Layer-based Privacy and Security Architecture for Cloud Data Sharing

Ishu Gupta, *Member, IEEE,* Niharika Singh, and Ashutosh Kumar Singh, *Senior Member, IEEE*

Original scientific paper

***Abstract*—The management of data while maintaining its utility and preservation of security scheme is a matter of concern for the cloud owner. In order to minimize the overhead at cloud service provider for applying security over each document and then transfer it to the client, we proposed a layered architecture. This approach maintains security of the sensitive document and privacy of its data sensitivity. To make a balance between data security and utility, the proposed approach categorizes the data according to its sensitivity. Perseverance of various categories require different algorithmic schemes. We set up a cloud distributed environment where data is categorized into four levels of sensitivity: public, confidential, secret, top secret and a different approach has been used to preserve the security at each level. At the most sensitive layers i.e. secret and top secret data, we made a provision to detect the faulty node that is responsible for data leakage. Finally, experimental analysis is carried out to analyze the performance of the layer-based approach. The experimental results show that time taken (in ms) in processing** 200 **documents of size** 20 **MB is** 437, 2239, 3142, 3900 **for public, confidential, secret, and top secret data respectively when the documents are distributed among distinct users, which proves the practicality of the proposed approach.**

***Index Terms*—Cloud Computing, Data Leakage, Data Privacy, Data Sensitivity, Information Security, Guilty Client.**

I. Introduction

In today's on growing world, there is a need to share the information of the organization among various entities such as employees, business partners, customers, etc. [1], [2], [3]. With the emergence of cloud computing technology, connectivity enabled by the Internet is accomplished to allow the users having the potential to utilize the distributed and scalable computing environments. But this information can be attained by unauthorized access while transmitting the data or it can be intentionally or unintentionally leaked by the receiving party, and then it can be misused by some malicious entities [4], [5]. It can cause a serious threat to the organization's goodwill and reputation [6], [7]. Due to this reason, data security and leakage detection have become critical challenges for any organization. There is a need of mechanism that can preserve the security of the data being shared and can detect the malicious entity causing data leakage.

Though, from the year 2013, number of cloud users are projected to be increased from 2.4 billion to 3.6 billion in the year 2018 [8]. Hence, by the time, data availability has gradually gotten a hike that needs security and privacy. Maintaining the privacy is important to protect the data from leakage [9]. According to a study, number of leaked sensitive data records had reached to 1.1 billion during the years 2011 to 2014. It has kept on increasing as the number of cloud users are increasing, also the malicious users [10]. Thus, to cope up with increasing cyberattacks, cybersecurity requires an approach that can manage, secure and locate the malicious agents and activities.

Conversely, limiting the sharing of information in order to maintain security results in reduced data utility which may affect the performance of the organization [11], [12]. Security mechanism is applied over the whole data while transferring it to the cloud and then to the user which incurs high computational costs. When stronger security mechanism is applied to the whole data, it reduces data utility, processing speed and increases the overhead. If, no security mechanism is applied, then security and privacy can be compromised and there can be chances of data leakage and data misuse. To minimize the overhead while maintaining the data security and data utility, we proposed a layer-based security and privacy architecture.

Traditional research has focused on transmitting the encrypted data from the owner to the cloud. We enhance the traditional architectural behavior and provide a layer-based architecture for securing the data that flows among three party system i.e. cloud, owner and client. The approach, we present in this paper, to preserve the data is related to maintain data utility issue while ensuring data security. Our solution to maintain privacy and security contributes in the following directions:

1) The paper proposes a layer-based privacy and security architecture to preserve the cloud data confidentiality, when the data is shared among multiple entities. In order to reduce the computational overhead of applying security mechanism over the whole data, the stored data is categorized as per its level of confidentiality and then appropriate level of security is applied when the data is retrieved. Data utilization and security requirements may be quite different for different data. To make a balance between information utility and protection, the layer-

Manuscript received September 10, 2018; revised February 04, 2019. Date of publication April 02, 2019. Date of current version June 3, 2019. The associate editor Prof. Toni Mastelić has been coordinating the review of this manuscript and approved it for publication.

Ishu Gupta and Ashutosh Kumar Singh are with the Department of Computer Applications, National Institute of Technology, Kurukshetra, India (e-mails: ishugupta23@gmail.com, ashutosh@nitkkr.ac.in).

Niharika Singh is with Universiti Teknologi PETRONAS, Malaysia (e-mail: niharika.academics@gmail.com).

Digital Object Identifier (DOI): 10.24138/jcomss.v15i2.617.

1845-6421/06/617 © 2019 CCIS

based mechanism classifies the data in four categories named as Public, Confidential, Secret, Top Secret. At each layer, a different integrated combination of multiple technologies is utilized to fulfill the privacy and security requirements as per the data sensitivity. Each successive layer provides stronger security in addition to the security at the previous layer. Watermarking technique is utilized in case of secret and top secret data to identify the leaker responsible for leaking the sensitive data. Furthermore, we adopt the message authentication scheme to verify the identified leaker in case of most critical data.

2) We evaluate the computation cost in terms of computation time required for processing the document at each layer. Furthermore, in the experimental evaluation, it has been represented that how the computation time can be reduced by effectively sharing the data.

The rest of the paper is organized as follows: Section II discusses the related work. Section III describes the threat model and the design goals of our system. Section IV highlights the proposed model which is followed by performance evaluation in section V that explains the experimental analysis and includes the results. Section VI discusses the conclusion of the work presented.

II. Related Work

In today's emerging world, various organizations are shifting their data to the cloud because of a long list of its advantages [13], [14]. Data storage analysis is done and is tabularized in Table I and Table II [15], [16], [17], [18]. Table I represents storage specifications of cloud data in percentage terms (%) for standard data (consisting both sensitive and non-sensitive data) resulting in the highest percentage of 'Relational' data type (with 34%). Furthermore, Table II specifies the storage specification for sensitive data only having maximum range of 73% customer data stored in the cloud.

As more and more data is being stored in the cloud and shared among the users, it requires resistant security services and leakage detection mechanism. The proposed solutions in this field are grouped into five categories, i.e. access control mechanism, cryptography, fingerprinting, probabilistic evaluation and watermarking. [19], [20], [21], [22], [23] provide the security and privacy to data through access control mechanism. In [19], coalitions have been formed among data owners for distributing the data in a secure manner. To ensure the controlled transfer of data in a distributed environment while preserving well defined policies, usage control enforcement systems are given in [20], [21]. [22], [23] provide the access control policies to secure the data in a cloud environment. Although this method can control the release of sensitive data and protect the information. But this method cannot stop an unauthorized access to obtain and misuse the data. Also, providing security through this method results in reduced data utility.

Another method proposed for maintaining the security and privacy of data stored in the cloud is based on cryptography. This method is used to preserve the data from unauthorized disclosure during transmission. The aim of the method is to make the data difficult to understand by malicious entities [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]. Kao et. al. [25] proposed uCloud which is a user-centric key management scheme. It safeguards the keys in a manner that specifically stores private keys on user's mobile devices and presents through 2D barcode images. Later, to control the information leakage rate, Qin et. al. proposed PKE based approach which was precisely more efficient than the other related proposals [26]. To achieve resource sharing and reduction in maintenance cost of specialized data center, third party is a better choice to outsource such services. Specifically, for secure sharing of Personal Health Records (PHR); [28], [33] proposed Ciphertext-Policy Attribute-Based Signcryption (CP-ABE). To provide collision resistance, unforgeability and CIA services, this CP-ABE combines the assets of encryption and digital signature. The combination of these two results in presenting an efficient scheme, but impractical due to high acquisition costs and performance limitations. Liang et al. proposed Ciphertext-Policy Attribute-Based Proxy Re-Encryption (CP-ABPRE) which is universally applicable for privacy in each type of network sharing applications [29]. This tackles the privacy problem efficiently by integrating dual system encryption technology with selective proof technique. In an attribute-based secure data sharing scheme with efficient revocation (EABDS) [31], attribute authority and key server generate the attribute secret keys of the user by adopting homomorphic encryption in addition with attribute based encryption to solve the key escrow problem. The approach prevents the entity from accessing the data alone. Wang et al. [36] proposed CP-ABE-ET scheme by adopting the concept of PKE-ET and CP-ABE techniques. Authors in [37], [39] proposed privacy-preserving reputation systems using secure multi-party cryptographic techniques for the evaluation of business entities trustworthiness and autonomous machines trustworthiness in the machine to machine network. These schemes ensure privacy, security and correctness and allow public verifiability without relying on a centralized trusted system. A homeomorphic cryptographic system is utilized in [38] for the protection of privacy. The design of a decentralized reputation aggregation system named "PrivBox" is reported that protects the privacy of the users without relying on any anonymous identities and trusted system. The proposed system has a small communication and computation overheads with the essential properties of decentralization and privacy-preservation. The cryptographic protocol based decentralized collaborative systems are proposed by Azad et. al. [34], [35] for the effective blocking of spammers who target multiple TSPs. The systems have evaluated the performance using the synthetic and real call detail records. To provide CIA services, Al-Haj et. al. [30] proposed two-crypto-based algorithms using whirlpool hash codes and internally generated symmetric keys. This results in reducing signal distortion and robustness against signal processing attacks. Although, the cryptographic method provides stronger security to protect the data, but the method does not give the guarantee that data can not be leaked by the receiving party, once the information is being passed to it. This method cannot protect the data when it is leaked to

TABLE I
STORAGE SPECIFICATIONS OF CLOUD DATA IN PERCENTAGE TERMS (%)

Data Type	Percentage	Example
Text	10%	UNICODE, ASCII etc.
Number	24%	Number System (base-10, base-2 etc.), Integers, Digits etc.
Image	15%	Jpg, png, gif etc.
Video	12%	Mp4 etc.
Audio	5%	Mp3 etc.
Relational	34%	MySQL, mSQL, JavaDB etc.

