JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Secure Cloud Computing Based Framework with Privacy Preservation and Indexing

[1]Nitin Gunvantrao Gulhane, ME Student, Computer Science & Engineering Department , Matsyodari Shikshan Sanstha 's , College of Engineering &Technology, Jalna

[2] Prof. Mr. Gajanan P Chakote, HOD ,Computer Science & Engineering Department , Matsyodari Shikshan Sanstha 's , College of Engineering &Technology, Jalna

***Abstract*—**

Storing private and professional data on the cloud server has a likely chance of losing possession of large size of data. Users convey data using Internet which can be affected by malicious attacks. So, to preserve the privacy of documents, data should be outsourced in encrypted format. In this proposed system, Secure Querying algorithm is used to retrieve top matching results from huge encrypted answer set relevant to user interest. To retrieve the documents of user interest, multi-keyword query has to be fired which will result in accessing the top k ranked documents. Lucene indexing technique has been developed to build an index of keywords from documents. Index will store hashed values of keywords. Blowfish algorithm is used for encryption and decryption of data.

Keywords – Secure Cloud Computing, Blowfish, Indexing, Lucene

I. INTRODUCTION

Cloud computing has become famous as of late because of a few benefits over conventional computing models. Common benefits incorporate adaptability, versatility, nimbleness, energy productivity, and cost saving. Thus, it has been relied upon to be a prevailing computing model later on. By utilizing cloud computing in shrewd matrices, we not just location the issue of huge data the board yet additionally give a high energy and cost saving stage. It is on the grounds that 1) the structure can scale exceptionally quick to manage changes in the measure of handling data and 2) it can give a high use of computing assets. All things considered, before our work, starting endeavors have been dedicated to demonstrate that cloud computing can fulfill necessities of data the board in these frameworks. Specifically, in, properties of savvy network and cloud computing were examined to demonstrate the connection between them. Besides, in, use instances of a shrewd framework were examined to comprehend nitty gritty prerequisites of data the executives, and cloud computing properties were considered to show that they meet the necessities. All things considered, none of these stirs thinks of a substantial plan for data the board in savvy frameworks other than rather unique examinations.

Placing our confidential data in hand of cloud storage is critical because of the term data confidentiality has outstanding importance. Our original plain data must be accessible by only trusted parties. Data must be encrypted and must not be accessible by untrusted third parties such as cloud providers, intermediaries, and Internet [1]. Achieving these goals has different levels of complexity. As type of cloud services changes, level of complexity changes.

To permit the centralized data repository, and access to the computer services or resources whenever desired by clients, cloud facilitates huge assemblage of remote servers to be in a network. Now a day's many people used cloud storage to save their official documents, health documents, financial documents, etc [2]. As documents are saved and transferred on cloud, owner of documents doesn't hold the physical possession of these data. So, robust privacy preservation scheme is required to ensure the safety of documents at cloud side as cloud is not confidential server. Therefore to ensure data confidentiality and unauthorized access to the data placed on cloud, owners have to adopt encryption process on data while placing confidential data on cloud.

User will accumulate encrypted data on cloud. So, traditional data searching process will not be effective and efficient for encrypted cloud storage [3]. To retrieve efficient results on encrypted data, multi-keyword query should be structured and fired. This multi-keyword query helps to find out top related data belonging to user concern.

Existing methods works on single query searching procedures or on single user with Boolean keyword search procedures. The existing searching methods for data retrieval are only restricted for single keyword queries. These searching methods fetch all the related data corresponding to the specified keyword without ranking the data of user interest.

II. RELATED WORK

Smart grid information management typically includes three essential assignments: information gathering, information handling, and information putting away. For information gathering, since smart grids need to gather information from heterogeneous gadgets at various areas, the principle research challenge is to construct effective correspondence engineering.

A few arrangements have been proposed to address this test and a large portion of them can be found in the new studies, for example, [5], [7], and [7]. As far as information handling, information coordination additionally lays a test as information can be gotten from various gadgets, which might utilize various information designs to deal with the information. Luckily, a proposition for normalization of information structures utilized in smart grid applications has as of late been proposed to resolve this issue of information between operability. Notwithstanding, how to handle a lot of information proficiently still remaining parts as a major test. Cloud computing seems to fulfill this need and furthermore fulfill difficulties of information putting away. Therefore, starting chips away at cloud computing and smart grids have been delivered. In properties of smart grid and cloud computing were broke down to demonstrate that cloud computing is a decent contender for information management in smart grids. Likewise, in [4], use instances of a smart grid were talked about to comprehend nitty gritty prerequisites of information management and cloud computing properties were considered to show that they meet the necessities. These two works are not quite the same as our own in that they just introduced investigation while we present a substantial plan for the stage just as a security answer for it.

In earlier systems, binary maps are used for effectual keyword searching which utilizes the public key encryption technique. Significantly, these existing methods work for single users. Queries used to search data are structured in conceptual approach and therefore lack in hiding of search patterns.
In some existing methods, for generation of queries user must have to gather all necessary information about all valid keywords along with its positions. Cloud storage acts as database repository as it stores all information and documents of users. One of the cryptographic primitive is searchable encryption which allows private keyword based searching over encrypted database.
Existing system has some of the disadvantages as follows:
1. Single-keyword search without ranking
2. Boolean keyword searching without ranking
3. Single-keyword search with ranking
4. Rarely sorting of the results i.e. no index creation and ranking
5. Single User search

A. Comparison of Secure Algorithms

Symmetric encryption is also called as secret key encryption which uses a single key for both encryption and decryption processes. Same key is utilized to convey data for encryption process and in decryption process. Conveying key using Internet can be affected by malicious attacks. So if both parties are aware about the keys then symmetric encryption technique is more appropriate for communication to transfer secrete data over internet.Asymmetric encryption is also called as public key private key encryption which utilizes two distinct keys for encryption and decryption processes, public key and the private key respectively. The key can be transformed into an array of bytes for the intention of storing and transferring secretes data. This process is called as wrapping. When key is required for decryption purpose then array of bytes is again converted in key. This process is called as unwrapping. There are various algorithms for data encryption using symmetric and asymmetric encryption keys. Few of them are stated and compared below:

DES (Data Encryption Standard) is the very earliest encryption standard developed by NIST (National Institute of Standards and Technology) and commonly used in the commercial, military, and other domains. It was emerged in 1947 by IBM and accepted as a national standard in 1997. DES standard is public & the design criteria used are classified. DES is a 64-bit block cipher which utilizes 56-bit key [2]. This algorithm contains permutation of sixteen rounds block cipher. DES algorithm carried out permutation and substitution procedures on every block of plaintext data which is afterwards EX-ORed with the given input. The same process is carried out 16 times with different sub keys. It is vulnerable to security attacks and secrete data can be easily retrieved.

