Techniques Used to Formulate Confidential Data by Means of Fragmentation and Hybrid Encryption

Olly Beckham a
Gord Oldman b
Julie Karrie c
Dorth Craig d

Article history:
Received: 18 June 2019
Accepted: 30 September 2019
Published: 15 October 2019

Abstract
Cloud computing is a concept shifting in the approach how computing resources are deployed and purchased. Even though the cloud has a capable, elastic, and consistent design, several security concerns restrain customers to completely accept this novel technology and move from traditional computing to cloud computing. In the article, we aspire to present a form of a novel architectural model for offering protection to numerous cloud service providers with the intention to devise and extend security means for cloud computing. In this work, we presented a two-tier architecture for security in multi-clouds; one at the client side, and other at the server side. The article presented a security domination outline for multi-clouds and supports security needs like Confidentiality, Integrity, Availability, Authorization, and Non-repudiation for cloud storage. Through this document we have anticipated, HBDaSeC, a secure-computation protocol to ease the challenges of enforcing the protection of data for information security in the cloud.

Keywords:
authorization; cloud computing; novel technology; protection; traditional;

Introduction
The abundance of the digital world continues to augment the necessity of novel storage space as well as net utilities, besides with a growing requirement for additional expenditure for effective handling of storage capacities and network bandwidth for information transmitted. The employment of the distant storage method is achieving esteem, specifically the cloud storage based services, as they grant beneficial architecture. This style supports the communication, storage and intensive calculation of outsourced information on a pay per use business model. This

a University Of Surrey, Guildford, UK
b University Of Surrey, Guildford, UK
c University of Huddersfield, Huddersfield, UK
d University of Huddersfield, Huddersfield, UK
extensive significance in cloud storage services primarily arises from agencies looking for further flexibility and outlay valuable systems. That is the profit of cloud espousal is very substantial in a later period of receptiveness, usefulness, and competence in “Information Technology” service delivery. Consequently, expending huge quantity of assets on buying high-priced application is no longer required. These reasonable profits present the chief crucial inspiration for cloud acceptance as they assist the enterprises plummeting the Capital Expenditure (CapEx), kept to procure permanent assets and the Operational Expenditure (OpEx) which is an ongoing expenditure for accepting a product business or a system.

The clause is structured as ensuing: Section 2 discusses the role of cloud computing and the need of multi-clouds. Section 3 illustrates the importance of security in clouds and principles. Section 4 provides the motivational scenario and the main contributions of the paper. Section 5 highlights the literature survey done. Further, Section 6 describes the techniques used to formulate confidential data by means of fragmentation, and hybrid encryption and discusses Blowfish and Homomorphic encryption in detail. Section 7 explains the proposed HBDaSeC system model and operations to be operated on files in multi-clouds. In section 7 a framework for secure data storage on the cloud is proposed. A detailed description of the framework along with all components and phases are described in the chapter. Section 8 provides notes on the implementation of the framework introduced in section 7 and it also describes the performance of the framework. Section 9 provides concluding remarks and as well as routes for possible enhancements.

Cloud Computing: Characteristics, Service Models and Deployment Model

Over the duration, computing models have transformed commencing scattered, similar, and a grid to cloud computing. Cloud computing can be named as an innovation used to convey benefits through the web as a medium. Cloud computing comes with numerous intrinsic abilities such as on-demand resource distribution, abridged management efforts, elastic pricing form, and simple applications and services stipulation.

Cloud computing comprises three key service models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) and their description are presented in Figure 2 below.

---

Cloud services can be provided as four basic cloud delivery models as shown in Figure 3. Five major cloud actors each concerned with performing diverse roles are cloud broker, the cloud provider and consumer, cloud auditor, and cloud carrier.

With the arrival of data, both individuals and enterprises are producing a huge amount of data every day, which profoundly influences our living style. On one hand, the explosively increasing data creates the limits for data storage and data transmission; on the other hand, data has become an important productivity resource. Therefore, it is essential to invent a proficient data sharing model to share huge amounts of data among different individuals or organizations. Obviously, cloud computing is certainly a perfect data sharing tool for its vast storage space and reliable distributed storage system. However, even if there are various advantages of cloud computing, security and privacy concerns are the primary obstacles to wide adoption.

Why Multiclouds are essential? The multi-cloud approach employs two or more clouds and consequently avoids dependence on one individual cloud. Vukolic stated that multiclouds encompass improved trust, protection, and dispersed reliability amid several cloud providers. Abu Libdeh prefers multi-clouds so as to elude “vendor lock-in” by storing user’s information amid various clouds. Multicloud scheme has the capability to lessen the hazard of service accessibility collapse, failure and corruption of information, privacy hindrance in addition to the view of malicious insiders in the single cloud, increased virtual power by merging the infrastructure of several cloud providers by offering application programming interface abstractions which has made easy the management of multiple cloud providers at the same time, and increased flexibility by offering hardware, software and infrastructure redundancy and seer traffic from different customers through the fastest possible parts of the network. It has incorporated diverse aspects of security like confidentiality, integrity, availability, proficient retrieval and information sharing.

