Adaptive Light Weight Encryption Algorithm for Securing Multi-Cloud Storage

Viswanath Gudditti[a], and P.Venkata Krishna[b]
[a]
Department of Computer Science, BharathiarUniversity, Coimbatore, Tamil Nadu,India,
[b]Departmentof ComputerScience,Sri PadmavatiMahilaVisvavidyalayam,Tirupati, Andhra Pradesh,India,

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: In several organisations the use of cloud computing has rapidly inflated. In terms of low cost and accessibility to information, cloud computing offers several advantages. Next generation design of the IT Company was described as Cloud Computing. It moves software and expertise to centralised huge data centres where management of data and services cannot be entirely reliable. Its cost-effectiveness is the main advantage of this technology. The disadvantage is significant, however, and the security problems associated with information stored in clouds are also disadvantageous. We first identified the potential security problems and the difficulties in fully expanding cloud information and restore availability. Previous efforts have ensured the integrity of distant knowledge, but lack public verification and access to knowledge. This ensures the integrity of knowledge and ensures public verification and accessibility. We have created an elegant verification with access rights to ensure the seamless integration of these two key options in our protocol style. This paper attempts to achieve public verification by changing the traditional Merkle Hash Tree into newly materialised multi-cloud or inter-cloud. It has been shown that we can ensure the availability of knowledge with the DepSky System model for multi-clouds.

Keywords: Cloud ,Merkle hash tree, Security, Storage, Availability, Optimisation.

1. Introduction

The dawn of the digital era has brought in the viability in sharing and securing the data. However there are some major concerns as well. With the introduction of cloud services opportunity to steal data has reached new heights and so has the concerns. A major disadvantage with security issues relating to the data stored in the cloud follows. Potential security problems and problems have been identified with full expansion of cloud data and service availability. While previous work ensures the integrity of remote data, public verification and data access are lacking. Therefore, the objective is to ensure public verification and accessibility of data integrity. The seamless integration of these two key features is proposed through an elegant verification of access rights.

One more protocol is introduced to enhance the data confidentiality. A protocol that takes security to every new level is designed as part of the four layered security protocol with an encryption algorithm. Every layer adds to the time needed to break the cipher to get the genuine information. This algorithm is lightweight and promises to provide the security needed for securing the sensitive data. It is a perfect replacement for most of the standard algorithms as it brings a set of revolutionary concepts to the desk.

The use of computer resources (hardware and software) supplied through a network is cloud computing. It offers distant services. The basic cloud architecture consists of three layers: service infrastructure, service platform and service software. Service infrastructure (SaaS).As storage space is outsourced Service Infrastructure (IaaS) it is crucial for data protection. Networking, data storage and computer services are available for the user. In other words, computer infrastructure is delivered as a service. The Amazon Web Service (AWS) [10], Go-Grid, Flexi Scale, and so on are examples of IaaS. Service Infrastructure (IaaS) is the lower layer from which the higher layers are abstracted. IaaS can be described as, “a model for enabling 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 providerinteraction”.

There are few security concerns due to outsourcing of data by the enterprises on to the public Virtualization of the cloud infrastructure application is comprehensive and gives public cloud service customers or tenants a unique security concern. Virtualization changes the relationship between the operating system and the hardware underlying it. The possibility of compromising the software or "hypervisor" includes specific concerns. Only if proper defensive implementations are in place will cloud security architecture be effective. The cloud provider may neglect the stored data to save money and storage space, or delete the rarely accessed files that create serious issues for users. There are many schemes proposed to solve this problem under different systems and models. All schemes fall into two broad categories, taking account of the verifier's role: private verification and public verification. Since information and data are shared with third parties, cloud users want to avoid the unreliable cloud provider. Private and material data should be protected from attackers or insiders, such as credit card details or patient health records[17]. Furthermore, the potential for migration from a cloud to a multi-cloud environment is explored and safety issues in single and multi-cloud computing are investigated.

Since the Internet provides important communication tools for tens of millions of people and is being increasingly used as a trade tool, security is an enormous problem to address. With the launch of the cloud, the online services got a boost over the potential customers and the ability to diversify services. With this the cloud becomes the central place for hackers to whip the sensitive information. Hence, the organizations do not trust the cloud as the safe place to store confidential data . Various cloud services are managed by different people and hence there is further decline in the trust . Hence, the need to encrypt the data before storing it on the cloud is on rise. Issues concerning to performance have been discovered with the help of evaluation of infrastructure and architecture and a set of proposed algorithms . We introduce a protocol that provides the needed security without compromising the performance. In this work, we assume the first layer of security is provided by the cloud service provider by means of access restriction, intrusion detection and other set of security tools and countermeasures necessary for assuring the safety of the clients. The users or clients are provided with login details to secure the data from unauthorized persons. As the data is stored at a single location the intruder, if successful, may run away with confidential information. The best technique to secure the data would be to split the data into fragments and store the data into multiple cloudstorages.

