Data Security In Cloud Computing : A Survey

Devi T and Ganesan R

VIT University, School of Computing Sciences and Engineering, Chennai, India.
VIT University, School of Computing Sciences and Engineering, Chennai, India.

ARTICLE INFO

Article history:
Received 25 August 2014
Received in revised form
19 September 2014
Accepted 29 November 2014
Available online 12 December 2014

Keywords:
access control; cloud computing; data security; encryption

ABSTRACT

Cloud computing provides a wide range of services to consumers with lesser investment which are delivered over Internet. Cloud offers massive spaces for data storage and computations to the clients. Security is considered to be a significant problem in the development of cloud computing which actually an obstacle to the growth of cloud computing. The most challenging task in a shared environment like cloud is data security. Cloud enables the clients to store data on remote servers thereby preventing direct control over this data. So, proper data security mechanism must be maintained involving encryption of the data. The major objective is to transmit, store and retrieve data in a secure manner. The main focus of this paper is to highlight the various encryption techniques and proof of storage methods. Encryption techniques are used to secure sensitive data when transferring or storing data in cloud. Speed of processing and computational efficiency of encrypting data should also be considered since the cloud environment involves large amount of data transfer, storage and usage. Data integrity proof a challenge—response protocol where a user challenges the server to check the correctness of data. This paper presents the survey about data security issues in cloud computing and appropriate solutions with their pitfalls. Also, a hybrid model for data security in cloud computing is proposed.

INTRODUCTION

Cloud refers to storing the user’s data in a remote database instead of storing it in the hard disk of their own computer. The connection between the remote database and the user computer is provided by the Internet. Cloud delivers computing resources as a service in a scalable manner to the clients by means of Internet which eliminates the need of setting up company’s own data center or server. These resources are offered on demand and customers pay for their level of usage. The main objective of cloud computing is to make better use of distributed resources. Peter and Tim(2009) stated that “The term cloud computing comes from the early days of the Internet where we drew the network as a cloud. We didn’t care where the messages went. The cloud hid it from us”. Cloud made the words of John McCarthy in 1960 that computation may someday be organized as a public utility to become true.

NIST defines cloud computing as a model that facilitates convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be quickly provisioned and released with least management effort or service provider interaction. Components of Cloud are (i) the user, who makes use of data in cloud, (ii) the Cloud Service Provider (CSP), who offer the resources to the clients based on their demand, and (iii) the data owner, who possess the data and stores in cloud for further maintenance.

A. Grid Vs Cloud Computing:

Cloud Computing is based on certain study areas like virtualization, utility computing as well as grid computing. In case of grid, many computers are involved to solve a problem by working together. But, in cloud an application access the resources it needs through services. Cloud computing differs from grid computing by virtualization. Cloud allows maximum utilization of computing resources like CPU, storage, network, memory and database through techniques like virtualization, thus removing the workload from the clients. Virtualization is an abstraction of compute resources. It allows several virtual machines with dissimilar operating systems to run on the same host machine with the help of hypervisor. In grid, high utilization is achieved by allocating
multiple servers for a single task whereas in cloud, a single server is allocated to perform multiple tasks by virtualization of servers.

B. **Cloud Characteristics:**

Five important cloud characteristics are:

- **On-demand self-service:** A user will cater the resources based on his requirement without interacting directly with every provider.

- **Broad Network access:** Resources available over the network are open to users by means of their phones, laptops and so on.

- **Resource Pooling:** Resources are shared to aid many customers based on multi-tenant model. The resources are allocated and reallocated in a dynamic manner according to user’s necessity.

- **Rapid Elasticity:** Elasticity refers to scalability of resources according to the need of the user.

- **Measured Service:** The cloud provider monitors or measures and controls the services for various reasons like billing, resource optimization and planning.

Figure 1 shows 2008 IDC report stating the benefits in cloud computing.

![Figure 1: Benefits of Cloud Computing](source: IDC Enterprise Panel, August 2008, n=284)

The various deployment models in cloud are as follows:

- **Public:** The cloud infrastructure is functions for common public and is possessed by an organization promoting cloud services. Services are available for general user over the internet.

- **Private:** The cloud infrastructure functions for a private business. It’s like a virtualized datacenter operating within firewall.

- **Community:** The cloud infrastructure is made common to several organizations with the same principles and agreement consideration. It is possessed, administered, and controlled by a single or others of the organization in the community or a third party and it shall be on or off site.

- **Hybrid:** The cloud infrastructure is a grouping of clouds (public, private or community). The user may place less critical information in the public cloud while more critical data residing in the private cloud.

Figure 2 depicts the various delivery models in cloud like SaaS, PaaS and IaaS.

**Software-As-a-Service (SaaS):** Client can access the software and the data associated with it on cloud through a browser. The user does not handle or administer the cloud infrastructure as well as network and servers.

**Platform-As-a-Service (PaaS):** Client can utilize the provider’s applications running on a cloud infrastructure and may be accessed from user devices with the help of interfaces like web browser.

