Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2014, Vol. 6, No. 1

Protection of data in unsecured public cloud environment with open, vulnerable networks using threshold-based secret sharing

Rahul Dutta

Advanced Software Division, India Centre of Excellence, EMC Corporation

Bangalore, Karnataka (India)

Tel: 91-9035226841 E-mail: rahul.dutta@emc.com

Annappa B.

Department of Computer Science & Engineering, National Institute of Technology, Karnataka. Surathkal (India)

Tel: 91-9845715006 E-mail: annappa@ieee.org

Received: December 31, 2013 Accepted: March 11, 2014 Published: April 30, 2014

DOI: 10.5296/npa.v6i1.4862 URL: http://dx.doi.org/10.5296/npa.v6i1.4862

Abstract

With the recent popularity of cloud computing, public cloud infrastructure is now being offered by many vendors. Many small, medium as well as large enterprises are now moving into public cloud due to its significant business advantages, flexibility and reduced cost. However, public cloud environments are less secure and the complete setup consists of vulnerable public networks. Data is sent typically over the internet or very large, open networks which are vulnerable to security attacks. Protection of data is hence primarily important in a public, third-party cloud. This paper is an extension of the previously published work and discusses a general approach to distribute and transmit data in an unsecured environment. We propose an application layer technique of protecting important data in public clouds which are transmitted over public networks. We use the concept of secret sharing to derive a technique which provides an efficient and effective solution to the above problem. We show how data can be protected without any complex cryptographic techniques that are generally inefficient in cloud environment.

Keywords: Network Security, public cloud, secret-sharing, data processing, privacy, storage.

1. Introduction

In recent years, cloud computing has gained much importance in the technological industry. Cloud computing enables substantial economies of scale as it allows users to perform processor or data intensive tasks at a much lower cost. Many definitions of cloud computing exists, but the broad definition outlined by the National Institute of Standards and Technology (NIST) is as follows:

Cloud computing is a model that enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

One of the major service models in cloud computing that has gained popularity is Infrastructure-as-a-service (IaaS). Storage, as a service is a part of IaaS offerings. IaaS offers elasticity, pay-as-you-use and data management facilities to organizations and thus help them reduce infrastructure cost. IaaS helps organizations save a lot of expenditure on infrastructures. Cloud data centers have gained importance and popularity as they offer clients with storage facility that can be scaled up or down as required along with very high bandwidths to access the data. Cloud computing presents several challenges of networking, of which security is most important, especially in public or hybrid cloud environments. Massive volume of data is transmitted over open, unsecured networks that are prone to various threats and attacks. The first step for this is to ensure privacy of the transmitted data so that they cannot be captured or misused over the network of by service providers. In general, several techniques based on conventional cryptography have been proposed to address the issue of privacy in cloud environments. These techniques are based either symmetric or asymmetric keys. However, owing to the volume and velocity of data, these techniques are slow and unsustainable in a cloud network or storage.

The idea of secret sharing was first introduced by Shamir (A. Shamir, 1979). Since then, several secret sharing schemes for different purposes have been proposed by many researchers. Many secret sharing schemes have been proposed in both visual and field domains, and applied to address various security problems. Secret sharing schemes have been used for developing authentication techniques, anti-phishing frameworks and for various other purposes. However the technique of secret sharing for protection of data from misuse over in a cloud environment and unsecure networks has been less explored. This work is aimed at adapting threshold-based secret sharing scheme to provide a performance efficient solution, which addresses the issue of data protection in public clouds with unsecured networks.

1.1 Challenges of network security in public cloud

There are several challenges of ensuring secure end-to-end communication in a public cloud environment. Two main concerns are outlined below.

1.1.1 Geographic distance

In most cases, the service providers are located over geographically remote places from the clients they serve. In that case the only medium of communication is via the internet and a secured, dedicated communication channel is not possible.

1.1.2 Lack of transparency

Clients are not sure of the security profile of the public cloud service providers. Also, there may be multiple clients accessing data or connected in the same network. Hence, clients are concerned about the security of their data.

1.2 Drawbacks of cryptographic techniques in cloud networks

Traditional cryptographic techniques present a lot of challenges and difficulties in a cloud environment. Some of these drawbacks are highlighted below.

1.2.1 Performance Overhead

Commonly used good cryptographic algorithms like AES, DES, Blowfish, and Salsa make use of finite fields and require complex computation. These computations are time consuming and degrade the performance in real time application. This happens mainly because cloud computing involves massive amount of data. A lot of research is being carried out to develop highly efficient, yet robust cryptographic techniques especially suitable for cloud environments.

1.2.2 Lacks of parallelism

The operations involved in a cryptographic algorithm are mostly interrelated and follow a sequence. This makes parallel operations difficult, especially for stream ciphers and block ciphers with chaining.

1.2.3 Key Management

Every cryptographic algorithm use keys for encryption and these keys need to be preserved for future decryption and retrieval of data. This presents a lot of problems.

Firstly, in a cloud environment, where there is huge amount of data, storing the key is a problem. The size of the keys used becomes extremely large due to the vast amount of data in cloud. The keys can either be stored with the client or managed by a third-party. Key management by a third-party increases the client's concern of trust.

Secondly, the key can become a single point of failure. It the key is lost or corrupted, the entire data is lost. This increases the client's risk of losing the data and diminishes trust in third-party.

Thirdly, if the keys are compromised or hacked, the entire security system fails and the data can be misused. For an attacker, it is easier to hack a central system having all the keys. To address this issue, again, key distribution mechanisms are needed which increases

the complexity of the system.

