CRUPA: Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud

Geeta C M · Usharani Rathod · Shreyas Raju R G · Raghavendra S · Rajkumar Buyya · Venugopal K R · S S Iyengar · L M Patnaik

Received: date / Accepted: date

Abstract Cloud repository is one of the most important services afforded by Cloud Computing where information is preserved, maintained, archived in distant servers and made available to the users over the Internet. Provided with the cloud repository facilities, customers can organize themselves as a cluster and distribute information with one another. In order to allow public integrity auditing on the information stored in semi-trusted cloud server, customers compute the signatures for every chunk of the shared information.

When a malicious client is repudiated from the group, the chunks that were outsourced to the cloud server by this renounced customer need to be verified and re-signed by the customer present in the cluster (i.e., the straightforward approach) which results in huge transmission and reckoning cost for the customer. In order to minimize the burden of customers present in the cluster, in the existing scheme Panda, the semi-trusted Cloud Service Provider (CSP) is allowed to compute the Re–sign key. Further, the CSP audits and re-signs the revoked customer chunks by utilizing the Re–sign key. So, it is easy for the CSP by colluding with the revoked customer to find the secret keys of the existing customer. We introduce a novel Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud (CRUPA) by making use of the concept of regression technique. In order to secure the secret keys of the existing customers from the CSP, we have allowed the information proprietor to compute the Re–sign key using the regression technique. Whenever the information proprietor revokes the customer from the cluster, the information proprietor computes the Re–sign key using the regression technique and sends to the CSP. Further, the CSP audits and re-signs the revoked customer chunks using the Re–sign key. The Re–sign key computed by the information proprietor using regression method is highly secure and the malicious CSP cannot find the private information of the customers in the cluster. Besides, our mechanism achieves significant improvement in the computation cost of the Re–sign key by information proprietor. Further, the proposed scheme is collusion resistant, supports effective and secure customer repudiation, multi-information proprietor batch auditing and is scalable.
Keywords Cloud Computing · User Revocation · Public Auditing · Proxy Re-signatures · Multi-Information Proprietor Batch Auditing · Regression Method.

1 Introduction

Cloud repository is one of the significant services provided by cloud computing [20]. It empowers the information owner to deploy their information to the cloud server. Many distributed computing service suppliers have been developed, such as Google App Engine, Dropbox that satisfies the requirement for data repository and high performance computation. With information repository and sharing services, customers are permitted to update and distribute the information saved in the distributed server in any place and at any moment [8]. Yet, security of the information has become a severe issue and one of the worrying factors of the information security is the integrity of the deployed information in the distributed server. Even though the cloud repository suppliers accomplish a trustworthy and secure repository maintenance to the customers, the honesty of deployed information might be adulterated due to the negligence of people or disruption of the hardware/software [25]. Apart from inherent hazards, external attacker can further impair the integrity of the deployed information in the cloud. Hence, public integrity verification is required to assure the customers that the deployed information is precisely deployed in the cloud.

Presently, optical networks [34], [35] have been deployed all over the globe for efficient information communication.

Numerous mechanisms have been suggested based on miscellaneous procedures [18], [42], [43] that assure the integrity of deployed information in an untrustable cloud. In all these mechanisms, signatures on every chunk of shared information are estimated by the Information proprietor (IP) and he deploys the information and the equivalent signatures to the distributed server, that permits the IP and public examiner to examine the integrity of the information in the distributed server without fetching the complete deployed information. Still, a large number of the earlier mechanisms deal with the ease of individual information, that implies the IP is the only modifier, who possesses the private key and can modify the information. Researchers are motivated to address the issue in cross domain areas such as Wireless Sensor Networks [28] and the Internet of Things [22].

Wang et al., [38] introduced Oruta, a public examining convention for distributed information in the cloud employing ring signatures. The scheme conserves identity privacy of the customers in the cluster from the public verifier at the time of verification. The limitation is that the mechanism does not bolster traceability and data freshness. Wang et al., [37] introduced Knox, based on cluster signatures that can conserve the identity secrecy of customers from the public verifier.