An organized data that is divided into fragments would not be the safest possible way to segregate the data. Hence, we introduce a protocol that makes use of a random formula to scatter the data into a number of files. Once the data is stolen, itcan be rearranged into a file and the entire file may be fed to the text processing tools to shuffle it until it makes proper sense. To overcome this we introduce a concept called garbage insertion as a second layer. The algorithm makes sure that even after extensive processing of randomized text the intruder will not get the exact information by attaching fake or garbage data to the real data thus making the boundaries between real and fake data invisible. The deciphering module holds all the manipulations that aretobedonefordiggingoutthereal data. Now withfakeandrealdatamixedup, the task of intruder is infinitely hard. To take the real information further apart from the intruder we make use of encryption. The encrypted text makes no senseto anyhumans. The cipher generation is based on the key provided by the user, making it difficultfortheintrudertodecipheritevenifhegetsavisual.

This proposed security protocol provides a four layered security to embed trust into cloud services and increases the safety of the user's data or information. With all these features the algorithm eliminates the need for key sharing. This algorithm makes use of a concept called intelligent scan technique to get the key out of the cipher. The key is not included into the cipher but the arrangement of cipherholds a computational mystery that can be solved by the decipheringmodule.

2. Literature Survey

NIST(http://www.nist.gov/itl/cloud/)expresscloudcomputingas"amodelforenablingconvenient,on-demand network access to a shared pool of configurablecomputingresources(e.g.networks,servers,storage,applications, and services) that can be rapidly provisionedand released with minimal management effort or serviceproviderinteraction". [16]

Kamara and Lauter both contributed (2010) There are two types of cloud infrastructure, namely public and private clouds. The easy-to-use infrastructure is hosted in the public cloud. Information accessed and controlled by trusted Users is available in the secure and secure non-public cloud, whereas infrastructure managed and controlled by the cloud service provider is available in the very public cloud. This information, in particular, is not managed by the user and is managed and shared with unsafe and untrustworthy servers (Kamara et al. 2010). [7]

The various edges of the public cloud are linked by reliability and availability, as is the low price (Kamara et al. 2010). However, there are some issues with public cloud computing, particularly issues with information integrity and confidentiality. Every customer is concerned about the safety of sensitive data such as medical records or financial information (Kamara et al. 2010). [7]

Cloud service providers must ensure the security of the information their customers provide and can be held liable if security risks affect their customers' service infrastructure. A cloud provider provides a wide range of services to its customers, including quick access to their information from anywhere, quantification, pay-per-use, information storage and recovery, hacker protection, on-demand security checks, and the use of network and infrastructure services (Subashini and Kavitha 2011).

Transferring to a larger data centre presents a number of security challenges (Wang et al., 2010), including virtualization vulnerabilities, accessibility vulnerabilities, privacy and management issues related to third-party information, integrity, confidentiality, and information loss or larceny. Coding techniques and secure protocols are insufficient to protect cloud data transmission. Cloud intrusion information obtained by hackers and cyber criminals via the Internet should also be addressed, and cloud entry should be secure and personal to buyers (Subashini and Kavitha 2011). [11]

Balakrishna S, Saranya G, Shobana S, and Karthikeyan S (Wang et al., 2009a) are considering the task of allowing a 3rd party auditor (TPA) to verify the integrity of the information contained within the cloud on behalf of the cloud consumers to ensure the correctness of the information. The use of public keys is frequently accomplished through homomorphic criticism combined with random masking of privacy protection. For batch

auditing, the additive combination signature technique is used. Batch auditing reduces computation overhead. Because cloud knowledge is used by so many industries, information modification is unavoidable. It contributes by supplying an AN External Auditor Privacy Protection Protocol for creating audit checks on outsourced knowledge without learning its contents. They also provide dynamic knowledge-based operations and target error correction. My solo reed technique (Wang et al. 2009a). [13]

The third-party auditor instituted the next step in auditing in order to ensure public verifiability (TPA). TPA can audit knowledge storage in the cloud using public verification without difficulty, feasibility, or resources for users. During this model, a stimulating question is whether we can build a theme to ensure the public accuracy of dynamic knowledge and storage. Qian Wang, Cong Wang, Jin Li, KuiRen, or Wenjing Lou identify the problems and potential safety issues of direct extensions with completely dynamic knowledge updates from previous works and show a way of building a chic, verifiable theme for seamless integration of those two key protocol options into their style. This is a new paradigm concerning a number of new security challenges that are not well understood. This paper investigates the issue of ensuring the integrity of cloud computing information storage. To achieve economic knowledge dynamics, the proof of retrievability model is improved by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Intensive analyses of safety and performance show that the proposed subject is very cost-effective and undeniably safe (Wang et al. 2009b). [14]