1.2.4 Difficulty in searching and retrieval of data

In cloud applications, vast amount of data is stored and processed. Searching for a particular data is extremely difficult in case the data is encrypted. The only way is to decrypt all the data and then look for the required data. It is very difficult to devise any indexing technique on encrypted data for fast location of a required data. Hence processing queries and retrieving data in cloud becomes practically infeasible using cryptography.

1.3 Concept of secret sharing

Traditionally, cryptographic encryption and decryption techniques have been used to provide security and data confidentiality. However such techniques are not suitable in a cloud environment primarily due to their complex mathematical nature which is time consuming.

The concept of secret sharing was first introduced by Shamir in 1976 and later extended to visual cryptography also. The idea is that, given a data item (secret) S, create n meaningless shares of that secret and distribute it to n different participants. Each of these secret shares individually does not reveal any information about the original secret. Only when all the secret shares are put together, the original secret is revealed.

There are several variations of secret sharing schemes which are suitable for different kinds of applications. One of them is threshold-based secret sharing. A (k, n) threshold-based secret sharing scheme divides a secret S into n shares such that any number of shares less than k does not reveal any information about the secret being shared. When k or more shares are available, only then the original secret can be reconstructed. Based on Shamir's secret sharing technique, Thien and Lin [2] proposed a threshold-based technique for secret sharing of images.

In general, the share construction and secret reconstruction algorithms involve simple mathematical operations as compared to conventional cryptographic methods. Hence, secret sharing schemes are better suited for protecting huge amount of data in a cloud environment. Additionally, secret sharing schemes do not involve any key and thus there is no burden of key management as in case of cryptographic algorithms.

In the rest of the paper, we discuss the related works in this area with details of secret sharing and then introduce our proposed technique. This is followed by the detailed simulation results and analysis of the performance from various aspects of networking. Finally, we conclude the work with the scope of future enhancements that can be done.

2. Related Works

Privacy and trust in cloud has been an active area of research in the recent past and present. On-demand integrity checking mechanisms, cryptography, monitors with remote attestation and various auditing schemes has been proposed. Schemes that are already proposed are discussed and their drawbacks are highlighted. This is followed by a survey of

various secret sharing schemes and their characteristics. Thien & Lin's secret sharing scheme [2] is discussed in details with example.

2.1 Security and privacy in cloud environment

Privacy preservation and trust in cloud using encryption has been proposed by Varalakshmi [3]. The technique relies on a third party for encryption & decryption and uses dynamic small keys which makes key management difficult. Encryption and decryption is also a time consuming process and the security is highly dependent on the confidentiality of the key. Thus, it is not very effective in enhancing the customer's trust in the service providers.

On-demand verification of data integrity in cloud has been proposed by Cong Wang et. al [4]. This provides a mechanism for auditing the information stored in cloud, but does not provide any real-time protection of data. In many cases, such as financial domain and e-commerce, auditing may not be possible by the customer. This scheme is effective in preventing data modification attacks and byzantine failures. But, data is still visible to any third-party, especially insiders of the service provider. Thus, it does not address the issue of confidentiality.

Bingyu Zou and Huanguo Zhang [5] proposed an idea to improve security using real-time monitoring and reporting to the cloud user using remote attestation. In this work, measurements of system configurations stored in platform configuration registers of TPM along with runtime states of application in cloud user's virtual machine are reported to corresponding cloud user through remote attestation. However, it does not take care of the correctness and confidentiality of the data stored in virtual machines. Also it does not provide any security that can prevent loss or damage to any sensitive data.

The idea of secret sharing to distribute data amongst multiple service providers has been explored by Divyakant [6]. In this work, Shamir's secret sharing algorithm [7] has been adapted to build an order preserving polynomial to construct the secrets. Such schemes reduce communication costs for queries, but do not provide any efficient indexing of the secret shares for fast retrieval of data. Moreover, results of a practical implementation of the scheme have not been explored. Based on this work, ALzain [8] proposed a new architecture called NetDB2-MS for database services and evaluated the results for small data. However, these works do not take into account the storage overhead incurred by the secret sharing scheme and the possibility of indexing the secret shares for very large data with a primary key (index).

2.2 Secret Sharing schemes

The idea of secret sharing was first introduced by Shamir (A. Shamir, 1979). Since then, there has been lot of research in this area and several schemes have been proposed for secret sharing. Naor and Shamir (Naor, et. al., 1994) extended the secret sharing concept into image research, and referred it as visual cryptography. Visual cryptography is a perfect secret sharing scheme, and requires stacking any k image shares (or shadow images) to show the original image without any cryptographic computation.

2.2.1 Shamir's Secret Sharing Scheme

Shamir developed the idea of a (k, n) threshold based secret sharing technique ($k \leq n$). The technique allows a polynomial function of order (k−1) constructed as,

$f(x) = d_0 + d_1x + d_2x^2 + \ldots + d_{k-1}x^{k-1} \pmod p$ ……. Eq. (1).

Where the value d_0 is the secret and p is a prime number.

The secret shares are the pairs of values (x_i, y_i) where $y_i = f(x_i)$, $1 \leq i \leq n$ and $0 < x_1 < x_2 \ldots < x_n \leq p - 1$. The polynomial function f(x) is destroyed after each shareholder possesses a pair of values (xi, yi) so that no single shareholder knows the secret value d0. In fact, no groups of $k - 1$ or fewer secret shares can discover the secret d_0. On the other hand, when k or more secret shares are available, then at least k linear equations $y_i = f(x_i)$ can be set for the unknown di's. The unique solution to these equations shows that the secret value d0 can be easily obtained. Shamir's SSS is regarded as a PSS scheme because knowing even $(k - 1)$ linear equations doesn't expose any information about the secret.