Wang et al., [39] proposed public verifying mechanism to bolster effective customer revocation utilizing intermediary re-signatures, that acknowledge the distributed server to transform the signatures estimated by the replicated customer into signatures of the current customer within the cluster. The cloud knows in advance the re-signing keys of any two customers in the cluster. This procedure leads to the following two severe security issues. Initially, a mischievous CSP may immediately transform signatures between two customers utilizing the re-signing keys. Further, conspiracy amidst the cloud and the replicated customers might disclose the private keys of all the current customers in the cluster. The reckoning cost of verification increases with the size of the cluster.

Considering these two security problems of [39], we propose a novel Collusion Resistant User Revocable Public Auditing of Shared Data (CRUPA) mechanism.

Motivation: In the existing scheme [39], the semi-trusted CSP is allowed to figure out the Re-sign key by employing the secret keys of the existing customers in the cluster. Since the CSP knows the secret keys of the customers, it is very easy for the CSP to know and retrieve the sensitive data cached in the server. Moreover, when the revoked customer colludes with the CSP, they can further hack or misuse the information cached in the distributed server. Hence the existing scheme [39] is not secure and is not collusion resistant. Motivated to secure the Re-sign key from the semi-trusted CSP, in the proposed scheme, after revoking the malicious customer from the cluster the IP who is the head or manager of the respective cluster is allowed to compute the Re-sign key using regression method such that the key computed is highly secure. Then, the IP transmits the Re-sign key to the CSP and allows him to audit the revoked customer chunks and re-signs the chunks using the Re-sign key. Since the semi-trusted CSP receives the Re-sign key by the IP, it is not possible for the CSP to learn the private keys of the customers present in the cluster and the information stored in the
server is highly secure. We have enhanced the existing system to multiple clusters with the respective information proprietors’ scenario.

Contributions: In this paper we introduce Collusion Resistant User Revocable Public Auditing (CRUPA) of Shared Data scheme that reduces the computation cost of the Re – sign key using regression method by the IP that is highly secure and also supports multiple clusters with their respective IP. Specifically, our contributions are outlined as follows:

(i) Secure Re – sign key generation: The IP, manager of the respective clusters is allowed to compute the Re – sign key securely using the regression method.

(ii) Effective and secure customer repudiation: Once a malicious customer is repudiated from the cluster by the IP, the chunks signed by the repudiated customer can be effectively re-signed. On behalf of the existing customers, the CSP efficiently and securely audits and re-signs the repudiated customer chunks using the Re – sign key sent by the IP and the repudiated customer can no longer estimate the valid signatures on the shared information.

(iii) Privacy preserving and collusion resistant: The CSP (possess the Re – sign key sent by the IP), by colluding with the revoked customer, cannot find the secret keys of the existing customers from the Re – sign key. Thus, the scheme preserves the privacy of the customers and is collusion resistant.

(iv) Public auditing: The Third Party Auditor (TPA) audits the requests sent by every IP of all the clusters individually called as individual auditing. The TPA also performs multi-information proprietor batch auditing for the requests of all IPs simultaneously.

(v) Scalability: Cloud information is effectively distributed among the existing customers of multiple clusters.

Organisation: The rest of the paper is arranged as follows: Related works and Background work are discussed in Section 2. Several preliminaries are introduced in Section 3. Problem definition, System model are discussed in Section 4. Mathematical Model using Regression Method, Security Analysis and Adversary Model are explained in Section 5. Scheme details of Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud (CRUPA) and the construction of Homomorphic Authenticable Proxy Re-signature Scheme (HAPS) using Regression Method are discussed in Section 6. In Section 7, Performance Evaluation results are analysed and Section 8 contains the Conclusions.