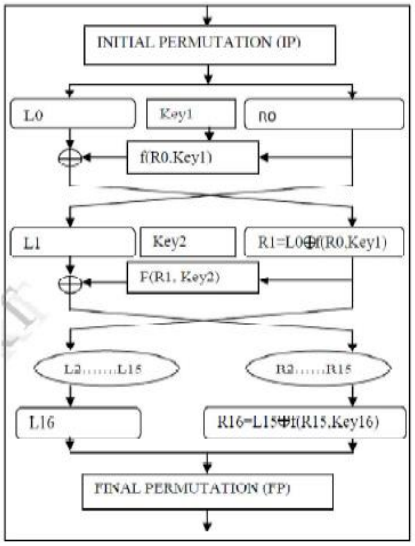


Figure 1: DES algorithm

AES (Advanced Encryption Standard) was discovered by scientists Joan and Vincent Rijmen in the year 2000. It replaced DES as the official standard of US National Institute for Standards and Technology (NIST). AES works on variable key sizes and variable block sizes of 128, 192 or 256 bits. AES uses Rijndael block cipher and Rijndael key [5]. Due to the number of permutations and combinations in AES contributing to its higher complexity, AES has a higher security as compared to DES.

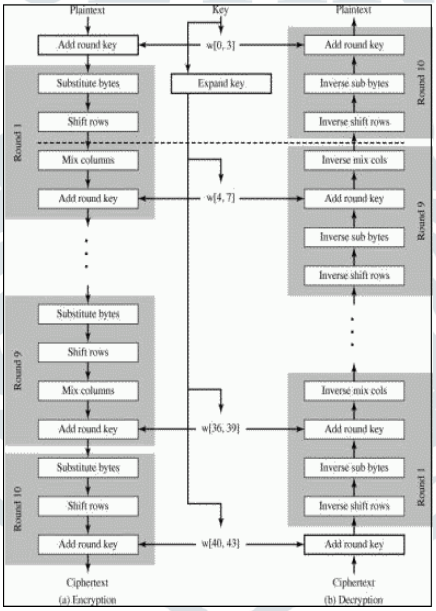


Figure 2: AES algorithm

Blowfish is a flexible algorithm which can run on PC, smart Card microprocessors or dedicated encryption hardware. It utilizes 128, 192 or 256 bit keys and operates on 128 bit-blocks [8].

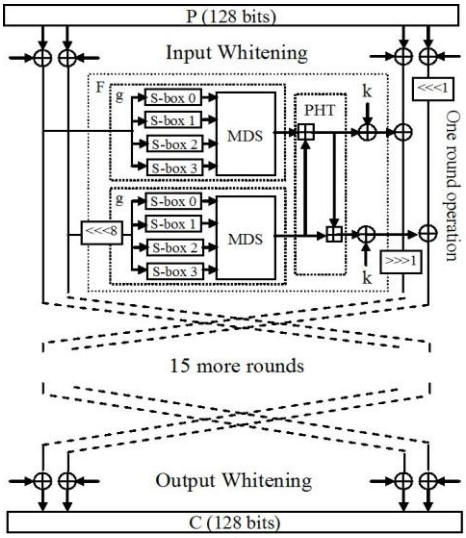


Figure 3: Blowfish algorithm

Advantages of Blowfish algorithm over other algorithm:
1)Blowfish is a fast running substitute forDES and IDEA.
2) Also Blowfish provides a good encryption rate in software.
3) Blowfish is unpatentedand available in public domain. This has led to itspopularity.

III. PROPOSED METHOD

A. System Development

Major objective of the proposed system is to extract the top relevant data similar to users query as well as to enhance experience of users while searching. Single keyword searching technique generates unwanted traffic of data which is not that much related to users query. Hence we need to develop robust system which will provide efficient results with multi-keyword searching.

As cloud storage is not trusted to keep our plaintext data, we need to save data in encrypted form to ensure privacy. User will register on cloud storage by registration process. Key generation center (KGC) will provide login credentials to user. After successful login user can store their data on cloud. User can upload his confidential data and documents on cloud. Next time whenever user will ask for uploaded document, it will retrieve in encrypted form. So that only user of data is able to decrypt the encrypted data with key.

Once document get uploaded on cloud server, proposed system extracts attributes of files such as file name, file date, author name, title of paper, etc,. Results are ranked and retrieved using indexing techniques. Lucene indexing is efficient among other indexing techniques and hence used in the proposed system. Top relevant results are also retrieved matching with user queries using top-k query.

