Granule based File Storage System with Secure Transparent Availability

[1]A. Suthan and [2] D. Kesavaraja

[1]HoD / Department of Computer Applications , [2]Lecturer,Department of CSE
Dr.Sivanthi Aditanar College of Engineering, Tiruchendur
83sutha@gmail.com, dkesavaraja@gmail.com

Abstract

The main aim of many security algorithms is to provide security to the data that we store. These algorithms provide security only untill the intruder gets hold of the file.With advancements in technology, any security mechanism can be cracked within a specified time period. In our paper we propose an alternative mechanism to this kind of security provision. We propose that the file be splitted into n number of particles and these particles be distributed within the system providing transparency to the user.The GFMS makes sure that the file is splitted and distributed inside the system in such a way that, even if some part of the file is retreived no information can be recognized. Inorder to increase the Security features the file is encrypted and then is being splitted.The information about the file pieces are stored in the GFMS_DB, which can be queried only if all the process other than the required system processes have been stopped. This is to ensure that the GFMS is free from impact of spywares. The implementation of this system has been tested for Image,Documents,video and Audio files.

Keywords: *Granule, File Merging, File Splitting,File Storage, Security*

1. Introduction

Maintaining a file with proper security has been a problem right from the inception of the file system concept. With the advancement of computing and emerging new technologies, the complexity increases[1][2]. Whatever security protocols that are being introduced do not provide security in the way that we expect. The Security protocols imopose a lot of restrictions on how we use the system and the file[3][4][5]. The following are the major issues when you consider implementing security for file system.

1. Spywares
2. Encryption
3. File Splitting

Spywares with their arrival introduced new challenges into the security arena. Earlier any hacker has to log into the system and browse the entire system for any valuable information. But the spywares made it very easy that they give you all the informaton about a file, where it is located and many other details,therby reducing the complexity of the hacker[6][7].

With the introduction of Encryption the threat was a little bit reduced but not completely removed, since any skilled programmer can write a program to decrypt the file only varying in time based on the length of the key used for encoding[8]. Also the process uses a software to encrypt your file, and if your system is infected with a spyware, then getting the key would still be a very easy process.

File splitting can be used at some areas inorder to provide security for the file system. Since the file is splitted into many parts, even if the user gets hold of some parts of the files, he will not be able to find out the entire content of the file[9][10][11]. The File can be encrypted before it is split, to provide more security. But here too the usage of spyware can nullify the effects of the security system. A spyware properly designed for the purpose can get hold of the information about the keys used for encryption and also about the various parts of the file,where they are stored and any other data that are required[12][13].

Also when we decide to split a file, certain criteria have to be considered like the type of file, size of a file,extension etc. These decide how efficiently the file is splitted. For example splitting a 2 mb file into 30 pieces and a 30 mb file into the same number of pieces would not be a correct perspctive of security. Thus these criterias bring upon their importance in deciding the file splitting process.

2. GFMS System Design

The process of GFMS is the very vital part of the system. When the process has to be initiated, the system gets from the user the necessary data for initiating the process,eg: a authentication.This authentication is to check whether the user is permitted to start the process.After the authentication is validated, the GFMS gets details of the file to be inputted and then moves to the silent mode.Before the silent mode starts, the state of the system is stored(GFMS_STATE). Once the silent mode starts, the GFMS kills all unnecessary process that are running in the system,both user initiated and system initiated which are not necessary for the GFMS to function. For example the Xwindow system can be stopped,since it has no relevance in the working of the GFMS. The GFMS in the silent mode functions on its own, without needing any manual interruption from the user other than a secondary authentication just to test whether the user initiated the process.This authentication ID will be a different one fron the authentication initially taken.This authentication is necessary since it is possible that a Robot program that was running in the system, without the knowledge of the user could have started the GFMS and could have supplied the Authentication details with the help of a Spyware. But after the system has moved into the silent mode, these spyware and robot programs would have been stopped and only the user can manually enter the Passcode.If the passcode is verified, then the GFMS starts the process. Once the GFMS finishes the process, it starts all the process that were running earlier to bring the system into the initial state. Thus the GFMS keeps data secure.

The working of the GFMS is as given below in the figure.