2 Related Works

Provable Data Possession [2], authorizes the auditor to publicly validate the integrity of information without fetching the whole information. Improving their earlier work for dynamic operations on data, Atienza et al., [3] constructed another PDP scheme using symmetric keys. This scheme does not support public verification. Erway et al., [12] suggested dynamic verifiable information possession mechanism by using authorized lexicons. Zhu et al., [47] introduced a public verifying scheme that uses the chunk format to reduce the depository of signatures. The mechanism uses Index Hash Table (IHT) that empowers customers to perform effective operations. Tian et al., [32] introduced a non-repudiation dynamic verifiable information possession scheme. The scheme supports identity authentication and non-repudiation. The disadvantage of the mechanism is that it does not support batch auditing. Wu et al., [40] present a Non-Repudiable Provable Data Possession with Designated Verifier (DV – NRDPD) scheme. The scheme addresses the non-renunciation issue and resolves the controversy among the clients and distributed repository servers. The disadvantage of the scheme is that it has high reckoning cost of examining a proof.

Raghavendra et al., [23] have presented a reliable multi-proprietor information distribution for effective association in the cloud. The advantage of the scheme is that the repository space is efficiently utilized and has reduced the time to query documents from the cloud. The drawback is that the convention does not bolster multi-media documents. Tian et al., [30] introduced a public verifying mechanism for secure cloud repository utilizing Dynamic Hash Table (DHT). The proposed mechanism supports dynamic data verification, privacy preservation and batch verification. Dynamic Hash Table (DHT) is used to archive the details of the data for verification and as a result it accomplishes prompt verification and effective data restoration. The limitation is that the scheme does not support different types of cloud data.

Luo et al., [21] have presented a public validation convention for the integrity of collaborative information with pervasive and conspiracy resistant customer repudiation. Polynomial based validation marks are generated that support secure and compelling public validation. The cumulative overhead of the examining scheme is comparatively small. Tian et al., [31] have introduced an extensive public verification mechanism for distributed information in cloud. The mechanism supports the customer’s identity privacy, information pri-
vacy and identity trackability. The drawback of the
mechanism is that it has larger communication cost.

Dong et al., [11] have achieved data confidentiality
against the semi-trusted cloud. They designed a pro-
tected, adequate and flexible data co-ordinated scheme.
The mechanism does not accomplish information consis-
tency. Yaun and Yu [36] have designed an auditing
mechanism for distributed data sharing utilities illus-
trated by multi-user alterations, public auditing, ade-
quate user repudiation and pragmatic reckoning audit-
ing performance. The mechanism overcomes customer
impersonation assault. The limitation is that it does
not realize dependability and error detection.

Geeta et al., [14] have performed extensive review
on the latest methods in information auditing and se-
curity in cloud computing. Shen et al., [26] have sug-
gested an effective public verification function. The
proposed convention supports batch verification, block-
less verification and lazy update. The limitation of the
scheme is that the transmission cost is more in verifica-
tion phase. Zhu et al., [48] have presented a secure
anti-conspiracy information sharing mechanism for dy-
namic clusters in the cloud. The repudiated customer
cannot fetch the original document though he conspires
with the CSP. The proposed mechanism bolsters guar-
anteed key allocation, fine-grained access control and
secure customer repudiation. Li et al., [19] have presented
a security model and a formal definition for Cipher-
text Policy-Attribute-Based Encryption (CP-ABE)
scheme with effective attribute repudiation. The pro-
posed mechanism is secure against conspiracy attack
launched by the prevailing customers and the renunci-
ated customers. The limitation of the scheme is that it
takes more time in the Setup phase.

