A Periodic Approach in Access Control for Multi Authority Based on Cp-Abe Using Cloud Computing

P. Mathivanan[1], D. Mohana Priya[2]
[1,2] *Assistant Professor, Department of Information Technology, Manakula Vinayagar Institute of Technology, Puducherry, India*

***Abstract*— Cloud computing makes storage outsourcing become a rising trend, which promotes the secure remote data auditing as a blistering topic. Our research considers the problem of secure and proficient public data integrity auditing for shared dynamic data. The existing scheme provides an proficient public integrity auditing with secure group user revocation based on ciphertext policy attribute based encryption (CP-ABE) Commitment. But still this scheme is not consistent for secure group user revocation and also for dynamic cipher text database. In this project, we propose Secure Hash Algorithm (SHA-2) that supports dynamic cipher text and proficient user revocation. Additionally this work wrapped up with the properties, such as confidentiality, efficiency, count ability and traceability of secure group user revocation Finally, in the comparison of experimental analysis reduces the security complexity using this proposal.**

***Keywords*- Cloud Computing, ciphertext policy attribute based encryption(CP-ABE), secure Hash Algorithm (SHA-2), Encryption, Public Auditing**

I. INTRODUCTION

Cloud computing is the computing technology that helps us to make use of various services provided by the cloud, which is nothing but a remote server. The services offered include hiring servers, storage space, networking platform, database platform and so on. It also offers innovative tools, resource flexibility and good economy of scale. We pay only for what we use on the cloud. Various other benefits of cloud include ability to scale, upgraded performance, and elimination of hardware and software expenses and a limited security of data.

Cloud computing, also on-demand computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services)

Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture and autonomic and utility computing have led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease.

Cloud computing has become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per annum, but due to being in a stage of infancy, it still has pitfalls that need proper attention to make cloud computing services more reliable and user friendly.

Cloud computing is the result of the evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.

The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.

An important term associated with cloud computing is the encryption. Encryption is the method of encoding any data in a form that can be accessed only by those who are authorized to do so. Encryption provides security to data. As encryption is essential for cloud computing, usually encryption is performed by the cloud provider itself and they maintain the keys. This does not include the intervention of the data owner. Unfortunately, this methodology has high chances of data breach and theft, even the cloud provider itself can misuse the documents of data owner. Researchers have been continuously investigating solutions for storing data on public or private clouds but, where the private keys are under the control of the data owner itself. Waters et al. researched the searching of data on encrypted audit logs. Many such inventions focused only on single keyword search. Boneh et al. had focused on keyword searching which used public key encryption without revealing the content to suspects.

Cloud computing shares characteristics with:

- Client–server model—Client–server computing refers broadly to any distributed application that distinguishes between service providers (servers) and service requestors (clients).
- Grid computing—"A form of distributed and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks."
- Fog computing—Distributed computing paradigm that provides data, compute, storage and application services closer to client or near-user edge devices, such as network routers. Furthermore, fog computing handles data at the network level, on smart devices and on the end-user client side (e.g. mobile devices), instead of sending data to a remote location for processing.
- Dew computing—In the existing computing hierarchy, the Dew computing is positioned as the ground level for the cloud and fog computing paradigms. Compared to fog computing, which supports emerging IoT applications that demand real-time and predictable latency and the dynamic network reconfigurability, Dew computing pushes the frontiers to computing applications, data, and low level services away from centralized virtual nodes to the end users.

Cloud computing exhibits the following key characteristics:

- **Agility** improves with users' ability to re-provision technological infrastructure resources.
- **Cost** reductions claimed by cloud providers. A public-cloud delivery model converts capital expenditure to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and need not be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained, with usage-based options and fewer IT skills are required for implementation (in-house).[41] The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.
- **Device and location independence** enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- **Maintenance** of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.
- **Multitenancy** enables sharing of resources and costs across a large pool of users thus allowing for:
    - **Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - **peak-load capacity** increases (users need not engineer for highest possible load-levels)
    - **utilization and Efficiency** improvements for system that are often only 10–20% utilised.
- **Performance** is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.
- **Productivity** may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.
- **Reliability** improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery.
- **Scalability and elasticity** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time (Note, the VM startup time varies by VM type, location, OS and cloud providers), without users having to engineer for peak loads.