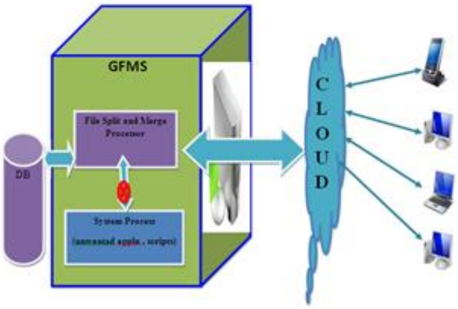


Fig 1 : GFMS Architecture

As shown in the figure above, the GFMS has the following sequences of process to complete its function.

All the processes are carried out in the silent mode only and also after the second authentication succeeds.

1. File Encryption process
2. File split process
3. Duplicate Naming Process
4. Granule Distribution Process
5. Data Updation Process
6. File Merge Process

2.1. File Encryption process

The file is encrypted with the AES algorithm.This process of encryption introduces the complexity for the hacker in identifying the data even if he gets hold of a piece of the file. The Encryption key is generated based on the authentication ID the user enters during the two steps of Authentication. This is to ensure that the Encryption key varies from one user to another, thereby making it impossible for one user to access the files of the other users. The Encryption Key is also stored in the GFMS_DB.

2.1.1. AES Algorithm for Security

1. KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule
2. Initial Round
    1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor
3. Rounds
    1. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
    2. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
    3. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
    4. AddRoundKey
4. Final Round (no MixColumns)
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey

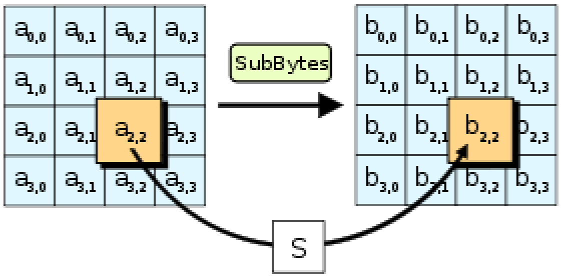

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

Fig 2 : AES Structure[15]

In Figure 2 describes detailed AES Structure of Sub bytes processing. The SubBytes step, one of four stages in a round of AES. By the use of AES Algorithm the files are transmitted in a secure way.

2.1.2. Key Generation Using Merkle–Damgård construction or Merkle–Damgard hash functiontitle

The Secure Key for AES Files Security , is generated by using Merkle–Damgard hash construction. With the help of Merkle–Damgard hash construction the message blocks are digested and generate an hash key. This hash key is used as a key for AES Encryption Process.

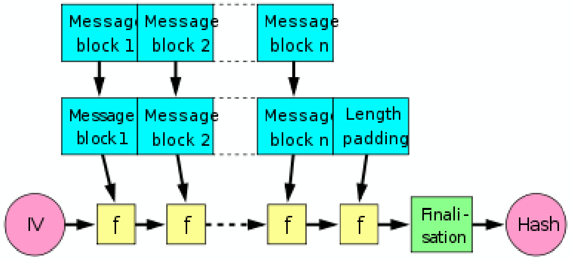


Fig 3 : Merkle–Damgard Hash Construction[14]

The Merkle–Damgard construction was independently proved that the structure is sound , if the function f is collision-resistant, then the hash function will be also. In order to prove that the construction is secure, Merkle and Damgard proposed that messages be padded with a padding that encodes the length of the original message. This will provide an efficient key for encryption process.

2.2.File Split Process

The file in consideration is inputed into the system. The System classifies the files based on the size of the file. The file is then splitted into the specified number of Granules depending upon its file size. The Granules are then renamed by the Duplicate name process. The Architecture of the File Splitting process is as follows.

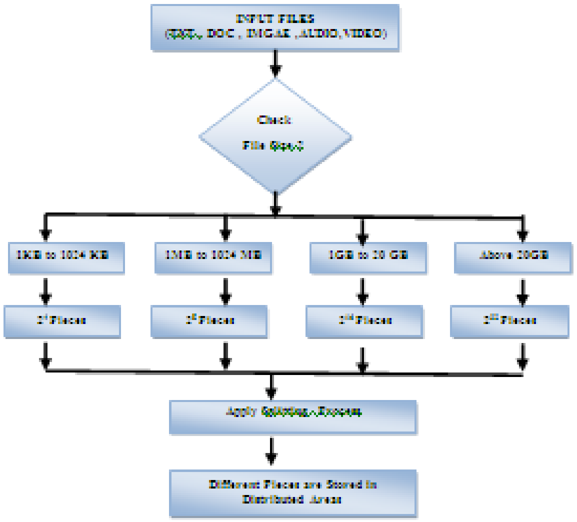


Fig 4 GFMS Flow Representation

2.3.Duplicate Naming Process

The file pieces are provided with duplicate names generated by the system. These information about the file pieces and their duplicate names are stored in the DB.