Yang et al., [44] have designed a framework for pub-
lic auditing for shared information in distributed reposi-

tory supporting identity secrecy and trackability. The
mechanism achieves data privacy by utilizing blind sig-
nature method. The limitation is that the mechanism
incurs little overhead to accomplish the identity track-
ability. Hall et al., [15] have presented a protocol which
achieves the cryptographic definition of security, when
the only output are the regression coefficient estimates.
The protocol guarantees the confidentiality of the input
information. Homomorphic encryption is utilized in
constructing the protocol for regression analysis. Chen
et al., [9] introduced two conventions that can autho-
rize protected and effective outsourcing of linear regres-
sion problems to the cloud. The conventions are efficient
and also preserves the client's data confidentiality. The
drawback of the mechanism is that it does not support
identity tracking problems related to computation
outsourcing to the cloud.

Verifiable data proprietorship mechanism [29] pro-
vides trustworthiness and individuality in an active,
multi-user framework. By exploiting trustworthy hard-
ware on the server, forking and rollback intrusions are discarded. The proposed design does not consider load
stabilizing over various servers. Venugopal et al., [36]
have proposed a number of soft computing techniques
for security requirements. Jin et al., [17] have intro-
duced the integrity auditing scheme that supports pub-
lic verifiability, efficient data dynamics and fair disputes
arbitration. Fair arbitration protocols are designed so
that any possible dispute can be fairly settled. The
scheme incurs reasonable overhead of data dynamics
and dispute arbitration.

Dong et al., [10] have suggested a confidentiality
preserving and secure data collaboration procedure in
distributed computing. The convention does not leak
any features of the clients to the cloud. The procedure
is adequate and has low overhead. The mechanism is
not executed on real cloud platform. A comprehensive
analysis of miscellaneous data trustworthiness proce-
dures for distributed computing has been carried out
by Garg and Bawa [13]. They have examined that the
maximum of the prevailing procedures concentrate on
integrity checks to distinctive data repository strategy.
Simulations are carried out on C++ platform [33].

Raghavendra et al., [24] have proposed an effective
token creation method, that enhances immune and pro-
ductive label construction phase. A systematic compo-
sition is refined to encode the ordered keywords for se-
cure label construction. The method reduces the cost of the
information proprietor. Xu et al., [41] have intro-
duced multi-authorization proxy re-encoding method.
The scheme greatly reduce the computation cost of
the creation of key constituents and the termination of the
customers retrieving authority. The algorithm needs
prolonged computation duration Setup phase.

Hwang et al., [16] have outlined a group signature
mechanism supporting the manageable connectivity. The
convention supports reliability properties for e.g., confi-
dentiality and connectivity. Privacy is not preserved by
global linkability. Yu et al., [45] have suggested a dis-
tributed data integrity auditing with identity privacy-
conserving convention for mobile cloud repository. The
scheme affords anonymity to Third Party Auditor (TPA)
and reliable label-updating. The mechanism incurs min-
imum reckoning, transmission and repository overhead.

Shen et al., [27] outlined a distant information in-
tegrity auditing mechanism that realizes information
distribution with sensitive information hiding. Authors
have utilized a sanitizer that is used to sanitize the
sensitive information of the document. The mechanism
supports information data sharing with sensitive infor-
motion hiding. The limitation of the mechanism is that the computation cost of TPA in proof verification is more.

Table 1 shows the comparison of recent existing schemes for Public Honesty Verification with Group Customer Repudiation.

2.1 Background Work

Wang et al., [39], have suggested public auditing scheme for the integrity of shared information with adept customer repudiation. By exploiting the concept of agent re-signatures, the cloud is permitted to re-sign revoked customer chunks on behalf of current customers at the time of customer repudiation, to prevent current customers to retrieve and re-sign chunks by themselves. Further, the public examiner examines the integrity of the distributed information without retrieving the entire information from the cloud, though CSP re-signs few chunks of distributed information. The scheme also supports batch auditing. The limitation of the scheme is that it does not preserve the privacy of the customers in the cluster and is not collusion resistant i.e., the revoked customer colludes with the cloud.