Merkle's signature theme converts every single signature theme into a multiple theme. This extension consists of complete binary trees, most of which are Merkle trees. Merkle trees have found a variety of applications in the construction of theoretical science as a result of the terrible foundation of this thesis. The most important role they play is to verify leaf prices in relation to the root value recognised in the public, as well as to authenticate individual leaf knowledge. This knowledge has one node price at each height, wherever those nodes are linked by the siblings of the nodes along the path. Merkle Tree Crossing is the task of discovering economic algorithms for sequential leaves in order to generate authentication knowledge. They're as beautiful as Merkle trees, but that's to be expected. One reason is that recognised cross-cutting techniques necessitate massive amounts of computation or storage. This implies that only the smallest trees will be extensively used. Merkle trees, on the other hand, could become compelling with many economic transverse techniques{ once again once again|again|again|again,}, The size of the general public key is the Lamport–Diffie theme's best disadvantage day. All inspectors would like to have a genuine copy of this public key in their nursing so that the authenticity of any signature can be confirmed. According to Ralph Merkle, the foundation is that in our digital signature topic, the public key and also the leaves are the only non-public keys. It is divided into three stages: key generation, output, and verification (Ederov 2007). [5]

[5] They argue that, in addition to public verification, customers can delegate integrity check tasks to TPA because they are either untrustworthy or unwilling to perform the required computer resources activity continuous verification. On untrusted nursing servers, the proposed protocol has been shown to be safe against Associate. Furthermore, it is private against third-party verifiers. [6]

As a result, it proposes a remote knowledge integrity testing protocol for cloud storage that can be regarded as a collaborator in Sebe' et al's protocol nursing adaptability. It proposes a proto-gap that supports the dynamic of information as well as public verification and privacy against third-party checks, while not requiring the use of a third-party auditor at the same time. They provide the projected protocol security analysis, demonstrating that it is secure against an untrustworthy server and private against third-party verifiers (JaisonVimalraj and Manoj 2012). [6]

In addition, Tribhu-wan et al. (2010) planned another thesis in 2010 to ensure the accuracy of user knowledge within the cloud. Tribhuwan et al. (2010) propose a method for verifying knowledge using a homomorphic token and distributed erasure-coded knowledge. They claim that their theme is a combination of storage correctness insurance and the location of a misbehaving server (s). They accept delete code in the preparation for file distribution to provide all redundancies and ensure knowledge reliability. When compared to older file distribution techniques, this structure significantly reduces communication and overhead storage. Then another theme was mentioned, because the new two-way handclasp token management support was introduced to address cloud security information challenges. [12]

Change from one cloud to several clouds. Singh et al. (2011) have been researching efficient storage value in multi-cloud environments. They propose a model for providing customers with knowledge accessibility as well as secure storage that includes a low-cost distribution of information among market services. During this work, we've discovered that relying on a single Service Supplier to read from a customer's outsourced knowledge isn't very promising. Furthermore, greater privacy and accessibility of knowledge guarantees are achieved by dividing the user knowledge block into knowledge items and distributing them among the market Service providers in the simplest way possible, to ensure that only a threshold of service providers participate in the fortunate recovery of the total knowledge package. [9]

This new approach to cloud storage in concrete benefits the area unit by utilising fully versatile web storage. However, from a security standpoint, this poses new and significant security threats. Problems such as information loss or leakage, accessibility, trust, integrity, and malicious insiders are among the most well-documented concerns when deciding to manage cloud data. Thus, Andre (2010) projected the DepSky Systems

Model thesis on the accessibility and confidentiality of cloud victimisation information storage in December 2010. This model addresses the most significant issues with the relevance of cloud storage security that have been raised. He divides his addition into two sections. To begin, he employs the DepSky system because it enhances by including erasure codes in a library variety. Second, he examined the accessibility of four business cloud providers using the faller ion of DepSky. Later, an AN analysis was performed to assess and correlate factors such as latency, value, and geographical variations. André et al., 2010 [2]

Meanwhile, B et al. (2011) contribute DEPSKY, a system that improves knowledge supply, integrity, and confidence through secret writing, encryption, and replication in different clouds. In other words, DEP-SKY can be a computer storage cloud that users can access via a variety of cloud operations. [3]

3. Proposed solution for publicverifiability

The key concerns in the current cloud computing scenario are data or information security that is stored in clouds. The verification of data integrity on untrusted servers is one of the main storage concerns. Since private data is shared with a third party, there is always a risk that the data is lost or defective. Cloud services also delete unused or least accessed files deliberately to save money and cloud space.To maintain a reputation, cloud service providers have also been studied in hiding data errors, file handling, malfunctioning, and network or outside intrusion attacks . Many schemes are planned to solve these problems in different systems and models. Some of these obstacles can be overcome by various design solutions. In other words, there is no single design solution that addresses all these problems simultaneously. The cloud security survey is intended to identify faults and carries out a detailed study of the subset concerned.Once the fault is identified, the problem statement for the researchers is refined. Since these vulnerabilities are highlighted, certain protocols are required to secure information for users and clients. A new algorithm has been proposed for wireless communication with an aim of increased performance . Here data is divided into blocks where first block of data is encrypted andrest is properly arranged. Here the process starts by encrypting first blockof data using algorithms like AES. Then perform XOR on all data with text of first block these assumptions are used to define the model definitions as shown below, followed by details.