2.2.2 Thien and Lin's Secret Sharing Scheme

Thien and Lin (C.C. Thien et. al, 2002) proposed a (k, n) threshold-based image SSS by using Shamir's SSS (A. Shamir, 1979) to generate image shares. The essential idea is to use a polynomial function of order $(k - 1)$ to construct n image shares from an $l \times l$ pixels secret image (denoted as I) as,

$Sx(i, j) = I(ik + 1, j) + I(ik + 2, j)x \ldots + I(ik + k, j)xk-1 \pmod p$ ……Eq. (2).

Where, $0 \leq i \leq (l/k)$

This method reduces the size of image shares to become 1/k of the size of the secret image. Any k image shares are able to reconstruct every pixel value in the secret image.

An example of (2, 4) image secret share construction process is illustrated in fig. 1 where k = 2 and n = 4. According to the technique, a first order polynomial function can be created as

$Sx(i, j) = (110 + 112x) \pmod{251}$ ……...Eq.(3).

Where 110 and 112 are the first two pixel values in the Lena image.

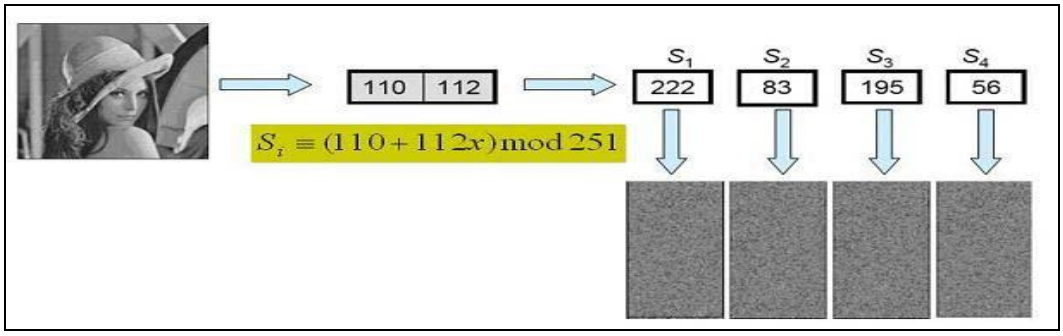

Figure 1 - Thien & Lin's Secret Sharing Scheme

For the four participants, four x values are picked randomly, and substituted in the polynomial function by setting p value to be 251 which is the largest prime number less than 255 (maximum gray image value). Four shares are computed as (1, 222), (2, 83), (3, 195) and (4, 56). They become the first pixel in four image shares. The second pixel is computed in the same manner by constructing another first order polynomial function using next two pixels in the Lena image. This process continues until all pixels are encoded. Four image shares are the bottom right images shown in fig. 1, and the size of each image share is half (1/2) size of the original image. None of the image shares appear to reveal information about the secret image.

However, the pixel values in a natural image are not random because neighboring pixels often have equal or close values. It is evident that the first two pixel values (110 and 112) are very close to each other. This creates a possibility that one image secret share may be used to recover the secret image by assuming neighboring pixels have same values in the first order polynomial function.

Since Thien and Lin's method reduces the size of image shares to become 1/k of the size of the secret image, the scheme cannot be qualified as a "perfect" image SSS. In fact, this method is a multiple-secret "ramp" SSS. In other words, the information about the secret exposed is proportional to the number of shares available until the number of shares becomes k or more. In addition, the pixel values in a natural image are not random because the neighboring pixels often have equal or close values. A secret image can be possibly recovered from less than k image shares because neighboring pixels are highly correlated. To address these security issues, Thien and Lin suggested an idea by permutation of the order of pixels (with a permutation key) in the secret image before the image shares are computed. Conversely, the secret image can still be reconstructed from any k image shares by solving the permutated image and applying inverse-permutation using the permutation key. Nevertheless, the permutation key becomes the single-point-failure in the system because the key can get lost or corrupted. This scheme also prevents real time processing because the permuted image has to be obtained before the secret image can be reconstructed.

2.2.3 Li Bai's Secret Sharing Scheme

Among several interesting properties of matrix projection SSS, an image application can be easily extended from this scheme's ability to share multiple secrets. Pixels in an image can be regarded as elements in a matrix. Although the technique is not a PSS scheme, it has strong protection on the secret even if the remainder matrix R is made public. However, matrix R can become single-point-failure if it is corrupted or lost. To overcome this problem, Li Bai's method uses Thien & Lin's method to share the remainder R without any permutation. Thien and Lin's method cannot protect matrix R securely, but it does not affect the protection capability on the projection matrix.

Amongst the several discussed secret sharing schemes, Thien & Lin's method is the most computationally less intensive SSS and it also keeps the size of shares low. Hence, this method is well suited for adaptation to cloud storage where enormous amount of data is stored. Although this method is primarily for secret sharing of images, it can be adapted to work for any type of data. It can also protect the data being shared strongly, without the need

of permutation if the data is not correlated, like in case of image pixels.

The idea of secret sharing has been used for authentication [10]. Secret sharing has also been used to define anti-phishing frameworks using captcha [11].

3. Proposed Technique

To incorporate trust in cloud computing, the data to be stored in the cloud is first converted to secret shares and then distributed to multiple, disjoint, smaller cloud storages managed by independent cloud vendors. The secret shares are indexed using a proposed scheme for fast lookup of shares and retrieval. However, in this work it is assumed that each data object contains a prime attribute that uniquely identifies each of them.