3 Preliminaries

This section discusses the foundations of our approach and are outlined below:

3.1 Bilinear Map:

Consider two cyclic multiplicative groups $G$ and $G_T$ of prime order $p$, $e : G * G \rightarrow G_T$ is a bilinear map with the subsequent properties [7]:

- **Bilinear:** for all $u, v \in G_1$ and $a, b \in Z_p$,
  $$e(u^a, v^b) = e(u, v)^{ab}$$
- **Non-degeneracy:** $e(g, g) \neq 1$;
- **Computability:** An effective algorithm prevails for estimating map $e$.

Computational Diffie-Hellman (CDH) Problem: Given $g, g^a, g^b \in G$ for unknown $a, b \in Z_p$, to estimate $g^{ab}$.

3.2 Homomorphic Authenticators

Homomorphic authenticators [2], permit a public validator to examine the integrity of information distribute d in the cloud server without fetching the complete information. The properties of homomorphic authenticable signature mechanism are as follows:

- Let the signer’s public/secret key pair be $(p_1, s_1)$, $p_1$ is the signature on chunk $b_1 \in Z_p$, and $p_2$ is the signature on chunk $b_2 \in Z_p$.
- **Blockless auditability:** Given $p_1$ and $p_2$, two arbitrary values $b_1, b_2$ in $Z_p$ and a chunk $b' = b_1b_1 + b_2b_2 \in Z_p$, an auditor audits the accuracy of chunk $b'$ without the knowledge of $b_1$ and $b_2$.
- **Non-flexibility:** Given $b_1$ and $b_2$, $p_1$ and $p_2$, two random values $b_1, b_2$ in $Z_p$ and a chunk $b' = b_1b_1 + b_2b_2 \in Z_p$, a customer with secret key $(s_k)$, is unable to produce a legitimate signature $\rho'$ on chunk $b'$ by joining $p_1$ and $p_2$.

Blockless auditability permits an auditor to examine the integrity of information hosted on the distributed server by generating the linear aggregation of all the chunks via a challenge-and-response convention. Hence the verifier need not download the whole information from the cloud. Non-flexibility illustrates that alternative entities who do not possess appropriate secret keys are unable to create legitimate signatures on combination of chunks by using the signatures that they possess.

3.3 Proxy Re-Signatures

Proxy re-signatures [5] permit a semi-trusted intermediary to accomplish as an interpreter of signatures amidst two customers. Conventional proxy re-signature mechanisms [4], [5], do not support blockless auditability, if we utilize these intermediary re-signature mechanisms in the public verification schemes, then the auditor has to retrieve the whole information to verify the integrity, that necessarily decreases the effectiveness of verification. Hence, we utilize Homomorphic Authentica ble Proxy Re-signature (HAPS) [39] mechanism, that satisfies blockless auditability and non-flexibility. In our paper, after repudiating malicious customer, the IP of respective clusters computes the Re – sign key and transmits it to the CSP. After acquiring the Re – sign key, the CSP checks the integrity of the revoked customer chunks and signs these chunks utilizing the Re – sign key sent by the IP.