**Infrastructure-As-a-Service (IaaS):** Client can order resources based on their demand and they can set up and run any software as well as operating systems and applications. The user handles the resources like operating system, storage and applications.
Fig. 2: Cloud Service Stack

Other emerging models are secure logging-as-a-service, storage-as-a-service and database-as-a-service (a prime example of SaaS). The database management software is installed and managed by the service provider. Data security can be provided by encryption techniques because the organizations are not sure that their data is secure or not.

This paper depicts various data security issues and solutions of cloud computing. Since data is stored in remote untrusted servers, security of data is a major concern in cloud. Section 2 presents vulnerabilities, threats and attacks to cloud computing. Section 3 presents significance of security in cloud. Section 4 presents the evaluation criteria for security models. Section 5 describes Class A security models and Section 6 deals with authentication and access control in cloud computing. Section 7 describes Class B security models and Section 8 presents brief discussion about the combination of class A and class B model. Section 9 describes conclusions derived out of this survey.

II. Vulnerabilities, Threats And Attacks To Cloud Computing:

A. Vulnerabilities in Cloud Environment:

Vulnerability is the probability an asset will not be capable to withstand the events of a threat agent. Xiao and Xiao(2013) discussed the vulnerabilities in cloud as:

VM co-residence: Since multiple users share the same physical infrastructure, several security issues like cross VM-attack may arise.

Loss of physical control: In cloud computing, user lack direct control on the data since it is outsourced to cloud servers. These customers may be unable to resist certain attacks as a result of which their data may be altered or lost.

Bandwidth Under-provisioning: A datacenter formed may be under-provisioned with a factor of 2.5:1 to 8:1, which means real network capability is not more than total host capability in the similar subnet.

Cloud pricing model: The customer pay for the usage of resources in cloud computing. The attackers may also harass the billing process. Table 1 presents the effects of vulnerabilities in cloud environment.

<table>
<thead>
<tr>
<th>Vulnerability</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulnerabilities in virtualization</td>
<td>Bypassing security barriers can allow access to underlying hypervisor</td>
</tr>
<tr>
<td>Vulnerabilities in Internet protocol</td>
<td>Allow network attacks</td>
</tr>
<tr>
<td>Unauthorized access to management interface</td>
<td>Intruder can gain access control</td>
</tr>
<tr>
<td></td>
<td>Access to administrative interface can be more critical</td>
</tr>
<tr>
<td>Injection vulnerabilities</td>
<td>Unauthorized disclosure of private data</td>
</tr>
<tr>
<td>Vulnerabilities in browsers and APIs</td>
<td>Allow unauthorized service access</td>
</tr>
</tbody>
</table>

Table 1: Effects Of Vulnerabilities


B. Threats to Cloud Computing:

Threats to cloud computing are violation of data, loss of data, seizing the account, insecure application programming interfaces (APIs), rejection of service, malicious insiders, abuse and disreputable use, lacking carefulness and common technology problems. Table 2 presents the effects of threats and specifies the cloud services affected by them.

<table>
<thead>
<tr>
<th>Threats</th>
<th>Effects</th>
<th>Affected Cloud Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious Insiders</td>
<td>Insider malicious activity bypassing firewalls and other security model</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>Data Loss and Leakage</td>
<td>Confidential data can be deleted, modified or compromised</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>Service hijacking</td>
<td>User accounts and service instances make a new base for attackers</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>Shared Technology Issues</td>
<td>Allow one user to interface other users' services by compromising hypervisor</td>
<td>IaaS</td>
</tr>
<tr>
<td>Identity Theft</td>
<td>Attacker can get valid user's identity to access resources and obtain benefits in that username</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
</tbody>
</table>

Table 2: Effects Of Threats
C. Attacks on Cloud Computing:

Attacks include DoS attack, service injection attack, man-in-the-middle attack and metadata spoofing attack. Table 3 highlights the types of attacks an adversary can commence in cloud with their effects and the services affected.

Table 3: Types of attacks on cloud with effects

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Effects</th>
<th>Service Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Injection attack</td>
<td>Instead of valid service, malicious service is provided to users. Affects service integrity.</td>
<td>PaaS</td>
</tr>
<tr>
<td>Man-in-the-middle attack</td>
<td>Affects data security and privacy</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>DoS/DDoS attack</td>
<td>Affects service availability.</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>Phishing attack</td>
<td>Affects privacy of user’s sensitive information</td>
<td>SaaS, PaaS and IaaS</td>
</tr>
<tr>
<td>Attacks on virtualization, VM escape and attacks on hypervisor</td>
<td>Allows an attacker to gain control over another user’s VM.</td>
<td>IaaS</td>
</tr>
</tbody>
</table>

III. Significance of Security In Cloud Computing:

Data in cloud actually remains in remote servers. User and data owner should be confident enough that the data has not been corrupted during transmission, storage and retrieval. Therefore, security is the major concern in cloud environment to ensure authorized access. Figure 3 shows IDC report 2008 clearly describing that security at the top of the list is the major area of focus in cloud environment compared to other parameters as recognized by organizations.