The main multiclouds models are DepSky to construct clouds of clouds and are considered to be the finest model as it provides all the security issues like availability, integrity, and confidentiality and avoids vendor lock-in problem. HAIL is High Availability and Integrity Layer which provides a software layer by combining cryptographic protocols with erasure codes to provide integrity and confidentiality. RACS is a redundant array of independent disks (RAID) like practice solving vendor lock-in problems but it fails to provide confidentiality and security issues. InterCloud Storage (ICStore) implements all client-side functionalities as a library and has three layers which offer confidentiality, integrity, reliability, and consistency. MultiCloudDatabase (MCDB) is considered to be the advance side of DepSky.
and uses distributed method to grant privacy with database management systems.

Security in Clouds

Cloud computing is mentioned to develop dynamically and diverse cloud aspects are mentioned to emerge. The document intends to recognize the security issues linked amid cloud storage and highlighted the significance of security principles used for cloud storage.

Importance of security in cloud computing

The control, suppleness, and easiness of employment of Cloud Computing arrive with a number of challenges/issues. In particular, security has been extensively reported to constitute the main factors which prevent migration to the cloud. Even if a corporation states to include peak class security and does not renovate its security policies from time to time, it will be vulnerable to security breaches in near Vista. In this view, via this manuscript, we intend a novel practice to alleviate the security challenges through Integrated Encryption approach.

The distance amid the user and the physical site of his information construct a hurdle as this information can be leaked by a third party affecting the privacy of customers information. The employment of conventional encryption schemes to encrypt the remote data prior to transfer to the cloud service provider have been extensively used to viaduct this security breach. However, the customer should provide the secret key to the server to decrypt the information prior to carry out the requisite calculations. Homomorphic encryption permits performing computations on encrypted data without decryption.

Cloud Computing Security Principles

Information security ethics at diverse stages of cloud applications are emphasized by enforcing confidentiality and protection of data. Ramgovind, Eloff, and Smith defined six information security requirements in Cloud computing (Ali et al., 2015).

a) Identification and Authorization: involves verification and validation of cloud user’s using security checks for user- names and passwords and then check the permission of access.

b) Authorization: Referential integrity is guaranteed through this process to control and privilege processes in the cloud.

c) Confidentiality: It is a chief prerequisite to preserve power over the information of numerous organizations that could be positioned across some distributed databases.

d) Integrity: is to assure the accuracy and consistency of information.

e) Nonrepudiation: It can be maintained through safety protocols and token provisioning, such as using digital signatures, timestamp, etc.

f) Availability: It is an important factor while considering the delivery model and selecting the service model and must be part of SLA.

The security concerns of distributing information to the clouds laid the driving force for the expansion of data security practice able of addressing the aforesaid critical issues.

Motivational Scenario and the Main Contributions of the Paper

Users deduce that cloud service providers certify that while information is in transit commencing clients premises to the cloud servers, its security constraints would not be distorted, and their information can be transmitted securely to corroborate an increased data security. However, the significant security challenges have augmented consequence and should be cautiously addressed. Thus, cloud clients deal with the challenges of selecting appropriate cloud service providers and assess security implementations depending on their security needs.

Key contributions of the paper are as follows. The subsequent is an outline of our effort:

a) We intend a procedure for safe allocation of cloud data. This practice adeptly offers the reliability assurance to customers as a cloud service provider is considered reliable in storing all the records and these records cannot be exposed to malicious or illegal users.

b) We verified that security in terms of confidentiality and availability of the projected system against attacks.
We authenticate the results of the planned method via analysis and assessment. Our system takes less storage space and execution time to perform the queries and offers better security and is highly efficient.

2 Materials and Methods

Security is a vital facet of all information processing conduct and every organization has to extend mechanisms and tools to sustain and guarantee the security of their information resources. Engoulou et al., (2014), had mentioned that a lot of research activities are being carried out to address cloud security threats. The paper motivated the need for effective cloud security countermeasures by discussing the cloud security issues. The motivation of use of multi-clouds was described along with the proposed set of four distinct Multicloud architectures. Each of the introduced architecture provides its own security merits and flaws were started with respect to the security requirements by using diagrammatical presentation. Case studies discussing real-life examples were also mentioned in each scenario. Security, feasibility, and regulation were considered for comparison. Fernández-Alemán et al., (2013), explored cloud security in terms of issues, solutions, and shortcomings. The author correlated data confidentiality and user authentication. Gubbi et al., (2013), suggested the usage of homomorphic encryption for storage on the cloud. The author described security and privacy constraints, issues, and approaches. Keshavamurthy et al., (2012), aimed to understand the security issues and highlighted the significance of data integrity schemes used for cloud storage. The paper had defined taxonomy to explain all aspects of data integrity considering its attributes like nature of data, deployment model used, and nature of metadata employed. The security attacks and mitigation techniques were discussed.