Security can improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford to tack.

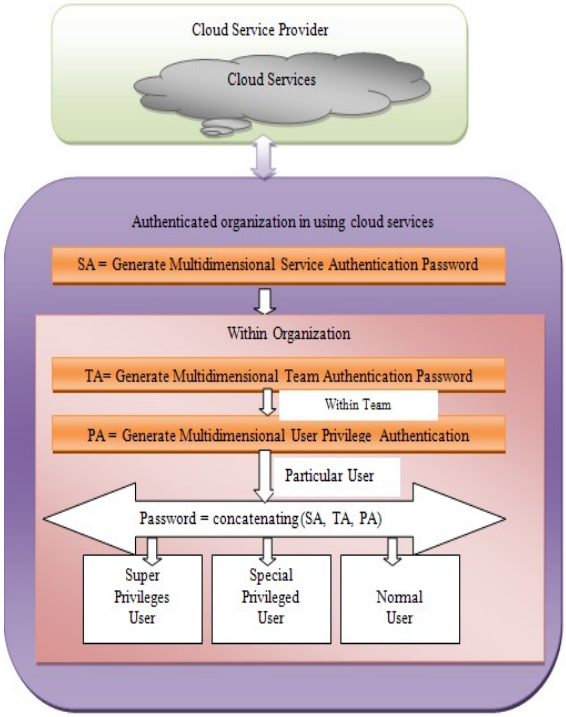

Fig: 1.1 System Model of Cloud Computing

Service Model

Cloud Computing is a process in which nothing but a particular design of computing where everything from processing energy to infrastructure, company applications etc. are offered "as a service". Cloud computing generally works on three types of architectures. These are: SAAS, PAAS and IAAS.

Software as a Service (SAAS)

Users are provided access to software and data source. Cloud providers handle the systems and facilities that run the programs.

Platform as a service (PAAS)

Cloud vendors provide a platform for computing, which includes the OS, coding language, execution environment, web server and the database. Developers can use the resources to build and run their software without buying expensive hardware.

Infrastructure as a service (IAAS)

Providers of IAAS offer computers – physical or virtual – and other resources. A good example of IAAS is dedicated servers provided by Web Hosting sites such as Bluehost, Hostgator etc.

II. RELATED WORK.

RAJANI SHARMA AND RAJENDER KUMAR TRIVEDI. LITERATURE REVIEW: CLOUD COMPUTING–SECURITY ISSUES, SOLUTION AND TECHNOLOGIES. INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH ISSN, PAGES 2319–6890, 2013

Cloud computing has grabbed the spotlight in the year 2013 at a conference in San Francisco, with vendors providing plenty of products and services that equip IT with controls to bring order to cloud chaos. Cloud computing trend is increasing rapidly so to make cloud computing more popular the very first step for the organization is to identify exact area where the cloud related threats lie. At an unusual pace, cloud computing has transformed business and government. And this created new security challenges. The development of the cloud service model provide business – supporting technology in a more efficient way than ever before .the shift from server to service based technology brought a drastic change in computing technology. However these developments have created new security vulnerabilities, including security issues whose full impressions are still rising. This paper presents an overview and study of cloud computing, with several security threats, security issues, currently used cloud technologies and security solutions.

Techniques Used:

A Markov chain is "a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. In probability theory and related fields, a Markov process, named after the Russian mathematician Andrey Markov, is a stochastic process that satisfies the Markov property (sometimes characterized as "memorylessness"). Roughly speaking, a process satisfies the Markov property if one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history, hence independently from such history; i.e., conditional on the present state of the system, its future and past states are independent.

A Markov chain is a type of Markov process that has either discrete state space or discrete index set (often representing time), but the precise definition of a Markov chain varies. For example, it is common to define a Markov chain as a Markov process in either discrete or continuous time with a countable state space (thus regardless of the nature of time) but it is also common to define a Markov chain as having discrete time in

either countable or continuous state space (thus regardless of the state space).