The Duplicate name are numerical and are generated randomly by the system. The generation of the duplicate names consists of the following procedure.

The Date, Month, Year, Hour, Minutes, seconds of the time of submission of the file is taken into acount along with the numerical user_ID and the result of the GFMS_CALC function is obtained. Then a randomly generated number is added with the output of the GFMS_CALC to form the name of the first Granule. For each granule a different random number is generated and added, with the GFMS_CALC, to produce GFMS_RAN. It is also ensured that the GFMS_RAN is unique by comparing with the previously generated and stored GFMS_RAN.The Series number of the Granule is finally concatenated with the GFMS_RAN, to produce GFMS_DUP. The GFMS_DUP produced for all the granules have to be stored in the DB for Future use.

2.4. Granule Distribution Process

The Granules that have been generated are distributed into the systems locations as hidden files. The granules are modified as hidden files inorder to prevent the user from accidentally deleting them, and also to ensure that the granules are not readily available for the hacker.

2.5. Data Updation Process

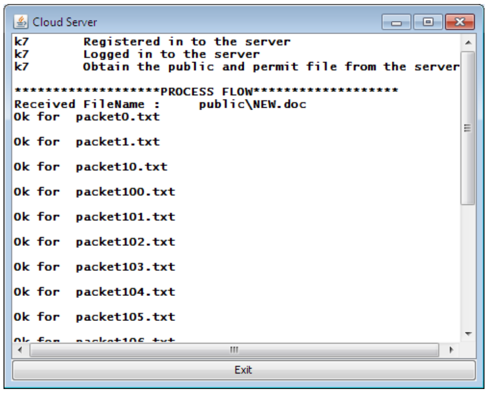
The GFMS_DUP is very essential, as it is the only key for identifying the granules that are related to the file requested for merging and also to find out the sequences for file merge. Also the information about the distribution of the granules, where it is located, etc are updated into the DB.This DB is secured by ensuring that it can be accessed only by a child process initiated by the GFMS. This feature is included to ensure that the DB should not be opened accidentally and the datas read.

2.6.File Merge Process

The merging of the file starts when the user requires to view or modify the file. The process starts with initiating the GFMS with inputing the file name required by the user and then the GFMS proceeds with the Same process of dual authentication, and starts accessing the DB. Then from the DB, the information about the number of file granules, their location, their duplicate names are extracted. Based upon this the file is merged and produced to the user in the original format it existed before the split process. Then the file is decrypted by retreiving the key from the DB. Once the file is ready for use by the user, the GFMS, brings the system to its original state, based on the information in GFMS_STATE.

3. Implementation

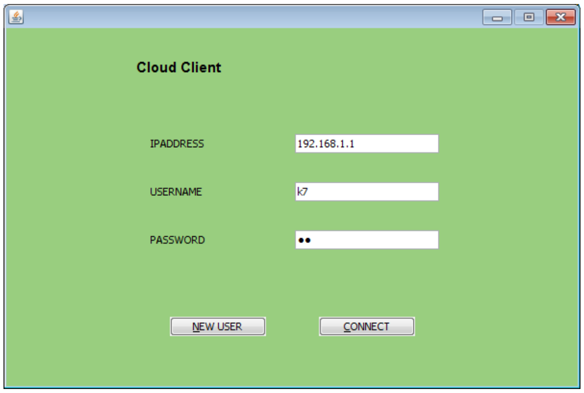
This GFMS scheme is implemented using java as a front end tool and My SQL as backend. The AES Security is implemented with the help of java.security package.



```
k7        Registered in to the server
k7        Logged in to the server
k7        Obtain the public and permit file from the server

******************PROCESS FLOW******************
Received FileName :     public\NEW.doc
Ok for  packet0.txt

Ok for  packet1.txt

Ok for  packet10.txt

Ok for  packet100.txt

Ok for  packet101.txt

Ok for  packet102.txt

Ok for  packet103.txt

Ok for  packet104.txt

Ok for  packet105.txt

Ok for  packet106.txt
```