Miorandi et al., (2012), presented a novel architecture of security which acts as an interface amid end users and private cloud service providers. The major contributions of the paper included high available cloud storage gateway (HASG) architecture. Data reliability and fault tolerance, file fragment algorithm utilizing information dispersal algorithm divided the file into redundant partitions and stored on different cloud storage. Mokhtar & Azab (2015), presented a pioneering method for safe data storage on the cloud through the collective use of cryptography and Disintegration Protocol and validated Integrity and Confidentially, security constraints against internal and external attacks. Mollah et al., (2017), proposed for a multi-cloud system for encrypting, splitting and storage of data. In case of any failure in existing CSP or two CSP’s, a model was proposed for efficient working. The system limitation is increased time but the security is assumed to increase. Nguyen et al., (2015), defined a new scheme called storage correctness and fine-grained access provision (SCFAP) which utilized hierarchical structure to allow users an exclusive access. The storage correctness verification of the outsourced data was done using a token granting mechanism. The future work mentioned was to deploy SCFAP scheme for outsourced data decryption techniques. Raza et al., (2013), emphasized an auditing process involving a TPA to achieve data integrity and privacy. The research on cloud data auditing was mentioned to focus on the verification, privacy preservation and integrity of stored data using cryptographic methods.

Sobh (2006), presented the architecture overview with algorithms for file slicing and encryption and file decryption and Merging. The comparison table of various existing schemes and the proposed model were listed in terms of Turnaround time, encryption process time, decryption process time, security features (Privacy, Insider attack, secret keys, confidentiality and data integrity) and Reliability features (File formats supported, collusion attack, Key Escrow, Malicious files, File size). The future enhancements suggested implementing dynamic data slicing using 3DES. Weinman (2017), mentioned the use of highly dispersed compute and storage elements in the form of “FOG”. Multicloud FOG is a hybrid architecture working together in an integrated fashion providing highly dispersed facilities. A real-life example is Walmart which uses OpenStack for its private cloud, Rackspace, Azure, and other public clouds.

Srivastava & Nandi (2014), provided a new efficient remote data possession checking (RDPC) protocol based on Homomorphic Hash function. The presented scheme was mentioned to be secure against several attacks. The performance analysis showed the computation and communication cost to be reduced. Ali et al., (2017), proposed a model called DaSCE (Data security for cloud Environment) with a semi-trusted third party which provided key management, access control, and file assured deletion functionalities. DaSCE is mentioned to utilize Shamir’s (k, n) threshold scheme. Scyther tool was used to graphically analyze and verify security protocols.

Zissis & Lekkas (2011), discussed the problems of big data storage in the cloud like data security, data authentication, data integrity, data availability and data de-duplication. Architecture, SecSVA: A secure storage, verification, and auditing for big data in the cloud environment were discussed supporting the data authentication, verification, auditing, integrity and confidentiality for cloud storage. An attribute-based security framework with
secure de-duplication for Big data storage in the cloud was discussed. The architecture comprised of various entities like client, data service provider, cloud service provider, trusted party auditor (TPA) and Kerberos server. From the analysis, the proposed scheme was mentioned to withstand different attacks. Wei et al., (2018), suggested for the Multi-cloud environment. The encrypted data block and key blocks are together distributed to multiple cloud service providers. The verification of correctness of data exchange is done using a cryptographic protocol. The erasure correcting code was employed to support reliable data storage for tolerating multiple failures among CSPs.

It evaluated the performance of the proposed large universe cipher-text based attribute-based encryption outsourcing scheme in two hardware platforms like Intel and ARM and employed key encapsulation variant. The security analysis and performance evaluation showed the planned scheme to be secure and efficient. It discussed generic cloud storage architecture and various security requirements considering confidentiality, integrity, and availability. The presented system encrypted only some bits of each data block instead of encrypting the whole file, thereby eliminating computation overhead. The Integrity was checked by the Meta-data created and inserted at the end of the original file. Metadata was encrypted by applying the AES-256 encryption algorithm and the SHA-256 algorithm was used for generating a hash of the original file. The implementation of the proposed approach was done using Amazon S3 Live cloud storage and the mechanism was used to validate the integrity check of the uploaded file.

3 Results and Discussions

3.1 Techniques Used to Formulate Confidential Data by Means of Fragmentation, and Hybrid Encryption

Encryption Techniques Employed

Encryption is the process of translating plaintext into cipher text and decryption is the reverse of the encryption process. Key size plays a significant role in the encryption and decryption process. Longer the key, more difficult is the process of decrypting the message.

Whenever a message is sent from the sender to the destination, an intruder denoted by 'I' can interfere with the communication medium who can affect the system in the following ways:

a) A message can be ‘blocked’ which violates the availability and thus the message never reaches the destination.

b) Confidentiality can be breached if the message can be ‘intercepted’ by the intruder which makes it no longer secret.

c) Integrity can be violated if ‘content’ of the message can be changed or a fake message is sent.

Homomorphic Encryption

Homomorphic encryption is the apt elucidation to resolve cloud computing security issues because its schemes facilitate to execute computations on encrypted information devoid of sharing the secret key essential to decrypt the data.

A Homomorphic encryption is: from Enc(P) and Enc(Q) it is possible to compute Enc(Func(P,Q)), where ‘Func’ can be one of the operations: +, ., exclusive of using the private key. It was developed by Ronald Rivest, Leonard Adleman, and Michael Detouzos in 1978. Homomorphic encryption (HE) is formed by four functions as shown in Fig.4.