Random walks on integers and the gambler's ruin problem are examples of Markov processes. Some variations of these processes were studied hundreds of years earlier in the context of independent variables. Two important examples of Markov processes are the Wiener process, also known as the Brownian motion process, and the Poisson process, which are considered the most important and central stochastic processes in the theory of stochastic processes and were discovered repeatedly and independently, both before and after 1906, in various settings. The algorithm known as Page Rank, which was originally proposed for the internet search engine Google, is based on a Markov process. Furthermore, Markov processes are the basis for general stochastic simulation methods known as Gibbs sampling and Markov Chain Monte Carlo, are used for simulating random objects with specific probability distributions, and have found extensive application in Bayesian statistics.

Disadvantages

- Traffic Hijacking Insecure Interface and APIs.
- Denial of Service.
- Malicious Insiders.
- Abuse of Cloud Services

Advantages

- Insufficient Due Diligence
- Shared Technology Vulnerabilities
- It consists of high protection of data sharing security.
- It is reliable to monitor the data in two way communication channel.

F.MESSINA,G.PAPPALARDO,D.ROSACI,C.SANTORO, AND G. M. L. SARN. A TRUST MODEL FOR COMPETITIVE CLOUD FEDERATIONS. IN COMPLEX, INTELLIGENT AND SOFTWARE INTENSIVE SYSTEMS (CISIS), 2014 EIGHTH INTERNATIONAL CONFERENCE ON, PAGES 469–474, JULY 2014

This paper discussed the problem to promote mutual users interactions and cooperation within thematic groups in open virtual (agent) communities in presence of heterogeneous knowledge's among the affiliated users (i.e. the associated agents). To this aim, a framework is proposed such that each thematic group is assisted by a Group Agent and, in turn, each user is assisted by a Personal Agent. More in detail, each Personal Agent is specialized only on a specific theme (i.e. topic) and manages a personal profile (resp. catalog) of its owner's knowledge and interests, such that users are supported by one or more Personal Agents. In such a context, Group Agents provide to their affiliated Personal Agents some basic services. Each group catalog is extensible by the delegated Personal Agent in order to take into account other topics of interest for its user. Then such further knowledge can be exploited by the Group Agents to enrich their respective common Thematic Catalogs of their groups. As future work, we will perform a number of simulations in order to verify the effectiveness of this proposal.

Techniques Genetic Algorithm

A genetic algorithm (**GA**) is a meta heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). In a genetic algorithm, a population of candidate solutions (called individuals, creatures or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

Disadvantages

- Query forms are designed and pre-defined by developers in information management systems.
- Difficult to design a set of static query forms to satisfy various ad-hoc database queries on complex databases.

Advantages:

- We propose a dynamic query form generation approach which helps users dynamically generate query forms.
- The dynamic approach often leads to higher success rate and simpler query forms compared with a static approach.

- The ranking of form components also makes it easier for users to customize query forms.

TASK-DEPENDENT VISUAL-CODEBOOK COMPRESSION AUTHORS: HONGXUN YAO ; WEI LIU ; XIAOSHUAI SUN; QI TIAN

In this project, an introduce a novel indexing scheme-query context tree (QUC-tree) to facilitate efficient query sensitive music search under different query contexts. Distinguished from the previous approaches, QUC-tree is a balanced multiway tree structure, where each level represents the data space at different dimensionality. Before the tree structure construction, principle component analysis (PCA) is applied for data analysis and transforming the raw composite features into a new feature space sorted by the importance of acoustic features. The PCA transformed data and reduced dimensions in the upper levels can alleviate suffering from dimensionality curse. To accurately mimic human perception, an extension called QUC +-tree is proposed, which further applies multivariate regression and EM based algorithm to estimate the weight of each individual feature. The comprehensive extensive experiments to evaluate the proposed structures against state-of-art techniques based on different datasets. The experimental results demonstrate the superiority of the technique.

Techniques EM Algorithm

An **expectation–maximization** (**EM**) **algorithm** is an iterative method to find maximum likelihood or maximum a posteriori(MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

The EM algorithm is used to find (local) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either missing values exist among the data, or the model can be formulated more simply by assuming the existence of further unobserved data points.

Disadvantages

- Retrieval using low-level (local) features: color, texture, shape at specific parts of music's (an music is a collection of descriptors of local features)
- Semantic approaches where an music is, in some way, represented as a collection of objects and their relations

Advantages

- Re-rank music list by exploring the contents of music and their associated tags.
- Music is represented by a vertex in the constructed hyper-graph, and the visual clustering results are employed to construct the hyper-edges.

