7 shows how the CPU time varies for each sizes of dataset say 500Kb, 5mb, 10mb.

2. File size vs Encryption time

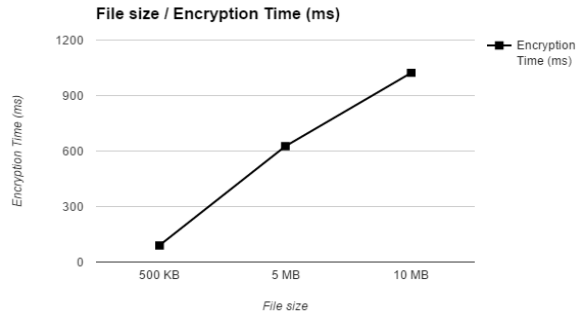


Fig.8. Comparing time taken for encrypting dataset for each of sizes of dataset.

Fig 8 shows how the encryption time is varying in case of each dataset depending on its size.

3. File size vs Decryption time

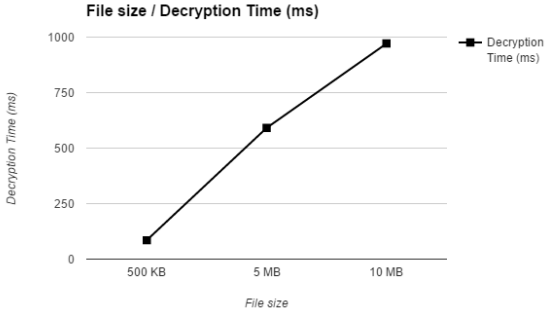


Fig.9. Comparing time taken for decrypting dataset for each of sizes of dataset.

Fig 9 gives how the time taken for decryption changes as the size of the dataset is varied.

Observation: Both encryption and decryption time are not same in case of each dataset. This change may be assumed because while we are encrypting the data each of the block must be encrypted sequentailly and whereas in the decryption we can apply XOR operation on the all the blocks paralelly.

4. File size vs Memory utilization.

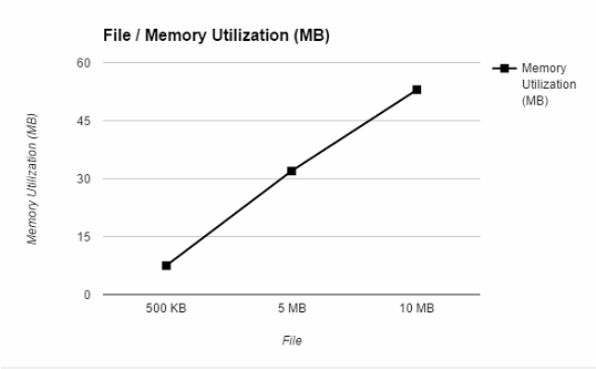


Fig.10. Comparing memory utilized for each dataset of different sizes.

Fig 10 gives the amount of memory that is been utilized for each dataset of different sizes.

Evaluation of Each Dataset

Another edge device-Laptop is considered so as to evaluate the best and worst cases for each of the datasets. It has features of Processor-I5, RAM-8GB, CPU@ 2.4 GHz..

Data Set-1 500KB

Table 1.

Device	CPU Time(ms)	Encryption Time(ms)	Memory Utilization(mb)
Laptop	91	87	7.5
Mobile	98	90	7.5

In the table 1, values for dataset which is of 500Kb with respect to each metric over two edge devices is given.

Best Case: Best case in the encryption time and CPU time.

Data Set-2 5Mb

Table 2.

Device	CPU Time(ms)	Encryption Time(ms)	Memory Utilization(mb)
Laptop	426	436	32
Mobile	630	626	32

In the table 2, values for dataset which is of 5mb with respect to each metric over two edge devices is given.

Best Case: In the encryption time.

Worst Case: In the CPU time.

Data Set-3 10Mb

Table 3.

Device	CPU Time(ms)	Encryption Time(ms)	Memory Utilization(mb)
Laptop	1103	973	53
Mobile	1240	1023	53

In the table 3, values for dataset which is of 10mbmb with respect to each metric over two edge devices is given.

Worst Case: In the encryption time, and CPU time.

Memory Utilization through AES in the mobile can be considered as average case.