Security in cloud consists of storage security, data security, middleware security, network security and application security. Armbrust et al(2010) states that top 10 difficulties in the development of cloud computing are availability of service, data lock-in, confidentiality of data and auditability, bottlenecks in data transfer, performance unpredictability, scalable storage, errors in large-scale distributed systems, scaling speedily, reputation fate sharing and licensing of software.

Subashini and Kavitha(2011) presented a survey in cloud computing focusing on the security concerns in service delivery models with detailed explanation of these security issues[9]. These surveys show that security is the most important problem in the growth of cloud computing. US research firm Gartner released a report entitled “Assessing the security risk of cloud computing” in June 2008 which lifts the fear of risk in storage, recovery, privacy, and integrity of data. Figure 4 depicts the complexity of security in cloud environment.
The security attributes are confidentiality, integrity and availability (Figure 5).

Confidentiality: In case of cloud, data confidentiality is associated to authentication of the user. SaaS offerings are accessed by the user over Internet through a web browser. So, protection of user’s data and network traffic from unauthorized access is the major concern. The development environment offered by a PaaS provider also has similar confidentiality concerns. Confidentiality in case of IaaS includes isolating the usage of resources among multiple users.

Integrity: Data integrity refers to consistency of the data stored on cloud servers. In SaaS, protection should be provided to the data-in-transit, data-in-rest and network traffic. The platform settings and configuration files are important in case of PaaS and IaaS, since a small change in the settings or files can affect the SaaS applications which are deployed on them.

Availability: The main feature of cloud computing is to provide services only when the user is in need of those. If certain service is not available at the time of need, the customer may lose faith in the cloud system. Resources like Domain Name System (DNS) are the targets of attackers in case of availability.

The security measures can be divided into external and internal security. The security measures provided by the cloud to access it comes under external security whereas the security measures cloud offers to secure and separate clients within the cloud comes under internal security. The taxonomy of services in cloud computing can be represented as a tree structure with the root representing all the cloud services, first level with SaaS, PaaS and IaaS followed by the characteristics of these services. Cloud security is the responsibility of both the provider and a user. Figure 5 presents a graph showing that data integrity has a major part in the entire service delivery model.

Fig. 5: Security attributes in cloud

A. Need for Data Security in Cloud Computing:

Generally, data security deals with data protection. Security involves protecting data from being lost or destroyed or corrupted or modified or disclosed. Instead of storing the data locally, users store them in cloud. So, correctness and availability of data must be assured. The primary concern in cloud computing is protection of user data which is the major research focus area. But new challenges arise when they are being employed. Open challenges in case of data security exist in: data-in-transit, data-at-rest, lineage of data, remanence of data, provenance of data, recovery of data, location of data, data breaches, and support in case of investigations.

Data is transferred over a network via internet. An adversary in the network can affect the data secrecy and reliability. The data that is not being moved from one system to another and stored is data at rest. Its major problem lies in loss of control. The path that data takes is data lineage. It is difficult to trace the path of data in cloud where the resources have been virtualized. Left out data is data remanence. This may also incur few security problems for the customers. Data recovery is the most challenging problem. Discovering the data location in cloud is a difficult task.

Data encryption and access control are familiar techniques to resolve security issue. To provide data security in cloud, various encryption techniques like homomorphic encryption, functional encryption, broadcast encryption, searchable encryption and data integrity proof such as proof of retrievability (POR), provable data possession (PDP) have been employed.

IV. Evaluation Criteria For Security Models:

Most of the data security models presented in the survey are based on encryption methods. The rest of models are based on integrity proofs. So, this survey categorizes the existing data security models in cloud computing into two major classes namely:

- Class A - Security Model Based on Encryption Techniques (SM-ET)
- Class B - Security Model Based on Data Integrity Proofs (SM-DIP)
Computational requirements and scalability play major role while employing security models in cloud computing environment. Clients store their file on cloud and do not want their files to be exposed. Security models should also offer privacy and security features to its users.

A. Evaluation criteria for Class A models:
Class A deals with security models based on encryption techniques. The following parameters are considered for comparing the existing security models.

1) Building Block
The building block specifies the basic theory of the discussed data security model for cloud computing. They may be mathematical or cryptographic principles which helps to identify the computational needs of the discussed data security models.

2) Authentication
Cloud server enables only authenticated users to access the resources. This parameter considers authentication issues in the discussed models.

3) Protection of Data
The models may provide complete or partial data protection due to some of the limitations of the techniques used in the discussed models. This parameter identifies the category of data protection as moderate or partial for existing data security models.

4) Integrity of Data
To remove the burden of local storage. clients upload data in cloud and lose their control over data. So, there must be a mechanism to verify the correctness of clients' data in cloud. Data integrity parameter is used to identify integrity verification issue in discussed models.