III. SYSTEM ANALYSIS

3.1 Existing Work

In Existing System, Cipher text-policy attribute-based encryption (CP-ABE) is a useful cryptographic method for data access control in cloud storage. All these CP-ABE based schemes enable data owners to realize fine-grained and flexible access control on their own data. However, CP-ABE determines users' access privilege based only on their inherent attributes without any other critical factors, such as the time factor. fine-grained access control for time sensitive data in cloud storage. One challenge is to simultaneously achieve both flexible timed release and fine granularity with lightweight overhead, which was not explored in existing works.

In the current system, to avoid collusion of revoked users and to provide data secrecy in the group users. A CP-ABE is a collection of three ex polynomial-time algorithms (VLR.KeyGen, VLR.Sign, VLR.Verify). Asymmetric Group Key Agreement (ASGKA) and group signatures are used to support cipher text data base update among group users and efficient group user revocation respectively. Only a shared cipher text key is used to encrypt or decrypt the messages by this ASGKA scheme. And the public key can be simultaneously used to verify signatures. The primitive of verifiable database with efficient update based on vector commitment is useful to solve the problem of verifiable data outsourcing.

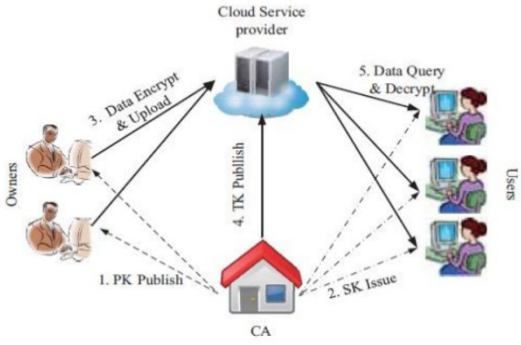

Figure : 3.1 Existing System model

- **The Central Authority (CA)** is responsible to manage the security protection of the whole system
- **The Data Consumer (User)** is assigned a security key from CA.
- **Cloud Service Provider (Cloud)** includes the administrator of the cloud and cloud servers.
- **The Data Owner (Owner)** decides the access policy based on a specific attribute set and one or more releasing time points for each file

Disadvantages

- CP-ABE protocol in does not consider the users 'authentication.
- It can implement the well-known man-in-the-middle attacks during the protocol execution.
- It only supports static group whose membership is fixed.
- New start round execution is required if any member leave or join the group.
- Efficient data secrecy is not maintained.
- This protocol does not allows users to join or leave the group at the same time.
- Computational cost is high.

3.2 System Architecture:

In this project, we propose Secure Hash Algorithm (SHA-2) that supports dynamic cipher text and proficient user revocation. Additionally this work wrapped up with the properties, such as confidentiality, efficiency, countability and traceability of secure group user revocation Finally, in the comparison of experimental analysis reduces the security complexity using this proposal.

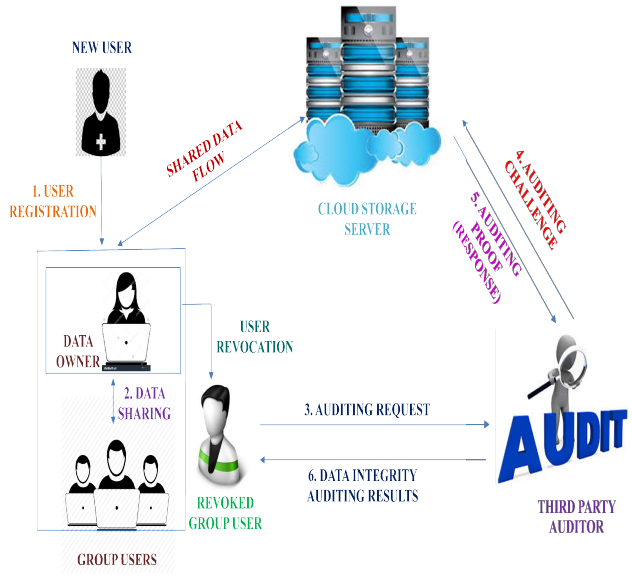

Fig:3.2 System architecture of Public auditing

1. User Registration: It is used to to create a account for the user.
2. Data Sharing: It is used to share a datas between the data owner and users registered.
3. Auditing Request: Its is a request sent by the admin to the auditee.
4. Auditing Challenge: Audited details are stored to the cloud services.
5. Auditing Proof: It is an Response sent to the Third party Auditor from cloud.
6. Data Integrity: It is the accuracy of audited data.

