Securing Cloud-Based File Storage System via Homomorphism

Adnan Tahir, M. N. A. Khan, Sheeraz Mughal
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
adnantycoon@gmail.com, mnak2010@gmail.com, sheeraz_mughal@hotmail.com

***Abstract*— Cloud computing is playing a much important role in comparison to other fields of IT, in providing Data storage, Data security, Quality of Services (QoS) etc. Many data security factors have also increased due to the fast evolution of cloud in the IT industry. Therefore many security models and techniques have been deployed and are been in execution for providing more & more security to the data, especially the sensitive & private one. Despite of that much security, many of the models/techniques lacks in one or more security threat measures. In this paper a new model have been proposed & designed which comprises of two techniques in parallel. This model adapts Homomorphic Encryption scheme for the encryption of data at the client side and SSL-128 in parallel as a secure tunnel for all the transmission and communication from the user to the cloud and vice versa. The data file being sent to the cloud vendor for storage will comprise of data's context and decryption keys, both of which will be encrypted separately with the same encryption algorithm. When retrieving this file, user will send query consisting of user's data request & decryption key values. Decryption key values will be separated by query engine and will be saved in Decrypter for decrypting the Encrypted Decryption keys attached with the data. These decryption keys, attached with data, will be decrypting the data's context when received at user's end through the Decryption Algorithm Component.**

Keywords- Cloud Computing, Cloud Data Security, Encryption/Decryption Engine, Homomorphism

I. INTRODUCTION

Cloud computing is rising with a very high pace because of its countless benefits like reduction of the cost of the IT infrastructure, high availability, excellent performance, etc. Organizations are moving towards the adoption of the cloud computing services with keen interest. In the cloud computing the end users don't need to install the software's or applications in their computers and they can access their data or files remotely from any computer through the internet. There is a cloud provider who manages the cloud and will provide the cloud services to its users. The cloud provider operates the cloud data center and manages its resources. In the cloud computing, all the storage and processing is done at the cloud data center. Everything is provided as a service over the internet through a web browser and you pay only for what you use. You can use the service for minutes, hours, days or months according to your needs on a utility basis. The services that are provided by the cloud providers are of three types which are: software as a service, infrastructure as a service and platform as a service. In the software as a service a user can access different software's and applications from the cloud provider by using a web browser. A user don't need to install the software, he'll get an instance of the software running in the cloud data center on his machine. In the infrastructure as a service a user can get the hardware as a service from the cloud data center. The hardware can be memory, disk space, processing power, etc. The user will use the hardware for as much duration as he wants and will pay for it. The demanded hardware will be allocated to the user in the cloud data center. In the platform as a service, the user will be provided with the set of software's, development tools or a platform. And by using that platform a user can develop or create different applications and products. A user can use different API's available at the cloud provider's end. The fast speed internet and different virtualization technologies have given emerge to the concept of cloud computing. The cloud can be a public, private or hybrid. In a public cloud anyone can by the services of a cloud through the internet. The private cloud offers services to a limited number of users. While a hybrid cloud is a mixture of a public and private cloud. The barrier towards the rapid growth of cloud computing are the security and privacy issues associated with it.

Cloud computing is facing many challenges due to the security and privacy issues and lack of standards. The major issue associated with this computing are the security and privacy issues. Reduction in the cost and expenses is important but the data and information is the biggest asset for any kind of organization. No one will ever want to hand over his business information in the hands of others without the proper proofs. It's the biggest barrier concerned with the cloud based computing. Security and privacy are the key to the success of the cloud computing. Lots of work is going on to resolve the security related gaps in the cloud

computing. End users and organizations should be ensured that their data and information is in the custody of the safe hands. And these issues will be hopefully resolved with the passage of time. The other big issue associated with the cloud computing is the lack of standards. There are no globally defined standards for the cloud computing, every provider defines and follows his own standards. So there is a need that the IT standards defining organizations should define the standards for it. The challenge is the management of the cloud and the cloud resources. Managing the cloud is a difficult task and expert and skilled individuals are needed for the management of the cloud and the cloud resources. They should be properly trained with the skills needed for the running of the clouds. The monitoring is, also a challenge, how to monitor the users and customers that are availing the services is also important. There will be a proper system for the protection from the attackers and intruders. The other challenge for the organizations is that they will not have the control over their information and it will be in the custody of the other party. There can be an issue if you move or switch your data while changing the cloud provider.

II. HOMOMORPHIC ENCRYPTION

Securing the data in the databases, whether they are in traditional environment or cloud environment, has remained the major concern and issue in the field of IT. In this regard Encryption approaches and schemes have played a very effective and efficient role. These encryption schemes and approaches have been helpful in Cloud environment where we didn't need to involve the 3rd party in the transaction and communication between the client/user and the cloud server. One of the encryption known as the Homomorphic Encryption is the best in this regard because without knowing or having the knowledge of the private key, this encryption can perform operation and computation on the encrypted data. This encrypted data or the result of the encrypted data, which is also in the encrypted form, can only be decrypted with the secret private key which the authenticated user/client have. When the results of this encrypted data are decrypted and matched with the original results, they appeared to be same. This states that through Homomorphic encryption we can store and retrieve the data in the encrypted form in the cloud and also can perform computation on it, which will be useless for the unauthorized user because all the work to be done is in the encrypted form, in fact the results the user gets is also in the encrypted form. The main flavor of the Homomorphic encryption is that the user has the access to deal with the encrypted cipher data directly without the intervention of any kind of 3rd party or any administrator authority body. The user can also assure the security of the data whereas anyone who didn't know the secret private key can't access the data or can decrypt it.