3.4 Regression Co-efficient

Regression co-efficient is an estimation of an independent variable in terms of the other. If $p_k$ and $s_k$ are correlated, the best fitting straight line in the least square sense gives a reasonably good relation between public key $p_k$ and secret key $s_k$. Similarly, in our scenario, the regression co-efficient secures the public key $p_k$ and secret key $s_k$ of the Re – sign key.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Concept</th>
<th>Performance</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tian et al., 2019 [31]</td>
<td>Public auditing for distributed cloud data with adept and reliable cluster management</td>
<td>The computational cost is significantly reduced in the verification phase.</td>
<td>Supports individuality privacy, data privacy and individuality trackability.</td>
<td>Low communication cost.</td>
</tr>
<tr>
<td>Shen et al., 2019 [27]</td>
<td>Individuality-based integrity auditing and information sharing with sensitive information hiding for reliable cloud repository.</td>
<td>The computation costs of TPA and CSP is higher with the increase of challenged blocks.</td>
<td>Supports information sharing with sensitive information hiding.</td>
<td>Computation cost of TPA in proof verification is high.</td>
</tr>
<tr>
<td>Tian et al., 2017 [30]</td>
<td>Public auditing mechanism for protected cloud repository based on Dynamic Hash Table (DHT).</td>
<td>The scheme has lower costs of storage, communication and computation.</td>
<td>Achieves higher updating efficiency and secure auditing.</td>
<td>Does not support various types of cloud data.</td>
</tr>
<tr>
<td>Xu et al., 2016 [41]</td>
<td>Multi-authority intermediary re-encryption based on CPABE for distributed repository system.</td>
<td>(MPRE – CPABE) reduces the estimation cost of the creation of key components.</td>
<td>Small reckoning cost of key allocation.</td>
<td>Additional computational period in Setup phase.</td>
</tr>
<tr>
<td>Wang et al., 2015 [39]</td>
<td>Public auditing for shared information with adept user renunciation.</td>
<td>No communication overhead to existing customers during customer repudiation, cloud has reduced computation cost.</td>
<td>Secure customer repudiation, public auditing.</td>
<td>Collusion of repudiated customer and cloud.</td>
</tr>
<tr>
<td>CRUPA</td>
<td>Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud(CRUPA)</td>
<td>Significant improvement in computation cost of (Re – sign) key by information proprietor, low processing time in Setup phase.</td>
<td>Supports multi-owner batch auditing, efficient customer revocation.</td>
<td>Average auditing time cost is more.</td>
</tr>
</tbody>
</table>

### 4 Problem Statement

#### 4.1 Problem Definition

Given a cloud storage model consisting of CSP, TPA and multiple clusters with their respective Information Proprietor's, the main objectives are:

(i) **Secure Re−sign key generation**: The IP, manager of the respective clusters is allowed to compute the \(Re – sign\) key securely using the regression method.

(ii) **Effective and secure customer repudiation**: Once a malicious customer is repudiated from the cluster by the IP, the chunks signed by the repudiated customer can be effectively re-signed. On behalf of the existing customers, the CSP efficiently and securely audits and re-signs the repudiated customer chunks using the \(Re – sign\) key sent by the IP and the repudiated customer can no longer
estimate the valid signatures on the shared information.

(iii) **Privacy preserving and collusion resistant:** The CSP (possess the \( Re – sign \) key sent by the IP), by colluding with the revoked customer, cannot find the secret keys of the existing customers from the \( Re – sign \) key. Thus, the scheme preserves the privacy of the customers and is collusion resistant.

(iv) **Public auditing:** The Third Party Auditor (TPA) audits the requests sent by every IP of all the clusters individually called as individual auditing. The TPA also performs multi-information proprietor batch auditing for the requests of all IPs simultaneously.

(v) **Scalability:** Cloud information is effectively distributed among the existing customers of multiple clusters.

4.2 Assumptions

(i) CSP is a semi-trusted entity.

(ii) Private channels (e.g., SSL) exist between each pair of entities.

4.3 System Model

As demonstrated in Fig. 1, the system framework comprises of three objects: the Cloud Service Provider (CSP), the TPA and multiple clusters with respective IP. The CSP provides information repository and distribution services to the customers. The TPA aims to audit the integrity of distributed information via challenge-and-response convention with the CSP. Each cluster consists of an IP and various customers in the cluster. The IP is the head or manager of the cluster (group of customers). The IP generates the private keys and public keys for all the customers in the cluster (See Function 1: GenerateKey). The IP also creates the Customer List (CL). The IP of the respective cluster generates and distributes information with other customers in the cluster through the cloud. Both the IP and customers in the cluster can retrieve and update the distributed information. The distributed information is divided into range of chunks. A customer in a cluster modifies a chunk by carrying out an insert, delete and update operations on the chunk.

Considering that the CSP is a semi-trusted party, it obeys the rules and does not corrupt the integrity of the information passionately as a mischievous attacker. However, it might also deceive the auditor regarding the inaccuracy of the distributed information so that the prominence of its information services is retained.