IV. PROPOSED ALGORITHM

In this project, we propose Secure Hash Algorithm (SHA-2) that supports dynamic cipher text and proficient user revocation. Additionally this work wrapped up with the properties, such as confidentiality, efficiency, count ability and traceability of secure group user revocation Finally, in the comparison of experimental analysis reduces the security complexity using this proposal.

A **cryptographic hash** is a kind of 'signature' for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text.

A hash is not 'encryption' – it cannot be decrypted back to the original text (it is a 'one-way' cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare 'hashed' versions of texts, as opposed to decrypting the text to obtain the original version.

Such applications include hash tables, integrity verification, challenge handshake authentication, digital signatures, etc.

'Challenge Handshake Authentication' (or 'challenge hash authentication') avoids transmission passwords in 'clear' – a client can send the hash of a password over the internet for validation by a server without risk of the original password being intercepted Anti-tamper – link a hash of a message to the original, and the recipient can rehash the message and compare it to the supplied hash: if they match, the message is unchanged; this can also be used to confirm no data-loss in transmission Digital signatures are rather more involved, but in essence, you can sign the hash of a document by encrypting it with your private key, producing a digital signature for the document. Anyone else can then check that you authenticated the text by decrypting the signature with your public key to obtain the original hash again, and comparing it with their hash of the text. Hash functions are not appropriate for storing encrypted passwords, as they are designed to be fast to compute, and hence would be candidates for brute-force attacks. Key derivation functions such as bcrypt or scrypt are designed to be slow to compute, and are more appropriate for password storage (npm has bcrypt and scrypt libraries, and

PHP has a bcrypt implementation with password hash). SHA-256 is one of the successor hash functions to SHA-1 (collectively referred to as SHA-2), and is one of the strongest hash functions available. SHA-256 is not much more complex to code than SHA-1, and has not yet been compromised in any way. The 256-bit key makes it a good partner-function for AES.

Digital signatures are rather more involved, but in essence, you can sign the hash of a document by encrypting it with your private key, producing a digital signature for the document. Anyone else can then check that you authenticated the text by decrypting the signature with your public key to obtain the original hash again, and comparing it with their hash of the text.

Notes on the implementation of the preprocessing stage:

- FIPS 180-4 specifies that the message has a '1' bit appended, and is then padded to a whole number of 512-bit blocks, including the message length (in bits) in the final 64 bits of the last block
- Since we have a byte-stream rather than a bit-stream, adding a byte '10000000' (0x80) appends the required bit "1".
- To convert the message to 512-bit blocks, I calculate the number of blocks required, N, then for each of these I create a 16-integer (i.e. 512-bit) array. For each if these integers, I take four bytes from the message (using charCodeAt), and left-shift them by the appropriate amount to pack them into the 32-bit integer.
- The charCodeAt() method returns NaN for out-of-bounds, but the '|' operator converts this to zero, so the 0-padding is done implicitly on conversion into blocks.

However, JavaScript bit-ops convert their arguments to 32-bits, so n >>> 32 would give 0. Hence I use arithmetic operators instead: for the most-significant 32-bit number, I divide the (original) length by 2^32, and use floor() convert the result to an integer.

The **Secure Hash Algorithm** are a family of cryptographic hash functions published by the National Institute of Standards and Technology(NIST) as a U.S Fedral Information processing Standard (FIPS) including

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.
- **SHA-1**: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic weaknesses were discovered in SHA-1, and the standard was no longer approved for most cryptographic uses after 2010.
- **SHA-2**: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224, SHA-384, SHA-512/224 and SHA-512/256. These were also designed by the NSA.
- **SHA-3**: A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

In the proposed system, we suggest that to contribute dynamic cipher text cryptography process with persistent public data integrity auditing in the cloud server storage. Secure Hash Algorithm (SHA-2) is designed for compatibility with increased security provided by the Advanced Encryption Standard (AES) cipher. It is a set of cryptographic hash functions that prevents collusion. Used to generate a digital signature on data and by a verifier to verify the authenticity of the signature. For both signature generation and verification, by SHA-2. An adversary, who does not know the private key of the signatory, cannot generate the correct signature of the signatory (ie) signatures cannot be forged. For data or message modification one must know the cipher text to upgrade it. Dynamically cipher text is generated and sent via mail to the group users who requested access to the data.