Moreover the Homomorphic encryption is further divided in two categories according to its behavior, i.e. Partial Homomorphic Encryption (PHE) and Full Homomorphic Encryption (FHE). However PHE have been somehow implemented and been practical used in the cloud computing but the FHE has not been implemented or been used practically due to its huge number of processing and computations which involves large and heavy resources to carry out this operation. In PHE the encryption method or operation used in any single one, i.e. whether it will perform additive Homomorphic encryption or multiplicative Homomorphic encryption. Whereas in the FHE the operation to be performed are single or the combination of more, i.e. it can be multiplicative or additive or combination of both such as NAND or XOR. But FHE, no doubt, is more powerful in security aspect as compared to PHE but because of requiring a much lot or computation processing it has not been implemented. As of FHE it has been theoretically, mathematically and empirically been proven to be more secure and threat proof than other encryption schemes been in use.

III. LITERATURE REVIEW

Kaur et al. [1] introduce an enhanced hybrid encryption technique which utilizes the three algorithms altogether, i.e. RSA, Triple DES and Random Number Generator. This technique provides more flexibility and security in multilevel than the other techniques already deployed in the cloud. This software will also help the cloud user un guiding and telling them that which encryption algorithm or technique will provide great performance, which will process the data in faster speed and which will provide most powerful encryption regardless of speed and performance. Through the guidance of this help the user will be able to select the encryption technique that best suit their data protection. The re-encryption mechanism also takes place in the proposed scheme which shows that the cipher-text data is encrypted once again for duality. The proposed scheme is very useful in cloud application where privacy is of great issue. This scheme provides security proof over the cloud and is advantageous in secret sharing, cloud-database integration etc. Due to its nature of double encryption on data, it require more processing than normal encryption methods. As all of encryption mechanisms need random number, prime number or both of them to perform their Key generation process, therefore in proposed technique RNG is used for this purpose which will be likely to produce non-periodic random number that will be long enough that the sequence of numbers couldn't be repeated. 3DES is used for the encryption of block of data. It is used in 2 ways, i.e. (-EDE) or (-EED). Both of the ways utilize total of 168-bit encryption key. RSA is used for establishing secure communication connection between users with authentication of their Digital Signature. This proposed Hybrid technique provides flexibility to the user in the sense that the user will be free to choose any of the 3 algorithms

to be applied on the data or all of them. The main disadvantage of the technique is that as the level of computation (encryption/decryption) increases, the performance of the Query processing decreases. But in future this computation time can be decreased by using some more appropriate methods. There are still many ways to provide security to the data in the databases in cloud.

Huang et al. [3] introduced and proposed a new encryption scheme which can also be applied to the databases in the cloud. This encryption technique follows the asymmetric encryption mechanism and uses the concept of Commutative Encryption & ElGamal Encryption methods. In commutative the encryption on data is done more than once & the order of public/private key used for encryption/decryption also doesn't matter. Whereas in ElGamal encryption on which proposed technique is based, asymmetric & Diffie-Hellman encryption schemes are used. The proposed scheme uses following four algorithms:

1) **System Setting**
2) **Key Generation**
3) **Encryption**
4) **Decryption**

The re-encryption mechanism also takes place in the proposed scheme which shows that the cipher-text data is encrypted once again for duality. The proposed scheme is very useful in cloud application where privacy is of great issue. This scheme provides security proof over the cloud and is advantageous in secret sharing, cloud-database integration etc. Due to its nature of double encryption on data, it require more processing than normal encryption methods. There are still many ways to provide security to the data in the databases in cloud. There are many techniques whose features could be merged to form a new powerful and light weight technique or model which could minimize many of the major security issues in cloud. Providing security to the data in the cloud is very much important because all of the transactions and processing being carried out on the cloud is based on these data and information. There are many techniques developed for providing highest level of securities and attain maximum protection to the cloud and preserve user's & organization's data. Because the nature of the cloud is predictable and number of users in cloud are increasing day by day. There is much work which is remaining for the betterment of achieving highest level of security in cloud. In the future I will propose a new technique or model which will provide maximum security to the cloud's data and will minimize many of the major security issues concerning with the cloud's data.

Pagano et al. [2] proposed a new technique for the privacy and security of sensitive data over un-trusted cloud environment. They named this technique as In-Memory DataBase (IMDB) encryption. The proposed technique suggests that:

1) There should be a synchronizer between the owner and the client seeking the data. Client will be required a key from the synchronizer to decrypt the encrypted shared data it receives from the owner of it in cloud. The purpose of the synchronizer will be to store the correlated shared data and the keys separately.
2) The main memory or RAM will be used for the data storage of database management system. This eliminates the use of cache in process of transferring data from the hard disk to memory. This also increases the performance as well. As these DBs will rely on the main memory so they will require much less space than traditional DBs. This proposed technique is called In-Memory DataBase (IMDB).