Figure 4. Homomorphic Encryption Functions
**Taxonomy of Homomorphic Encryption:**

**Multiplicatively Homomorphic:** When an allowable act on the encrypted information is controlled to multiplication, it is known as Multiplicatively Homomorphic. Notation:

\[ \text{Enc}_k(P_1 \Theta P_2) = \text{Enc}_k(P_1) \times \text{Enc}_k(P_2) \]  

(1)

where, \( P, P \) are plaintexts. **Privacy:** HE provides more privacy than the traditional cryptographic techniques as calculations are performed on the encrypted data.

**Solves Confidentiality constraint problem:** HE resolves the confidentiality constraint when shared information is to be operated by diverse clients who want to execute reverse operations on data.

**Paillier:** In 1999, Pascal Paillier developed a probabilistic asymmetric algorithm meant for public key cryptography (Paillier, 1999) and is shown below.

---

**Algorithm 1 Paillier**

BEGIN

**Key Generation:** The parameters are:

1) Choose two prime numbers \( p \) and \( q \).
2) Compute \( n = p \times q \).
3) Compute \( \lambda = \text{lcm}(p - 1)(q - 1) \).

**Additive Homomorphic Encryption:** When an allowable operation on the encrypted data is limited to addition, it is said to be Additively Homomorphic. Notation:

\[ \text{Enc}_k(P_1 \Theta P_2) = \text{Enc}_k(P_1) + \text{Enc}_k(P_2) \]  

(2)

**Types of Homomorphic Encryption**

Homomorphic encryption is of three types as shown in Table 1 (MahaTeeba et al., 2012). A somewhat Homomorphic performs limited addition and multiplication on encrypted information.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Partial HE</th>
<th>Fully HE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of operation supported</td>
<td>It allows either addition or multiplication scheme</td>
<td>It allows both addition and multiplication operations</td>
</tr>
<tr>
<td>Computation</td>
<td>It allows a limited number of computations</td>
<td>It allows an unlimited number of computations</td>
</tr>
<tr>
<td>Computational efforts</td>
<td>It requires less effort</td>
<td>Requires more efforts</td>
</tr>
<tr>
<td>Performance</td>
<td>It is faster and more compact</td>
<td>It has slower performance</td>
</tr>
<tr>
<td>Versatility</td>
<td>It is low</td>
<td>It has high</td>
</tr>
<tr>
<td>Speed</td>
<td>It is fast in speed</td>
<td>It is slow in speed</td>
</tr>
<tr>
<td>Ciphertext size</td>
<td>It is small</td>
<td>It is large</td>
</tr>
<tr>
<td>Example</td>
<td>Unpadded RSA, ElGamal</td>
<td>Gentry Scheme</td>
</tr>
</tbody>
</table>

**Homomorphic Encryption Applications**

HE is considered to be a consolidated element for information security in cloud computing. It has several benefits:

1) **Protection of mobile agents:** It can provide protection by either using computation with encrypted function or computation with encrypted data by supporting + and operations.
2) **Secret sharing scheme:** HE can be helpful in reconstruction of a secret.
3) **Banking:** Accounts can be encrypted by the private key and stored in the bank’s server.
4) **Private information retrieval** (PIR): Sensitive data could be obtained without describing the nature of
information. Any search engine could implement HE to protect users privacy. Choose $g$ such that $g \in \mathbb{Z}_n^*$ and the order of 'g' is a multile of 'n'.

5) Choose public key $(n, q)$ and private key $(p, q, \lambda)$

**The Encryption Process:** of message $m$ to generate ciphertext $c$ such that $m < n$.

1) Select a random number $r < n$ such that $r \in \mathbb{Z}_n^*$

2) Compute ciphertext $c$ such that $c = g^m r^\epsilon (\text{mod} n^2)$

**The Decryption Process:** of ciphertext $c$

1) Ciphertext $c < n^2$.

2) Plaintext $m$ is compute as $m = t c_{\lambda} \mod n^2$ (modn)

END

Homomorphic Properties of Paillier:
Paillier cryptosystem supports the property of additive homomorphism. In this cryptosystem, the product of two cipher-texts will decrypt to the sum of their corresponding plaintexts. Considering message to be encrypted as $m_1$ and $m_2$. Enc() and Dec() are the encryption and decryption functions respectively and $n$ is from the PublicKey=(n, q), and then the additive homomorphism property can be expressed as follows.

\[
\text{Dec(Enc}(m_1, n, q) \text{ Enc}(m_2, n, q)) \mod n^2 = m_1 + m_2 \mod n
\]

(3)

Assume $c_1$ and $c_2$ be the cipher-texts of the plaintexts $m_1$ and $m_2$.

The encryption procedure of Paillier Cryptosystem is,

- $c_1 = g^{m_1} r^{n_1} \mod n^2$, for some random number $r_1 \in \mathbb{Z}_n^*$
- $c_2 = g^{m_2} r^{n_2} \mod n^2$, for some random number $r_2 \in \mathbb{Z}_n^*$

Now $c_1 c_2 = g^{m_1 + m_2} [(m_1 r_1)(r_2)] \mod n^2$ represents a valid encryption for $m_1 + m_2$.