TABLE II
STORAGE SPECIFICATIONS OF CLOUD SENSITIVE DATA IN PERCENTAGE TERMS (%)

Data Type	Percentage	Example
Confidential Information (Text, Numbers, Images, Video, Audio)	15%	Legal investigations conducted by the University, Sealed bids etc.
Intellectual Properties (Text, Numbers)	4%	Copyrights, patents, trade-marks etc.
Customer Data (Relational)	73%	Name, Company, Address, E-mail, Contact etc. details stored in service centers
Health Records (Relational)	8%	Patient, Disease, Prescription etc.

some unauthorized access. Also, it cannot detect the leakage and the malicious entity who has leaked the data.

A content based fingerprinting approach is presented to preserve the information from unauthorized revelation [40], [41], [42]. Leakage is detected by extracting the patterns from the document and then matching these with the outgoing documents. The benefit of the scheme is that it does not reveal the entire sensitive document to the intermediate party, instead, only a set of keywords are exposed to semi-honest provider. But, the scheme can identify the leakages only that are caused deliberately by malicious entity. Another major disadvantage of the approach is that it becomes unable in detecting the leakage even when slight changes are performed in transmitting data.

To identify the guilty agent, [43], [44], [45] preferred to choose probabilistic approach which implements a variety of data distribution strategies. This improves distributor's chances of identifying leaker. The method does not depend upon alteration of the data, but the limitation of the probability method is that it cannot identify the exact leaker responsible for leaking the data.

For the proof of ownership and copyright protection, a special technique named watermarking has been presented by hiding the information [46], [47], [48]. Several algorithms have been proposed to hide the message in the data. Hiding the data using four different formats i.e. Text in Text, Image in Image, Image in Text and Text in Image requires specified watermarking technique. *When watermarking is called to process images and text bound as I, T respectively, it works over 2^n combinations; when $n = 2$ pairing $I+I, I+T, T+I$ and $T + T$.* In [49], [50], various traditional categories of this technique are explained. Singh et. al. [51] presented an algorithm for digital watermarking which is based on discrete wavelet transforms, Singular value Decomposition (SVD) and Discrete Cosine Transforms (DCT). Numerous known attacks are extensively tested and resulted robust and imperceptible performance in comparison to other existing methods. We use watermarking technique to trace the entity who has leaked the data. It can identify the exact malicious entity causing data leakage. This technique can not protect the data from unauthorized disclosure, when it is being transmitted to the user.

The aforementioned methods alone can't impose multiple security paradigms, therefore different technologies are innovatively combined to support multiple data security demands. To overcome the above shortcomings, we proposed a new layered based hybrid approach that categorizes the data according to its sensitivity and then applies the different security mechanism accordingly to obtain a better balance between data security and utility while reducing the overhead. At each layer, a different integration of cryptography, watermarking and hashing techniques are used according to the requirement by exploiting the benefits of these techniques. The key advantage of our proposed scheme is that it can be used for any data type for which watermarking scheme exists and it can be used with any existing watermarking technique without the need of any modification. The comparison of the research against the related work is summarized in Table III.

III. THREAT MODEL AND DESIGN GOALS

Our system model consists of three different entities O_{id}, Cl_{id} and C_{Sid} in the network that can be identified as follows:

- *Data Owner (O_{id}):* an entity, which has data to be stored in the cloud C_{Sid} and depends upon C_{Sid} for the maintenance and computation of data. Data owner can be either an enterprise or individual consumers.

TABLE III
COMPARISON OF VARIOUS ATTRIBUTES OF SECURITY AND PRIVACY APPROACHES

Approach	Attack Type	Security & Privacy Strength	Information Utility	Control Mechanism
Access Control Mechanism	During and After transmission	Strong	Low	Limiting the access of data
Cryptography	During transmission	Strong (with theoretical guarantee)	Low	Preserves CIA Triad credentials
Fingerprinting	During and After transmission	Compound strength choices Moderate to Strong	Moderate	Extracted patterns are matched with the outgoing data to maintain privacy and security of data
Probabilistic Evaluation	After transmission	Less Privacy; Strong security results	High	Works better for detection of faulty node during Data Leakage
Watermarking	During transmission (less participation) After transmission	Strong	High	Embedding information in documents and digital images

- *Client (Cl_{id}):* an entity, which retrieves the data shared by the owner O_{id} and can perform some task using it.
- *Cloud Server (C_{Sid}):* an entity that provides a high quality service using a number of servers $C_{S1}, C_{S2}, \ldots, C_{Sp}$ having considerable computation power and storage space.

Data flow among the three party system in our model is represented in Fig. 1. Our security model considers the most severe threat to the cloud data confidentiality when the data is shared among the clients. In this case, the model preserves the data confidentiality via securely sharing the cloud data and recognizing the malicious entity causing data leakage which may reduces the chances of occurring the data leakages. The entity Cl_{id} in our model is untrusted as, once the model provides the data to the client, it cannot guarantee that intended recipient will not leak the data. Also, when the data is received by the user, then no one can stop him from revealing it.

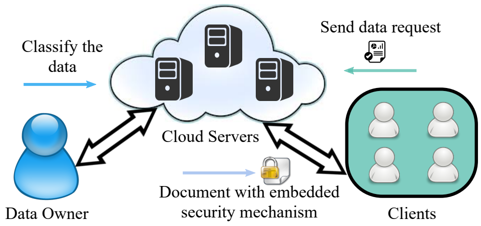


Fig. 1. System model.

The goal of the proposed method is to identify the guilty party in case of sensitive data leakage. The attacks that can occur in our model are: a) To lose the confidentiality of data by leaking it. b) To make the system disable in identifying the malicious entity. Therefore, our system considers the client Cl_{id} as an attacker who takes every possible step to leak the confidential information without being liable for their action. We can say that our security model considers the adversary to the malicious user that misuse the data and can publish it at unauthorized place. As the sender cannot trust the client, it embeds the unique code in the document every time when it provides the document to a client. However, we consider that the client tries to take out this identifying information in order to reveal the document safely without being caught.

The case that can arise, after embedding the information in the document, sender transfers the document to the receiving client, could keep a copy of this document with its embedded information, publish it and blame the receiving entity for it. There might be another possibility that it points to some other client by embedding its identifying information in the document, publish the resulting document without even transferring it to the receiving client. A different possible case that can arise is the refusal of the allegation. The guilty client can argue that he is innocent and blamed by the sending party. Our system requires the following properties that we expect from our protocol to fulfill and tolerate the failures with negligible probabilities only.

1) *Accuracy:* The guilty entity can be detected when both the sender and the receiver follow the protocol specification accurately and reveal their version of document only.
2) *No framing:* The sender cannot frame the receiving entities for its own leakage.
3) *No refusal:* If any document is published by the receiving entity then he could be provably involved in the leakage.
4) *Collusion resistance:* We also expect collusion resistance from our model i.e. it should be capable of bearing small number of colluding attackers.

The model considers the data owner O_{id} as trusted party. It believes, as the data owner is the first person who concern about the confidentiality of data so O_{id} cannot leak the data. As considerable research work has been done for cloud data security [2], [12], [13], [14], [25], [27], [28], [29], [32], [33], so we account the entity C_{Sid} is trusted in our model. Our model assumes that C_{Sid} follows all the security protocol, hence secures the critical data and can't leak it. User is an untrusted party and we have to secure the information from this entity. Transferring the documents to the clients involving untrusted entities is the pivotal phase of our model.

We also assume the reliable communication links among three party cloud systems. Data owner is required to define the access structure to specify which file can be accessed by which user. Data is provided to the user only-

- If the user id and password is correct.
- If the user malicious record is satisfied by checking the previous record of the user maintained in the database.
- If the user has permission to access the requested file.

Users should be provided that data only for which they are allowed to access without being access to unauthorized data. Our system also requires that users should not be capable to access the cloud data when their privileges have been revoked.

IV. Proposed Work

A. *Definitions and Mathematical Background*

To explain how the architecture works, the following definitions and assumption are needed.

1) Data Arrangement: In our model, three-party background of the cloud are denoted with O_{id}, Cl_{id}, C_{Sid} where the identifiers infer Data owner ID, client ID and cloud server ID respectively. Data owner O_{id} may choose cloud ID k that varies from 1 to p i.e. $k \in 1 - p$ to upload filtered data over a set of distributed cloud servers $C_{S1}, C_{S2}, \ldots, C_{Sp}$.