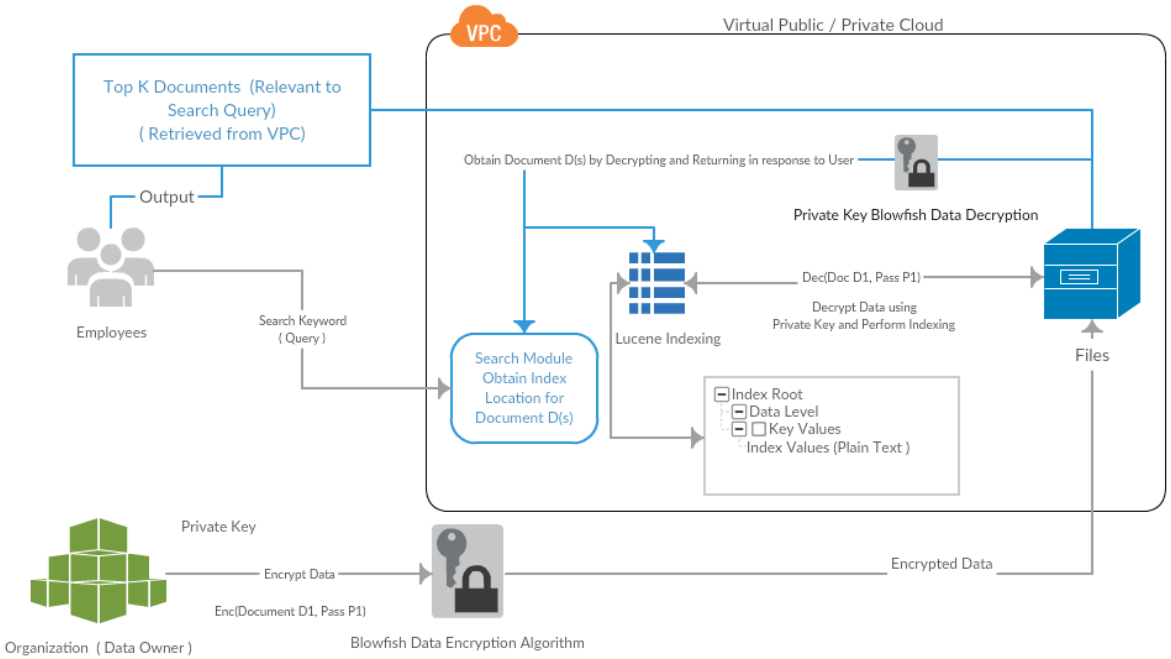


Figure 4: Architecture of proposed system

Proposed system will solve the problem of searching of information from encrypted data stored on cloud with the help of multi-keyword search query which also preserves system-wide privacy in cloud computing. Proposed system used Keyword relevance technique as an intermediate similarity measure which extracts keywords from search query to match with keywords stored on cloud. Ranking system is developed which provides the effective retrieval of results using lucene indexing. Top matching results are retrieved using Secure Querying so that unwanted network traffic gets avoided. Results relevant to user's interest are retrieved efficiently using multiple keyword search procedure. Single Keyword search query will yield too much unnecessary results so that there is necessity of supporting multi-keyword query.

B. Lucene Indexing

When person wants to upload his official or personal data on the cloud server, keywords are get abstracted from that documents and index will be generated for all abstracted keywords [1], [4]. Hashed values of keywords are stored as index. When document will get added and removed from cloud storage, index will get restructured accordingly.
In proposed method, for generating index of keywords extracted from documents, Lucene indexing technique is used. Lucene possesses full-text search engine architecture which delivers complete query and indexing engine. Lucene indexing technique delivers distinct advantages such as indexing is like file format and application platform independent. It also delivers searching of data over documents [9]. A Lucene technique provides continuous updations into indexes of fetched keywords when they are get added and removed from storage space.

Lucene is very wealthy and potent full-text search library written in Java. Lucene indexing is utilized to deliver full-text indexing over both database objects and documents in various formats.Lucene indexing API is independent from other file formats. Textual data from various resources such as pdf, HTML, Microsoft word, openDocument is get extracted first. In Lucene indexing, to construct an index, whole document gets scan first so as to produce list of postings. Occurrence of word in a document is get described by postings [10]. Posting generally consist of word, document identifier and frequency of the word within that document.

Indexing process is one of the main functionality delivered by Lucene. Figure shown below describes the overall indexing procedure carried out by lucene and use of each class. IndexWriter is the most significant and main module of the indexing process.

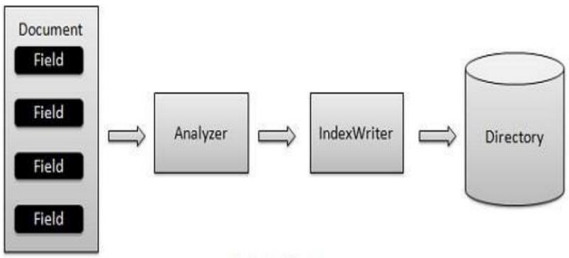


Figure 5: Indexing Process

Supporting full-text search using Lucene indexing executes two steps:
(1) generating a lucence index on the documents and on database objects and
(2) parsing the user query and looking up the prebuilt index to answer the query.

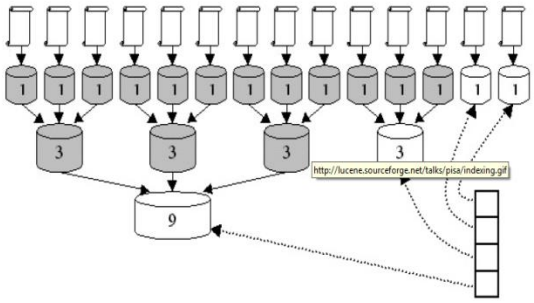


Figure 6: Working of Lucene indexing

Above figure describes complete working procedure of Lucene. As soon as the document will gets uploaded on to the cloud server index segment will get generated. In accordance with merge factor, index segments will get merged. Index contains hashed values of keywords

C. Secure Querying

Information retrieval systems utilize diverse approaches to rank query answers. Users are more apprehensive for the most significant i.e., Secure Querying response from large answer space. Some emerging applications claim for support of top-k queries. In case of the Web, Secure Querying can be used to provide effectiveness and efficiency of meta search engines. Distinct applications are also presents in fields of information retrieval and data mining.

Goal of this Secure Querying algorithm is to fetch top matching results from large record set. Secure Querying helps to search more precise answers from specified record set that equivalent with filtering keyword, and assemble relevant answers in accordance with their scores. Steiner tree is constructed with each result set which is basically an XML tree [3]. The constructed Steiner tree will holds the list of all records which are already assembled in accordance with their scores.