3.1 Secret sharing and storage of data in cloud

For creating secret shares and reconstruction of original data, Thien & Lin's threshold-based secret sharing scheme is adapted with some modifications to allow indexing of secret shares. In Thien & Lin's technique, the size of each share is 1/k of the original data and the secret construction, reconstruction algorithms are very fast and involve very less computation. Thien and Lin's method is a strong secret sharing scheme and it can highly protect the secret being shared.

Thien & Lin's method is used for the purpose of creating secret shares of data which is in text format. So, given any plain text data, it is first convert into numerical form using ASCII values. (Any other standard encoding technique can be used suitably). These numerical values are then used in place of the intensity values as coefficients for computation of the secret shares. The modular operation is eliminated from the equation of secret construction and it is required for proposing an indexing technique. For each record, the index attribute is separately converted to secret shares and appended to the secret shares of rest of the record. The client chooses suitable parameter values for each service provider and generates the secret shares for distribution. The size of each secret share is 1/k. The total size of all the shares is $\lfloor n/k \rfloor$ times of the original data. If n=k, then the combined total size of the secret shares is same as the original share.

The secret shares of the database records are stored on third-party clouds. Each share is stored on a different cloud storage managed by an independent cloud service provider. The architecture is illustrated in fig. 2.

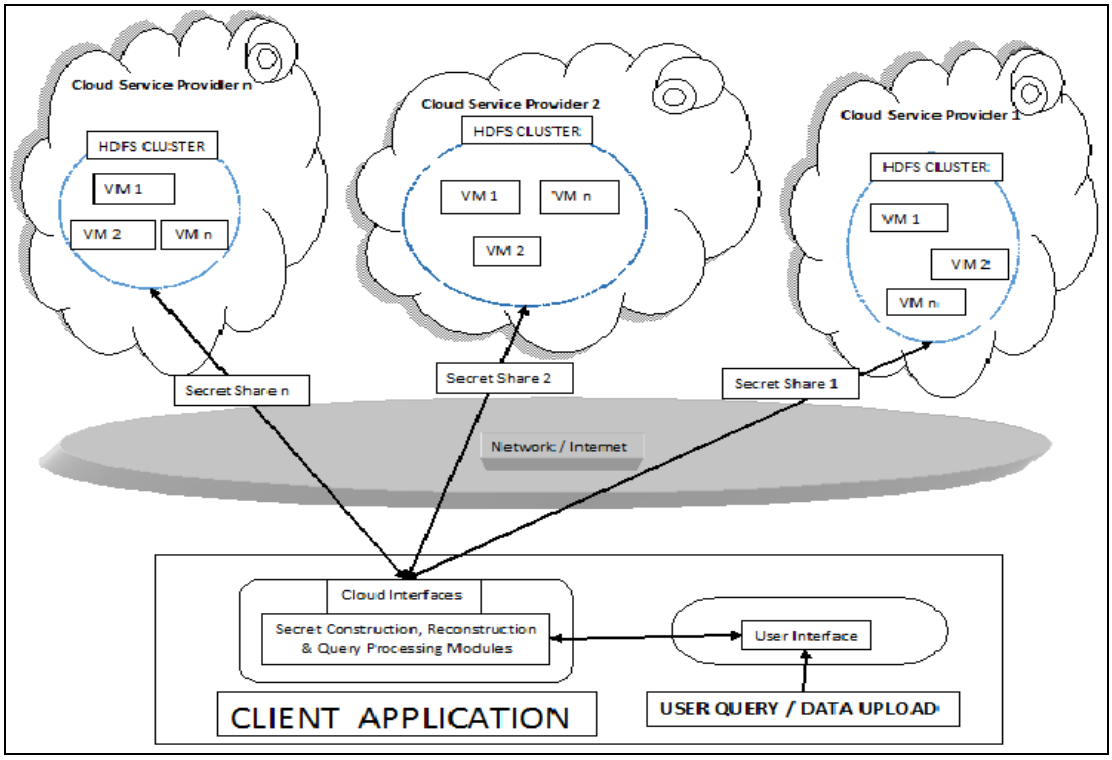

Figure 2 - Architecture for secret sharing in cloud

3.2 Parallel implementation of secret construction and data reconstruction

The following algorithms should be implemented in the application layer interface for the proposed technique

ALGORITHM 1 - *SHARE CONSTRUCTION*

Input: *Original data from database. n – Number of shares, k – Number of parallel tasks that can be executed*

Output: *Secret shares of the original data*

Start

Open DATABASE file
While (NOT end-of-file)
    Read 'k' records from file
    For each 'k' records do in parallel
        Convert record to ASCII values
        For i=1 to n do in parallel
            Generate the i^{th} secret share for the record.
        End-for
        Write each share to different cloud storage
    End-for
End-while

End

If the total number of records to be processed is R, the sequential time complexity of the above algorithm is O(R*n). If k records can be processed in parallel using k*n processors, the time complexity is O(R*n) / (k*n) = O(R/k).

During retrieval of data or query processing, the following algorithm reconstructs data from secret shares:

ALGORITHM 2 - *SECRET RECONSTRUCTION*

Input: *Secret shares of the data. n – Number of shares, k – Number of parallel tasks that can be executed.*
Output: *Original data of the database.*
Start
Open each secret share file S_1, S_2, S_3.....S_n
While (NOT end-of-file)
    Read 'k' records from each of the files S_1, S_2, S_3.
    For each 'k' records do in parallel
        Read n shares from S_1, S_2, S_3.....S_n to a nx1 matrix B.
        Reconstruct the original record R.
        Process the record R
    End-for
End-while
End

If the total number of records is R, the sequential time complexity of the above algorithm is O(R). If k records can be processed in parallel using P processors, the time complexity is O(R) / P = O(R/P).