C) Analysis:

Performance is evaluated over the datasets using different metrics. When each dataset is considered in mobile and laptop, there is lot of gap in time taken for CPU in case of larger datasets of 5mb and 10mb, but in case of smaller dataset, CPU time is almost equal compared in both the devices. This can convey that mobile can have a good CPU time utilization for smaller datasets rather than bigger datasets. When coming to the memory utilization, it will be the same for any device having RAM. Encryption time for the datasets is lower in laptop when compared to mobile that may be due to the difference in the speed of the processors. With respect to datasets, when taken in two different devices, it shows best results in high processing device. But as fog will be taken in every small device and as mobile is having best adaptability for security to smaller datasets, it gives good encryption and thus security in such edge device. This can show that fog can be protected and cloud system can be provided with second layer of protection through encryption in fog.

VI. Conclusion & Future Plan

Fog computing is considered to be one of the major part in the computing world, and as there are millions of devices connected and as IOT would be a major part of it, there may be a lot of issues on security. So our research here considered data security as the key factor and implemented Advanced Encryption Standard (AES) in the fog computing. This adds a second layer of security for data and makes difficult for intruder to sense the data. Different datasets are choosen and applied the AES algorithm for encryption and decryption for each of the dataset. Analysing of different metrics is done so as to evaluate the adaptability of AES in second layer of cloud system of fog. Consideration over time has also been undertaken to see that all the datasets could be processed within a fraction of time irrespective of its size and type.

As our future work, we would like to implement AES with key size of 512 bytes in fog.

References

[1] Zhidong Shen, Li Li , Fei Yan, Xiaoping Wu , Cloud Computing System Based on Trusted Computing Platform, International Conference on Intelligent Computation Technology and Automation, Volume 1, May 2010, On page(s): 942-945.

[2] Pearson, S., Benameur, A., Privacy, Security and Trust Issues Arises from Cloud Computing, Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference 2010, On page(s): 693-702.

[3] Rohit Bhadauria and Sugata Sanyal, A Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques. International Journal of Computer Applications, Volume 47- Number 18, June 2012, On page(s): 47-66.

[4] Mohammed, E.M, Ambelkadar, H.S, Enhanced Data Security Model on Cloud Computing, 8 th International Conference on IEEE publication 2012, On page(s): cc-12-cc-17

[5] Lizhe Wang, Gregor von Laszewski, Marcel Kunze, Jie Tao, Cheng Fu, Xi He, Andrew Younge, Cloud Computing: A Perspective Study, New Generation ComputingAdvances of Distributed Information Processing, Volume 28, Issue 2, April 2010, On page(s): 137-146.

[6] Sonali Khairnar , Dhanashree Borkar, Fog Computing: A New Concept to Minimize the Attacks And TO PROVIDE SECURITY IN CLOUD COMPUTING ENVIRONMENT, IJRET: International Journal of Research in Engineering and Technology, Volume: 03 Issue: 06, Jun-2014.

[7] Ivan Stojmenovic SIT, Sheng Wen, The Fog Computing Paradigm: Scenarios and Security Issues , Federated Conference on Computer Science and Information Systems pp, ACSIS, Vol. 2, 2014.

[8] Neha Shrikant Dhande, FOG COMPUTING: REVIEW OF PRIVACY AND SECURITY ISSUES, International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015

[9] Salvatore J. Stolfo, Malek Ben Salem, Angelos D. Keromytis, "Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud" IEEE, 2014, DOI 10.1109/SPW.2012.19.

[10] Sowmya Nag K , H.B.Bhuvaneswari , Nuthan A.C, IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD-192 BIT USING MULTIPLE KEYS, Research & Technology in the Coming Decades (CRT 2013), National Conference on Challenges in , vol., no., pp.1,7, 27-28 Sept. 2013

[11] Sumitra, "Comparative Analysis of AES and DES security Algorithms", International Journal of Scientific and Research Publications (IJSRP), Volume 3, Issue 1, January 2013 Edition.

[12] Sombir Singh, Sunil K. Maakar, Dr.Sudesh Kumar, "Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.

[13] Shah Kruti R., Bhavika Gambhava, "New Approach of Data Encryption Standard Algorithm", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[14] Karthik .S, Muruganandam .A, "Data Encryption and Decryption by Using Triple DES and Performance Analysis of Crypto System", International Journal of Scientific Engineering and Research (IJSER), Volume 2 Issue 11, November 2014.