Secure Querying algorithm utilizes scores documents against keywords. This algorithm is used to rank “Tuple Units”. A tuple unit consists of a set of similar tuples which again consists of query keywords. These tuple units are used to construct tuple set. When two tuples are directly associated with each other then they are merges into one single tuple. Direct scoring and indirect scoring techniques are used in top-k approach to rank every tuple.

Secure Querying scores every tuple according to two scoring methods, namely, Direct Scoring and Indirect Scoring. Direct Scoring depends on TF-IDF algorithm. TF-IDF stands for “Term Frequency, Inverse Document Frequency”. TF-IDE presents a way to score the words depending on how repeatedly they come into sight across various documents.

Scores are termed according to the following criteria:

1) If a word appears frequently in a tuples, it is termed as important and is scored high.

2) If a word appears in many tuples, it is termed as a unique identifier and is scored low.

Therefore, common words like "the" and "for", which appear in many tuples, will be neglected. Words that appear frequently in a single tuple will be scale up.

The second type of scoring method is Indirect Scoring. Indirect Scoring scores a tuple unit based on the keyword that is indirectly present in the tuple unit. The keyword is scrutinized against each tuple unit for indirect similarity.

D. Blowfish Algorithm

Encryption approaches are categorized in two parts as symmetric key encryptions and public key encryptions. Blowfish falls under Symmetric algorithms. Blowfish algorithm is developed by Bruce Schneider and also known as block cipher. As name in indicated, it divides message into fixed length blocks while encryption and decryption process. Blowfish algorithm can be used as quick alternative for some existing encryption processes and as it delivers robustness, none of the attack becomes successful against it. Blowfish algorithm consists of key expansion a technique which makes it more complex to break. It also takes less time for encryption process as compared to other encryption algorithms as AES, DES.
Blowfish contains 16 rounds. Each round holds XOR operation and a function. Each round consists of key expansion method and data encryption process [6]. Key expansion generally used for generating initial contents of one array and data encryption uses a 16 round Feistalnet work. Foundation of Blowfish Algorithm is Feistel Network which iterates simple encryption function 16 times [12].

Proposed system employ Blowfish encryption algorithm for encryption of data. Homomorphic encryption is implemented along with Blowfish algorithm. Homomorphic encryption permits computations on ciphertext directly. So that, generated encrypted outcome when again gets decrypted will exactly be similar because all operations are carried out on plaintext. Homomorphic encryption provides facility to perform complex mathematical computations on encrypted data without negotiating the encryption data quality.

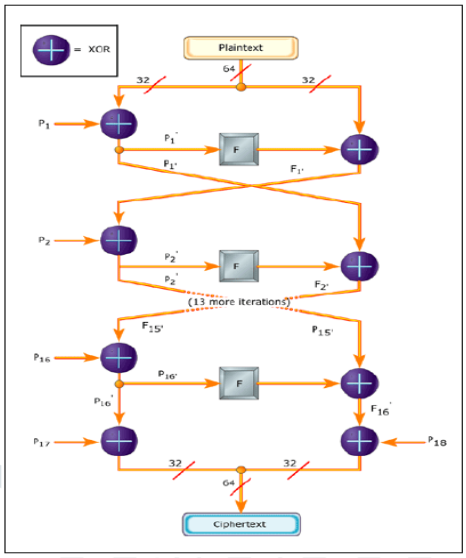


Fig 7: Blowfish Algorithm

Blowfish algorithm consists of 64-bit block cipher and variable length key. This algorithm requires less memory space and yields high speed as compared to other algorithms. This algorithm needs 32 bit microprocessor at a rate of one byte for every 26 clock cycles. It employs variable length key block cipher up to 448 bits. Blowfish contains total 16 rounds. Plain text and key are the inputs provided for Blowfish algorithms.

IV RESULTS AND DISCUSSIONS

This system basically uses the Blowfish encryption algorithm [12] to encrypt the data file. This algorithm is a 64-bit block cipher with a variable length key. This algorithm has been used because it requires less memory. It uses only simple operations, therefore, it is easy to implement. It is a 64-bit block cipher and is a fast algorithm for encrypting data. It requires a 32-bit microprocessor at a rate of one byte for every 26 clock cycles. It is a variable length key block encryption of up to 448 bits. Blowfish contains 16 rounds. Each round consists of XOR operation and a function. Each round consists of key expansion and data encryption. The key expansion generally used to generate initial contents of a matrix and the data encryption uses a network of 16 round of Feistal [14]. Simple text and key are the entries of this algorithm. 64 bit Normal text is taken and divided into two 32-bit data and in each round the given key is expanded and stored in 18 p-array and gives 32bit key as input and XORed with previous round data. The functionality consists in dividing a 32-bit input into four bytes and using them as indexes in an S matrix. Search results are aggregated and XOR together to produce the result. In round 16 there is no function. The output of this algorithm must be 64-bit encrypted text. It is having a function to iterate 16 times of network. Eachround consists of a permutation dependent on the key and a key and a substitution dependent on the data. All operations are XOR and additions in 32-bit words. The only additional operations are four index data search tables indexed for each round.

Function F

Divide xL into four eight-bit quarters: a, b, c and d
F (a, b, c, d) = ((S1,a + S2,b) XOR S3,c) + S4,d

Thus, each round includes the complex use of addition modulo 232 and XOR, plus Substitution using S-Boxes. The function divides a 32 bit input into four bytes and uses those as indices into an S-array. The lookup results are then added and XORed together to produce the output.