**Blowfish Algorithm:** It is considered to be the fastest symmetric block cipher used in place of DES/IDEA. It was developed in 1993 by Bruce Schenier. It is a 16 rounds Fiestal structure with changeable key length (32-448 bits). All operations are done as addition on 32-bit words and XOR ( ) (Seth, 2016). Blowfish S-Boxes are key dependent. It operates in three phases as depicted below.

Algorithm 2 Blowfish
BEGIN

- Blowfish has 16 rounds.
- The input is a 64-bit data element, data.

**Data Encryption:**

- Divide data into two 32-bit halves: $data_L$, $data_R$.
- For $i = 1$ to 16:
  1) Compute the value of $data_L$ by find $data_L \oplus P_i$.
  2) Compute the value of $data_R$ by find $F(data_L \oplus P_i)$. 2) Interchange values of $data_L$ and $data_R$.
- Following the sixteenth round, swap $data_L$ and $data_R$ again to undo the last swap.
  1) $data_R = data_R \oplus P_{16}$.
  2) $data_L = data_L \oplus P_{16}$.

3) Finally, recombine \textit{data}_L and \textit{data}_R to get the ciphertext.

\textbf{Data Decryption:}

\begin{itemize}
  \item $P_1, P_2, \ldots, P_{18}$ are used in reverse order and is the reverse of the encryption process.
\end{itemize}

END

Graphical representation of the Blowfish algorithm is shown in Figure 5(a) taking into account the fiestal structure. The $F$ function is shown in Figure 5(b).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{blowfish_algorithm.png}
\caption{(a) Blowfish algorithm (b) Function module $(F)$ (Manju, 2016)}
\end{figure}

The function divides 32-bit input into four bytes and uses them as indices into an S-array. The lookup results are XORed together to produce output. $P$ taken is an array of eighteen 32-bit integers. $S$ is a two-dimensional array of 32 bit and stored as $4 \times 256$.

\textit{Fragmentation}

The data fragmentation method gives more security to user’s data on the cloud. In this, user’s data is first divided into multiple parts based on size. It’s illustrated in Figure 6.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{data_partitioning.png}
\caption{Data Partitioning Architecture}
\end{figure}
Splitting the input file into segments is important for many reasons:

1) Distributing over multi-clouds is easier than storing a single file on the cloud.
2) Avoids cloud maximum size file problems.
3) Allows better utilization of bandwidth.
4) Facilitate load balancing.
5) Facilitate encryption and decryption process by reducing the size of computed files.

3.2 Proposed Hbdasec System Model

Cloud storage as a service is an emergent drift with attractive characteristics which are lacking in onpremise storage. Nonetheless, each organization is not competent of controlling huge secondary storage or constructs their confidential information centers due to the incurred expenditure of constructing and maintaining such infrastructure. Cloud storage elastic nature can be a vast provision to such organizations. Nevertheless, the failure of managing outsourced data is an intrinsic crisis. Even though the CSP is constrained by a Service Level Agreement to guarantee data security, clients cannot exclusively depend on these agreements. Moreover, dependence on a contractual commitment may fail to identify the malevolent activities of the service provider. So, in addition to the expediency supported by the cloud system, data security issues are also mandatory to be looked upon with a cloud storage service system. Our proposed cloud information storage scheme is illustrated in Figure 7.

The user can perform various operations like upload, download, delete and view the files.

![Figure 7. Proposed system model](image)

**HBDaSeC Components**

The HBDaSeC is the proposed secure secret, fragmented, and encrypted system consisting of 4 steps:

1) Registration
2) Encryption/Decryption
3) Fragmentation

Each time a new user wants to store data in the cloud, the authorized administrator performs the registration of the client. The blend of the encryption techniques helps to diminish the threat of information seepage in multiclouids even further, in the visage of inquiring or hacked CSPs. Furthermore, the availability of information located in the cloud is increased. The recovery procedure is analogous to the storage course involving authentication of Cloud Service Providers and reconstruction of encrypted files.

HBDaSeC involves:

1) **Secure Storage:** The subsequent stride is taken to amass the data given by a data storage provider securely using encryption and trusted authority:
   a) Define access rights for user’s w.r.t. the files based on role-based access control.
   b) Fragment the files into random numbers based on their type.
   Data files are encrypted using the Paillier algorithm at the client end and using the Blowfish algorithm at the server end and secret keys generated.
   a) Signatures for encrypted files are generated by SHA-1 etc.
   b) An encrypted chunk and its allied signature are sent to the independent CSPs.

2) **Secure Verification:** In the proposed HBDaSeC scheme, if a user desires to access the chunk stored by data storage provider, subsequent are the steps for uniqueness and access authentication:
   a) The request is sent by a client to the administrator for the desired segment.
   b) The administrator verifies the access privileges of the clients by role-based access control policy.
   c) The client is denied access on non-confirmation of access rights.