The solution proposed focuses on the two main problems of simultaneous verification and data availability. The proposed solution is based on certain assumptions. The hypothesis for developing the system model proposed is as follows:

• No Third party auditors (TPA) are not available for customer audit. The customer or other users who want to use the file must instead challenge verification.

Since no TPA is involved, all checks on the server side are performed. This is why the cloud service provider is entrusted with the assumption.

The file owner shares a general public key by his/her handy ways of viewing the file or the entire file challenge (email, SMS, etc.).

This section will also include different notes reflecting the results produced after algorithms or methods or functions are executed, in each stage of implementation. To implement the functionalities to produce results at any stage of the system model, a right-hand (arrow) technique or algorithm is used. The left ratings show the results of the respective system model step.

- (F')←Encrypt (F). This algorithm runs on the side of the server. The AES Symmetric key encryption algorithm is used for encryption. A clear format file is required to be uploaded as an input in the cloud.
- (pk, sk)←KeyGen().It's running on the server side. We generate the public and private key(pk) in this algorithm (sk). These keys are used to allow users to operate on a file.
- (ϕ)←FileSplit(F'). This function divides the encrypted file into data blocks (chunks) of the same size. The number of chunks to be made depends on the file size that is encrypted. Where¨denotes the set of chunks produced. These chunks are saved to avoid confusion in the respective directory of the same name as the file.
- (Ω)←HashGen(ϕ). The hash value for each of the chunk in ϕ isgenerated. These hashes are represented using Ω. Thus Ω is a set of hashes of all chunks. The algorithm implemented to generate the hashes in the proposed system is MD5 Checksum. Any similar algorithm can be used.
- R (Ω) ← (Ω) a hash tree is generated by combining the adjacent two nodes with the chunks obtained above to form their parent nodes. This is done repetitively until a single parent node i.e. the root node represented by R (Ω) is obtained.
- (Ƶ)←ZipDir(ϕ). The directory of the data blocks is zip to the server to maintain the efficiency of the space.
- {TRUE, FALSE}←Verify (Filename). This function returns the TRUE or FALSE value to challenge the CSP for the integrity of certain files. The file is intact when TRUE is returned and FALSE is not returned.
- (F)←ExecuteOp(Filename, K, OP_Mode). This function accepts a filename for a certain operation. The parameter OP Mode determines the type of operation to be carried out (verify, view, update, delete). Depending on the operating mode, the corresponding key K is requested.

4. Proposed construction for Public verifiability and data access

At each stage of implementation in the previous section, we discussed the outline and results. It also describes the effectiveness and efficiency arguments or parameters with which techniques or algorithms are executed. The

ratings show clearly what is supplied as input and what is produced as output. The previous section also describes the system model technique for implementing what functionality. Now we can see how the system model performs with a basic understanding of the ratings. The system proposed focuses on data security issues in the cloud. As information and data are shared with third parties, cloud computing users want to avoid an untrusted cloud provider. It is critical to protect the privacy and sensitivity of the patient from attackers or malicious insiders such as credit card information or medical records. This results in steps to verify the accuracy of the entered data or information. The customer can also easily access, update, and download cloud data. The system proposed includes a data integrity encryption mechanism. This is because the client uploads the file (F) clearly. The raw file is encrypted to prevent such errors. The encrypted file is marked with the letter F'.

The next problem is that if an encrypted file is stored in a single location and is accessible as a whole, intruders can easily tamper with it. As a result, the files are divided into chunks known as data blocks (ϕ), to avoid confusion; they are kept in their designated locations. The size of the file uploaded by clients determines the number of chunks into which it is divided. Verification is handled using the hashing technique. To begin, each data block (Ω), is given a hash value, which is then used to build a hash tree. According to the analogy, every tree leaf node represents the respective blocks' hash code. Furthermore, the hash values of each child node are used to calculate each parent node. Consider the diagram below, which shows how the hash tree is built. The hash values of the chunks of the encoded file are its nodes. Furthermore, the hash of the two binary nodes is the next parent of the nodes. This is repeated until the root node is reached. The root node hash value is referred to as the whole R(Ω) once it reaches the root node file hash value.