4. Results

4.1 Implementation

To evaluate the network performance of the proposed technique, a cloud platform is created using Hadoop Distributed File System (HDFS) and Matlab Distributed Computing Server (MDCS). For experimental studies, test data is created consisting of relational databases with prime attributes. We use layer 2 switches with 1Gbps bottleneck bandwidth and twisted pair cables for testing the performance of our proposed concept.

Fig. 3 shows the layered approach to the proposed architecture. The first 3 layers are owned by the client and it is the client's responsibility to incorporate the secret sharing and reconstruction of data. The clients interface only exposes the secret shares to the network, which is transmitted and stored in public clouds. The cloud service providers can only see the secret shares sent to the cloud storage. The summary of the simulation environment is as follows:

Number of clusters: 3

Data nodes per cluster: 2 each running Ubuntu 12.04 LTS.

System configuration for each node: Intel Core i7 3rd Generation, 2.93 Ghz, 8 GB DDR3 RAM, 500 GB SATA II Hard Drive, 1 Gbps Network interface.

Interconnect Switch: Cisco 24-port 1Gbps Ethernet switch.

<table>
<tr><td colspan="4">APPLICATION FRONT-END</td></tr>
<tr><td colspan="4">DATA AND QUERY PROCESSOR</td></tr>
<tr><td colspan="4">CLOUD INTERFACE (FTP CLIENTS)</td></tr>
<tr><td colspan="4">NETWORK</td></tr>
<tr><td>FTP SERVER 1</td><td>FTP SERVER 2</td><td>---------</td><td>FTP SERVER N</td></tr>
<tr><td>HDFS CLUSTER 1</td><td>HDFS CLUSTER 2</td><td>-------</td><td>HDFS CLUSTER N</td></tr>
<tr><td>VIRTUAL MACHINES</td><td>VIRTUAL MACHINES</td><td>-------</td><td>VIRTUAL MACHINES</td></tr>
<tr><td>CLOUD SERVICE PROVIDER 1</td><td>CLOUD SERVICE PROVIDER 2</td><td>-------</td><td>CLOUD SERVICE PROVIDER N</td></tr>
</table>

Figure 3 - Layers of the proposed architecture

4.1 Results of performance testing

4.1.1 Data upload with varying data size

The graph in fig. 4 and table 1 shows the data upload time to the cloud storage including secret share construction. We consider 3 shares for our experiment. Data upload is done using Matlab distributed computing with 8 workers on a single node. The time recorded includes share creation and network transmission time measured in real setup.

Table 1 - Data Upload time

Data Size	Seconds
128 MB	290.106871
256 MB	520.554789
512 MB	1033.025115
768 MB	1551.154689
1 GB	2119.678547
1.25 GB	2651.225548
1.5 GB	3170.254469
2 GB	4010.658891