3) **Secure Auditing:** In the planned design given in Fig.4, after step 1 and 2, the client verifies data integrity by the subsequent steps:
   a) The client seeks the encrypted information from the cloud service provider.
   b) The cloud service provider sends the encrypted data to the client. The client generates the hash checksum using GtkHASH² GUI. The generated checksums are checked against the checksum stored in its database.
   c) The data integrity is confirmed if a match occurs.
   d) After confirmation of data integrity, the download operation is said to be complete when the split shares of data are regenerated and merged to get the original file.

**System architecture overview**

In the next section, we will provide an outline of a design in Figure 4 comprising the storage course from a solitary Healthcare unit (HU) including the data flow and security measures. The system will consist of major components: Cloud service providers, Data users, and Trusted Third Party Auditor. A novel style of security, as a proficient distributable cloud storage gateway acting as an interface between end users and private Cloud service providers, is provided. In this work, we have presented an Innovative mechanism for Secure Data Storage in Multiclouds for end users to:

1) Provide transparency to users to rely on different cloud storage providers.
2) To enforce double encryption, availability, and integrity.

Our projected system has the subsequent advantages:

1) Permit the consumer to amass data in a safe manner on commercial CSPs since the data is stored in an encrypted form.
2) Let the customer describe an access control policy.
3) The method allows the client to alter the access policy vigorously through Third Party Authority without necessarily decrypting the data.

The progress of a security system HBDaSeC: “A Novel Approach integrating Homomorphic and Blowfish encryption techniques for Secure Data Storage in Cloud” combining Homomorphic and asymmetric encryption is illustrated in Fig.8 below. At the client end, the client sends the fragmented and encrypted data with personalized keys on the cloud storage. The work of the server is to re-encrypt the encrypted data. The provision for storing the encrypted file with and without compression is provided at the server side. The client requests for data to the cloud storage server. The server sends the double encrypted data to the client. The client decrypts the obtained results and reconstructs all the partitions to attain the original file.
Our approach will convey the latest facet to cloud storage. It assures confidentiality, integrity to the data as in no stage information is exposed in plaintext. The data fragmentation method gives more security to user’s data on the cloud. In this, user’s data is first divided into multiple redundant partitions based on size. After partitioning, the user’s data are encrypted and scattered each share over the Internet to different Cloud service providers. The use of multiple cloud servers provides more security to user’s data. If an attacker gets any part of the file, it is impossible to get complete data because the data will be divided and stored on different servers. The following architectural diagram shows the concept of data partitioning and storage on different cloud servers. The deployment of the cloud has twofold benefits for such an approach:

1) The user can retrieve his file in case of temporarily/permanently unavailability of the provider.
2) Providers can’t access the file stored within them.

Only the authorized users are supposed to have full control of the overall security of the data. The clients have the right to access their own specific data and can’t overlook other user’s data. The cloud service providers are not provided any knowledge about the stored information as the data is stored in encrypted form.

Role of Trusted Third Party Authority at the server side
Third party authority has been assigned the authority for changing the attributes is used to set/unset certain attributes to a file to secure accidental deletion or modification of important files and folders. The file can be made immutable i.e. no renaming, no execution, no symbolic link creation, no writable is allowed. We can secure the entire directory and its file content using different switches like +r with +i flag along with the full path of the folder. To synchronously update changes on the disk on any file modification, the ‘s’ attribute set can be used.

**HBDaSeC phases**

The scheme can be achieved by creating a safe and completely automated information storage and transfer protocol amid CSPs and users having access to numerous storage volumes on the cloud to accumulate his information. The protocol can be implemented in two phases:

1) Upload phase
2) Retrieve phase

The send phase occurs when the file is being saved or restructured in the public cloud storage and retrieve phase occurs when the file is accessed or downloaded from the cloud infrastructure. The file can be stored among available virtual volumes V1, V2, ..., Vn with each having certain storage space.

Upload phase: A file to be uploaded is split into numerous smaller chunks to be stored in the existing volumes such that reconstruction of the initial file becomes difficult.

Information associated is:

1) Index of the chunk in the original file which provides the precise position in the file.
2) Virtual volumes(s) where the chunk is stored.
3) Hash sum of the chunk which is used for integrity check.

---

The chunks are chosen arbitrarily for uploading and consequently, the chances of an attack are minimized.

Retrieving phase: The chunks are randomly selected in the retrieved phase. When all the chunks are obtained, an integrity check is carried out and using the information in the Index field, the chunks are precisely arranged to generate the original file.

Challenges involved:
1) How to split information into segments that carry the least information with respect to the original file.
2) Storing these portions on public cloud storage volumes in view to diminish the probability of illegal reconstruction.

Encryption and trust-based solution: How effectively the cloud service provider confidentially to cloud service user’s data is an important challenge.

The problem associated with leaving the keys with the CSPs or third party can be solved by making cloud service users hold the keys and assign keys to CSPs based on the trust mechanism. CSP provides data storage and data computation services which include searches, addition, modifications, insertions, and deletions.

Three types of operations conducted by CSUs are:
1) Encrypting its own data and send encrypted data to CSP.
2) CSP trustworthiness or availability is evaluated through delays.
3) For performing any data computation, sp will need encryption keys and these keys will be assigned by CSU according to the trust values of CSP.

Figure 9. HBDuSeC Upload Phase

Figure 10. Integrity check phenomenon
Why three parameters are chosen?