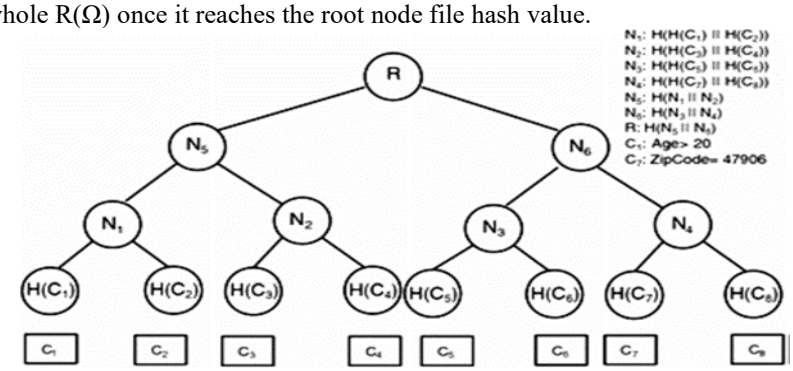

Fig 1 Hash Tree

The uploaded file chunks in fig 5.1 are C1, C2, C3,...,C8. H(C1), H(C2), H(C3),..., H(C4), H(C5), H(C6), H(C7), H(C8), H(C9), H(C10) (C8). The next step is N1, N2, N3 and N4. N1 is calculated using its children's hash values, H (C1) and H (C2) is calculated equally, and N2, N3, N4, N5, and N6 are also calculated. In conclusion, the hash value from the N5 and N6 nodes is the root hash value (R). R is the hash of the entire root file. If there is any malfunction or manipulation in one of the chunks in this kind of tree structure, the corresponding hash value will change. This change in the hash value affects the hash values across the entire tree structure, including the root hash value. There was a glitch, as shown by the root hash change. The next step in the proposed solution is public and private key generation (pk, sk). This is done to ensure that the use of resources is legal. This means that not every cloud user has access to all the resources of the cloud. The owner of the file will grant access to its file. For each file owner, the public and private keys and metadata and hash must be stored in the database. The owner must enter the private key to view, update or delete. Only if the entered private key is valid, can these operations be performed. Similarly, by giving them his public key, the owner can control who has access to his files. That is, if you have access to the public key, users can only view the file. The chart below shows the general structure or execution flow for uploading a file to the cloud (Fig.2).

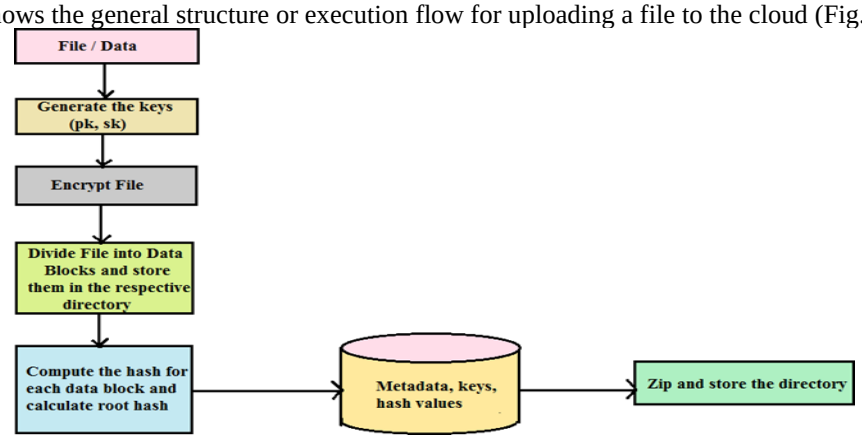

Fig 2 Schematic view of the execution of the proposed system

The solution proposed focuses on data integrity and confidentiality, which enables the user or owner to duplicate his/her information by challenging the cloud services provider. This means that the customer can ask the cloud provider to check its files to be correct. This check is used to see if a data block has been tampered with. The hash code of the respective chunk will be changed if the data block is affected by any malfunction, distortion, or deletion performed by the cloud provider. In this way, the integrity of the stored file/data is maintainedSimilarly, the user must provide a private key to update/modify an existing file. Only after inserting a private key can the user download the file. On the other hand, if any other user (i.e. someone who is not owner of the file) wants to retrieve the file, the public key is needed. The file can only be viewed if the user provides the public key.

The recovery procedure is as follows. If you want to view or update a file, the private/public key is requested. Once the relevant key is entered, the directory is searched in the cloud with the correct filename. The data blocks in the directory will be merged and decrypted when the directory is found. The requested file of the user is then sent to the decrypted file.

5. Implementation of proposed securityProtocol

Implementation of proposed protocol is built over four levels to provide ample security. Let's have a step-by-step implementation of the protocol by taking top-down approach as shown in fig3.

Level 1-Cloud Service Provider

At the top level basic set of security must be provided by the cloud service provider. The security modules may include the ways of restricting access to the personnel/confidential information and employing modules that alert whenever there isan attempt to infiltrate the remote storage. The cloud service provider must gain trustby handing over SLAs to the clients if the need arises. Once the client hands over the data, it must be provided with all the security agreed on over the SLAs.