Through the collaborated & combine use of above two proposed techniques, only the client's owned data will be stored in IMDBs while other (external) data to be added or deleted will be stored in synchronizer. Row-level encryption will be used in IMDB only for received data, which encrypts only received rows. The disadvantage in proposed technique is that the delay can occur only due to the communication effort with the central synchronizer. But this can be decreased and reduced by adopting group encryption & also minimizing communication between nodes & synchronizer.

Alzain et al. [4] has emphasized on the security of data as well as databases stored in the cloud environment. As the cloud computing and the sciences of cloud has recently been evolved in the recent years, therefore most of the security issues related to the privacy and thefts to the data and information in there have not been solved because of which many organizations are not transferring into cloud. In this paper authors have tried to resolve three main types of security issues, i.e. the integrity of data, data intrusion and the availability of service in cloud. Their proposed model suggests that the data and information that is sensitive and is to be secured & hided as well from the external unauthorized access has to be stored in multiple clouds or cloud databases. One of the secret sharing algorithms, i.e. "Shamir's Secret Algorithm" have also been used and utilized in the proposed model this algorithm plays very important role in the whole process of this model. The data that is to be hided from the internal or external unauthorized user or from malicious hacker, is first divided into chunks or parts and is then stored on multiple CSPs in cloud. This software will also help the cloud user un guiding and telling them that which encryption algorithm or technique will provide great performance, which will process the data in faster speed and which will provide most powerful encryption regardless of speed and performance. Through the guidance of this help the user will be able to select the encryption technique that best suit their data protection. The

re-encryption mechanism also takes place in the proposed scheme which shows that the cipher-text data is encrypted once again for duality. The proposed scheme is very useful in cloud application where privacy is of great issue. This scheme provides security proof over the cloud and is advantageous in secret sharing, cloud-database integration etc. Due to its nature of double encryption on data, it require more processing than normal encryption methods. After storing, the Shamir's secret algorithm generates a polynomial function against each chunk that is stored in different CSPs. This provided more security to data because the attacker/hacker retrieving or seeking data should have all the values of polynomial functions to decrypt the data he requires, otherwise having value of one polynomial function of one chunk doesn't retrieve any data unless it would have all polynomial values of all the chunks stored at different multiple CSPs. This model has resolved the three main security issues discussed earlier, i.e. intrusion to the data, availability of service and the integrity of data. As compared with the single cloud database models, this proposed model is far more superior then that of them and has resolved much security issues and factors.

The privacy and protection of data in cloud has higher priority than all other services. As the users in cloud store as well as transfer their data that could be very sensitive and brittle, so it should be protected and secured as well so that confidentiality of such data should be hided from the malicious attack and unauthorized access. For this purpose, Delettre et al. [5] proposed a technique to resolve such protection and security issues of data especially in the data bases residing in the cloud. In this model they have introduced a concealment concept. This model suggests that the real data is integrated and merged with the visual fake data to make the real data's volume fake or falsify. Meanwhile it allows the authorized users to easily differentiate and separate the fake data added and find the required real data. This technique somehow increases the overall volume of real data but provide some enhanced security to the private data of cloud' user and organizations. The objective of this proposed technique is to make the real data safe and secure from the malicious outside unauthorized user or attacker. To make surety of this objective, a water marking method is used which uses a key for the real data. Only the user that is authorized owner of the original data has that key of water marking. Through the collaborated & combine use of above two proposed techniques, only the client's owned data will be stored in IMDBs while other (external) data to be added or deleted will be stored in synchronizer. Row-level encryption will be used in IMDB only for received data, which encrypts only received rows. The disadvantage in proposed technique is that the delay can occur only due to the communication effort with the central synchronizer. But this can be decreased and reduced by adopting group encryption and also minimizing communication between nodes & synchronizer.

IV. PROPOSED MODEL

There are two main components of the proposed model/system:

- ⇨ User-defined File Specific Encryption&Decryption keys & Engine @ User side
- ⇨ File-Specific Decryption Engine @ Client side

File-based System

It is to be clarified that the proposed system uses the file-based mechanism. In the research paper, file-based or file-specific means any type of information or data that is to be securely stored at cloud storage. The *file* does not mean that the data to be stored or retrieved to/from the cloud should be in the form of file/s, it can be in any form that could be stored in either cloud database and cloud file storage system. The main aspect of the proposed model is to secure the file, which is basically the original data or information in a form that could be stored in cloud database or cloud file storage, and perform query computation on encrypted data. Moreover as homomorphic mechanism is adopted where all of the computation and processing is performed on the encrypted data rather than the actual data, therefore, it does not matter whether the encrypted information is a file or data. Therefore, in all of the further discussion and explanation, wherever the word/phrase file-specific, file-based, or file will be stated, it means the data to be securely stored in the cloud database or cloud file storage. Here file storage means any type of storage used for storing encrypted data on cloud, i.e. database or file storage. For our purpose we have conducted experimentation on cloud file storage but the storage is still the encrypted data name or stated as *file*. Cloud database storage will be used in our future work experimentation. It should also be clarified that rather than encrypting and sending the whole database or file storage in cloud we will be encrypting data, stated as *file,* and then sending it in the cloud for secure storage.