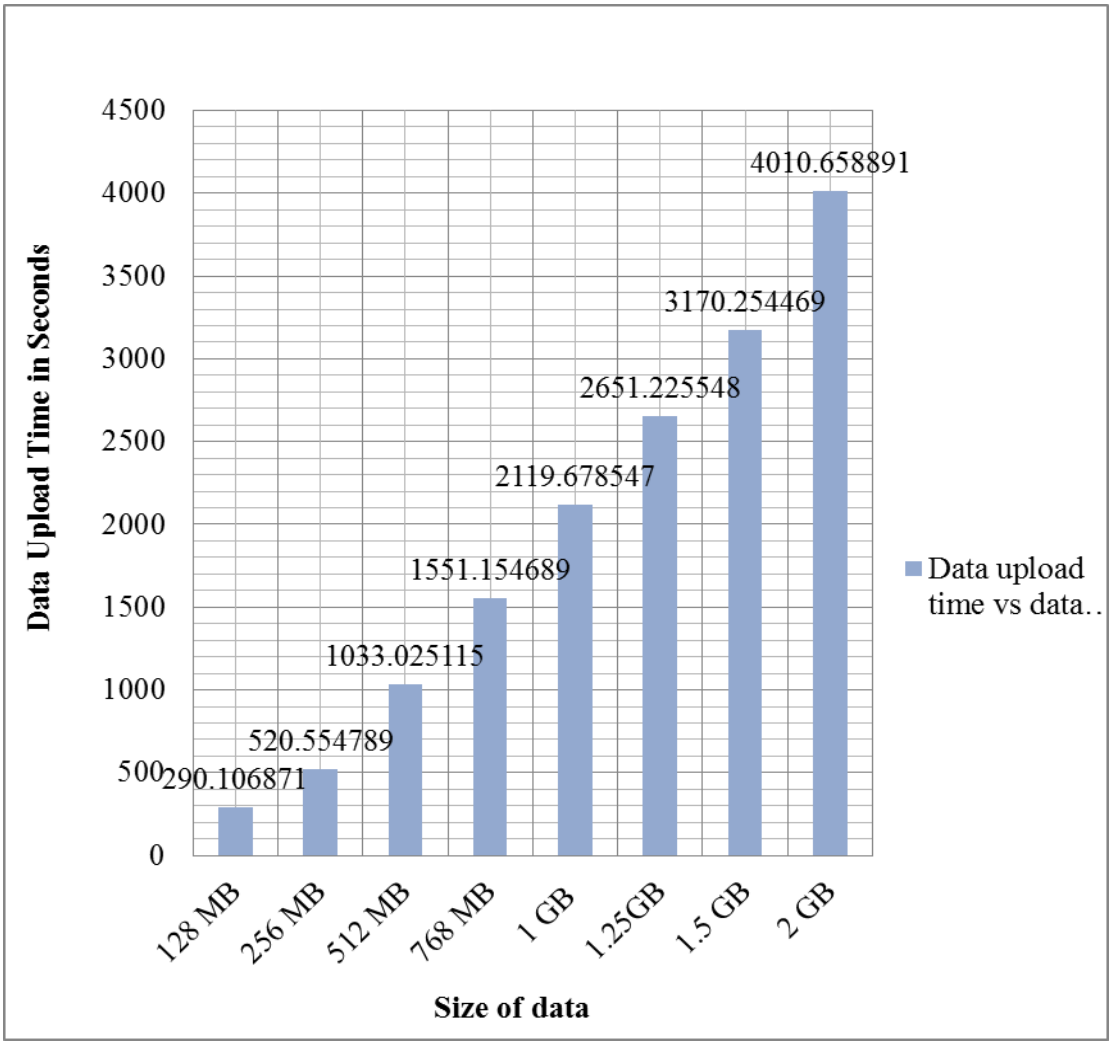

Figure 4 - Data upload time for different data size using 3 shares

4.2.2 Data upload with varying shares

Experiments are performed for data upload with 256 MB fixed size data and varying the number of shares constructed. It is found that there is an almost linear relationship between the numbers of shares and upload time. Increasing size of data increases both computation and communication cost of the shares proportionally. The results are illustrated in table 2 and fig. 5.

Table 2 - Data upload with varying shares

Number of Shares	Upload Time (in seconds)
3	516.5125575
5	721.298747
10	1149.369346

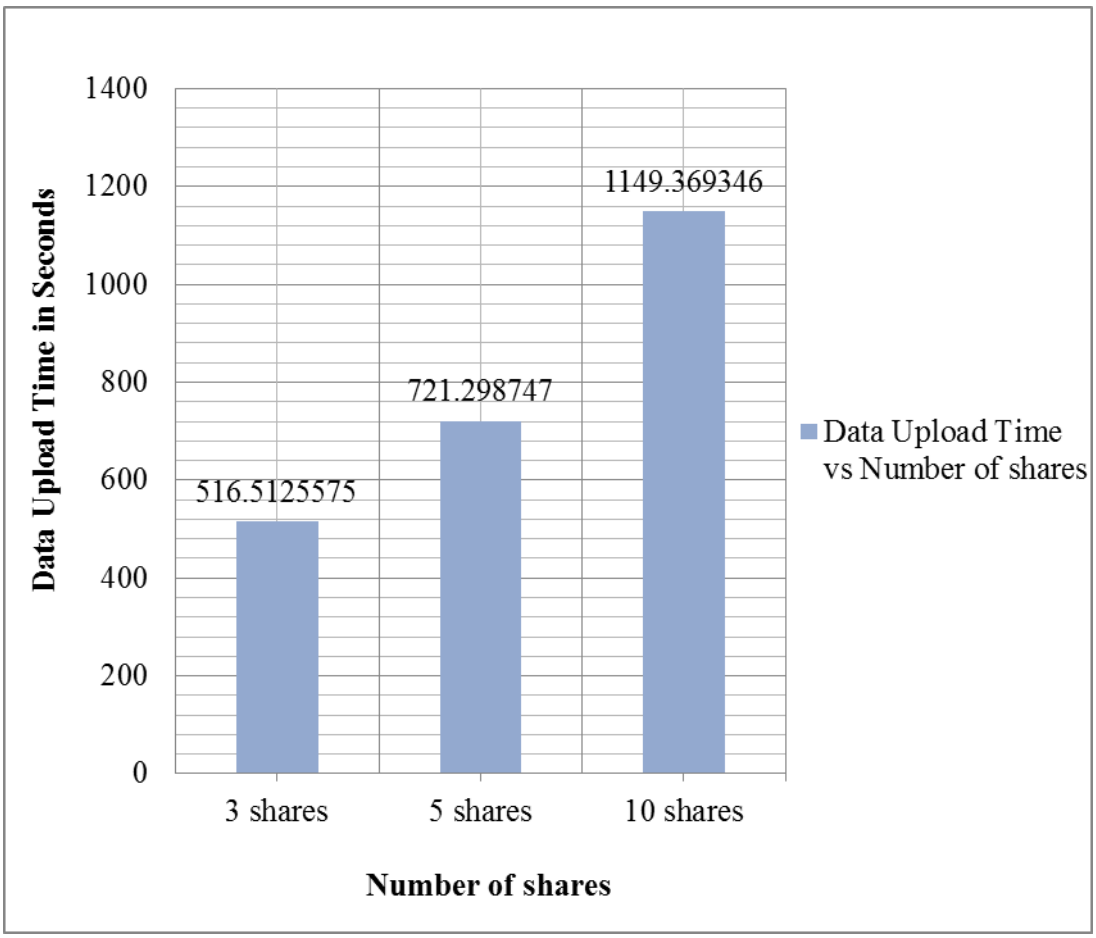

Figure 5 - Data upload with varying shares

4.2.3 Response time for queries

The network delay for queries performed is experimentally determined for varying number of shares. Results are shown in table 3 and fig. 6.