Definition 1: Let D is the data of various forms as given in (1). Data D is distributed among m clients $Cl_{id} = Cl_1, Cl_2, \ldots, Cl_m$ on their demand with the expectation of not leaking the data at some unauthorized place. Our goal is to secure data D_i $(i = 1, 2, \ldots, n)$ from any malicious entity and to preserve its confidentiality.

$$D = D_1 \cup D_2 \cup \ldots \cup D_n = \bigcup_{i=1}^{n} D_i \qquad (1)$$

Definition 2: If the data item D_i; $i \in \{1, 2, \ldots, n\}$ is found at some unauthorized place, then it is supposed to be leaked content. The client Cl_{id}; $id \in \{1, 2, \ldots, m\}$ responsible for leaking the data contents is named as guilty client G_C.

2) Modular Elements and Components: In our model, we assume a setting where distributor monitors a set of resources. Once an event triggers to transport data from owner organization to cloud, it needs to filter the data according to its sensitivity.

Assumption 1: We assume that O_{id} defines data sensitivity category for $D_i \in D$ $(i = 1, 2, \ldots, n)$ before sharing it with the cloud servers and stores the information in the server directory.

Data is collected from various resources and then filtered according to its sensitive category by O_{id}. The desired data accessibility between owner and cloud employs an attribute vector $\hat{v}$ that frames several important properties of data item $D_i \in D$ $(i = 1, 2, \ldots, n)$. Let vector is denoted as $\hat{v} = (s_i, z_i, \mu_i)$. Here, s_i represents sensitivity of data item, z_i is the term defining size of the document. The term μ_i indicates server ID where data is to be stored. Finally, filtered data is sent to the cloud where data is distributively stored on different servers and a server directory is maintained to store meta data $\hat{v}$ of D_i $(i = 1, 2, \ldots, n)$.

Definition 3: Let Cl_{id} requests D_i, server directory specifies the server ID μ_i containing D_i and the category s_i of D_i. The query is sent to the cloud server $C_{S\mu_i}$ and the algorithm is applied for zero, single, double and triple layer depending upon data sensitivity category s_i.

The parameters are passed by Cl_{id} to demand particular data set or document from data distributor. Client gets the data with specific layered algorithm depending upon the category of the requested data. If zero layer is denoted as $\boxed{l_z}$ than for next layer l_s, encryption q is applied by considering $\boxed{l_z}$ as a base frame producing $l_s = \boxed{\boxed{l_z}\ q}$. Similarly, the pattern is followed for other two layers as represented in Fig. 2.

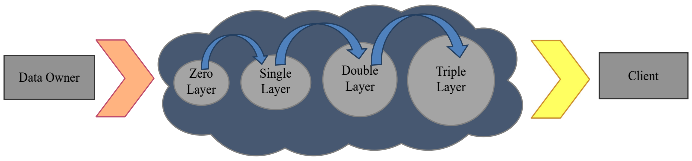


Fig. 2. Process of layered based approach.

Definition 4: To proceed with the technicalities of our model, the process utilizes the following varieties of keys in the layer-based architecture. a) Secret Key K_S to encrypt a document using AES-256 and client ID using AES-128 algorithm b) Public key K_P which encrypts the secret key K_S using RSA. Doing this reduces complexity as we are encrypting the key only rather than encrypting the whole document with this algorithm c) Private Key K_{PV} is used to decrypt the secret key K_S d) Encrypted key K_E is defined as the Secret key K_S encrypted by RSA Public Key K_P.

B. *Architecture Model*

The notations used in the paper are summarized in Table IV. Collected data is divided into ψ logically separated categories. This division is based upon data sensitivity measure. On a standard sensitive measure scale, we take $\psi = 4$ which are specified as Public $(\mathcal{P})$, Confidential $(\mathcal{C})$, Secret $(\mathcal{S})$ and Top Secret $(\mathcal{TS})$. These documents are then shared with the cloud distributor and is assigned a cloud server ID C_{Sid}. Fig. 3 explains the block diagram to define the architecture flow step by step. Client Cl_{id} sends request query req to retrieve data stored in a cloud distributed environment. Processing data from owner to client includes layered approach according to sensitivity category of the document i.e. Public, Confidential, Secret and Top Secret. Accessing data through specified category $(\mathcal{P}, \mathcal{C}, \mathcal{S}, \mathcal{TS})$ applies a different scientific approach depending upon data sensitivity. Sensitivity of data increases, as we move from category $\mathcal{P}$ to $\mathcal{TS}$ which requires stronger security to protect the data in comparison to the mechanism at previous layer. When we move from layer zero to layer three, each layer provides enhanced security in addition to the previous layer by strengthening the security mechanism. The proposed framework reduces the overhead and the computational cost of performing operation on the whole data.

For the Public Data $(\mathcal{P})$, cloud provides a uniform log-in registration whose credentials are given by cloud itself. For the Confidential Data $(\mathcal{C})$, cryptography is applied to convert data from plain text to cipher text. Secret Data $(\mathcal{S})$ is secured by using cryptography and watermarking algorithms that produce Watermarked Crypto-Document (WCD). Then, Top Secret

TABLE IV
NOTATIONS

Symbols	Description
m	number of clients
n	number of data objects
p	number of cloud servers
O_{id}	data owner
$Cl_{id}\ (id = 1, 2, \ldots, m)$	clients
$C_{Sid}\ (id = 1, 2, \ldots, p)$	cloud servers
D	data owned by O_{id}
$D_i\ (i = 1, 2, \ldots, n)$	data objects
G_C	guilty client
s_i	sensitivity category of the document
z_i	size of the document
μ_i	server ID where the document D_i is stored
K_P	public key
K_{PV}	private key
K_S	secret key
K_E	encrypted key
ψ	number of data category
$\mathcal{P}$	public data
$\mathcal{C}$	confidential data
$\mathcal{S}$	secret data
$\mathcal{TS}$	top secret data
SD	simple document
CD	cryptographic document
WCD	watermarked crypto document
$AWCD$	authenticated watermarked crypto document
WD	watermarked document
req	request query
cat	categorization
$E(Cl_{id})$	encrypted client ID
AD	authenticated document
AWD	authenticated watermarked document
C	cipher text
P	IP address of client
H	embedded hash code
H'	hash code generated from the extracted Cl_{id}

($\mathcal{TS}$) inherits the scheme of secret data along with authentication technique over the document. This technique generates message authentication code and after the process, it produces Authenticated Watermarked Crypto-Document ($AWCD$). If we have the case of getting encrypted data from the owner at that time there is no need to apply the encryption again on the data. After the process, data is sent to the client and is given a one-time password along with the document so that the document would be accessed by single machine only. The technical model description of the architecture is given in the following division.

C. Layered based Approach

In order to see, how our model constraints interact in distributed environment, in this division, we study detailed scenario. In subdivisions we are explaining how security and privacy is preserved at each layer.

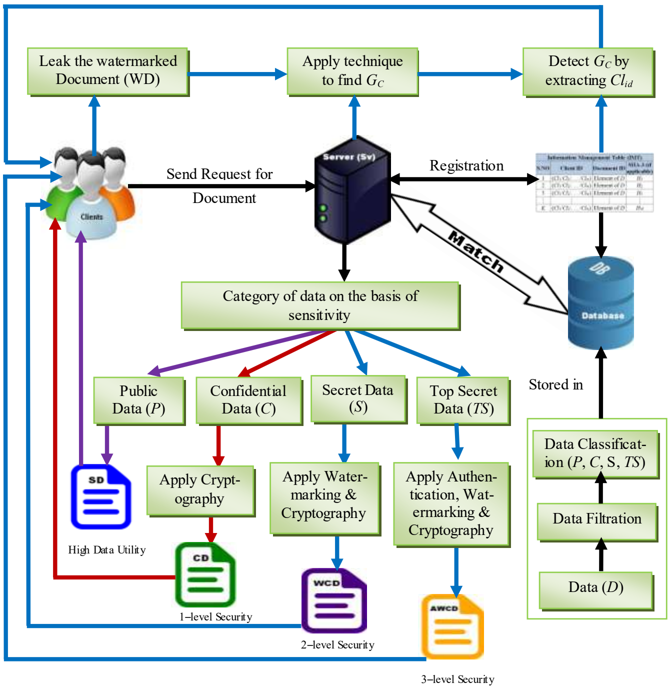

Leak the watermarked Document (WD)
Apply technique to find G_C
Detect G_C by extracting Cl_{id}
Clients
Send Request for Document
Server (Sv)
Registration
Match
DB
Database
Category of data on the basis of sensitivity
Public Data (P)
Confidential Data (C)
Secret Data (S)
Top Secret Data (TS)
Stored in
SD
High Data Utility
Apply Cryptography
Apply Watermarking & Cryptography
Apply Authentication, Watermarking & Cryptography
Data Classification (P, C, S, TS)
Data Filtration
Data (D)
CD
1–level Security
WCD
2–level Security
AWCD
3–level Security

Fig. 3. Architecture flow block diagram.

1) Policy/Architecture Composition: Our model includes m number of authorized clients whose authorization table is associated with the owner-cloud agreement and is stored at cloud server C_{Sid}. When a client generates request query req to C_{Sid} for some document D_i, server accepts the request. It checks whether the client is an authorized entity and applies searching algorithm to check the availability of data.

If the requested document is found, then server combines the matched server ID (where the document is stored) and document ID (requested document). It generates the client ID (client who have requested the document) if it does not exist and then inserts all these information (server ID, document ID and client ID) along with the hash code generated for client ID in the Information Management Table stored in the cloud server. The sensitivity category of the requested document is identified whether $s_i \in \{\mathcal{P}, \mathcal{C}, \mathcal{S}, \mathcal{TS}\}$ from the server directory. Data is transferred to the clients by generating the authenticated data according to ψ^{th} sensitivity level. Algorithm 1 presents the layered wise strategy applied for security and privacy.