Level 2-Fragmented Text

At the top level the security is provided by the cloud serviceprovider to make sure that the clients trust the cloud and stealing the information is no easy task. The intruder hasto surpass all these obstacles to get confidential data. The level two of the protocol is active in the area where the intruder successfully breaches all the walls. The basic threat of data theft can be eliminated if all the data is not present in same place. If we divide the data into fragments and store it using multiple usernames within single cloud storage or by having multiple usernames among multiple cloud storages. With this the first task of the intruder would be discover multiple user accounts and the breach every single account to collect the complete information. The security is increased in proportional with that of the number of user accounts. The proposed algorithm that is a part of this protocol provides this functionalityby dividing the data into multiple files. As the data is fragmented into files nowthe task of intruder is to combine all the data once he gets all the files that carry the information. Hence to make sure the intruder does not get the information the data is randomly divided and randomly arranged into files. Hence when the intruder gets all the files he is still far away from the authenticinformation.

Randomizing the information gives another edge against theintruder as there is no easyway to find the pattern of randomization. The task of intruder is now in a deadlock until he finds a way out. Hence this second leveladds considerable security to the information. This is completed in no time as the operation is very basic and involves no processing ofdata.

Garbage Insertion

As the data is scattered over a set of files and cloudsthetask of getting to data is tricky and time consuming. But if the intruder possesses the high-end processing capability then he might end up with the confidential information. Hence we introducethe concept of Garbage Insertion into this protocol. The taskat this level is to make sure that the information is perfectly mixed with fake information. The objective is to eliminate the boundary between the real and fake data to make sure the task of filtering fake data out of file is eliminated. As the fake and real data appear alike, it's hard to mark the real part of the file. The algorithm generates random fake data to be embedded with real data which is scatteredinto multiple files along with real data. There will be no static patternof data distribution which makes prediction of possible location of real data a work of fiction. By employing the concept of Garbage Insertion the intruder is left with no choice but to quit. The tools are being constantly upgraded to get the information out of shuffled data files. Hence the concept of Garbage insertion is used to eliminate the crypt-analytical tools that deal with text processing. The garbage information may be formatted to appear like real information with fake details. With this we can engage the intruder into an infinite loop of unwanted data.

Encryption

As the information in fragmented, shuffled and mixed with fake data but still the data isreadable and makes sense to anyone who reads. So to eliminate the characters that make sense and increase the level of security we employ the idea of encryption. The encryption algorithm employed is none of the standard algorithm out in the market. This particular algorithm has evolved after months of research to invent a way of cipher generation that is fast and generates a high standard encrypted text. This algorithm follows the basic rule of converting text into something that does not make any sense based on the key provided. The algorithm tries to eliminate the threat of intruder noting down the key by getting a visual over the keyboard orthe screenby not using the key directly for

cipher generation. The algorithm is designed to generate a key based on the key submitted by the user. The decryption module is in sync with that of the encryption to be able to decipher the text. With these layers of security thetask of retrieving the real information from encrypted, faked, and fragmented data is rarely possible. Hence this protocol provides a four layered security structure for securing the data at different levels as explained instable 1 and table 2.

No key sharing: As the need to share the key over the network or exchanging the key on a paper are defenseless against the intruders nagging around, the best way to take out this risk factor is to eliminate the need of key sharing. The decryption module can be modified to be able to detect the key all by itself. The effect on the performance is negligible and reduces the amount of input to be provided.

Infinite key length: As the amount of security provided by any algorithm depends on the key length, this algorithm has no limit to its key length. The max key length can be specified and the key used may be of the specified length or smaller. The smaller key length does not create any performance issue, neither the extra bits in the key exceeding the limit are truncated, in the later case.

Algorithm

Table 1 Algorithm for Encrypter

Step 1: Start.
Step 2: Prompt for file location.
Step 3: Prompt for key.
Step 4: Generate actual key based on the key provided by the user. Step 5: Carry out character by character cipher generation.
Step 6: Generate fake and garbage data to insert into output file.
Step 7: take the output file for fragmenting the information into multiple files.
Step 8: Stop.

Table 2 Algorithm for Decrypter

Step 1: Start.
Step 2: Prompt for location of all files.
Step 3: Prompt for first sequence file.
Step 4: Combine all the fragmented information from all file as indicated by the
Sequence information in first file and store it in a single file.
Step 5: Perform intelligent scan to obtain garbage code.
Step 6: Based on garbage code pick out the real data.
Step 7: Perform intelligent scan to obtain the mystery equation.
Step 8: Carry out needed manipulation over the mystery equation to obtain the hidden key.
Step 9: Carry out character by character decryption.
Step 10: Store the decrypted information into a file.
Step 11: Stop.