Fig 5 : Cloud Server

Figure 5 gives the implementation of java swing structure of cloud server of the registered user "k7" and perform file spilt operation on the file "NEW.doc"



Cloud Client

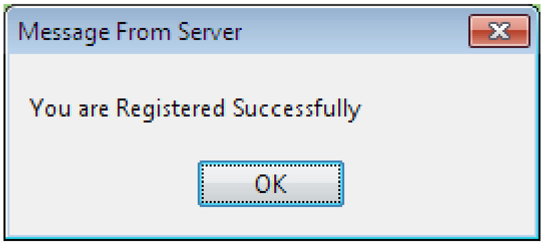
IPADDRESS 192.168.1.1

USERNAME k7

PASSWORD ••

Fig 6 : Cloud Server Connection

Figure 6 shows that the cloud connection to the server ip 192.168.1.1 with the help of username”k7”.Simply its an cloud client login page.

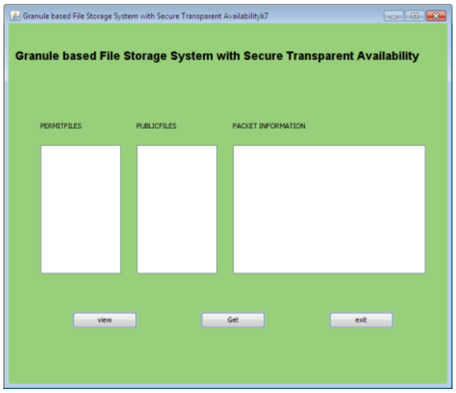


Message From Server

You are Registered Successfully

Fig 7 : Cloud Server Message

Figure 7 shows the java.swing message of validating the registered user status.This shows the valid user status .

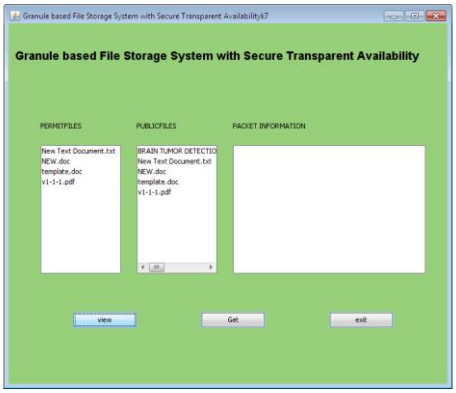


Granule based File Storage System with Secure Transparent Availabilityk7

Granule based File Storage System with Secure Transparent Availability

Fig 8 : Cloud Server File Process

Figure 8 gives the Local and Cloud Directories for retrieving and splitting files and its general packets. It will run inside the cloud server.



Granule based File Storage System with Secure Transparent Availabilityk7

Granule based File Storage System with Secure Transparent Availability

PERMITFILES

New Text Document.txt
NEW.doc
template.doc
v1-1-1.pdf

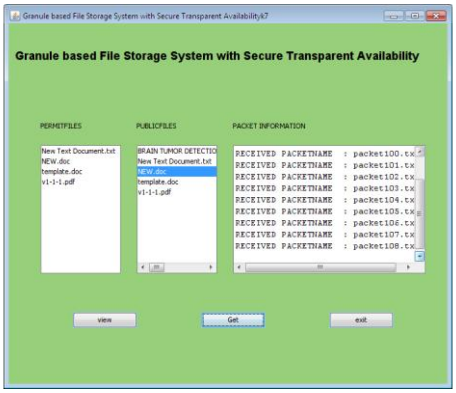
PUBLICFILES

BRAIN TUMOR DETECTIO
New Text Document.txt
NEW.doc
template.doc
v1-1-1.pdf

PACKET INFORMATION

Fig 9 : Cloud Server File Process

Figure 9 gives the Local and Cloud Directories for retrieving and splitting files and its general packets with sample datas. It will run inside the cloud server.