Encryption Algorithm

Divide x into two 32-bit halves: xL, xR
For i = 1to 32:
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR

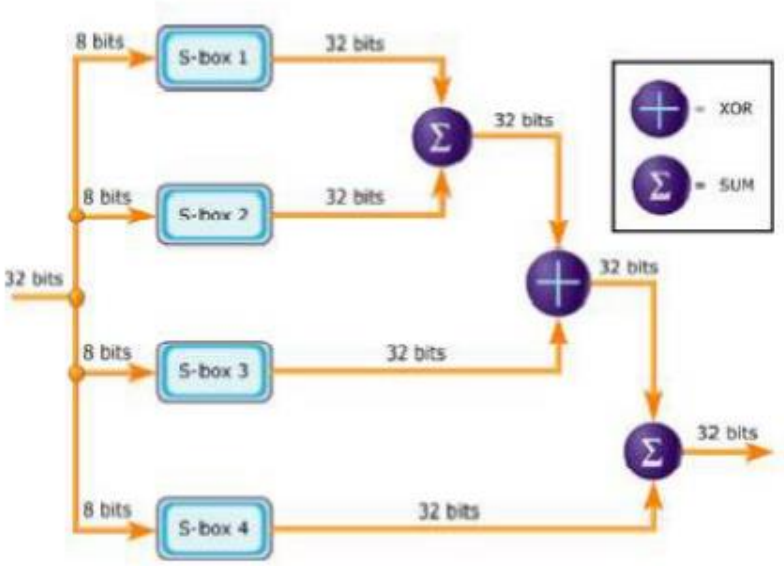


Figure 8 Modified Function F(x)

Decryption

For decryption, the same process is applied, except that the Pi subclasses must be supplied in reverse order. The nature of the Feiestel network [12] ensures that each half is exchanged for the next round (except, here, for the last two sub-words P17 and P18).

The proposed algorithm of Blowfish can achieve an efficient data encryption of up to 4 bits per clock. In this design, we avoid limited I / O restrictions by modifying the 64-bit I / O to 16 bits.

The proposed architecture should satisfy the need for high-speed data encryption and can be applied to several devices, respectively.

Decryption Algorithm

Divide x into two 32-bit halves: xL, xR
For i = 1 to 16:
xL = xL XOR P19-i
xR = F (xL) XOR xR
Swap xL and xR
Next i Swap xL and xR (Undo the last swap.)
xR = xR XOR P2
xL = xL XOR P1
Recombine xL and xR

Time complexity: O(1)