Architecture of protocol

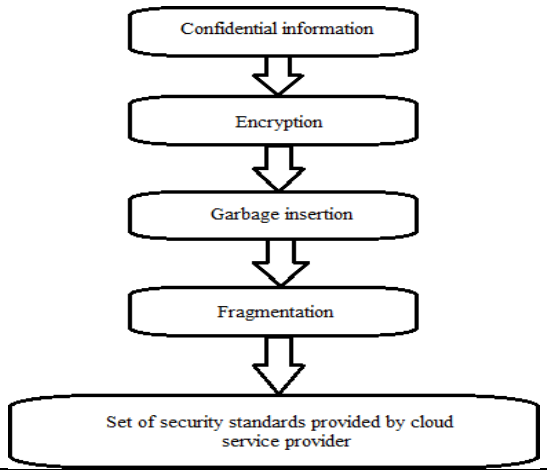

Fig 3 Protocol for preprocessing the data

6. Results and discussion

The following table 5.3 is derived by assuming the experiments conducted on two clouds.

Table 3 Results obtained for public verifiability

	File name	Owner	Verification	Status
User1	file1	Yes	Yes	No
User2	file1	No	Yes	Yes
User3	file1	No	No	Yes

Table 3 shows the results obtained for public verifiability. As in the table, the owner of the file is User1 (i.e. the file is created by User1). The file verification permission is given by the check column in the table, and the status column shows whether the file is checked or not. Note that the displayed status is "Yes" while user3 does not have any verification rights, which means the file receives verified by others.

Table 4 Results showing Access Rights to perform various operations

	File name	Owner	Authorized	View	Update	Delete
User1	file1	Yes	Yes	Yes	Yes	Yes
User2	file1	No	Yes	Yes	No	No
User3	file1	No	No	No	No	No

The permission rights for each user for different operations on stored data are explained in Table 4. Table 4 shows in the authorized column the authority or access rights given to a specific user. For instance, user2 is not the proprietor, but can only access the file. In contrast, user3 is not the owner and is not authorized to perform all operations.

Table 5 Service Availability through Clouds

No of requests	File name	Users served by	Users served by
		Cloud1	Cloud2
4	file1	4	-
6	File2	5	1
10	File3	5	5

Table 5 provides the service availability factor. The first column does not show simultaneous file requests in column 2. The results are presented in the following table, which simulates the experiment with only 5 requests for 10 users and for each cloud. When the number of requests exceeds 5, cloud2 serves 6th user requests automatically. This is because multi-cloud data are redundant. Due to the redundant availability of data in several clouds, the denial of service can be reduced. You can argue that. This reduces service denial at the cost of cloud storage.

Also the implementation of the proposed light weight algorithm brings out fascinating results. As the algorithm has a promise to perform the work of encryption and decryption swiftly, it delivers it by carrying out a set of simple calculations without consuming much of the resources. As a bunch of standard algorithmsout in the market take up lots of valuable time of the users these algorithm areused only with most valuable information. Most of the times the task of encrypting the data is handed over to the assistants or the tasks are outsourced. To

reveal the difference between the proposed and existing algorithms we compare AES, RSA and the proposed algorithm based on the standard C++ implementation of all three. The proposed algorithm offers many advantages:

1. The key length is large enough to support the key of almost any length with nofixed key length.
2. Application is faster than AES and RSAalgorithms.
3. Acts as additional layer of security for cloud contents.
4. Eliminates the need of key sharing which eliminates one area of vulnerability.
5. Even with the correct key the intruder will not get the original information.
6. As the decryption algorithm is with designated persons the chance of confidential information being stolen is largely degraded.

These results will make sure that the content of the cloud is safe. The algorithm is small, sweet and user friendly with dynamic decisions over fragmentation, garbage insertion and encryption key. In this situation the hackers must find an innovative and auto adaptive way to deal with this dynamic aspect of the algorithm. Comparison of the performance of the proposed algorithm with that of AES and RSA as carried out in the labs:

Table 6 Comparison of RSA, AES and NEW algorithm.

DATA SIZE	TIME TAKEN BY AES ALGORITHM	TIME TAKEN BY RSA ALGORITHM	TIME TAKEN BY PROPOSED ALGORITHM
6KB	114ms	112ms	4ms
10KB	116ms	216ms	5ms
15KB	116ms	270ms	5ms
25KB	117ms	409ms	6ms
30KB	122ms	511ms	7ms

The algorithm that is being introduced provides a set of extra features to increase the level of security. The result of employing the concept of garbage insertion is fascinating as it makes task of getting real data out of the file that is encrypted and fragmented with garbage a long process. Every feature increases the level ofsecurity as every level adds extra time needed to decipher the information. Hence when the information is deciphered the intruder will be long gone, not around to use it or the information is no longer valid.