Algorithm Function

```
public string EncryptText(string input, string password) {

byte[] bytesToBeEncrypted =
Encoding.UTF8.GetBytes(input);

byte[] passwordBytes = Encoding.UTF8.GetBytes(password);

// Hash the password with SHA256

passwordBytes =
SHA256.Create().ComputeHash(passwordBytes);

byte[] bytesEncrypted = AES_Encrypt(bytesToBeEncrypted,
passwordBytes); string result =
Convert.ToBase64String(bytesEncrypted); return result;

}

public string DecryptText(string input, string password) {

byte[] bytesToBeDecrypted =
Convert.FromBase64String(input);

 byte[] passwordBytes =
Encoding.UTF8.GetBytes(password);
```

passwordBytes =
SHA256.Create().ComputeHash(passwordBytes);

byte[] bytesDecrypted = AES_Decrypt(bytesToBeDecrypted, passwordBytes); string result = Encoding.UTF8.GetString(bytesDecrypted);

return result;

Advantages

- ✓ Group user's authentication is possible.
- ✓ It is resistant to all possible attacks.
- ✓ It completely eliminates collusion of the group users.
- ✓ The encryption hash used in SHA-2 is significantly stronger.
- ✓ Encryption is applied to the entire message plus hash code, confidentially is improved.
- ✓ Provides constant security for the group member's communication.
- ✓ SHA provides greater security than ASGKA using dynamic cipher text sharing.
- ✓ High speed processing at low cost of investment.
- ✓ Widely implemented in popular security applications and protocols like SSL, TLS, IPSec, SSH, PGP etc.

V. IMPLEMENTATION

5.1 Module Description

Account Creation

In the login page admin can create Auditor and Audit their account by register option and can log onto through their account by entering their details by using log in option.

Create Assignment

The admin recommend distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together or individual shares are of no use on their own.

Key Generation

Key generation is the process of generating keys for cryptography. The key is used to encrypt and decrypt data whatever the data is being encrypted or decrypted. Modern cryptographic systems include symmetric-key algorithms (such as DES and AES) and public-key algorithms (such as RSA). Symmetric-key algorithms use a single shared key; keeping data secret requires keeping this key secret. Public-key algorithms use a public key and a private key.

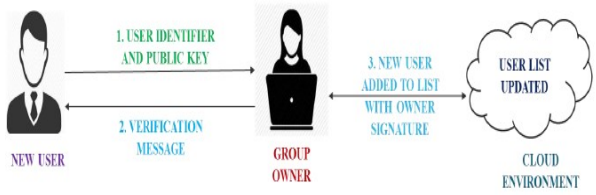

Fig: 3.3 Key Generation

The public key is made available to anyone (often by means of a digital certificate). A sender will encrypt data with the public key; only the holder of the private key can decrypt this data. Since public-key algorithms tend to be much slower than symmetric-key algorithms, modern systems such as TLS and its predecessor SSL as well as the SSH use a combination of the two in which:

1. One party receives the other's public key, and encrypts a small piece of data (either a symmetric key or some data that will be used to generate it).
2. The remainder of the conversation (the remaining party) uses a (typically faster) symmetric-key algorithm for encryption.

In this project cryptography keys are integers. In some cases keys are randomly generated using a random number generator (RNG) or pseudorandom number generator (PRNG), the latter being a computer algorithm that produces data which appears random under analysis. Some types the PRNGs algorithms utilize system entropy to generate a seed data, such seeds produce better results, since this makes the initial conditions of the PRNG much more difficult for an attacker to guess. In other situations, the key is created using a passphrase and a key generation algorithm, using a cryptographic hash function such as SHA-5.

Dynamic Data Sharing and Data Uploading

Uploading Data is one in which an Audit used the website and they used to give tag names ,its description and divide its category and etc…It is used to upload any types of image but not video files.