Table 1 File size pre and post encryption/ decryption

Before Encryption	After Encryption	After Decryption
160kb	576kb	160kb

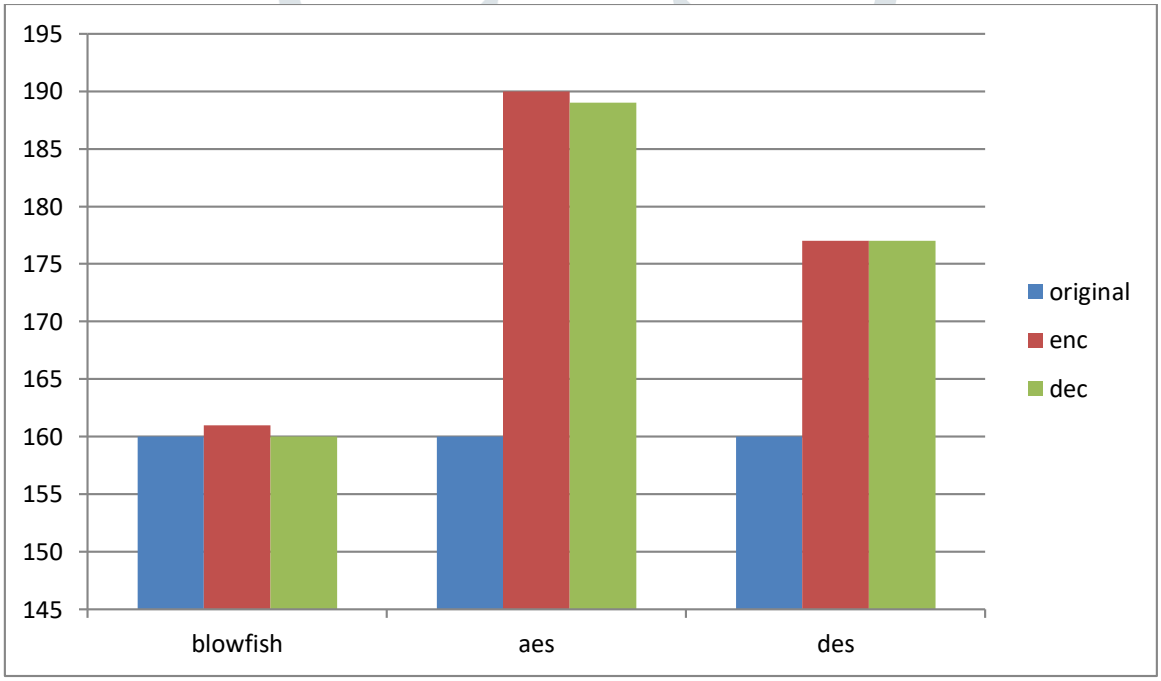


Figure 9 Comparison of AES,DES and Blowfish algorithm

Table 2 Execution time pre and post encryption / decryption

After Encryption	After Decryption
1383	1093

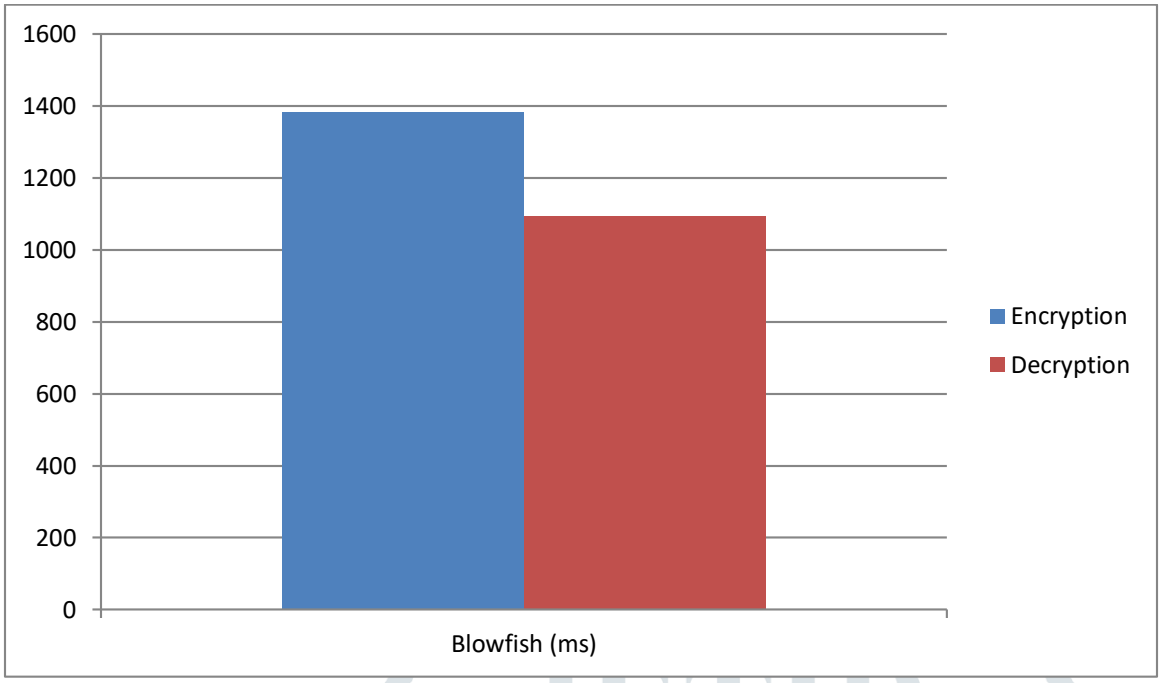


Figure 10 Encryption Decryption Graph for Improved Blowfish Algorithm

Figure 11 Encryption Decryption Response Time and CPU Usage

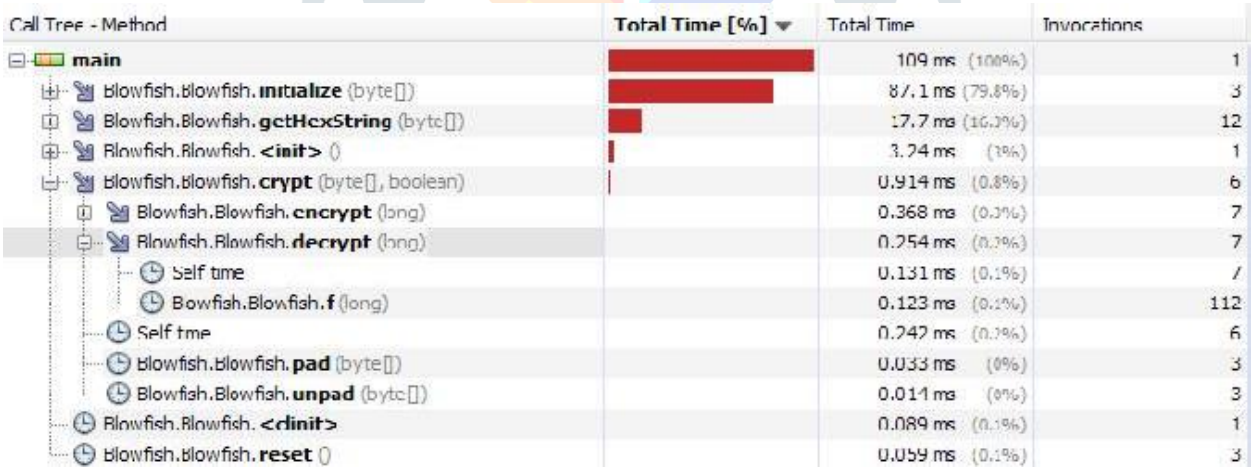


Figure12 HexStringInitialization