Granule based File Storage System with Secure Transparent Availabilityk7

Granule based File Storage System with Secure Transparent Availability

PERMITFILES

New Text Document.txt
NEW.doc
template.doc
v1-1-1.pdf

PUBLICFILES

BRAIN TUMOR DETECTIO
New Text Document.txt
NEW.doc
template.doc
v1-1-1.pdf

PACKET INFORMATION

```
RECEIVED PACKETNAME  : packet100.tx
RECEIVED PACKETNAME  : packet101.tx
RECEIVED PACKETNAME  : packet102.tx
RECEIVED PACKETNAME  : packet103.tx
RECEIVED PACKETNAME  : packet104.tx
RECEIVED PACKETNAME  : packet105.tx
RECEIVED PACKETNAME  : packet106.tx
RECEIVED PACKETNAME  : packet107.tx
RECEIVED PACKETNAME  : packet108.tx
```

Fig 10 : Cloud Server File Process

Figure 10 gives the Local and Cloud Directories for retrieving and splitting files and its general packets with an packet splitting empirical data’s . It will run inside the cloud server.

4. Performance Analysis

The performance analysis of the GFMS is tested with the help of Spearman's rank correlation coefficient[14] , In statistics, Spearman's rank correlation coefficient denoted by the Greek letter ρ (rho) . It assesses how well the relationship between two variables can be described using a monotonic function. If there are no repeated data values, a perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other data. In practice, however, a simpler procedure is normally used to calculate ρ. The n raw scores Xi, Yi are converted to ranks xi, yi, and the differences di = xi − yi between the ranks of each observation on the two variables are calculated from packets.

Table – 1 Distribution Comparison

File Distribution	Correlation Coefficient
0	0.2000
1	0.4330
2	0.5000
3	0.5330
4	0.5700
5	0.5888
6	0.8000
7	0.8000
8	0.8000
9	0.9660
10	1.0000

Table 1 gives the distribution comparison of data and its correlation coefficient values .In the distribution of 10 its gives +1 correlation coefficient.

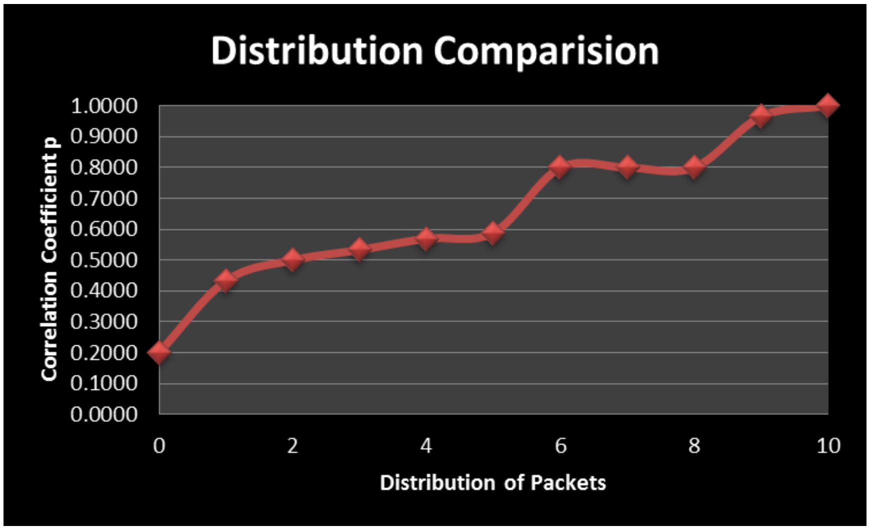


Fig 11 : Performance Analysis

Figure 11 shows that the graphical representation of distribution comparison of data and its correlation coefficient values. In the distribution of 10 its gives +1 correlation coefficients.

5. Conclusion

The system provides a valid file storage system with necessary security thereby making it a reliable one. Based upon the performance analysis it can be found that the system provides a reliable outcome. The system can be used in areas where reliable and secure file storage is expected. In this New approach also provides possibility of exploring a possible solution to the recent problems of attacks on highly secure data in the government offices by other countries. The system also seems to provide a solution to the problem of security while storing valuable data in cloud based storages.

Acknowledgement

We would like to thank our management and co faculty members for helping and bearing with us during the period in which we were involved with this work. Also we would like to thank all the authors of the papers we have referenced to bring up our work, without which it would have been very difficult to complete our work

We are the heaviest burden to our parents in many cases yet their passion upon us gave us support to get this work done. We dedicate this research to our beloved parents.