Normally, the inaccuracy of shared information might be due to hardware/software breakdown or human misinterpretation. Because of these aspects, the customers do not totally rely on the cloud with the integrity of distributed information.

The integrity of the distributed information is preserved by appending a signature to every chunk of the shared information, that is estimated by anyone of the customer’s present in the cluster. Particularly, when the IP originally generates the shared information in the cloud, the total signatures on the shared data are estimated by the IP. Hereafter, when a customer changes a chunk, this customer additionally requires to sign the revised chunk with his secret key. By distributing the data amidst the cluster of customers, distinct chunks may be signed by various customers due to modifications by distinct customers.

While the customer in the cluster leaves or misconducts, the cluster has to remove this customer. Usually, as the originator of the shared information, the IP acts as the cluster manager and he has an authority to repudiate the customer from the cluster. When a customer is removed, the signatures computed by this eliminated customer become insignificant to the cluster, and the chunks signed by this renounced client ought to be re-signed by the prevailing user’s secret key, so that the accuracy of the complete distributed information is validated with the public keys of the current customers.

5 Mathematical Model

5.1 Computation of \( Re\)-sign key \((\tau_{Re – key})\) by the Information Proprietor using Regression Method:

In the existing scheme [39], the authors have allowed the semi-trusted CSP to estimate the \( Re – key \) utilizing the secret keys of the existing customers in the cluster. Thus, it is very easy for the CSP to know and access the sensitive data cached in the server. Moreover, when the revoked customer colludes with the CSP, they can further hack or misuse the information cached in the cloud server. Hence, the existing scheme [39] is not secure and is not collusion resistant.

In the proposed scheme, we have not allowed the semi-trusted CSP to compute the \( Re – sign \) key. In order to secure the secret keys of the existing customers, we have allowed the IP of the respective clusters to compute the \( Re – sign \) key \((\tau_{Re – key})\) using the regression method. When a customer is repudiated from the cluster, the IP of the respective cluster computes the \( Re – sign \) key and transmits to the CSP. The CSP receives the \( Re – sign \) key, verifies and re-signs the re-
voked customer chunks with the \( Re – sign \) key sent by the IP.

The Information Proprietor (IP) uses secret key \( \tau_i \) and public key \( \langle pk_j \rangle \) of customers \( c_i \) and \( c_j \) respectively. The identities of customers \( c_i \) and \( c_j \) are \( id_i \) and \( id_j \) respectively where \( \{i,j\} \subset [1,c] \). \( H \) is a hash function with \( H:\{0,1\}^* \to G_1 \). The computation of \( Re – sign \) key using regression technique is as follows:

In order to secure secret key and public key, the IP substitutes \( \tau_i \) and \( pk_j \), along with hash of id of \( i^{th} \) customer and id of \( j^{th} \) customer in the variables \( a_1 \) and \( a_2 \) respectively.

\[
a_1 = (H(id_i)) \tau_i; \quad a_2 = (H(id_j)) pk_j
\]

By using \( a_1 \) and \( a_2 \) compute \( X_1, Y_1 \) and \( Z_1 \):

\[
X_1 = 2(a_1)^{\alpha_2}; \quad Y_1 = 2(X_1)^{\alpha_2}; \quad Z_1 = X_1 - Y_1
\]

The following steps shows the computation of \( Re – sign \) key (\( \tau_{Re-key} \)) using the Regression method:

\[
X_2 = 2(Y_1)^{\alpha_2}; \quad Y_2 = 2(X_1)^{\alpha_2}; \quad Z_2 = X_2 - Y_2
\]

\[
S(X) = X_1 + X_2; \quad S(Y) = Y_2 + Y_2; \quad S = Z_1 + Z_2
\]

\[
mX = SX/2; \quad mY = SY/2; \quad mZ = SZ/2
\]