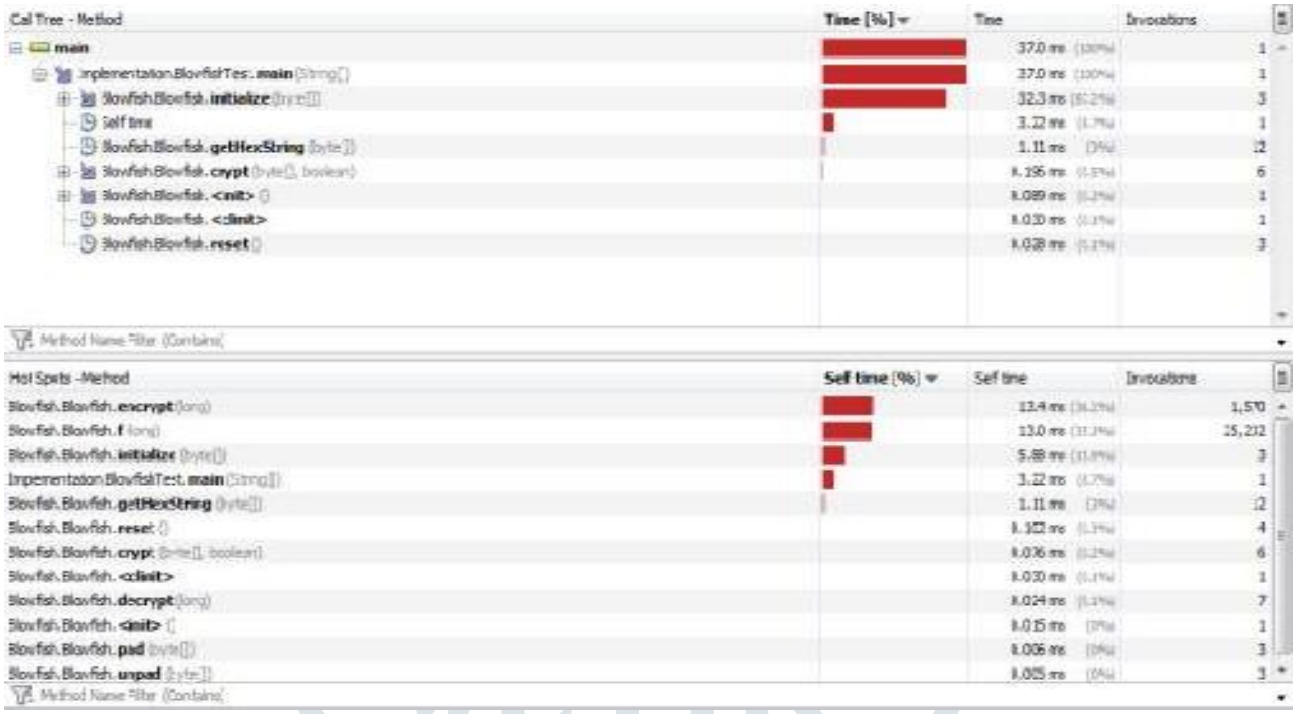


Figure13Encryptiontimeofstandardalgortihm&modified algorithm

Table3Throughputefficiency

File	Size	Throughputin millisecond(StandardBlowfishAlgorithm)	Throughputinmillisecond(ModifiedBlowfishAlgorithm)
BlockSpecimen1	3KB	3.53	2.24
BlockSpecimen2	5KB	5.80	2.33
BlockSpecimen3	7KB	8.21	1.73
BlockSpecimen4	11KB	13.65	7.87
BlockSpecimen5	16KB	21.89	8.64
BlockSpecimen6	21KB	27.99	17.99

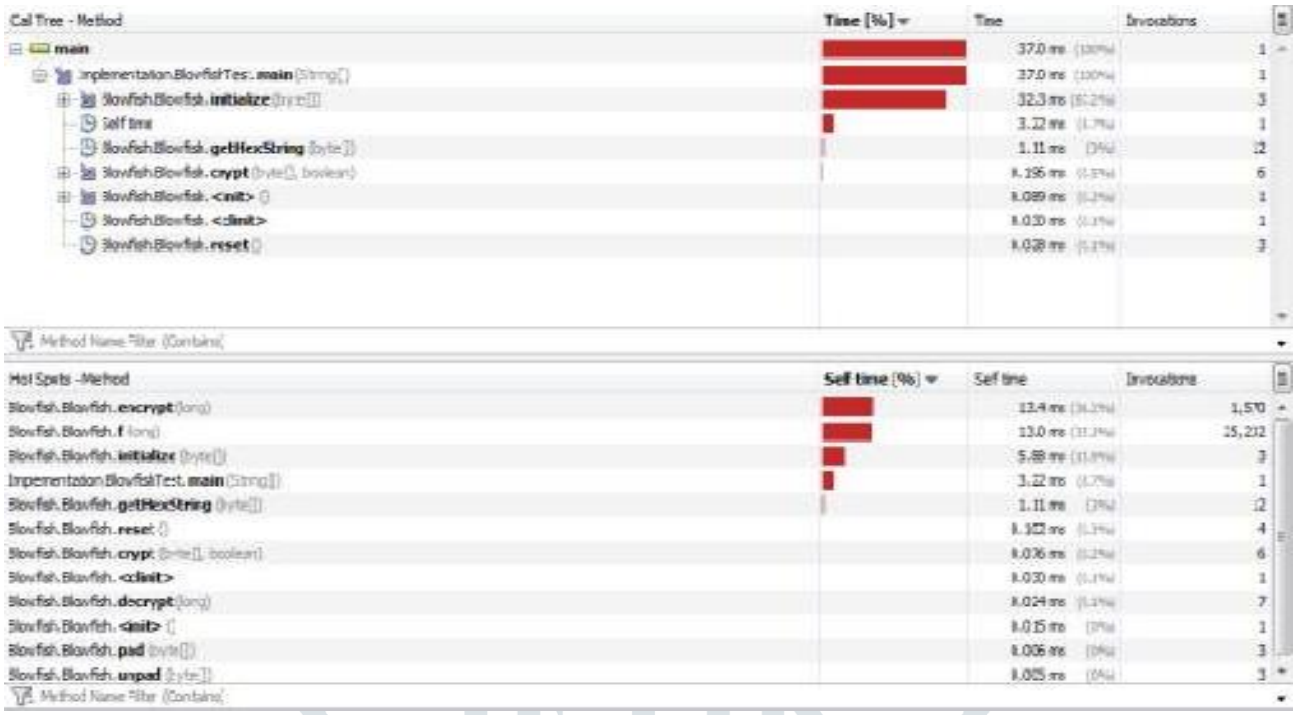
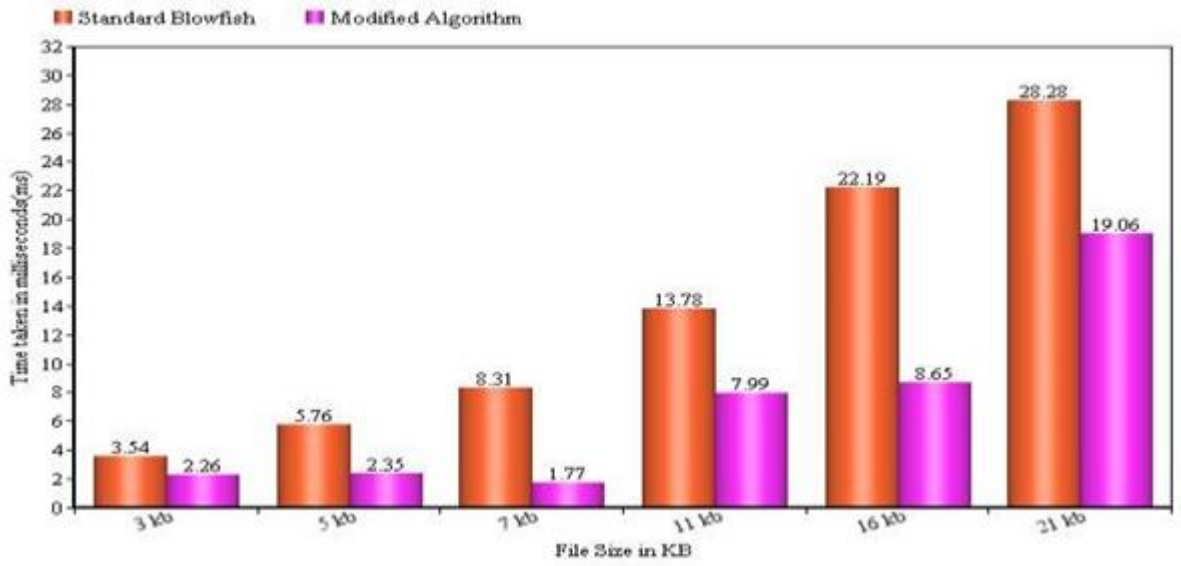
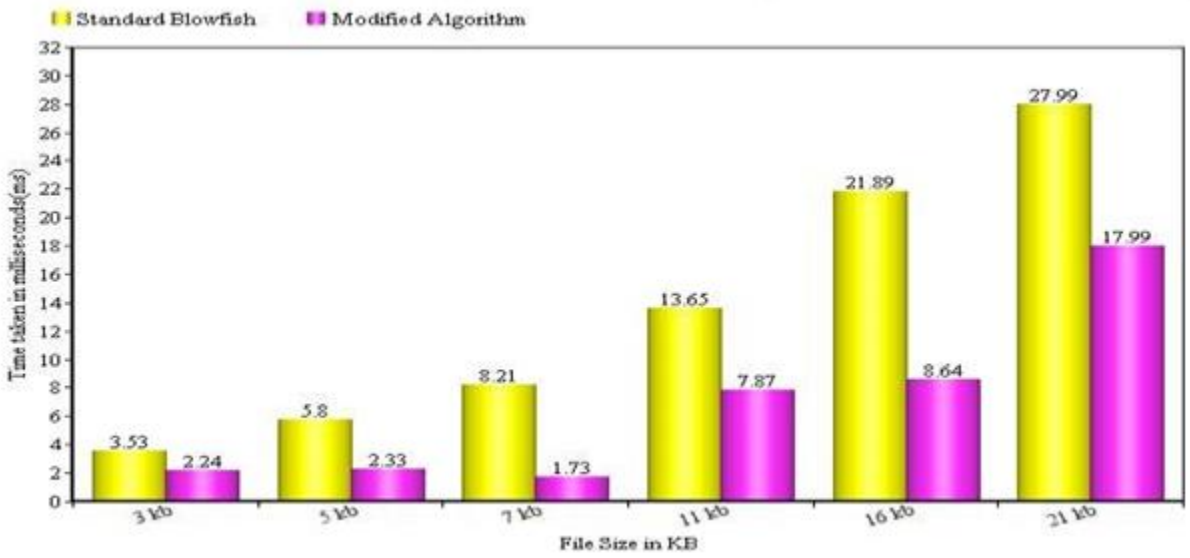