The Proposed Model

The proposed model is based on the File storage system in cloud which means that the data of the user will be stored on cloud in form of files. The reason for not choosing other systems, like MySQL based cloud databases, is that **1)** the data cannot be encrypted by the user with his own defined encryption algorithms or schemes instead he is only bound to use the built-in encryption schemes and functions of MySQL. **2)** Other reason was related to the vulnerability and stated that even the cloud vendor cannot be trusted for not being making misuse of the data that the user has handed over to him with trust. Even the administrator of the vendor on cloud cannot be fully trusted in keeping the data secure because may be he will misuse the data for any reason at any time or an attacker can have access to the

administrator privileges by any unlawful means and can exploit the user's data. So, therefore, for these reasons and drawbacks, DBMS based system in cloud is not being selected in the proposed model. Moreover when the owner or the user will not trust the cloud vendor and will be trying not to be dependent on it then in sense of data's security, then the level of trust worthiness between these two parties, i.e. user and the cloud vendor, will not remain live & active and successful transmission will not be carried out then. Therefore to avoid all of these headache problems and provide user a well vendor dependent-free environment (security on data not defined/deployed by vendor) in which he will be able to implement his own encryption method/algorithm with the data and not to rely on vendor for this purpose, as like in DBMS system, a model has been proposed which will be utilizing the File storage system of "OpenStack". The advantage of file storage system in our model is that it provides the opportunity to define our own encryption/decryption algorithm and attach it with the data, which will only be decrypted by the owner of it or by the user to whom the owner has shared the keys for its decryption. In this way the cloud vendor providing the services of file storage will also be unaware of the data that it will be receiving from the user's end.

A. User Defined Encryption/Decryption Keys & Engine

The working of the proposed model as discussed above is divided into two major components. In this section the working of the first component that is on the client side is discussed. First of all the component will receive the data or file that the user want to encrypt and store it in the cloud file storage. The function of the component here will be to encrypt that data/file with the user-defined encryption algorithm. One thing to be clarified is that the file before being sent to the cloud will compose of two portion/parts. One part will be containing the encryption of the context of original data/file and the other part will contain the decryption keys of this encrypted data/file. Initially the context of the data/file will be encrypted using the encryption algorithm defined by the user and after that the decryption keys to this encrypted data's context will be encrypted with the same algorithm used for the data's encryption and will be attached to encrypted data. Now this whole will be the final encrypted file, composed of two parts, to be sent to the cloud. It is to be noticed here that as the Encryption and Decryption's algorithms is defined by the user, therefore, these algorithms are said to be **User Specific**. Moreover as each file will have its own encryption attached to it, therefore, the decryption keys will also be definitely be different for each and every one of it and hence Decryption keys are said to be **File Specific**. Up till now all of these functions and processing are performed on the user's end.