2) Layer-wise Linearity/ Security and Privacy Cover: As the value of ψ^{th} level increases, the following security layers would be applied accordingly:

Zero Layer: For $\psi = 1$ (Public data $\mathcal{P}$), either no security or login authentication system is installed, this allows clients to use data as open access records. A uniform log-in system is provided by the cloud server C_{Sid}. If, client enters the correct user id and password, it is declared as authenticated entity. Then, Simple document SD is transferred to the client resulting in high data utility.

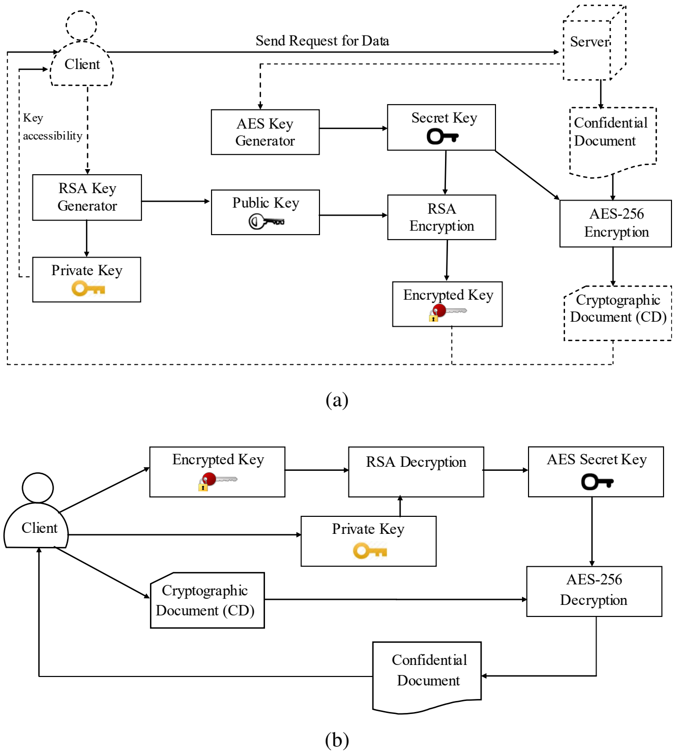
Single Layer: Aimed at $\psi = 2$, confidential data would be encrypted initiating single layer security of cryptography technique. Accessible data belonging to this category, enciphers

Algorithm 1 Layered Based Security Mechanism

```
Input: Data request by any client
Output: Generated authenticated document to be transferred
    to client
begin
    $D_i \leftarrow D$ where $i = 1, 2, \ldots, n$
    $\psi \leftarrow cat(D)$
    $D_i \in \psi \; \forall \; i = 1, 2, \ldots, n$
    $C_k \rightarrow req(D_i)$ where $i \in \{1, 2, \ldots, n\}$ & $k \in \{1, 2, \ldots, m\}$
    switch $(s_i)$ do // Additional parameters may be used
        case 1 : $\mathcal{P}$
            $SD \leftarrow \mathcal{P}$
        case 2 : $\mathcal{C}$
            $CD \leftarrow$ AES-256('$K_S$','$\mathcal{C}$')
            $K_E \leftarrow$ RSA('$K_P$','$K_S$')
        case 3 : $\mathcal{S}$
            $E(Cl_{id}) \leftarrow$ AES-128('$K_S$', '$Cl_{id}$')
            $WD \leftarrow$ DWT('$\mathcal{S}$', '$E(Cl_{id})$')
            //DWT – Digital Watermarking Technique
            $WCD \leftarrow$ AES-256('$K_S$', 'WD')
            $K_E \leftarrow$ RSA('$K_P$','$K_S$')
        case 4 : $\mathcal{TS}$
            $AD \leftarrow$ SHA-3($\mathcal{TS}$)
            $E(Cl_{id}) \leftarrow$ AES-128('$K_S$', '$Cl_{id}$')
            $AWD \leftarrow$ DWT('AD', '$E(Cl_{id})$')
            $AWCD \leftarrow$ AES-256('$K_S$', 'AWD')
            $K_E \leftarrow$ RSA('$K_P$','$K_S$')
    $C_k \leftarrow Transfer(D_i)$ for $k \in \{1, 2, \ldots, m\}$
end
```

and deciphers dynamically between three party distributed environment (owner, cloud, client). This requires to work over two processes enciphering $E(M) = C$ and deciphering $D\big(E(M)\big) = M$ to get cipher text $C = E(M)$ and plain text $M = D(C)$ respectively. It needs to use standard algorithms to encipher document D_i and to encrypt cryptographic keys. To reduce the complexity, we used AES-256 for encryption of the document. Furthermore, the secret key K_S of AES-256 is encrypted using RSA to provide stronger security. RSA encryption algorithm generates private and public keys K_{PV}, K_P respectively by selecting two prime numbers and then applied over generated K_S. The process results in cryptographic document CD which is transferred to the client and then decrypted by it. See Fig. 4(a) and 4(b), where Fig. 4(a) explains how confidential data records are encrypted and Fig. 4(b) explains the decryption process using three keys (secret key K_S, private key K_{PV} and encrypted key K_E).

Double Layer: If the data sensitivity of D_i falls $\psi = 3$, on the scale of sensitivity measure, it requires a stronger security. At this layer, we provide the mechanism for data leakage detection and leaker identification responsible for leaking the data by considering previous layer as the base layer. We apply embedding algorithm by adding watermarked logo in the document which is transfered to the client. The extraction algorithm detects the guilty client G_C. The security mechanism applied at double layer is represented in Fig. 5(a) and 5(b).



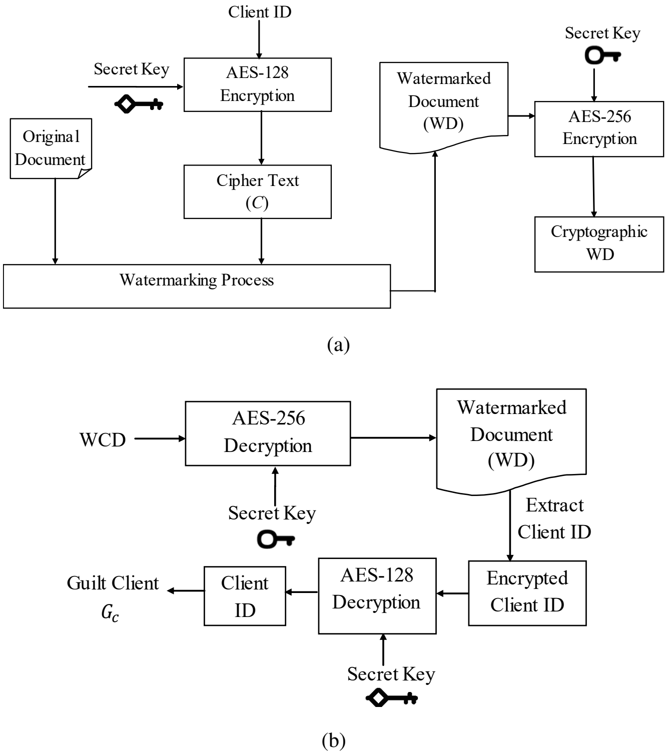
(a)

(b)

Fig. 4. (a) Encryption process of confidential data at single layer (b) Decryption process of confidential data at single layer.

At this stage, when Cl_{id} calls for data D_i, client IP address is captured and is stored in Information Management Table (IMT). Let P be the IP of Cl_{id}, is encrypted using K_S to get cipher data $C = E(P)$. This encoded IP is embedded in the logo of the client's organization and then the document is encrypted, results in watermarked cryptographic document WCD as shown in Fig. 5(a). This WCD is transferred to client Cl_{id} and decrypted by it, to obtain the watermarked document WD. In case of data leakage, when document is found at some unauthorized place, client ID $(P = E(C))$ is extracted from the document as shown in Fig. 5(b) and the guilty client G_C is identified. The steps for embedding and extraction of the watermark is explained in Algorithm 2 and 3 respectively.

Triple Layer: This layer inherits the concept of layer 2 along with authentication technique. If data belongs to $\psi = 4$ category specifying Top Secret $\mathcal{TS}$ then, message authentication code generated via hashing technique is called to verify the leaker. Stepping to Top Secret level $(\psi = 4)$, due to security measure scale, SHA-3 for message authentication is used in addition with AES-128. We encrypted the client ID Cl_{id} using AES-128 generating $E(Cl_{id})$ or C and then hash value H of Cl_{id} is calculated using SHA-3 with 512 bits. Hash code H is embedded along with cipher text C in the document and then resulting document is forwarded to client by encrypting it as represented in Fig. 6(a). On receiving the document, if the client reveals it in the unauthorized hands, then detection mechanism is applied. The information H and



(a)

(b)

Fig. 5. (a) Watermark embedding process at double layer (b) Watermark extraction and guilty client identification process at double layer.

$E(Cl_{id})$ is extracted to identify the leaker and verified using authenticated code as shown in Fig. 6(b). H' is calculated using Cl_{id} which is obtained by decrypting $E(Cl_{id})$. If H and H' values are equal, then extracted Cl_{id} is matched with the client ID stored in IMT and guilty client G_C is detected and verified.