Figure4.2Encryptiontimeofstandardalgortihm&modified algorithm



Graph4.1Comparisonofencryptionofstandardblowfish&modifiedimplementation



Graph4.2Comparisonofdecryptionofstandardblowfish&modifiedimplementation

V. CONCLUSION

The previous work [1] focused mainly on providing privacy to the data in the cloud in which multiple-word classified search was used on the data of the coded cloud using an efficient similarity measure of the coincidence of coordinates. The previous work [4] also proposed a basic idea of the use of a safe calculation of the internal product. There was a need to provide more real privacy than this document presents. Blowfish has a better performance than other common encryption algorithms used. Since Blowfish has not any known security weak points so far, this makes it an excellent candidate to be considered as a standard encryption algorithm. In this system, strict privacy is provided by assigning access restriction and the files and user details are hidden from the cloud service provider and from the external user to protect the user's data in the CSP cloud and from the external user. Therefore, by hiding the attributes of the user and the data of the organization, the confidentiality of the data is maintained. This work may be useful in several contexts designing 'secure' symmetric block cipher algorithms. As a future work we can consider a ranked scheme that can support latent semantic searching that can use only True positive values as vectors for indexing and thereby try to increase the efficiency of the system.

REFERENCES

[1] JoonsangBaek, QuangHieu Vu, Joseph K. Liu, Xinyi Huang, and Yang Xiang, "A Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid" ,IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, APRIL/JUNE 2015

[2] Prachi Jain, Prof.ShubhangiKharche Effectuation of Blowfish Algorithm using Java Cryptography", International Journal of Scientific & Engineering Research Volume 4, Issue 5, May-2013.

[3] Ning Cao, Cong Wang, Ming Li, kuiren, Wenjing Lou , "Privacy Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data" *IEEE Transactions on parallel and Distributed Systems*, Vol. 25,January 2014

[4] Ayad Ibrahim, Hai Jin, Ali A.Yassin, deqingzou, "Secure Rank-Ordered Search of Multi-Keyword Trapdoor over Encrypted Cloud Data" *IEEE Asia-Pacific Services Computing Conference* 2012

[5] Yanjiang Yang, "Towards Multi-User Private Keyword Search for Cloud Computing" *IEEE 4th International Conference on Cloud Computing*. 2011.

[6] Qin Liu, Guojun Wag, Jie Wu, "An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing" *IEEE International Conference on Computational Science and Engineering*, 2009.

[7] Blowfish : A 128 Bit Block Cipher by Bruce Schneier ,John Kelseyy, Doug Whitingz David, Wagnerx Chris Hall

[8] "Analysis of AES and Blowfish Encryption Schemes" *IEEE Transaction* 2011.

[9] Yong Zhang, Jian-lin Li, "Research and Improvement of Search Engine Based on Lucene" *IEEE Transaction* 2009.

[10] QIAN Liping, WANG Lidong, "An Evaluation of Lucene for Keywords Search in Large-scale Short Text Storage" *IEEE Transaction* 2010.

[11] GovindS.Pole, MadhuriPotey, "A Highly Efficient Distributed Indexing system based on large cluster of commodity machines" *IEEE Transaction* 2012.

[12] MsNehaKhatri – Valmik, Prof. V. K Kshirsagar, "Blowfish Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 80-83.