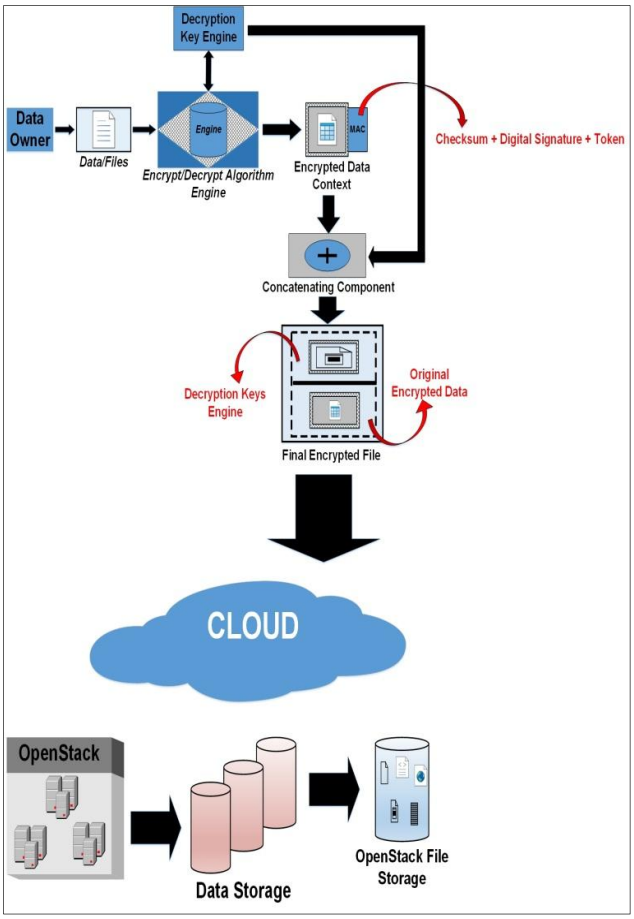

Fig. 1: Encryption Process at User Side

B. SSL-128 Tunneling

SSL tunneling is basically the establishment of the SSL connection and handshake protocols that are being deployed between the client and cloud or web server. This SSL when activated and established will create a tunnel like connection, in which the client and the web server can communicate securely without the intervention of outside unauthorized user or attacker. If for any reason or circumstances attacker get access to the data being communicated through the SSL tunnel, even then the data he will be accessing will be useless for him because it will be encrypted by such algorithms which will only be known to the owner or the authorized user of it. All the communication being communicated between the client and the cloud/web server will be in encrypted form. The data being communicated will comprise of dual encryption, i.e. encryption performed on data at client side and the encryption performed by the SSL tunnel or protocol. In the proposed model all the transmission is carried out through the SSL tunnel, i.e. from the enrollment or authentication of the user to the cloud directory till the retrieval of the data

from the Cloud "OpenStack" file storage system. The question rises what basically SSL/SSl-128 is.

Secure Socket Layer (SSL) works over public key and is a certification-based general purpose protocol, developed by Netscape for encryption of information. SSL allows an SLL enabled server to authenticate itself to an SSL enabled client and vice versa, enabling both machines to establish an encrypted connection. Server operating securely generally obtains an SSL key and certificate pair from an issuing authority. It then makes these available on the server itself and announces the availability within the protocol exchanges between the server and the client. The exchange between the client and the server is initially started with a Handshake of SSL known as SSL Handshake where both the machines exchange information about the encryption algorithm to be used throughout their connection by the SSL certificate. Once the handshake protocol has been completed between the web server and the client then the information being communicated between them will be in encrypted form and can be decrypted by them only. Anyone else other than the client and server trying to access the data being communicated will receive the encrypted information which will be useless for him because the decryption keys will only be known to the authenticate parties of the SSL tunnel. This is the reason for which the proposed model has utilized SSL-128 specifically. SSL-128 requires more length of bits than the previous SSL bit version. These numbers of more bits make it complex enough for the brute force attacker to decrypt it. The beauty of the SSL-128 is that it requires 2^{88} combinations to break this encryption tunnel or layer with the brute force attack which is more than enough.

C. Owner/User Authentication/Enrollment

After the successful establishment and deployment of the SSL connection and its handshaking protocol, the user is needed to authenticate before the actual retrieval or manipulation of data. If the user is accessing the data for the first time and is not registered in the Cloud Directory then he will be needed to first enroll himself in the Cloud Directory then after the successful enrollment and authentication as well he will be allowed to retrieve the data. In this phase the user who wants to send the data to the cloud database for the storage will be needed to authenticate him to the cloud directory where he would have registered himself before. If the user is registered then the cloud directory will check its existence in its directory and if he matches with the directory then he will be redirected to the cloud storage site. In case if the user is accessing the data for the first time or is not registered then he will be redirected to the user registration or enrollment site where he will have to first enroll himself and then after successful enrollment he will be redirected to the data for which he would have requested. Moreover it could be clarified from the figure below that how the enrollment/registration and authentication process occurs. To provide somewhat more security to the data, the user authentication will be requiring some more things than the usual authentication, i.e. with the username & password it will also be requiring the secret question which was asked at the time of registration and is only known to the authenticate user.

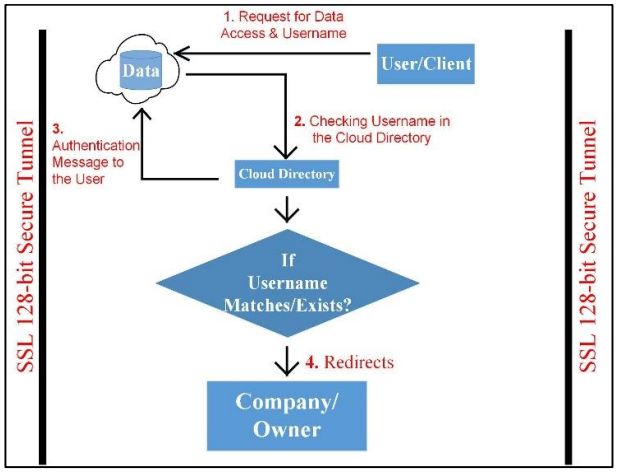

Fig. 2: User's Authentication from Cloud Directory

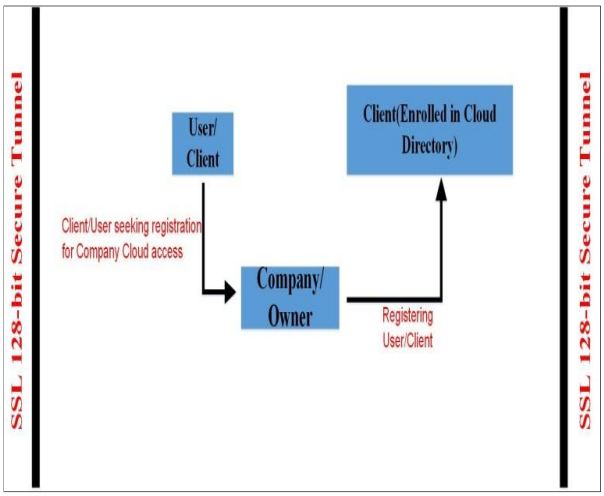

Fig. 3: User's Registration Process to Cloud Directory

D. File Specific Decryption Engine

After the successful establishment of the SSL tunnel and the authentication/enrollment of the user into the cloud directory, he will now send the encrypted data/file to cloud vendor's storage area.