This layer provides a hybrid approach to detect and prevent data leakage. Specifically, data leakage prevention is done by assigning one-time password (OTP) for each document, when it is downloaded by the client. Data D_i is accessible for single client ID only. When C_{Sid} sends data to Cl_{id}, it requires a code generated by data distributor which has accessibility limit of time $\tilde{t}$. If the time limit is crossed, the request query req for D_i is aborted. Though, Cl_{id} is evaluated by matching mechanism and guilty client can be captured in the track list of distributor.

Remark: The watermarking technique used is robust enough that fulfills our requirements and it becomes typical for the client to remove the embedded information from the document. Any other robust cryptography, hashing and watermarking schemes that fulfill our system requirements can be used.

V. Performance Analysis

In this section, we present system and experimental setup that includes overall machine settings. Then, we present standard benchmarks that help in evaluating performance and implementing our architecture.

Algorithm 2 Embedding Algorithm

```
Input: Original image $L_o(X,Y)$, Information to be embedded $I_m$
Output: Watermarked image $L_{oW}(X,Y)$
begin
    Convert $I_m$ in ASCII format denoted as $A(I_m)$.
    Read $A(I_m)$ and convert it in bit stream of $k$-bits as $B_s$.
    Read $k$-bit $B_s$, $(B_s = 1-k)$ to be hidden and $L_o$ of size $X \times Y$.
    Generate $k$ different long-tailed distribution based $PN$-sequences $P_{Ns}, (P_{Ns} = 1-k)$ of length 22 (for 22 mid-band DCT coefficients) using shape parameter $\mu$ as secret key and scale parameter $\nu$ to reset the Random Number Generator $R_{NG}$.
    Transform original image $L_o$ using $8 \times 8$ blocks 2D-DCT.
    For $B_s = 1-k$, hide the $B_s^{th}$ bit and modulate the $B_s^{th}$ DCT block of $L_o$ using Eqs. (2a) and (2b) respectively for a '0' or a '1' bit.

    $L_{oW}(i,j) = \begin{cases} L_o(i,j) + Q\,W_s(i,j), & \text{if } i,j \in F_{\cdot m}, \\ L_o(i,j), & \text{if } i,j \notin F_{\cdot m}, \end{cases}$    (2a)

    $L_{oW}(i,j) = \begin{cases} L_o(i,j) - Q\,W_s(i,j), & \text{if } i,j \in F_{\cdot m}, \\ L_o(i,j), & \text{if } i,j \notin F_{\cdot m}, \end{cases}$    (2b)

    Where $I_m-$ Information to be embedded; $Q-$ Gain Factor is used to specify the strength of embedded $I_m$; $W_s-$ Appropriate Pseudo Random Noise Sequence $(PRNS)$ on the $B_s^{th}$ hidden bit; $L_o(i,j) - 8 \times 8$ DCT block of original image $L_o$; $L_{oW}(i,j)-$ Corresponding marked DCT block.
    Inverse transform each of the marked DCT blocks, $L_{oW}(i,j)$ using $8 \times 8$ block inverse 2D-DCT to get the final watermarked image $L_{oW}(X,Y)$.
end
```

A. Experiment Settings

The experiments were performed on the machines equipped with an Intel ®core ™i7-2600 CPU @ 3.6 GHz having 8 GB RAM. Implementation is performed using 4 machines. We used these machines as clients and servers. Our prototype system is implemented in ASP .NET using language C# and SQL server 2014 on Visual Studio 2015 framework. Combination of AES and RSA is used for encrypting the data. We construct required hash functions for guilty client G_C authentication by using robust SHA 3 512 bit technique. For the dataset variation, we use benchmarks that are explained in following subsection.

B. Benchmark

In our experiments, we used SherWeb benchmarks and cloud servers to establish a virtualized atmosphere. In a few submodules, to collect datasets, we use authenticated open data available on data.gov and data.gov.in. The benchmark, we designed for our architecture is dealing with medical data

Algorithm 3 Extraction Algorithm

```
Input: Watermarked image $L_{oW}(X,Y)$
Output: $I_m$
begin
Read watermarked image $L_{oW}(X,Y)$.
Generate $k$ $PN$-sequences of length 22 (for 22 mid-band DCT coefficients) after resetting $R_{NG}$ using the same secret key $\mu$ as in watermark embedding process.
Transform watermarked image $L_{oW}(X,Y)$ using $8 \times 8$ blocks 2D-DCT.
Generate one dimensional array of size $k$ denoted as $corr(k)$.
For $B_s^{th} = 1 - k$, calculate the correlation between the mid band coefficients of the $B_s^{th}$ $PN$-sequence and the $B_s^{th}$ block and store this value in $corr(s)$.
Calculate the average of all the values stored in the array $corr(k)$.
Extract the $B_s^{th}$ hidden bit $\chi_s$ using the Eq. (3)
    $\chi_s = \begin{cases} 0 & \text{if } corr(s) > average\big(corr(k)\big) \\ 1 & \text{if } corr(s) \leq average\big(corr(k)\big) \end{cases}$    (3)
Rearrange the extracted bits $B_s$.
Regenerate $A(I_m)$ and Convert it to obtain $I_m$.
end
```

records. These records are varied over two attributes: size of file and type of data. Our proposed model is applicable to every type of data for which robust watermarking scheme exists, but in experimental scenario, we consider the documents in the form of text and image such as .doc, .xls, .pdf, .jpg, .png files etc. Although, we use text and image files only, the same mechanism can be applied for other data types.

We run the processing operations in a 'client/server'configuration. We place database systems and processing rules on different machines that are interconnected through a network. Each implementation is processed in its defined rules and regulations of layered modules to provide concurrent accesses.

C. Results Evaluation

We evaluate the computation time at each layer with respect to different parameters to analyze the performance of the proposed approach. We do not analyze the transmission of data, instead the computation cost of processing the document at every layer. We assess analytical results on the basis of filtering medical datasets out of multiple dataset streams. Medical datasets are universally accounted as most sensitive in nature. Various legal laws and regulation are aspired to control the misuse of EMRs (Electronic Medical Records). The orientation of different layered operations is done in section IV. Mainly 7 operations are applied that are: 1. Time utilized during cloud database search 2. Check client ID generation (Generate if required) 3. Perform required encryption process on Cl_{id} 4. Calculating hash value of Cl_{id} for authentication purpose 5. Embedding watermark for malicious agent identification 6. Perform document encryption 7. Secret key K_S encryption using K_P. We are taking $n = 200$ documents for experimental evaluation. We conduct a comparative study of time consumed in performing the required operations at various layers by varying a) Number of documents b) Size of documents c) Number of users.

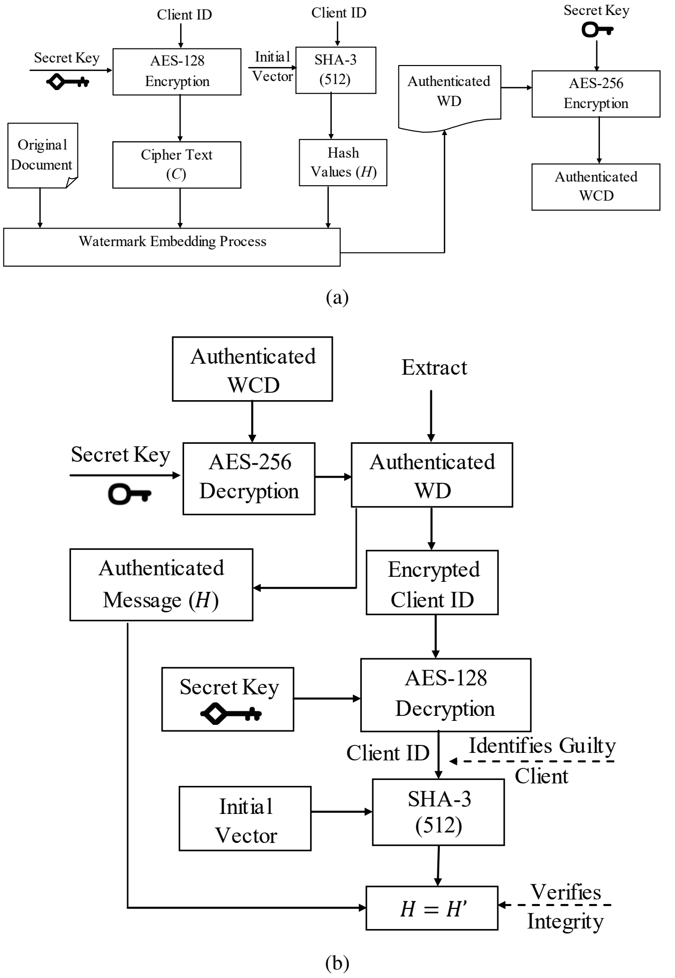
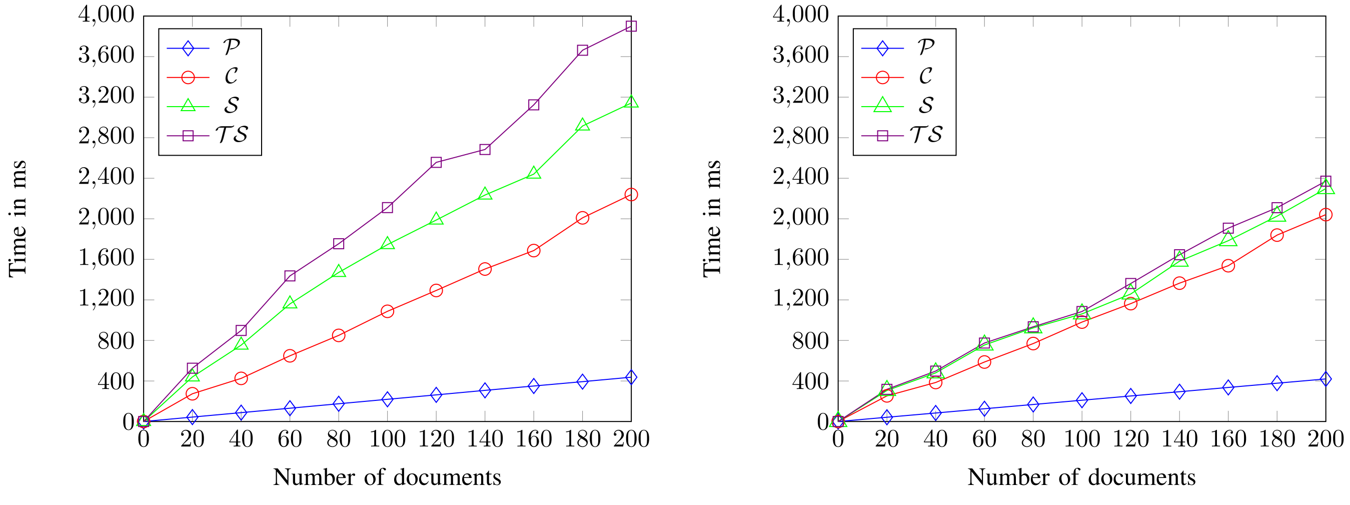