5) Scalability
The model is said to be more scalable if the number of users can be increased without affecting the performance. If the model depends on centralized server, scalability is considered to be fair, else its poor.

6) Access of Data
Automated access allows clients to share their encrypted files on cloud and authenticated users can use them automatically. Semi-automated access enable users to access the uploaded file by the use of passwords or key exchanges.

B. Evaluation criteria for Class B models:
Class B deals with data security models based on integrity proofs. The following parameter have been selected to evaluate the existing models:

1) Type of Proof
This parameter identifies the type of proof discussed in the existing data security model.

2) Security Aspects
The security aspects identifies the security features of the models discussed in cloud computing. Aspects are data security, privacy, integrity, authentication or risk management.

3) Scalability
Scalability parameter is considered more if the model adapts increased number of users without degrading the performance of the model. Scalability is fair if the model depends on centralized server, else its poor.

V. Encryption Techniques In Cloud Computing (Class A Models):
Tang et al.(2010) discussed that cryptography is the key technology to solve the problems related to data security in cloud environment. Clouds don’t possess walls to secure their data as in case of data centers. So, the only solution to replace these physical walls and to protect their data against unauthorized access and data tampering when stored on untrusted servers is encryption. The major role of encryption is to ensure that cloud hosted data remains secret via encrypted transmission of data and encrypted storage of data. Clear data is converted into ciphertext by using encryption and this ciphertext is decrypted by only an approved individual with a legitimate decryption key.

Cloud characteristics like scalability, multi-tenancy and virtualization make traditional security mechanisms to perform worse to make data safe in cloud. A responsibility to provide security in cloud is shared by both the provider and the users. This may differ according to the service delivery models. IaaS users are more responsible for confidentiality of data whereas IaaS providers deal with data integrity issue. Providers and users are both equally responsible for protecting data and confidentiality issue in PaaS and SaaS. Platform availability is dealt by providers.

Two major means to ensure confidentiality of data in any IT environment are encryption and access control. So, management of these cryptographic keys is a challenging complex problem in cloud environment where physical and logical control of resources is divided between the actors of cloud like consumer, broker, and
provider and so on. Since the users are not skillful enough to handle the keys hence key management is given to the providers. States of various types of keys and management issue of keys for IaaS, PaaS and SaaS are discussed in detail by Chandramouli et al. (2013). Figure 6 depicts how key management functions are implemented by cloud providers.

![Fig. 6: Key management functions by cloud providers](image)

**A. Homomorphic Encryption:**

Data security model proposed uses fully homomorphic symmetric encryption algorithm and focuses on cipher text retrieval process and data integrity verification. Figure 7 shows the model which aims at offering various IT services for SMEs as discussed by Sun and Wang (2013). Homomorphic encryption is a public key system where a user encrypts message with public key published online and only the user with secret key decrypts it. This type of encryption prevents the need of encrypting the data before sending to the cloud by enabling a user to carry out functions on the data that is encrypted without decrypting that data. The necessity of knowing the secret key is thus avoided as discussed by Peter et al. Homomorphic encryption (HE) allows a user to protect its data from an untrusted cloud by encrypting its data using encryption function and the cloud does computation on encrypted data. Algebraic functions applied to the ciphertext will be reflected in the equivalent plaintext.

![Fig. 7: Structure of Data Security Model](image)

Tebaa et al. (2012) discussed that additive and multiplicative homomorphic encryption methods have been proposed to perform addition and multiplication calculations on encrypted data. Additive homomorphic encryption is used in case of electronic voting where the vote of each person is encrypted but the total number of votes is obtained by decryption. Homomorphic encryption is used in SMC (secure multi-party computation). According to Craig (2009) a fully homomorphic encryption is capable of evaluating random number of additions and multiplications enabling to compute every kind of operation on data that is encrypted. One of the major disadvantages of multiplicative homomorphic encryption method using RSA cryptosystem is that it is not semantically secure. Cloud performs computation on encrypted data. There is no means to find that the
computation result produced by the server is correct or not. Performance is one of the disadvantages as the computation on larger ciphertexts is slow.

B. Incremental Encryption:

The cloud service provider has full control over the entire computing infrastructure. Hence it is difficult to share data. Gansen et al.(2010) proposed that secure sharing of data (Figure 8) via cloud storage service provider can also be achieved by using progressive elliptic curve encryption (PECE) to control the integrity and confidentiality of data.

![Fig. 8: Secure Sharing over Cloud Storage](image)

The main idea of incremental encryption is to encrypt the data prior to its storage in cloud and while sharing with other authorized user the encrypted data is again re-encrypted with a different encryption key. Data is encrypted several times with diverse keys which makes key management a difficult task. The cipher text is finally decrypted by only one operation with a distinct key. For encrypting and decrypting data, by elliptic curve cryptography is employed. User shall access data only when the data owner issues credentials to the authorized user. Incremental cryptography does not suit all applications and moreover an incremental signature may reveal the information about the way the document was written which violates the privacy of the user.