The test results indicate that the AES Andréa algorithmsare slower in encrypting the information. Hence the proposed algorithm is best possible alternativeto use as this algorithm provides the security at four layers which adequate enough to handle any threat.

7. Conclusion

Obviously, while cloud computing has rapidly increased, cloud security remains the key issue. Customers do not want to lose privacy because of malicious cloud insiders. In addition, many customers have recently experienced a loss of availability. Therefore, the multi-cloud concept responds to service availability defects. Since the data is stored on different cloud servers, many competitors can simultaneously request the service. And services without distortion can be used easily. Data intrusion also causes cloud users numerous problems. Public verification and permission is essential to ensure safe cloud data storage. Public verification also allows customers to delegate tasks and permits for integrity control to address the issue of data intrusion.

Our construction is designed deliberately to achieve these three important objectives while keeping efficiency in mind. Provide public verification by eliminating third-party auditors, and improve data accessibility for other authentic Multi-Cloud users later. The files are either accessed by other users in the existing system by loading or sending them through or downloading the link once. This means that the file is either exposed to downloadable social networks or that more critical data are shared via emails by providing an email address to the recipient. The file system proposed contains files for other authentic users by entering a public key to get the file. In addition, a four layered security protocol with a lightweight encryption algorithm that is8 times faster than other traditional algorithms like AES and RSA isproposed.

Hence, I conclude that the adoption of the proposed protocol across all the platforms for speeding up the process of encryption without any expense of resources is an efficient solution. Integration of this protocol, as an encryption option to the users of CSaaS remains as a futurescope.

References

1. AlZain MA, Pardede E, Soh B, Thom JA (2012) Cloud computingsecurity: from single to multi-clouds. In: Proceedings of the 45thinternationalHawaiiconferenceonsystemsciences
2. Andre FM (2010) Availability and confidentiality in storage clouds.MS in Information Technology-Information Security, Informa-tion networkingInstitute,CarnegieMellonUniversity,December2010
3. BessaniA,CorreiaM,QuaresmaB,Andre´F,SousaP(2011) DEPSKY: Dependable and secure storage in a cloud-of-clouds.UniversityofLisbon,FacultyofSciences,Portugal
4. CachinC,KeidarI,ShraerA(2009)Trustingthecloud.ACMSIGACTNews40:81–86
5. Ederov B (2007) Merkle tree traversal techniques. Bachelor Thesis,Darmstadt University of Technology, Department of ComputerScienceCryptographyandComputerAlgebra,April2007
6. JaisonVimalraj T, Manoj M (2012) Enabling public verifiability anddata dynamics for storage security in cloud computing. Dha-nalakshmiSrinivasanEngineeringCollege,India
7. Kamara S, Lauter K (2010) Cryptographic cloud storage. In: C'10:Proceedings of the 14th international conference on financialcryptographyanddatasecurity,pp136–149
8. RochaF,CorreiaM(2011)Lucyintheskywithoutdiamonds:stealing confidential data in the cloud. In: Proceedings of the 1stinternational workshop of dependability of clouds, data centersandvirtualcomputingenvironments,pp1–6
9. Singh Y, Kandah F, Zhang W (2011) A secured cost-effective multi-cloudstorageincloudcomputing.DepartmentofComputerScience, North Dakota State University, Fargo, ND 58105 IEEEINFOCOM2011Workshoponcloudcomputing
10. SuyogBankar (2018) Cloud Computing Using Amazon Web Services (AWS) Volume-2 Issue-4, June 2018, pp.2156-2157
11. Subashini S, Kavitha V (2011) A survey on security problems in commission delivery models of cloud computing. J NetwComputAppl thirty four (1):1–11
12. Tribhuwan ,Bhuyar VA, Pirzade S (2010) making certain information storage security in cloud computing through 2 method handshaking supported token management. In: 2010 International conference on advances in recent technologies in communication computing. IEEE Society
13. Wang Q, Wang C, Li J, Ren K, Lou W (2009a) facultative public verifiability and information dynamics for storage security in cloud computing. Supported partly by the USA National Science Foundation below grant CNS-0831963, CNS-0626601, CNS- 0716306, CNS-0831628 and CNS-0716302
14. Wang Q, Wang C, Li J, Ren K, Lou W (2009b) facultative public verifiability and information dynamics for storage security in cloud computing. In: Proceedings people national science foundation below Grant CNS-0831963, CNS-0626601, CNS-0716306, CNS- 0831628 and CNS-0716302
15. Wang C, Wang Q, Ren K, Lou W (2010) making certain information storage security in cloud computing. In: ARTCOM'10: Proceedings of the international conference on advances in recent technologies in communication and computing, pp 1–9
16. NIST http://www.nist.gov/itl/cloud/. Accessed fifteen November 2012
17. Malek B Salem et al. A Survey of business executive Attack Detection analysis