Fig. 6. (a) Embedding process in triple layer case (b) Guilty client identification process at triple layer.

In our first experimental scenario, we consider the size of documents as 20 MB. Table V represents the time taken by each operation for one document ($\mathcal{P}, \mathcal{C}, \mathcal{S}, \mathcal{TS}$). We see that the results sum up in very less time. For public ($\mathcal{P}$) document, we simply apply one time password facility for user authentication. It includes time for database search and client ID generation. We observe that there is a insignificant difference among each of the operations being performed. For the database searching, it takes 2.1 ms and during client ID generation, it requires 0.1 ms. There is another case when ID is not required to be generated that takes 0.08 ms in searching the client ID. Confidential data ($\mathcal{C}$) includes encryption time of data and secret key encryption in addition to the operation at layer Zero. Secret data ($\mathcal{S}$) involves encryption of client ID and watermark embedding in addition to the operation at layer one. The most sensitive data protecting layer considered as top secret includes calculation of hash value in addition to the previous operations. It takes 25.5 ms in processing one

TABLE V
TIME TAKEN (IN MS) BY EACH OPERATION AT VARIOUS LAYERS

Sequence of operation	Operation name	Public document	Confidential document	Secret document	Top Secret document
1	Cloud database search	2.1	2.1	2.1	2.1
2	Client ID generation	0.1	0.1	0.1	0.1
3	Performing required encryption process on Cl_{id}	NA	NA	0.77	0.77
4	Calculating hash value of Cl_{id} for authentication purpose	NA	NA	NA	0.75
5	Embedding watermark for leaky agent identification	NA	NA	8.6	11.3
6	Document encryption	NA	9.1	9.4	9.5
7	Secret key encryption	NA	0.98	0.98	0.98



(a)

(b)

Fig. 7. Processing time comparison on the basis of variety of documents (a) to distinct user (b) to single user.

document. The time distribution for processing each operation is given in Table V.

The stats mentioned in Fig. 7 gives the comparative view of the time consumed by each layer to process equal number of documents when the documents are provided to distinct users and single user in Fig. 7(a) and 7(b) respectively. We observe that as the sensitivity increases, the time consumed in processing the document increases. We also observe that the computation time reduces at each layer, when documents are provided to single user in Fig. 7(b) as compare to other case in Fig. 7(a). The reason can be explained as most of the operation such as Cl_{id} generation, Cl_{id} encryption, hashing on Cl_{id}, watermark embedding and encryption of K_S using K_P becomes constant.

Our propose scheme is focused to reduce the overhead by sharing the load. In Fig. 8, we shared the processing among 4 cloud servers symbolized as $\{C_{Sid} || C_{Sid}| = 1, 2, 4\}$. When the load is shared on single server, it takes time for accessing each task consecutively. When the load is shared among 2 servers, work is performed in parallel. On one server zero and triple layer tasks are executed and on the other, tasks of single and double layers are performed. Both have the same load to be shared. Practical evaluation reflects ups and downs, but at each instant, time is gradually increasing. We observe that overall, the computation time decreases when the load is shared among 2 servers as compare to the single server. Furthermore, when the load is shared among 4 servers, these servers perform work in parallel for each layer task execution. We find that balancing the overhead among 4 servers progressively decreases the processing time and gives efficient results.

In our experiments, comparison among various layers is done in three ways: a) On the basis of variety of documents (as represented earlier) b) On varying size of documents. c) On varying number of users. Time taken in processing by varying the size of one document in megabytes (MBs) is represented in Fig. 9. It involves the working of each layer specified in the architecture. In case of public document, processing is done for client ID generation and database searching, which is independent of document size variation. This results in horizontal line unlike other categories document processing. Similarly, when we are providing same document of size 20 MB to variety of users as shown in Fig. 10, it helps in reducing computation time. As database searching time and encryption task would be administered for single time only.

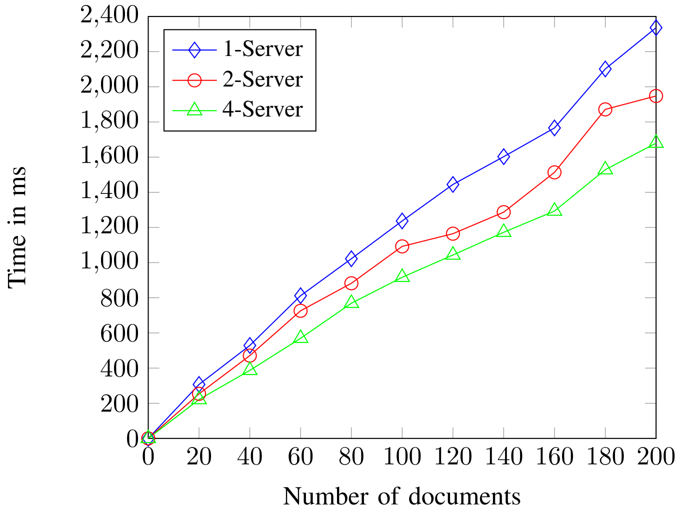


Fig. 8. Comparison of balancing task overheads among different servers.

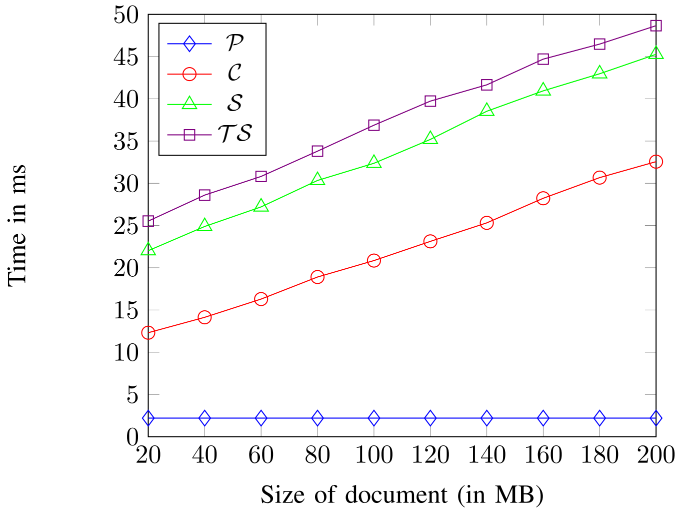


Fig. 9. Processing time comparison on the basis of variable size of documents.

D. Computation Complexity

We present the computation costs in terms of the computation time required for processing the document at each layer. The computation cost is $\mathcal{O}(1)$ at layer 0, while it is $\mathcal{O}(z_i)$ for the layers 1, 2 and 3 where z_i is the size of document. At layer 0, computation time stays constant independent of the size of document. The execution time in processing the document is linear and increases slowly with respect to document size at layers 1, 2 and 3, as most of the operation performed remains consistent independent of document size. Furthermore, the computation time in processing the multiple documents can be minimized by sharing the data in effective manner as depicted in 7(b) and 10.

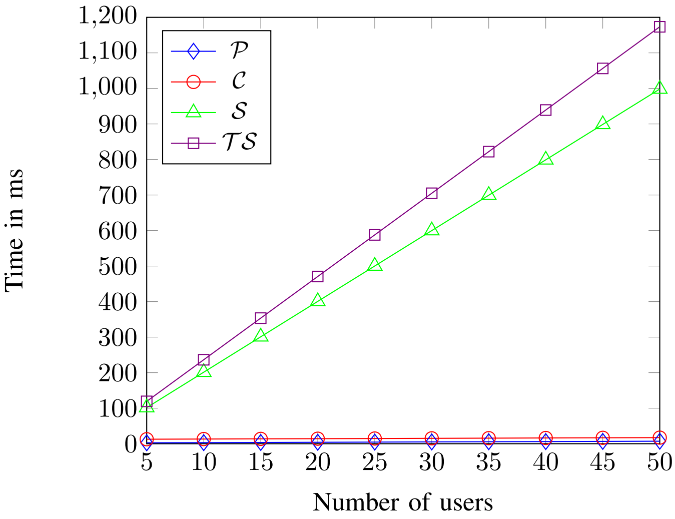


Fig. 10. Processing time comparison when same type of document is shared with variety of users.