C. Broadcast Encryption:

Fiat and Naor(1993) proposed broadcast encryption which allows delivering encrypted content over a broadcast channel allowing only qualified users to decrypt the content. These users can change in each broadcast emission. Message is encrypted by the broadcaster only for privileged set of users. Users in privileged set with his/her private key decrypts the broadcast. If the users external to this privileged set collude, they cannot get any data regarding the broadcast contents. Broadcast encryption can be used for sharing of files in encrypted file systems, email encryption for large mailing lists, and content protection in case of DVD.

Liu et al.(2013) proposed that secure data sharing in cloud is achieved by employing group signature as well as dynamic broadcast encryption techniques. Figure 9 shows company enables its staff to share file with the help of cloud.

![Fig. 9: System Model](image)

Group signature does not reveal the source within a set of legitimate users whereas group encryption does not reveal the decryptor within a set of legitimate receivers. Feng et al.(2011) discussed that group encryption is
also used to propose Multi-Party Non-Repudiation (MPNR) protocol for safe storage of data in cloud environment.

Kumbhare et al. (2011) proposed cryptonite architecture for data repository service in a secured manner makes use of digital signature deployed in file manager, for auditing and verification of data integrity uses audit manager and for distributing the signature keys use broadcast encryption and for searching in an encrypted file without decrypting it uses searchable encryption.

D. SSL (Secure Socket Layer) Encryption:

Sandeep Sood (2012) proposed a data security model using various techniques to provide data security in cloud. The data owner categorizes data based on three parameters namely confidentiality, integrity and availability and user assigns values to the data from which the SR (sensitive rating) is calculated. Before storing data in cloud, an index is built based on the keywords relating to the list of documents (Figure 10). Both data and the index are encrypted by 128-bit SSL encryption. Message authentication code (MAC) generated by owner or user is transmitted along with the data that is encrypted to cloud for data integrity checks.

**Fig. 10:** Client side encryption in cloud

In case of requests from users to retrieve a file from cloud (Figure 11), cloud verifies for user authentication by using digital signature. User requests by specifying keywords. Cloud responds the user by sending encrypted record of positions corresponding to the keywords from the index. User decrypts these entries by decryption key and again sends download request to cloud to retrieve the required file. Cloud sends the file and the user decrypt the file by decryption key. Data encrypted asymmetrically are hard to break than the data that is encrypted symmetrically. Similar key is utilized for encrypting and decrypting data.

**Fig. 11:** File retrieval process

E. Searchable Encryption:

The users with whom the data owners share their data may need to retrieve only certain data files which is accomplished by keyword-based search. Figure 12 depicts such a scenario. A ranked searchable encryption scheme can overcome the problems of retrieving the entire file. Instead it returns the corresponding files in a
ranked order based on a significance criteria. Since only the relevant files are retrieved, it reduces the cost of sending the entire file back if the search was unsuccessful in case of traditional searchable encryption and also reduces the search over the entire file. Inverted index is used to store the connections from these keywords to set of files containing the keywords which include the relevance score. This relevance score is calculated by a ranking function. Ranking done on the user side takes huge computation and post-processing overhead. It also takes two round-trip times for every request for search made by each client. Order-Preserving Symmetric Encryption (OPSE), an encryption scheme is used for efficient ranked search over encrypted data by providing the relevance scores and ensures ciphertexts preserve the order of the plaintexts is deterministic in nature.

![Fig. 12: Search over encrypted cloud data architecture](image)

But the range of cipher text is large and it needs to be fixed before. Key management is bit hard because of the usage of different encryption keys for different data. Wang et al. (2010) proposed One-to-many OSPE to reduce the amount of information leaked. Figure 13 shows the algorithm in detail. But, pre-knowledge of encryption techniques is necessary when a user tries to apply these techniques to cloud. There is no information regarding confidentiality and integrity of data, hence not suited for security.

![Fig. 13: One-to-many Order preserving mapping algorithm](image)

Searchable encryption is used to encrypt the index. But this searching may leak information to the service provider. In symmetric searchable encryption (SSE), party searching for data is the one who generated it. The advantages are efficiency and security. Its disadvantage lies in its functionality. The server may try to learn which document contains the data being searched for with the help of tokens. The major disadvantages of this scheme are the search time and the updates to the index. In asymmetric searchable encryption (ASE), both the parties involved in generating data and searching are different. Its advantage lies in functionality whereas the drawbacks include lack of efficiency and fragile security assurances. The token may be learnt easily. Efficient ASE (ESE) makes the time for search more efficient with the disadvantage of token being learnt. Multi-user SSE (mSSE) is single writer/many reader scheme. The user’s search privilege shall be added or revoked by owner.

The unique key is encrypted by attribute-based encryption. Users are given the secret keys built on set of attributes. When a file is encrypted with keys of the attributes, it will be decrypted only by the user who has all
the attributes. To check the correctness of data, data verifier uses proof of storage protocol. Client can run this protocol randomly with the server to ensure whether data is tampered by server or not. Data integrity is confirmed privately only by client or publicly by anybody who posses client’s public key. Major drawback of cryptographic cloud storage is the need to retrieve the entire data for decryption and integrity verification. These methods are employed for searching data that is in encrypted form but may enlarge the system’s intricacy.