References

[1] S.Thomson and T.Narten, “IPv6 Stateless Address AutoConfiguration”, RFC 2462, Dec 1998.

[2] C.Perkins, J.Malinen, R.Wakikawa, E.Belding-Royer, and Y.Sun, “IP Address AutoConfiguration for Ad Hoc Networks”, IETF Internet Draft, Draft-ietf-manet-autoconf-01.txt, November 2001.

[3] R.Wakikawa, J.Malinen, C.Perkins, A.Nilsson and A. Tuominen, “Global Connectivity for IPv6 Mobile Ad Hoc Networks”, IETF Internet Draft, draft-wakikawa-manet-globalv6-03.txt, October 2003.

[4] I.K. Park, Y.H Kim and S.S.Lee, “IPv6 Address Allocation in Hybrid Mobile Ad Hoc Networks”, The 2nd IEEE workshop on Technologies for Embedded and Ubiquitous Computing Systems, May 2004, pp.58-62.

[5] Johnson,D.Maltz and Y.Hu.The Dynamic Source Routing Protocol for Mobile Ad hoc Networks .Internet Draft:draft-ietf-manet-dsr-09.txt,2003

[6] Z.Haas and M.Pearlman.The Performance of Qury Control Schemes for the Zone Routing Protocol.IEEE/ACM Transactions on Networking,9(4):427-438,2001

[7] C.Ho,K.Obraczka,G.Tsudik and K.Viswanath.Flooding for Reliable Multicast in Multi-hop Ad hoc Networks.International Workshop in Discrete Algorithms and Methods for Mobile Computing and Communication,64-71,1999.

[8] J.Gomez,A.T.Campbell,N.Naghshineth,andC.Bisdikian,"Conserving Transmission power in Wireless Ad hoc Network" Proc.Ninth IEEE Int'1 Conf.Network Protocols(ICNP '01),pp.24-34,2001

[9] J.Dorsey and D.Siewiorek, "802.11 Power Mnagement Extension to Monarch ns"Technical Report CMU-CS-04-183,School of Computer Science,Carnegie Mellon Univ,Dec.2004.

[10] VINT Project, The UCB/LBNL/VINT Network Simulator-ns(version 2),http://www.isi.edu/nsnam/ns,2009

[11] Majid Khabbazian,and Vijay K.Bhargava,"Efficient Broadcasting in Mobile Ad hoc Networks"IEEE Trans.on mobile computing,vol. 8,no 2,February 2009.

[12] Y.Ko.and N.Vaidya.Location-aided Routing(LAR) in Mobile Ad hoc Networks.Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking(MOBICOM),66-75,1998.

[13] Y-C Tseng,S-Y.Ni,and E-Y.Shih,"Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network"IEEE Trans,Computers,Vol. 52,no.5,pp.545-557,May 2003

[14] http://en.wikipedia.org/wiki/Spearmans_rank_correlation_coefficient

[15] http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Authors

A. Suthan has completed his B.Tech Degree from the Department of Information Technology from Arignar Anna College of Engineering ,Chennai , Under Madras University Chennai in 2004. He completed his M.Tech from the Department of Information Technology from Sathyabama University, Chennai in 2008. He is currently acting as the HOD at the Department of Computer Applications, in Dr Sivanthi Aditanar College of Engineering. His research interests include Open Source Software's , Distributed Computing , Grid and Cloud Computing , Developing Applications of these technologies in Real Time .Mr Suthan.A is the life member of ISTE . He has presented and published many national and international conferences and journals.

D. Kesavaraja has completed his B.E Degree from the Department of Computer Science and Engineering from Jayaraj Annapackiam CSI College of Engineering, Nazareth, Under Anna University Chennai in 2005.He has completed his M.E Degree from the Department of computer science and Engineering from Manonmaniam Sundaranar University , Tirunelveli in 2010.He is currently working as a Lecturer at the Department of Computer Science And Engineering, in Dr Sivanthi Aditanar College of Engineering , Tiruchendur. He is a co-author of a book titled "Fundamentals of Computing and Programming" ,ISBN 978-81-8472-099-0.His research interests include Intrusion Detection, Web Development and Cloud Computing. He published his papers in 3 National and 8 International journals and many national and international conferences.