1) Communication Overhead: As per the model consideration, there is no communication overhead involved at layer 0. Assuming 128-bits & 1024-bits security level for AES and RSA encryption, the communication overhead at layers 1, 2 and 3 can be computed as follows: First the receiver sends the public key (128 bytes) to the sender. The sender in returns sends the encrypted key (16 bytes) to the receiver. Thus, the model includes a total communication overhead of 144 bytes during single transmission at layers 1, 2 and 3 while it does not comprise any communication overhead at layer 0 which is found to be practical.

2) Storage Overhead: Our proposed model involves no storage overhead at layers 0 and 1, while there is need to store some information during the processing at layers 2 and 3 to identify the malicious entity. At layer 2, it is needed to store the ID of m clients and AES secret key, while at layer 3, embedded hashed values are required to be stored in addition to the storage overhead at layer 2.

VI. Conclusion

In this paper, we proposed a sensitivity based layered architecture that works for securing data and preserving its privacy in cloud environment. The proposed architecture reduces the overall overhead at the cloud service provider by implementing the layered based security mechanism. Considering the fact that data utility and security requirements may be quite different for different data, our adapt method maintains security and privacy at each layer differently while considering data utility. At each layer, hashing, encryption and watermarking schemes are innovatively and differently combined together to obtain a better balance between information security and utility. For the most sensitive data i.e secret and top secret data, the approach provides the mechanism to identify the malicious entities in the case of data leakage. Embedded ID in the document is extracted and then this ID is tracked from Information Management Table (IMT). If ID matches, then the leaker is identified resulting in the cancellation of its authorization by the owner. Furthermore, in case of top secret data, leaker is verified also using SHA authentication. The hash value of extracted ID is matched with the embedded hash value. If both are same, then the guilty client is verified. The experimental results prove the correctness, practicality, reliability and efficiency of the proposed approach. Our research work can be further extended via considering the case that captures real world leakage scenarios and for the case when security protocols are not fulfilled by any entity.

References

[1] A. Shabtai, Y. Elovici, and L. Rokach, “A Survey of Data Leakage Detection and Prevention Solutions,” Springer-Verlag NY, 2012, DOI: 10.1007/978-1-4614-2053-8.

[2] X. Dong, J. Yu, Y. Luo, Y. Chen, G. Xue, and M. Li, “Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing,” Future Gener. Comput. Syst., vol. 42, pp. 151-164, 2014, DOI: 10.1016/j.cose.2013.12.002.

[3] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, “A survey on data leakage prevention systems,” Journal Network Comput. Applicat., vol. 62, pp. 137-152, Jan. 2016, DOI: 10.1016/j.jnca.2016.01.008.

[4] B. Hauer, “Data and Information Leakage Prevention Within the Scope of Information Security,” IEEE Access: The Journal for rapid open access publishing, vol. 3, pp. 2554-2565, Dec. 2015, DOI: 10.1109/ACCESS.2015.2506185.

[5] A. M. Nia, S. Sur-kolay, A. Raghunathan, and N. K. Jha, “Physiological Information Leakage: A New Frontier in Health Information Security,” IEEE Trans. Emerg. Topics Comput., vol. 4, no. 3, pp. 321-334, Sept. 2016, DOI: 10.1109/TETC.2015.2478003.

[6] M. Backes, N. Grimm, and A. Kate, “Data Lineage in Malicious Environments,” IEEE Trans. Dependable Secure Comput., vol. 13, no. 2, pp. 178-191, Mar./Apr. 2016, DOI: 10.1109/TDSC.2015.2399296.

[7] B. C. Fung, K. Wang, and P. S. S., “Anonymizing Classification Data for Privacy Preservation,” IEEE Trans. Knowl. Data ENG., vol. 19, no. 5, pp. 711-725, May 2007, DOI: 10.1109/TKDE.2007.1015.

[8] Statista, “Number of consumer cloud-based service users worldwide in 2013 and 2018 (in billions),” [Online] Available: https://www.statista.com/statistics/321215/global-consumer-cloud-computing-users/.

[9] T. M. Payton and T. Claypoole, “Privacy in the age of big data,” U.S.A.: Rowman and Littlefield, 2015.

[10] Risk Based Security (RBS), “2014 Data Breaches – A Billion Exposed Records – A New All Time High,” [Online] Available: https://www.riskbasedsecurity.com/2015/02/2014-data-breaches-a-billion-exposed-records-a-new-all-time-high/.

[11] A. Harel, A. Shabtai, L. Rokach, and Y. Elovici, “M-Score: A Misuseability Weight Measure,” IEEE Trans. Dependable Secure Comput., vol. 9, no. 3, pp. 414-428, May/June 2012, DOI: 10.1109/TDSC.2012.17.

[12] J.-J. Yang, J.-Q. Li, and Y. Niu, “A hybrid solution for privacy preserving medical data sharing in the cloud environment,” Future Gener. Comput. Syst., vol. 43-44, pp. 74-86, 2015, DOI: 10.1016/j.future.2014.06.004.

[13] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Comput., vol. 62, no. 2, pp. 362-375, Feb. 2013, DOI: 10.1109/TC.2011.245.

[14] X. Zhang, C. Liu, S. Nepal, S. Pandey, and J. Chen, “A Privacy Leakage Upper Bound Constraint-Based Approach for Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1192-1202, June 2013, DOI: 10.1109/TPDS.2012.238.

[15] Ernst & Young, “Data loss prevention – Keeping your sensitive data out of the public domain,” Insights on governance, risk and compliance, Oct. 2011, [Online] Available: https://www.ey.com/Publication/vwLUAssets/EY_Data_Loss_Prevention/$FILE/EY_Data_Loss_Prevention.pdf.

[16] M. Hilbert, “What is the content of the world's technologically mediated information and communication capacity: how much text, image, audio and video?,” The Inform. Soc., vol. 30, no. 2, pp. 127-143, 2014, DOI: 10.1080/01972243.2013.873748.

[17] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, “Information Security in Big Data: Privacy and Data Mining,” IEEE Access: The Journal for rapid open access publishing, vol. 2, pp. 1149-1176, Oct. 2014, DOI: 10.1109/ACCESS.2014.2362522.

[18] J. Hua, A. Tang, Y. Fang, Z. Shen, and S. Zhong, “Privacy-Preserving Utility Verification of the Data Published by Non-Interactive Differentially Private Mechanisms,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 10, pp. 2298-2311, Oct. 2016, DOI: 10.1109/TIFS.2016.2532839.

[19] F. Salim, N. P. Sheppard, and R. Safavi-Naini, “A rights management approach to securing data distribution in coalitions,” Proc. 4th Int. Conf. Netw. Syst. Security, pp. 560-567, 2010, DOI: 10.1109/NSS.2010.94.

[20] A. Pretschner, M. Hilty, F. Schütz, C. Schaefer, and T. Walter, “Usage control enforcement: Present and future,” IEEE Security Privacy, vol. 6, no. 4, pp. 44-53, Jul./Aug. 2008, DOI: 10.1109/MSP.2008.101.

[21] F. Kelbert and A. Pretschner, “Data usage control enforcement in distributed systems,” Proc. 3rd ACM Conf. Data Appl. Security Privacy, pp. 71-82, 2013, DOI: 10.1145/2435349.2435358.

[22] M. Nabeel and E. Bertino, “Privacy Preserving Delegated Access Control in Public Clouds,” IEEE Trans. Knowl. Data ENG., vol. 26, no. 9, pp. 2268-2280, Sept. 2014, DOI: 10.1109/TKDE.2013.68.

[23] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, and R. Buyya, “Attribute-based data access control in mobile cloud computing: Taxonomy and open issues,” Future Gener. Comput. Syst., vol. 72, pp. 273-287, 2017, DOI: 10.1016/j.future.2016.08.018.

[24] V. Velichkov, V. Rijmen, and B. Preneel, “Algebraic cryptanalysis of a small-scale version of stream cipher Lex,” IET Inform. security, vol. 4, no. 2, pp. 49-61, 2010, DOI: 10.1049/iet-ifs.2009.0118.

[25] Y.-W. Kao, K.-Y. Huang, H.-Z. Gu, and S.-M. Yuan, “uCloud: a user- centric key management scheme for cloud data protection,” IET Inform. Security, vol. 7, no. 2, pp. 144-154, 2013, DOI: 10.1049/iet-ifs.2012.0198.

[26] B. Qin, S. Liu, and K. Chen, “Efficient chosen-ciphertext secure public-key encryption scheme with high leakage-resilience,” IET Inform. security, vol. 9, no. 1, pp. 32-42, 2014, DOI: 10.1049/iet-ifs.2013.0173.