All the security models discussed have been presented with the evaluation parameters in Table 4.

<table>
<thead>
<tr>
<th>Building Block</th>
<th>Authentication</th>
<th>Protection of Data</th>
<th>Integrity of Data</th>
<th>Scalability</th>
<th>Access of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homomorphic Encryption (Sun et al. (2013))</td>
<td>Yes</td>
<td>Partial</td>
<td>No</td>
<td>Fair</td>
<td>Semi-automated</td>
</tr>
<tr>
<td>Incremental Encryption (Gansan et al. (2010))</td>
<td>Yes</td>
<td>Partial</td>
<td>No</td>
<td>Poor</td>
<td>Semi-automated</td>
</tr>
<tr>
<td>Broadcast Encryption (Liu et al. (2010))</td>
<td>Yes</td>
<td>Complete</td>
<td>Yes</td>
<td>Fair</td>
<td>Automated</td>
</tr>
<tr>
<td>SSL Encryption (Sood (2012))</td>
<td>Yes</td>
<td>Partial</td>
<td>Yes</td>
<td>More</td>
<td>Automated</td>
</tr>
<tr>
<td>Searchable Encryption Wang et al. (2010)</td>
<td>No</td>
<td>Complete</td>
<td>Yes</td>
<td>More</td>
<td>Semi-automated</td>
</tr>
</tbody>
</table>

### VI. Authentication And Access Control In Cloud:

Yan et al. (2009) combined Role-based Access Control (RBAC) with Attribute-based Access control (ABAC) to provide protection to sensitive data of users in cloud. Users are provided or denied permissions using access control list (ACL). But, security policies are defined by the providers which limit the users from storing all types of data on cloud. Tian et al. (2009) proposed that re-encryption can provide flexible access control in cloud storage. Access control matrix is employed to store the access control policies. In case of attribute based encryption, the data loss will be minimal even if the storage of data is compromised. Usually for a user, the server provides access to files. But in case of ABE, access-control policy is associated with a user and his/her data. So, only the user with a right key can access the information. Access policy defines what kind of user can access the data involving the attributes. Su et al. (2012) discussed that ABE overcomes the disadvantages of traditional cryptographic methods by achieving one-to-many encryption.

A malicious user can change the ciphertext of one message the ciphertext of another message and can again transform into simple ciphertext with the help of transformation key. This malicious behavior the cloud cannot be detected by a user. So, ABE scheme with verifiability is important to ensure that the recovered plaintext from a transformed ciphertext is a correct one. Encryption and decryption of data is based on attributes of the user. ABE may also be employed to attain fine-grained sharing of data and better access control. Lai et al. (2013) proposed that ABE schemes are also employed with decryption being outsourced where the adversary cannot learn anything about the message that is encrypted. ABE scheme with secure outsourced decryption guaranteed verifiability of cloud’s transformation. Any user can get back the plaintext from the changed ciphertext.

Two kinds of attribute-based encryption schemes are (i) CP-ABE (Ciphertext Policy Attribute Based Encryption) and (ii) KP-ABE (Key Policy Attribute-Based Encryption). In CP-ABE, each client private key will be linked with set of attributes and each ciphertext is related with an access principle. The encryptor fixes the policy and decrypts the message that is encrypted. Usage of one public key and the master private key makes more restricted private keys. Reduces the necessity to process as well as store public key certificates as in conventional methods. In KP-ABE, each ciphertext will be labeled with set of features in addition these keys are related with the access policy. Access structure is used to control the ciphertext which the user can decrypt.

ABE schemes can resolve key escrow and user revocation problems and comprises of an authority, data owner and data user. According to the attributes, an authority generates keys; attributes of public key and master key are produced by an authority. Data will be encrypted by the data owner by means of public key as well as set of attributes. This encrypted data is decrypted by user’s private key sent from an authority and by this way a user can obtain the data he needs. Wan et al. (2012) proposed HASBE to attain scalability due to hierarchical arrangement and also allows access control of data in a fine-grained manner that is outsourced to cloud.

Barua et al. (2011) proposed that Patient Health Information (PHI) put in cloud reduces the burden of local storage. This method for secure cloud storage uses ciphertext-policy attribute-based encryption for access control and identity based encryption for communication of data in a secured manner and provides secure patient-centric access control (ESPAC) scheme. Figure 14 depicts that the major steps include PHI collection, secure data communication, processing of PHI and transferring PHI to cloud. The main focus is on confidentiality of data and access control.
Fig. 14: Major steps of proposed ESPAC scheme

Liu et al. (2011) presented that by using re-encryption key, the ciphertext encrypted by sender’s public key is converted to ciphertext encrypted by receiver’s public key. This allows only authorized parties to get the data without delay. Cloud servers do not wait for the commands to be received from data owner; rather they routinely re-encrypt this data based on the internal clock.