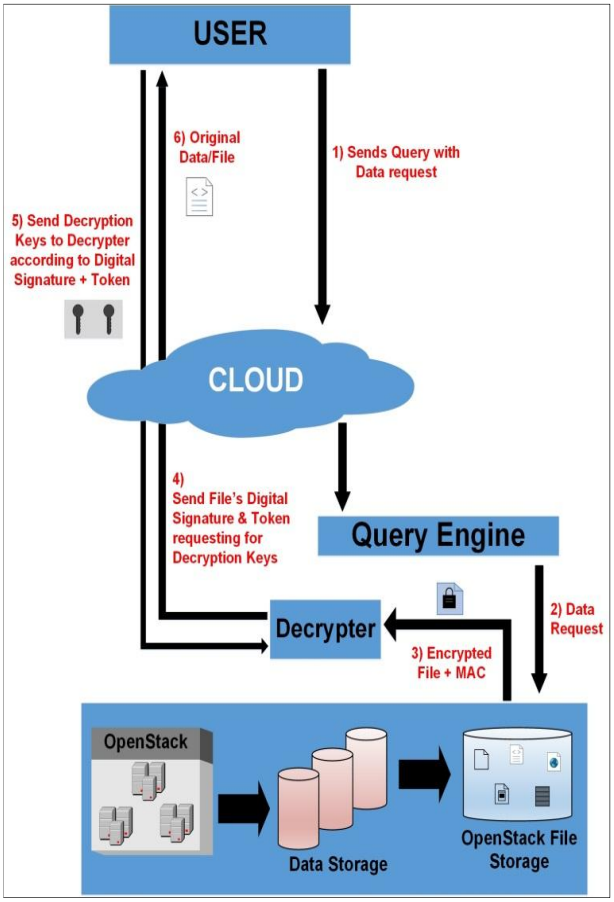

Fig. 4: Data Retrieval Process against User's Query

As discussed above that there are two main components of the proposed model, one will be on client side and other will be relying on the cloud server side. As the "OpenStack" is file based storage system therefore after receiving the data/file from the user it will store it into its file storage unit without knowing that what relies inside the encrypted file. Now the second main component of the proposed model will come in action when the retrieval of the file is being requested. For the retrieval of the data/file the user will be requesting it in the form of queries. The query he will be sending to the cloud vendor will be consisting of two things, one will be the request for the data and the other will be the decryption keys or values of the encrypted file requested. Now the question arises here that if the decrypting keys or values are sent in this why inside the query then they could be exploited easily or could be used against the encrypted data by the unauthorized users or attackers, so here it is important to mention that now worries should be taken because these decrypting keys or values will not be in a simple way that could be easily used by the attacker so easily instead they will be concatenated with the query in such a way that it will only be known after computation on the query by the decryption engine of the proposed model at the cloud vendor side. As the file consists of two parts, i.e. the encrypted decryption keys part and the encrypted data itself, the decryption keys obtained from the query will be used to decrypt the first part. As the decryption keys of the encrypted data are now been decrypted, therefore, they will now be sent along with the encrypted data/file as an attachment to the user who requested. User will have the original context of data/file after decrypting it with the decryption keys it received along with the encrypted data.

V. VALIDATION

To give credibility to the proposed model and work, it has been validated by conducting its implementation through some mathematical equation and some simulation. The implementation of proposed model found working perfectly on any Windows or Linux operating environment and having access through the browser (Firefox, chrome etc.) to the cloud (OpenStack) storage. At the client side we used Intel Core i7 CPU E7500 @ 2.5GHz with 8 GB RAM. VPN network was also utilized to link client directly to the cloud (enterprise). For this purpose we have used Gladinet, which works as a Cloud Accessing solution. This Gladinet will be already installed on user's system or he will have to install it for access to the OpenStack Cloud storage. User will be sending the files/data to the OpenStack through Gladinet interface and then the OpenStack will be storing in its file storage system through OpenStack Swift. Computer Algebra System MAGMA tool was used in logarithmic process of encryption at the client side.

Keeping everything else constant a little overhead of around 5% more processing time has been observed when the data was encrypted with Symmetric algorithm i.e. Advance Encryption Standard (AES) with the block size 128-bit which can be ignored or can be cater by increasing the resources such as RAM and CPUs.

As most of the work and processing to take place occurs in the backend or background, therefore, user cannot view it in frontend other than the Owner of it. Also frontend processing of the working proposed model is shown in the following figures.