[27] S.-H. Seo, M. Nabeel, X. Ding, and E. Bertino, “An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds,” IEEE Trans. Knowl. Data ENG., vol. 26, no. 9, pp. 2107-2119, Sept. 2014, DOI: 10.1109/TKDE.2013.138.

[28] J. Liu, X. Huang, and J. K. Liu, “Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption,” Future Gener. Comput. Syst., vol. 52, pp. 67-76, 2015, DOI: 10.1016/j.future.2014.10.014.

[29] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. yang, Y. Yu, and A. Yang, “A secure and efficient Ciphertext-Policy Attribute-Based Proxy Re-Encryption for cloud data sharing,” Future Gener. Comput. Syst., vol. 52, pp. 95-108, 2015, DOI: 10.1016/j.future.2014.11.016.

[30] A. Al-Haj, G. Abandah, and N. Hussein, “Crypto-based algorithm for secured medical image transmission,” IET Inform. security, vol. 9, no. 6, pp. 365-373, 2015, DOI: 10.1049/iet-ifs.2014.0245.

[31] Q. Huang, Z. Ma, Y. Yang, J. Fu, and X. Niu, “EABDS: attribute-based secure data sharing with efficient revocation in cloud computing,” Chinese J. Electron., vol. 24, no. 4, pp. 862-868, 2015, DOI: 10.1049/cje.2015.10.033.

[32] Y. Lu and J. Li, “A pairing-free certificate-based proxy re-encryption scheme for secure data sharing in public clouds,” Future Gener. Comput. Syst., vol. 62, pp. 140-147, Sept. 2016, DOI: 10.1016/j.future.2015.11.012.

[33] Y. S. Rao, “A secure and efficient Ciphertext-Policy Attribute-Based Signcryption for Personal Health Records sharing in cloud computing,” Future Gener. Comput. Syst., vol. 67, pp. 133-151, Feb. 2017, DOI: 10.1016/j.future.2016.07.019.

[34] M. A. Azad and S. Bag, “Decentralized privacy-aware collaborative filtering of smart spammers in a telecommunication network,” ACM Symp. Appl. Computing, pp. 1711-1717, 2017, DOI: 10.1145/3019612.3019792.

[35] M. A. Azad, S. Bag, S. Tabassum, and F. Hao, “privy: Privacy preserving collaboration across multiple service providers to combat telecoms spam,” IEEE Trans. emerging topics computing, 2017, DOI: 10.1109/TETC.2017.2771251.

[36] Q. Wang, L. Peng, H. Xiong, J. Sun, and Z. Qin, “Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing,” IEEE Access, vol. 6, pp. 760-771, 2018, DOI: 10.1109/ACCESS.2017.2775741.

[37] S. Bag, M. A. Azad, and F. Hao, “A privacy-aware decentralized and personalized reputation system,” Comput. & Security, vol. 77, pp. 514-530, 2018, DOI: 10.1016/j.cose.2018.05.005.

[38] M. A. Azad, S. Bag, and F. Hao, “PrivBox: Verifiable decentralized reputation system for online marketplaces,” Future Gener. Comput. Syst., vol. 89, pp. 44-57, 2018, DOI: 10.1016/j.future.2018.05.069.

[39] M. A. Azad, S. Bag, F. Hao, and K. Salah, “M2m-rep: Reputation system for machines in the internet of things,” Comput. & Security, vol. 79, pp.1-16, 2018, DOI: 10.1016/j.cose.2018.07.014.

[40] Y. Shapira, B. Shapira, and A. Shabtai, “Content-based data leakage detection using extended fingerprinting,” 2013, arXiv preprint arXiv:1302.2028.

[41] X. Shu, D. Yao, and E. Bertino, “Privacy-Preserving Detection of Sensitive Data Exposure,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 5, pp. 1092-1103, May 2015, DOI: 10.1109/TIFS.2015.2398363.

[42] X. Shu, J. Zhang, D. Yao, and W.-C. Feng, “Fast Detection of Transformed Data Leaks,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 3, pp. 528-542, Mar. 2016, DOI: 10.1109/TIFS.2015.2503271.

[43] P. Papadimitriou and H. Garcia-Molina, “Data Leakage Detection,” IEEE Trans. Knowl. Data ENG., vol. 23, no. 1, pp. 51-63, Jan. 2011, DOI: 10.1109/TKDE.2010.100.

[44] A. Kumar, A. Goyal, A. Kumar, N. K. Chaudhary, and S. K. S, “Comparative Evaluation of Algorithms for Effective Data Leakage Detection,” Proc. IEEE Conf. Inform. Commun. Technologies (ICT), pp. 177-182, 2013, DOI: 10.1109/CICT.2013.6558085.

[45] S. Sodagudi and R. R. Kurra, “An Approach to Identify Data Leakage in Secure Communication,” Proc. 2nd Int. Conf. Intelligent Computing and Applicat. (ICICA), pp. 31-43, 2015, DOI: 10.1007/978-981-10-1645-5_4.

[46] M. Shehab, E. Bertino, and A. Ghafoor, “Watermarking Relational Databases Using Optimization-Based Techniques,” IEEE Trans. Knowl. Data ENG., vol. 20, no. 1, pp. 116-129, Jan. 2008, DOI: 10.1109/TKDE.2007.190668.

[47] H. V. Singh, S. Rai, A. Mohan, and S. P. Singh, “Robust copyright marking using Weibull distribution,” Comput. Elect. Eng., vol. 37, no. 5, pp. 714-728, Sept. 2011, DOI: 10.1016/j.compeleceng.2011.04.006.

[48] G. Bhatnagar and Q. M. J. Wu, “Biometrics inspired watermarking based on a fractional dual tree complex wavelet transform,” Future Gener. Comput. Syst., vol. 29, pp. 182-195, 2013, DOI: 10.1016/j.future.2012.05.021.

[49] A. K. Singh, M. Dave, and A. Mohan, “Wavelet Based Image Watermarking: Futuristic Concepts in information Security,” The Nat. Academy of Sciences, vol. 84, no. 3, pp. 345-359, june 2014, DOI: 10.1007/s40010-014-0140-x.

[50] I. J. Cox and M. L. Miller, “A review of watermarking and the importance of perceptual modeling,” Proc. Electron. Imaging, vol. 97, pp. 92-99, 1997, DOI: 10.1117/12.274502.

[51] A. K. Singh, M. Dave, and A. Mohan, “Hybrid Technique for Robust and Imperceptible Image Watermarking in DWT–DCT–SVD Domain,” Natl. Acad. Sci. Lett., vol. 37, no. 4, pp. 351-358, July-Aug. 2014, DOI: 10.1007/s40009-014-0241-8.

Ishu Gupta received the BCA, M.Sc and MCA (Gold Medalist) degrees in Computer Science from Kurukshetra University, India. Currently, she is working as a Ph.D. student in the Department of Computer Applications, National Institute of Technology, Kurukshetra, India. She is awarded with Senior Research Fellowship (SRF) by the University Grant Commission (UGC), India. She had been entitled for the awards such as Merit Award – 08, Roll of Honour, and College Color – 02 for securing positions in the university during her graduation and master. Her recent research interests include the areas of Cloud Computing and Information Security.

Niharika Singh is currently working towards Ph.D. in the Department of Electrical and Electronics Engineering of Universiti Teknologi PETRONAS, Malaysia. She received M.Tech in Computer Engineering and B.Tech in Computer Science and Engineering from Kurukshetra University, India, in 2016 & 2014 respectively. She has an experience of more than 1 year in academia. She is actively engaged in research activities. Her current research interests include power systems and communication, high performance computing, cloud computing etc.

Ashutosh Kumar Singh is working as a Professor and Head in National Institute of Technology, Kurukshetra, India. He has more than 15 years research and teaching experience in various Universities of the India, UK, and Malaysia. Prior to this appointment, he has worked as an Associate Professor and Head of Department Electrical and Computer Engineering in School of Engineering Curtin University Australia offshore Campus Malaysia, Sr. Lecturer and Deputy Dean (Research and Graduate Studies) in Faculty of Information Technology, University Tun Abdul Razak Kuala Lumpur Malaysia, Post Doc RA in the Department of Computer Science, University of Bristol, Faculty of Information Science and Technology, Multimedia University Malaysia and Sr. Lecturer in Electronics and Communication Department at NIST, India. He has obtained his Ph. D. degree in Electronics Engineering from Indian Institute of Technology, BHU, India, Post Doc from Department of Computer Science, University of Bristol, UK and Charted Engineer from UK. His research area includes Web Technology, Big Data, Verification, Synthesis, Design and Testing of Digital Circuits. He has published more than 160 research papers now in different journals, conferences and news magazines and in these areas. He has co-author of six books with reputed international publishers such as Pearson Education Malaysia, Lap Lambert Academic Publishing and Scholar Press Germany. He has worked as principal investigator for four sponsored research projects and was a key member on a project from EPSRC (UK) “Logic Verification and Synthesis in New Framework”. He has delivered the invited talks and presented research papers in several countries including Australia, UK, South Korea, China, Thailand, Indonesia, India and USA. He had been entitled for the awards such as Merit Award – 03 (Institute of Engineers), Best Poster Presenter – 99 in 86^{th} Indian Science Congress held in Chennai, INDIA, Best Paper Presenter of NSC’99 INDIA and Bintulu Development Authority Best Postgraduate Research Paper Award for 2010, 2011, 2012.