We have chosen three cloud security aspects abbreviated as CIA i.e. Confidentiality, Integrity and Availability out of six security principles. Their significance is shown in the Fig.11 below:

![Cloud security aspects](image)

Confidentiality: Confidentiality refers to the prevention of unauthorized access of the data and making sure that only the user who has the permission can access the data. This way the CSP can guarantee the user that his data doesn’t get into wrong hands and also increases the users trust in cloud computing and helps it grow faster. Data confidentiality can be ensured through better encryption technique.

Integrity: When the integrity of outsourced data is to be checked, it is not practicable to download all records from the distant server and confirm as it incurs huge communicne and computational cost. To evade this great cost, generally, the integrity schemes execute “blockless verification” which allows downloading only metadata and not the real information from the cloud; which is generated by the CSP and verified at the client end. Efficiency is a major issue in data integrity schemes is calculated as computation, communication and storage expenses incurred.

Data integrity at the client side is done using a GUI called GtkHash. The combination of hashing algorithms are used to verify data integrity like MD5, SHA1, and SHA256 of the file before sending to the server are generated. After the file is being downloaded, the checksums are recomputed. The integrity of data is ensured by the validation of checksums performed at the metadata inserted at the end of the files at the client end. Metadata was encrypted using double encryption and SHA1, SHA256 and MD5 algorithms were used hash of the original file. The comparison between the previously stored and newly generated checksums is made. In case of any breach, the checksums will not match lest a check-mark is associated with the three specific fields.

Availability: Availability is ensured by storing the files both on the clients and server.

Delay incurred: There is always a difference in the traffic movement, at specific time the traffic flow is more whereas at some stage the traffic movement is less. Depending on current buffer usage at server and the data extents to be delivered by client node, the required memory banks can be activated, the memory banks are supplied power which leads to reduced congestion. An optimized approach to minimize delay is to distribute the memory into a number of chunks and then power the memory essential for data transmission. Our proposal will greatly reduce delay.

3.3 Implementation and Analysis

Implementation

Oracle Virtual Box is a free and open source hypervisor currently developed by Oracle corporation which is easily available and executes as a Virtual Machine. It can be associated with popular operating systems such as Windows XP and Vista, Macintosh and Linux, while additionally associate a large number of visitor operating systems such as Red Hat, Fedora, and Ubuntu, etc. Ubuntu is a free and opensource operating system and Linux distribution based on Debian. For security, Fog can be used to offer an effortless access and service attuned platform. Fog is Ruby-based cloud library which allows setting up a license file to join numerous service providers and function equally for generating on-demand service as required.

Three scenarios are considered in HBDaSeC framework:

1) Storage of encrypted data over the cloud using Paillier Homomorphic technique.
2) Store encrypted information over the cloud storage with Blowfish encryption and compression.
3) Store encrypted data over the cloud storage with Blowfish Encryption and without compression.
We implemented an archetype of HBDaSeC and evaluated the results of HBDaSeC based on time consumption parameters using Ubuntu 16.04 on Oracle VM VirtualBox 5.1.20. Performance evaluation is done along with the experimental setup. Python scripting language 2.7.3 is used for encryption algorithms.

**Analysis**

The implemented protocol is said to proficiently provide In-tegrity guarantee to users considering Cloud service providers to be loyal against malevolent or unauthorized users. The integrity, availability and confidentiality security constraints against internal and external attacks were validated. NetStress tool was employed to check the attacks.

Table 2 illustrates the client side encryption done using Paillier algorithm on different file types. We have chosen image, text, document, video, and pdf file types in bytes.

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Size (Bytes)</th>
<th>Encrypted File Size (Bytes)</th>
<th>Decrypted File Size (Bytes)</th>
<th>Encryption Time (S) Paillier</th>
<th>Decryption Time (S) Paillier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>7180</td>
<td>7475</td>
<td>7180</td>
<td>0.035</td>
<td>0.59</td>
</tr>
<tr>
<td>Image</td>
<td>23596</td>
<td>23596</td>
<td>23596</td>
<td>0.045</td>
<td>0.622</td>
</tr>
<tr>
<td>Pdf</td>
<td>483694</td>
<td>483987</td>
<td>483694</td>
<td>0.29</td>
<td>0.027</td>
</tr>
<tr>
<td>Video</td>
<td>4254956</td>
<td>4255251</td>
<td>4254956</td>
<td>0.155</td>
<td>0.154</td>
</tr>
<tr>
<td>Docx</td>
<td>23679</td>
<td>23971</td>
<td>23679</td>
<td>0.013</td>
<td>0.009</td>
</tr>
</tbody>
</table>

Table III presents the server side encryption employing Blowfish algorithm.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Encrypted File Size (Bytes) Paillier</th>
<th>Encrypted File Size Without Compression Blowfish</th>
<th>Encrypted File Size With Compression Blowfish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>7475</td>
<td>7538</td>
<td>7462</td>
</tr>
<tr>
<td>Image</td>
<td>23891</td>
<td>23954</td>
<td>23886</td>
</tr>
<tr>
<td>Pdf</td>
<td>483987</td>
<td>484059</td>
<td>484110</td>
</tr>
<tr>
<td>Video</td>
<td>4255251</td>
<td>4255314</td>
<td>4256534</td>
</tr>
<tr>
<td>Docx</td>
<td>23971</td>
<td>24034</td>
<td>23958</td>
</tr>
</tbody>
</table>

Table IV depicts the encryption time computation done on files with and without compression using Blowfish technique at the server side.