Figure 5: Login to Gladinet Account

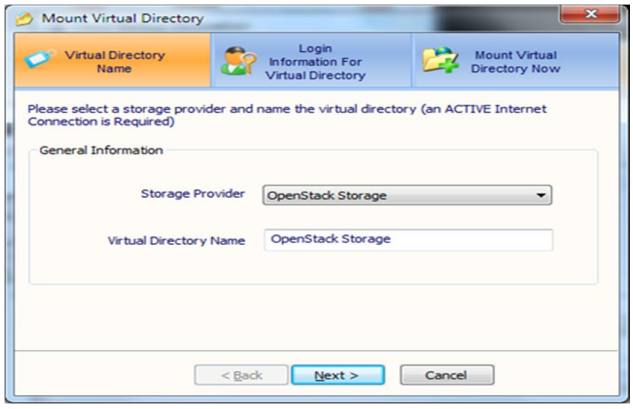

Figure 6: Linking Virtual Directory on Gladinet to OpenStack Storage

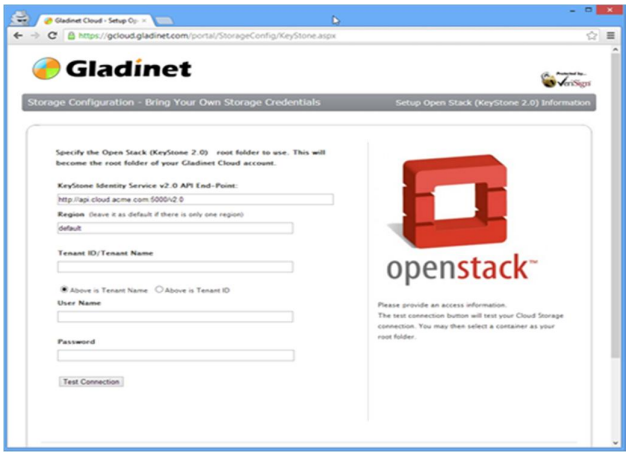

Figure 7: Gladinet-OpenStack Test Connection Setup

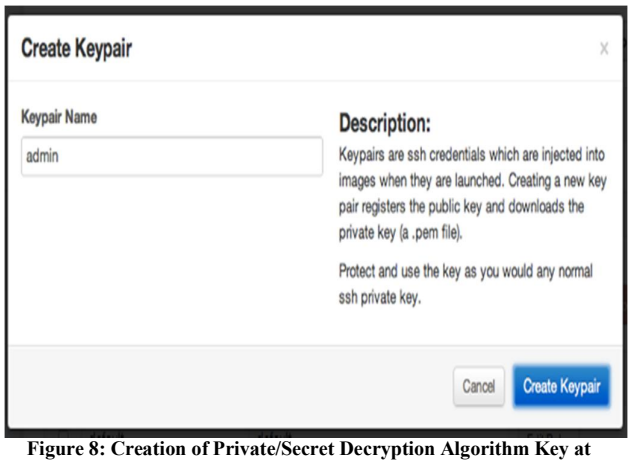

Figure 8: Creation of Private/Secret Decryption Algorithm Key at User's end

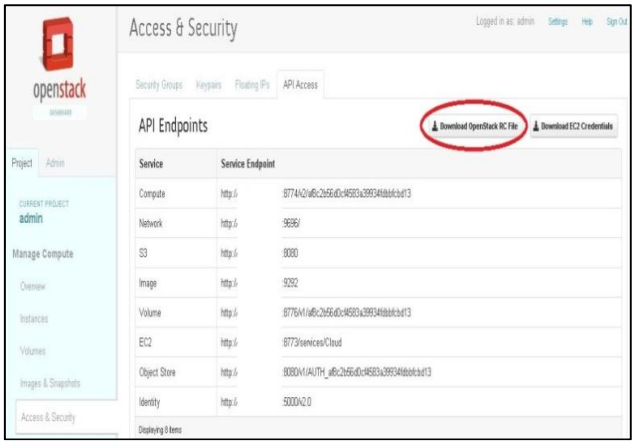

Figure 9: Downloading of File Credentials to be used in future

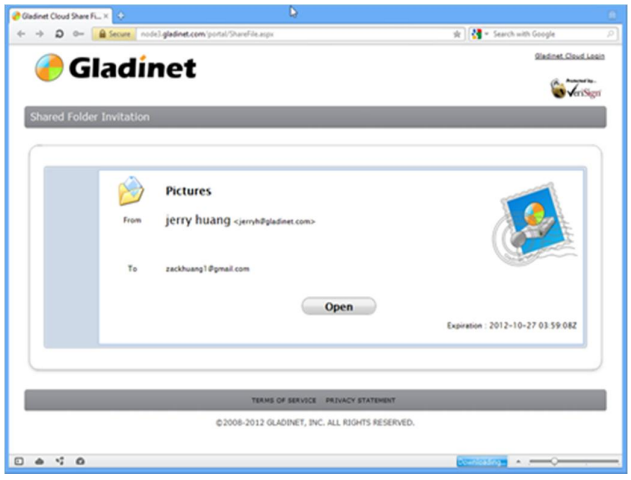

Figure 10: Original file retrieval after all decryption processes

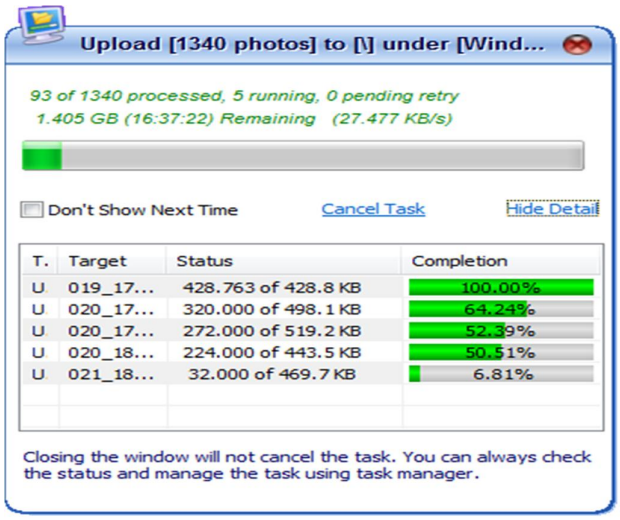

Figure 11: File uploading to OpenStack Storage through Gladinet view

VI. CONCLUSION & FUTURE WORK

In this paper, a new model has been proposed which comprises double encryption schemes on the data that is to be stored on the cloud vendor's storage system. The storage system that has been selected is the "OpenStack" File Storage System due to many of its advantageous features over others system like DBMS based etc. Dual encryption will be working in such a way that first data will be encrypted by him and then it will be again encrypted by the SSL-128 tunnel for transmission. Advantage & benefit of the proposed is that the encrypted file it sends to the user is composed of two parts, i.e. encrypted data's context and encrypted decryption keys. The decryption key/value of both will only be known to the owner or authenticated user only, which he will be sending in form or query to the cloud vendor for execution. The model has been designed in such a way that each & every file will have its own encryption/decryption algorithm and keys. Here the Encryption engine will be User-specific and Decryption keys will be File-specific. With the new design framework, the proposed model, also provides much more security than the already been deployed or been used models in the cloud environmental architecture. Also the proposed model has been graphically validated with different parameters against security.

The future work for the proposed model will be to make it more enhance, efficient and robust that it could also be manageable & provide security proof mechanism against the malicious and other vulnerable attack advancing in IT industry.

REFERENCES

[1] A. Kaur and M. Bhardwaj, "Hybrid Encryption for Cloud Database Security", *International Journal of Engineering Science & Technology* [IJESAT], Vol.2, pp.737-741, 2012.