[15] Majithia Sachin, Dinesh Kumar, "Implementation and Analysis of AES, DES and Triple DES on GSM

Network", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.1, January 2010.
[16] Gurpreet Singh and Supriya., "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", .International Journal of Computer Applications, 67(19):33-38, April 2013.
[17] Rishabh Jain, Rahul Jejurkar, Shrikrishna Chopade, Someshwar Vaidya, Mahesh Sanap, AES Algorithm Using 512 Bit Key Implementation for Secure Communication, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), Vol. 2, Issue 3, March 2014
[18] Kundankumar Rameshwar Saraf, Vishal Prakash Jagtap, Amit Kumar Mishra, Text and Image Encryption Decryption Using Advanced Encryption Standard, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 3, May – June 2014
[19] Advanced Encryption algorithm and its implementation. Accessed:July7,2015 https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
[20] Amazon-coco cola dataset with the company stocks. Accessed:July7,2015http://www2.stetson.edu/~jrasp/data.html
[21] Hospital Charge Data, Federal Government Accessed: July 7, 2015. http://www.data.gov/health/
[22] Sample Data set-CrimeStat, National Institute of Justice website.Accessed:july7,2015. https://www.icpsr.umich.edu/CrimeStat/download.html

Authors' Profiles

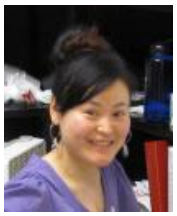
Akhilesh Vishwanath, Graduate Research Assistant at Kennesaw State University, born in India on 10th March 1993. Akhilesh is pursuing his Masters of Science in Computer Science Department at Kennesaw State University, Kennesaw, GA, USA with graduation in May 2016. Akhilesh has done Bachelors of Technology in field of Information Technology at Jawaharlal Nehru Technological University, Hyderabad, India, 2014.

He is currently working as a **Graduate Research Assistant** in the Department of Graduate Business. He worked as a Student Assistant with role of web developer in the department of CS, Kennesaw State University. He was offered with an employment by Infosys Private Limited- One of the world's biggest IT Company after his under-graduation. He worked as a web-developer intern at Internfever in his under graduation. After his under graduation he worked as Cloud Administrator for ITPlexus Pvt Ltd. He had presented various papers on Internet of Things and Fog Computing in Graduation seminars. His research interests are Internet of Things, Big Data, Cloud Computing and Mobile Computing.

Ramya Peruri, Graduate Research Assistant at Kennesaw State University, born in India on 4th may 1992. Peruri is pursuing Masters of Science in Computer Science at Kennesaw State University, Kennesaw, GA, USA with graduation in May 2016. Peruri has done Bachelors of Technology in field of computer science at Jawaharlal Nehru Technological University, Hyderabad, India, 2014.

She is currently working as a **Graduate Research Assistant** in the Department of Computer Science and doing Thesis degree in masters under the research area of computational models for HPC. She was offered with an employment by Tata Consultancy Services-Asian biggest IT Company after her under-graduation. She worked as a web-developer and Lab Assistant of Information Technology during her Under-Graduation. Several papers and posters on HPC were presented by her in Graduation seminars. Her Previous research interests include Mobile Computing and cloud computing with current Research Interests of Computational Models, HPC, Big Data, and Mobile Security.

Ms. Ramya was a member of IEEE in the past and currently she is a member of ACM Student Chapter. She was awarded for her presentation on "Concurrency in HPC" in the seminar of ACM. Her android project of Expense Tracker was awarded with merit of Excellence on poster day at Kennesaw, ACM Student Chapter.



Dr. Jing (Selena) He is currently an Assistant Professor in the Department of Computer Science at Kennesaw State University. She received her PH.D from the Department of Computer Science at Georgia State University. Her dissertation title is "Connected Dominating Set Based Topology Control in Wireless Sensor Networks", and her academic advisors are Prof. Yi Pan and Prof. Ying shu Li. She is now an IEEE student member and an N^2 Women member.

Before joining GSU, she received a B.S. degree in Electronic Engineering from Wuhan Institute of Technology, Wuhan, Hubei China. She received M.S. degrees in computer science concentrated in Artificial Intelligence from Utah State University and emphasised in Wireless Networking from Georgia State University. Her research interests include computational intelligence, machine learning, wireless networks, wireless sensor networks, network optimization, and cognitive radio networks. She also had intern experiences in IT industry. She worked with Electronic Arts in the Salt Lake Branch as a software engineer.

How to cite this paper: Akhilesh Vishwanath, Ramya Peruri, Jing (Selena) He,"Security in Fog Computing through Encryption", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.5, pp.28-36, 2016. DOI: 10.5815/ijitcs.2016.05.03