Table 3 – Response time for queries

Number of Shares	Upload Time (in seconds)
3	0.010572
5	0.015047
10	0.019025

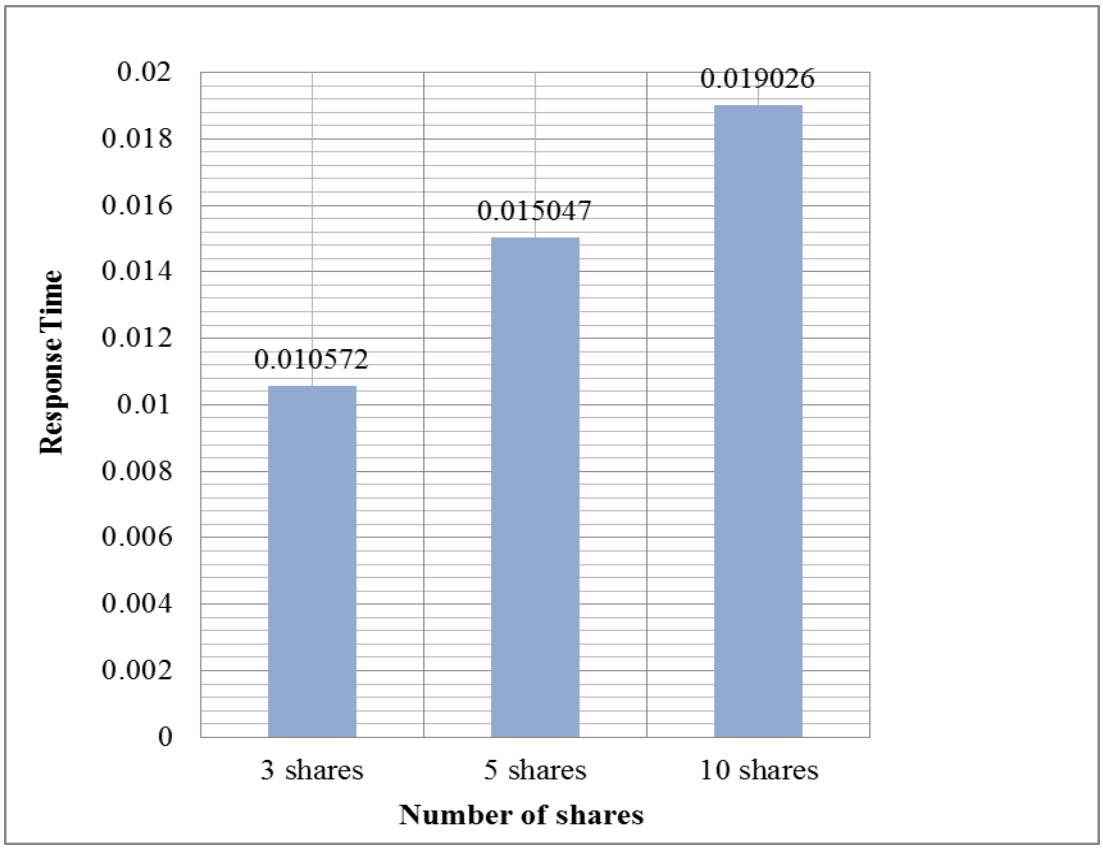

Figure 6 – Response time for queries

4.1 Analysis of results

The results obtained show that share creation and data uploading together takes good amount of time but still it is far more efficient than encryption for remote storages provided by a third party. Using well-known and popular block ciphers is also not feasible. In case of cryptographic methods query processing is also infeasible as indexing is not possible. The network delay in query processing is minimal. Although this method ensures security and privacy at the cost of speed, the overhead incurred is minimal. Based on requirements it is also possible to tune the number of secret shares which is discussed in the next section.

5. Optimizing the number of secret shares

To compute the optimum number of shares, an equation is proposed based on certain parameters. Attempt is made to compute the best possible values of k (threshold) and n (total number of shares) for the secret sharing of an application.

The following parameters are considered in order, whose requirements are taken into consideration for deciding the ideal number of shares for any specific application.

1. Cost
2. Security

3. Response time
4. Communication cost
5. Reliability / fault tolerance
6. Scalability

5.1 COST

Let us consider the total infrastructure setup cost for the entire project is N. Let P be the average cost for setting up single cloud storage.
Then, the maximum allowable number of secret shares

$X_{max} = {}^{N}/_{P}$ …….. Eq. (4).

5.2 SECURITY

For Thien & Lin's method, the probability of reconstruction of the secret if one share is available = $({}^{1}/_{S})^{n-1}$ where S is the maximum possible numerical value of the data and n is the number of shares.
If the security parameter of the application (allowable chance of any reconstruction which is extremely small) be P, then

$P = ({}^{1}/_{S})^{n-1}$ which implies

$n = -(\log_S P + 1)$ ……... Eq. (5).

Let X_2 = n subject to maximum value of X_{max}.

5.3 RESPONSE TIME

The coefficient of correlation between average response time and number of shares is calculated using Pearson's method from the experimental results. The coefficient of correlation is 0.999941.
Let D_{min} be the minimum achievable response time. Then, D_{min} occurs when number of shares is minimum (i.e. 2).
D_{min} = 2.43 in case of our setup.
Let D be the desired response time for the application. Then the desired number of shares X_3

$= \{{}^{D}/_{1.215}\} * 0.999941$ ……... Eq. (6).

5.4 COMMUNICATION COST

Let the fixed communication cost (to initiate, maintain and terminates a connection) be α. Let N be the number of optimum shares.
Let β be the cost for communicating each secret share. As the number of shares increase, β decreases proportionally. (More number of shares, lesser the size of each share).
If T is the total communication cost, then
$T = \alpha N + \beta$

$\rightarrow N = {}^{(T-\beta)}/_{\alpha}$ ..... Eq. (7).