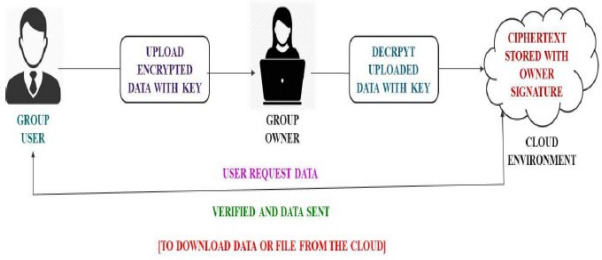

Fig:3.3 Dynamic Data Sharing

A program that searches for and identifies items in a database that correspond to keywords or characters specified by the

user, used especially for finding particular sites on the Internet.

User Revocation and Collusion Resistance

Users can easily store and share their data with each other using the cloud technology. Large numbers of users are not assured about integrity of their data by the reason of threats to security in a cloud. Many mechanisms are proposed and being used to verify the integrity or the correctness of single owner shared data. They suggest attaching signatures to the data. This will provide public auditing on multi-owner shared data. When user is revoked from the group, there must be some method to resign those blocks that are signed by that revoked user. It will also provide efficient user revocation with collusion resistance i.e. even if cloud colludes with any revoked users; it will not understand the contents of the data which is stored on cloud.

V. CONCLUSION AND FUTURE ENHANCEMENTS

With the cloud storage services, users can easily form a group and share data with each other. Given the fact that the cloud is not trustable, users need to compute signatures for blocks of the shared data to allow public integrity auditing. Once a user is revoked from the group, the blocks that were previously signed by this revoked user must be re-signed by an existing user, which may result in heavy communication and computation cost for the user. Proxy re-signatures can be used here to allow the cloud to do the re-signing work on behalf of the group. However, a malicious cloud is able to use the re-signing keys to arbitrarily convert signatures from one user to another deliberately. Moreover, collusions between revoked users and a malicious cloud will disclose the secret values of the existing users. In this project, we propose a novel public auditing scheme for the integrity of shared data with efficient and collusion-resistant user revocation utilizing the concept of Shamir secret sharing. Besides, our scheme also supports secure and efficient public auditing due to our improved polynomial-based authentication tags.

REFERENCES

[1]. A. Akavia, S. Goldwasser, and V. Vaikuntanathan, “Simultaneous hardcore bits and cryptography against memory attacks,” in Proc. 6th Theory Cryptography Conf., 2009, pp. 474–495.
[2]. S. S. Al-Riyami and K. G. Paterson, “Certificateless public key cryptography,” in Proc. 9th Int. Conf. Theory Appl. Cryptol., 2003, pp. 452–473.
[3]. M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, “Certificate based (linkable) ring signature,” in Proc. Inf. Security Practice Experience Conf., 2007, pp. 79–92.
[4]. M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, “Malicious KGC attacks in certificateless cryptography,” in Proc. 2nd ACM Symp. Inf., Comput. Commun. Security, 2007, pp. 302–311.
[5]. M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., 1998, pp. 127–144.
[6]. A. Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryption with efficient revocation,” in Proc. ACM Conf. Comput. Commun. Security, 2008, pp. 417–426.
[7]. D. Boneh, X. Ding, and G. Tsudik, “Fine-grained control of security capabilities,” ACM Trans. Internet Techn., vol. 4, no. 1, pp. 60– 82, 2004.
[8]. D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” in Proc. 21st Annu. Int. Cryptol. Conf., 2001, pp. 213– 229.
[9]. R. Canetti and S. Hohenberger, “Chosen-ciphertext secure proxy re-encryption,” in Proc. ACM Conf. Comput. Commun. Security, 2007, pp. 185–194.
[10]. H. C. H. Chen, Y. Hu, P. P. C. Lee, and Y. Tang, “NCCloud: A network- coding-based storage system in a cloud-of-clouds,” IEEE Trans. Comput., vol. 63, no. 1, pp. 31–44, Jan. 2014.
[11]. S. S. M. Chow, C. Boyd, and J. M. G. Nieto, “Security-mediated certificateless cryptography,” in Proc. 9th Int. Conf. Theory Practice Public-Key Cryptography, 2006, pp. 508–5
[12]. J. Lähteenm äki, J. Leppänen and H. Kaijanranta, “Interoperabil ty of Personal Health Records”, 31st Annual Intern ational Conference of the IEEE E MBS, pp. 1726-1729, 2009.