Grouping decryption keys into a single key is key aggregation. Unique key encrypts data in data sharing applications. To access certain files a data owner re-encrypts files with a new key and share that new key or can allow the user’s to access the original key. Computation of many files in re-encryption schemes is expensive. Key management problem arise if the user is given access to the original key. Chu et al. (2014) proposed a key aggregate system, a public-key encryption which is actually slight modification of broadcast encryption. Files are encrypted with unique public keys. The data owner computes an aggregate key for a set of shared file. This aggregate key decrypts files that belong to the set. Key management system combining both attribute-based encryption and re-encryption helps in securing mobile applications in cloud. Here encryption takes place with the help of group key, thereby reducing the burden on mobile data owner.

VII. Data Integrity Proofs In Cloud Computing (Class B Models):

The main advantage of cloud storage is that clients pay only based on their usage of resources. At the same time, a client needs to trust the third party services which may impose several security risks like deletion or corruption of data and leakage of data. So, client needs a way to check that the data is intact and stored securely in cloud without any corruption. These issues can be addressed with data integrity proof methods.

A. POR (Proof of Retrievability):

POR (Proof Of Retrievability) model allows the user to prove the correctness of data stored remotely without retrieving those. PORs are used to verify backed up files. POR scheme eliminated the need to download entire file in case of integrity verification. Juels et al. (2007) proposed a scheme that combines spot checking (for detecting large corruptions) and error-correcting code (for detecting small corruptions) to guarantee both ownership and file retrieval on archive systems. It detects illicit alterations to sections of file upon retrieval. POR is a challenge-response protocol wherein a user can challenge the server on which data owner stores data (Figure 15). Each client asks the server for POR. The server proves to the owner of the file that the file has not been corrupted. After which client’s data can be extracted from any prover that go by a confirmation test.

Fig. 15: POR system
The major phases of POR scheme is setup phase and verification phase. The user while storing a file in the server computes a hash value along with his/her private key. By this the user stores only his/her private key in the local storage and stores his data in remote server storage. In verification phase, the user challenges the server. The server generates a short response or proof upon receiving the challenge query based on user data file and the hash value is returned. Finally, the user verifies this hash value. Xu and Chang (2012) proposed the usage of keyed-hash function often used as message authentication code (MAC).

Shacham and Waters (2008) proposed sentinel-based POR scheme, a POR protocol encrypts the data file of the user and applies error-correcting codes to portions of the file to protect against corruption. After which check blocks are embedded randomly named as sentinels. The user tests the prover by indicating the locations of sentinels and asks that server to send back those related sentinel values. POR proof generated makes the user to understand whether the verifier has modified or deleted specific portion of file. The scheme uses efficient homomorphic authenticators based on pseudo-random functions (PRF) for private verifiability and BLS (Boneh-Lynn-Shacham) signatures for public verifiability. User fragments an erasure encoded file into blocks thereby generating a secret key and authentication value for each block. The block of file along with the authenticators is stored on server. Client executes POR protocol for data integrity. But it is efficient only for data that is static.

Computation problems arise in case of encrypting large data. Storage overhead occurs at the server by the recently added sentinels and also by error-correcting codes added with the data. Another disadvantage is large storage requirements on the prover and bandwidth consumption during checks. The insertion of sentinels and error-correcting codes do not support updates. And also the client can query only in a limited manner. Applications of POR to a distributed system remain unexplored since it uses the concept of file redundancy within a server for verification.

Bowers et al. (2009) proposed HAIL (High-Availability and Integrity Lay for Cloud Storage) that uses set of servers that are distributed to prove integrity and retrievability of file which is actually an extension of POR protocol. It improves the security of existing POR tools deployed on individual servers. HAIL provides protection only to static files but overcomes some of the limitations of using POR schemes.

**Fig. 16:** Encoded files with parity blocks added

Figure 16 shows that the main idea is to replicate each data file on many servers by using the basic principles of RAID. PORs are used by the clients to detect file corruption and these clients reallocate resources only in times of need. When a client detects a fault in the server with challenge-response protocol, the client by communicating with other servers recovers the corrupted share from cross-server redundancy and resets the faulty server with the correct share. In order to spread the file blocks across servers, HAIL uses dispersal code which in turn is based on hash functions and error-correcting codes. This may incur storage overhead at the server. The file blocks in the server are also encoded with error-correcting code for corruption. But, this may also cause storage overhead at the server.