Let $X_4 = N$

The final value of k is the average of the previous 3 number of share calculated = ${}^{(\sum_{i=2}^{4} Xi)}/_{3}$

..... Eq. (8).

5. RELIABILITY

Let the maximum tolerable node failures be N.
Then
$X_5 = k + N$ ..... Eq. (9).

Subject to a maximum of X_{MAX}

6. SCALABILITY

Let us consider the fractional value of the scalability factor of the system as Q (this is the fraction by which the system is expected to grow in future)
Then,
$X_6 = X_5 * (1-Q)$ ..... Eq. (10).

X_6 gives the optimum value of the total number of shares.

5. Conclusion

In summary, it can be said that the secret sharing technique gives a reasonably good performance in cloud environment. The modified Thien and Lin's secret sharing technique incurs little computational overhead and minimizes storage overheads if threshold value is equal to the total number of shares. The transmission delay over the network is also very less. The amount of data that travels over the network is nearly equal to the original data as the total size of secret shares is equal to the size of the original data. Hence, it can be concluded that the proposed modified Thien & Lin's secret sharing is best suited for use in cloud environment.

The secret sharing scheme in cloud can protect the privacy of the data as well as the transactions taking place on third-party cloud storages. The cloud service providers and any potential hacker in the network only have access to meaningless secret shares from which no transaction details of information about the original data can be determined. The minor computational overhead and some extra cost incurred by the client justify the usage of secret share for greatly enhancing the trust on third-party vendors. Using secret shares gives the clients a sense of control over their own data since no one else can see or misuse the data. In case there are chances of data being modified, additional secret shares can be created and verification can be done by reconstructing the data from different sources.

Initially, when setting up the system, optimizing the number of secret shares can be done with the discussed methodology. However, the optimization method discussed finds a value for the number of shares that is suitable taking all requirements into consideration. It may not guarantee full satisfaction of the actual desired requirements due to trade-offs, but gives a value that can nearly achieve the desired features of the system.

Future work can be focused on creating secondary indexes, speeding up query processing for non-key aggregate queries and range queries. Efficient techniques to process complex queries such as joins, semi-joins can also be developed. A fully capable query processing engine for SQL can be developed in future.

This technique can be implemented, deployed and tested on various types of platforms and using different technologies that can achieve better results. In this project, the implementation is limited to small setups, but this technique can be tested on various hardware platform and operating systems. The performance can also be measured with various transport layer protocols like Data Center TCP and other variants of TCP.

Optimization of the number of secret shares can also be enhanced. Several other parameters can be included for different types of system and the other mathematical optimization techniques can be used.

Acknowledgement

We would like to thank the entire staff of Computer Science department, NIT Karnataka for their help and continuous support for this work.

References

[1] Dutta R, Annappa B., “Privacy and trust in cloud database using threshold-based secret sharing ”. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013. August 2013. http://dx.doi.org/ 10.1109/ICACCI.2013.6637278

[2] C -C Thien et. al. “Secret image sharing,” Computers & Graphics, vol 26, no 5, pp. 765–770, 2002.

[3] Varalakshmi, et. al. “Integrity checking for cloud environment using encryption algorithm,” International Conference on Recent Trends in Information Technology, pp. 228-232, 2012. http://dx.doi.org/10.1109/ 10.1109/ICRTIT.2012.6206833

[4] Cong Wang, et. al. “Towards Secure and Dependable Storage Services in Cloud Computing,” IEEE transactions on service computing, Vol 5, No 2, pp. 220-232, 2012.

http://dx.doi.org/10.1109/TSC.2011.24
[5] Bingyu Zou, et. al. “Towards enhancing trust in cloud computing” 2nd International Conference on Control Instrumentation and Automation (ICCIA), pp. 364-366, 2011. http://dx.doi.org/10.1109/ICCIAutom.2011.6183990.
[6] Agrawal Divyakant et. al. “Database Management as a Service Challenges and Opportunities,” IEEE 25th International Conference on Data Engineering, pp. 1709-1716, 2009. http://dx.doi.org/10.1109/ICDE.2009.151.
[7] A. Shamir, "How to share a secret " Communications of the ACM, vol 22, No. 11, pp. 612-613, 1979. http://dx.doi.org/ 10.1145/359168.359176.
[8] Mohammed A. ALzain, “Using Multi Shares for Ensuring Privacy in Database-as-a-Service,” Proceedings of the 44th Hawaii International Conference on System Sciences, pp. 1-9, 2011.
[9] Mohammed A. ALzain, “A New Approach Using Redundancy Technique to Improve Security in Cloud Computing,” IEEE International Conference on Cyber Security Cyber Warfare and Digital Forensic (CyberSec), pp. 230-235, 2012.
[10] Jaya, et. al. “Novel Authentication System Using Visual Cryptography,” World Congress on Information and Communication Technologies (WICT), pp. 1181-1186, 2011.
[11] Kai Zhao, et. al. “Anti-phishing mutual authentication using the visual secret sharing scheme,” 2010 International Symposium on Information Theory and its Applications (ISITA), pp. 560-565, 2010.
[12] NIST. “NIST definition of cloud” http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
[13] Matlab [Online]. Available: http://www.mathworks.in/support/product/DM/installation/ver_current/
[14] KapilBakshi, “Considerations for Big Data: Architecture and Approach”, IEEE Aerospace Conference, pp. 1-7, March 2012.

Copyright Disclaimer