<table>
<thead>
<tr>
<th>Encryption Time (S) Without Compression Blowfish</th>
<th>Encryption Time (S) With Compression Blowfish</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.751</td>
<td>7.85</td>
</tr>
<tr>
<td>419.65</td>
<td>74.88</td>
</tr>
<tr>
<td>69.185</td>
<td>34.251</td>
</tr>
<tr>
<td>66.985</td>
<td>55.059</td>
</tr>
<tr>
<td>43.231</td>
<td>41.474</td>
</tr>
</tbody>
</table>

Table V shows Encryption time using both Paillier and Blowfish with and without compression.

<table>
<thead>
<tr>
<th>Encryption Time (S) Paillier &amp; Blowfish Without Compression</th>
<th>Encryption Time (S) Paillier &amp; Blowfish With Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>613.34</td>
<td>214.052</td>
</tr>
</tbody>
</table>
Table VI and Table VII represent transfer rate and delay with memory bank enable and disable for varied number of files.

Table 6  
**Transfer rate (memory bank enable and disable)**

<table>
<thead>
<tr>
<th>No. of Files</th>
<th>Transfer Rate (Kbps) (Memory Bank Disable)</th>
<th>Transfer Rate (Kbps) (Memory Bank Enable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5.907</td>
<td>6.2505</td>
</tr>
<tr>
<td>50</td>
<td>4.6915</td>
<td>4.4635</td>
</tr>
<tr>
<td>75</td>
<td>3.025</td>
<td>3.5085</td>
</tr>
<tr>
<td>100</td>
<td>2.524</td>
<td>3.0725</td>
</tr>
</tbody>
</table>

Table 7  
**Delay (memory bank enable and disable)**

<table>
<thead>
<tr>
<th>No. of Files</th>
<th>Delay (ms) (Memory Bank Disable)</th>
<th>Delay (ms) (Memory Bank Enable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0.02851</td>
<td>0.001592</td>
</tr>
<tr>
<td>50</td>
<td>0.01954</td>
<td>0.001635</td>
</tr>
<tr>
<td>75</td>
<td>0.0344</td>
<td>0.001492</td>
</tr>
<tr>
<td>100</td>
<td>0.04386</td>
<td>0.001537</td>
</tr>
</tbody>
</table>

Figure 12 – Figure 16 below shows the graph for encrypted text files in bytes using Paillier, and Blowfish (with and without compression). The figure shows the graph for encrypted Image files using Paillier, and Blowfish (with and without compression). The figure shows the graph for encrypted docx files using Paillier, and Blowfish (with and without compression). It is clear from Table I that there is a significant saving in storage space for text, image and docx files. Compression of data before transmitting it to the client reduced costs and preserved data integrity.

[Figure 12: Encrypted text file size using Paillier and Blowfish (with and without compression)]

[Figure 13: Encrypted image file size using Paillier and Blowfish (with and without compression)]
Figure 14. Encrypted docx file size using Paillier and Blowfish (with and without compression)

Figure 15. Encrypted all files using Blowfish (with and without compression)

Figure 16. Encrypted text file size using Paillier and Blowfish (with and without compression)

Figure 17 and Figure 18 represent transfer rate and delay with memory bank enable and disable for varied number of files.

Figure 17. Transfer rate (memory bank enable and disable)
4 Conclusion

Cloud computing is a concept shifting in the approach how computing resources are deployed and purchased. Even though the cloud has a capable, elastic, and consistent design, several security concerns restrain customers to completely accept this novel technology and move from traditional computing to cloud computing. In the article, we aspire to present a form of a novel architectural model for offering protection to numerous cloud service providers with the intention to devise and extend security means for cloud computing. In this work, we presented a two-tier architecture for security in multico; one at the client side, and other at the server side. The article presented a security domination outline for multi-clouds and supports security needs like Confidentiality, Integrity, Availability, Authorization, and Non-repudiation for cloud storage. Through this document we have anticipated, HBDaSeC, a securecomputing protocol to ease the challenges of enforcing the protection of data for information security in the cloud. To the paramount of our acquaintance, it is the foremost effort that together utilizes two encryption techniques for data storage security and computation in the cloud. Our execution and assessment by numerous experiments suggest the convenient viability and ease of use of the system. By the extensive security analysis and performance simulation in our developed SecHDFS prototype, it is apparent that our procedure is effectual and competent for achieving a secure cloud computing. In addition, we intend to execute them in the real cloud platform such as NetCloud. Also, storage capacity improvements in case of pdf and video files are considered as future work.

Conflict of interest statement
The authors declared that they have no competing interest.

Statement of authorship
The authors have a responsibility for the conception and design of the study. The authors have approved the final article.

Acknowledgments
We thank the editor of IRJMIS for their valuable time, support and advice.

References