**B. PDP (Provable Data Possession):**

Ateniese et al. (2007) proposed PDP (Provable Data Possession) model that gives a proof that the particular file is stored on a remote server but does not guarantee on file retrievability. The main goal is to detect the misbehavior of a server when it tries to modify or delete a fraction of file by the generated proof of possession. While storing data on a remote server, the user pre-calculates homomorphic verification labels for every file block and stores on server along with the file (Figure 17(a)). Client challenges server against a set of file blocks. Server makes a proof of possession build on the blocks which are questioned and their respective labels. The client verifies the response with a portion of metadata which is locally stored (Figure 17(b)). This scheme also eliminates the need for storing the local replica of the file on the client side. Before deletion, the client can even execute data possession challenge to ensure that the particular remote server had effectively stored this data. The
disadvantages may be the modular exponentiations included in the scheme, weaker assurance than a POR and the necessity to prove all the blocks for maximum assurance. O(1) costs for server and client computation time as well as for complexity measures also.

Fig. 17(a): Pre-processing and storage

Fig. 17(b): Verify server possession

Erway et al.(2009) proposed DPDP (Dynamic Provable Data Possession)[40] that widens the PDP model to sustain verifiable updates to outsourced data. It overcomes O(1) costs for computation times in case of PDP scheme used for storing static data by providing O(log n) costs. DPDP uses a novel version of authenticated dictionaries which make use of the rank information available at the nodes. Two kinds of data structures are used to check file integrity namely rank-based authenticated skip lists(Figure 18) and RSA trees.

Fig. 18: Rank based skip list example

Skip lists are just like doubly linked list of elements used mainly for dictionary implementation. Each node stores the rank of it which denotes number of nodes underneath them that can be reached from this node. An authenticated skip list has the hash value of the nodes in its path. So insertion, deletion or modification will only affect the nodes in the search path of the skip lists. The file along with the skip list is stored in the server and continues with the querying from the server side. It supports authenticated operations at block level and guarantees data possession of a data file and hierarchical file system. RSA tree is a dynamic authenticated data structure to which the rank information is added by storing it in the internal nodes. Blockless verification of data is involved but suffers from increased time for updates. Client can store massive data on remote servers in public cloud infrastructure and checks the integrity of data periodically.

Wang(2013) proposed PPDP (Proxy Provable Data Possession in Public Clouds) that delegates the task of data integrity at remote servers to some proxy. But the limitation is the reservation of ciphertext classes when number of ciphertexts increases. The data integrity proof with evaluation parameters is presented in Table 5.

<table>
<thead>
<tr>
<th>Type of Proof</th>
<th>Security Aspects</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof Of Retrievability[33]</td>
<td>Data security</td>
<td>Fair</td>
</tr>
<tr>
<td>Provable Data Possession[39]</td>
<td>Data integrity</td>
<td>More</td>
</tr>
</tbody>
</table>
VIII. Combination Of Encryption and Data Integrity Proof:

Popa et al. (2011) proposed CloudProof [42] which provides a protected storage system for boosting security over cloud. The model allows the client to identify breach of integrity and confidentiality and confirm the incident of breach to the third party. It also uses cryptographic tools to identify the misbehaving clouds. The major design principle is to move work to cloud as much as possible but should be able to verify it. Most of the SLAs guarantee only availability and there is no security guarantee in it. Security properties identified for secure storage are confidentiality, integrity, write-serializability and read freshness. Customer record is kept confidential (confidentiality), cannot be altered by illicit users (integrity), data remains consistent even after updates by authorized users (write-serializability) and the data is fresh as of the last update (Figure 19).

![Fig. 19: Read and Write Throughput](image)

Table 6: Evaluation of combination of Class A and Class B model

<table>
<thead>
<tr>
<th>Security Model</th>
<th>Protection of Data</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudProof[42]</td>
<td>Complete</td>
<td>Fair</td>
</tr>
</tbody>
</table>

Conclusion:

Cloud offers various services to the client and the client expects that these services are of greater benefit to them. Client stores his/her data in cloud to remove their burden of local storage. Data in cloud is stored in multiple locations which makes the need of strong security mechanisms to maintain the data safely. Data should be secured during transmission, storage and retrieval because data loss or leakage can cause both revenue and legal problems. As described in the paper, complete data protection is not achievable. Encryption techniques and proof of storage methods so far used in cloud are suffering from certain limitations. Unless there is a proper data security algorithm, data cannot be protected absolutely. This security algorithm should cater to all the problems occurring in cloud computing. Though there are lots of realistic difficulties concerning data security in cloud, the research is much concerned to draw a hybrid model that aims at these ideas and to present a practical solution. A proper hybrid model should be designed and deployed in cloud computing environment in order to keep customers' data safe.
The proposed hybrid model is based on HyperElliptic Curve Cryptography (HECC), Secure Hashing Algorithm (SHA) and hyperelliptic curve based digital signature algorithm. HECC is employed because it provides more data security compared to RSA and ECC by using minimal key size. To provide data integrity, SHA-3 algorithm is adopted as it takes minimal computation time and it allows maximum message size as input. Hyperelliptic Curve Digital Signature Algorithm (HECDSA) enables only authenticated users to transmit, store and retrieve data as it gives same level security compared to other DSA’s with minimal key size and signature size. This hybrid model will definitely provide data security, data integrity and user authentication in cloud computing.

REFERENCES