[2] F. Pagano and D. Pagano, "Using In-Memory Databases on the cloud", *1st International Workshop on Securing Services on the Cloud* (IWSSC), pp.30-37, 2011.

[3] K. Huang and R. Tso, "A Commutative Encryption Scheme based on ElGamal Encryption" *International Conference on Information Security & Intelligence Control* (ISIC), pp.156-159, 2012.

[4] M. A. AlZain, B. Soh and E. Pardede, "MCDB: Using Multi-Clouds to Ensure Security in Cloud Computing", *9th IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp.784-791, 2011.

[5] C. Delettre, K. Boudaoud& M. Riveill, "Cloud Computing, Security and Data Concealment", IEEE Symposium on Computers and Communications (ISCC), pp.424-431, 2011.

[6] R. Hayward and C-C. Chiang, "An Architecture for Parallelizing Fully Homomorphic Cryptography on Cloud", *IEEE 7th International Conference on Complex, Intelligent and Software Intensive Systems*, pp.72-77, 2013.

[7] S. K. Sood, "A Combined Approach to Ensure Data Security in Cloud Computing", *Journal of Network and Computer Applications*, Vol.35, pp.1831-1838, 2012.

[8] D. Purushothaman, and Dr. S. Abburu, "An Approach for Data Storage Security in Cloud Computing", *International Journal of Computer Science Issues (IJCSI)*, Vol.9, Issue 2, pp.100-105, 2012.

[9] C. Gentry, "A fully Homomorphic Encryption Scheme", 2009.

[10] Y. Gahi, M. Guennoun, Z. Guennoun, and K. El-Katib, "Securing Internet Application Using Homomorphic Encryption Schemes", *Journal of Theoretical and Applied Information Technology (JATIT)*, Vol.47, pp.479-495, 2013.

[11] B. K. Mohanta, and D. Gountia, "Fully Homomorphic Encryption Equating to cloud Security: An Approach", *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol.9, Issue 2, pp.46-50, 2013.

[12] A. A. Atayero, and O. Feyisetan, "Security Issues in Cloud Computing: The Potentials of Homomorphic Encryption", Vol.2, pp.546-552, 2011.

[13] M. Asad, "A Framework for Resource Allocation Strategies in Cloud Computing Environment",*35th IEEE Annual Computer Software and Applications Conference Workshops*, pp.261-266, 2011.

[14] C. Delettre, K. Boudaoud, & M. Riveill, "Cloud Computing, Security and Data Concealment", *IEEE Symposium on Computers and Communications (ISCC)*, pp.424-431, 2011.

[15] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data", *International Conference on Data Engineering*, pp. 1156–1167, 2012.