\[
Sx^2 = (X_1)^2 + (X_2)^2; \quad Sy^2 = (Y_1)^2 + (Y_2)^2;
\]

\[
SZ = (Z_1)^2 + (Z_2)^2
\]

\[
\delta x^2 = S(x^2/z) - (mX)^2
\]

\[
\delta y^2 = S(y^2/z) - (mY)^2
\]

\[
\delta z^2 = S(z^2/z) - (mZ)^2
\]

\[
\tau_{Re-key} = 2(\Delta + 4)/(2\sqrt{\delta x^2 \delta y^2 + 4}) \tag{1}
\]

where \( \Delta = \delta(x^2) + \delta(y^2) + \delta(z^2) \).

The \( \tau_{Re-key} \) computed consists of secret key and public key implicitly and the key is highly secure where it is difficult for the semi-trusted CSP or the revoked customer to break the key and know the secret keys of the existing customers in the clusters.

5.2 Security Analysis

Theorem 1: The CSP by colluding with the revoked customer, will not be able to find the secret keys of the existing customers from the \( Re – sign \) key.

Proof: In the proposed scheme, the IP is the manager of the cluster. The IP generates the secret keys and the public keys of all the customers present in the cluster [See Function 1: GenerateKey]. In order to secure the secret key \( \tau_i \) and public key \( \langle pk_j \rangle \), the IP substitutes \( \tau_i \) and \( pk_j \), along with hash of id of \( i^{th} \) customer and id of \( j^{th} \) customer in the variables \( a_1 \) and \( a_2 \) respectively [See Section 5]. Further in the regression technique, \( a_1 \) and \( a_2 \) are substituted in \( X_1 \) and \( Y_1 \). This procedure continues, and the final \( \tau_{Re – sign} \) (\( \tau_{Re-key} \)) Eq. no. 1, consists of secret key and public key implicitly and the \( Re – sign \) key computed is highly secure and the CSP will not be able to break this key. The steps in the computation of \( Re – sign \) key using the regression technique proves that the regression technique tightly secures the secret key and the public key and hence it is impossible for the adversary by colluding with the CSP to find the secret key and public key of the customers present in the cluster.

Let us assume that the revoked customer (malicious customer) colludes with the mischievous CSP. Now the CSP is possessing the \( Re – sign \) key sent by the IP. The CSP and the malicious customer tries to break the...
6 The Algorithm

6.1 System setup

Let $G_1$, $G_2$ and $G_T$ be multiplicative groups of prime order $p$, $g$ be a generator of $G_2$, $e: G_1 \times G_2 \rightarrow G_T$ be a bilinear map. $H(\cdot)$ is a secure map-to-point hash function: $\{(0,1) \times G_1\}$ that map strings consistently to $G_1$. Another hash function $h(\cdot): G_1 \rightarrow Z_p$ maps group element of $G_1$ evenly to $Z_p$. The overall number of chunks in the distributed information is $n$ and the distributed information is represented as $S=(b_1, b_2, ..., b_n)$. The total number of customers in the cluster is $c$.

The Algorithm 2, CRUPA (Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud) consists of two phases:

**Phase I:** Secure Re-signing of Revoked Customer Blocks by CSP.

**Phase II:** Secure Multi-Information Proprietor Cluster Auditing for Shared Information by the Third Party Auditor.

6.2 Phase I: Secure Re-signing of Revoked Customer Blocks by CSP

The Function 1: GenerateKey illustrates the generation of secret and public key parameters of the system. There are $D$ Information Proprietors (IP's) of respective clusters in the system, and each Information Proprietor $d$ has a document $F_d=(b_{d,1}, ..., b_{d,n})$ to be deployed in the distributed server, where $d \in \{1, ..., D\}$. For a specific Information Proprietor $d$, the private key is $s_d=\tau d \in Z_p$ and the corresponding public specifications are $(v_d, w_d, g_d, J_d)$. Every IP of their respective clusters generate secret keys and public keys for all their existing customers in the cluster. He also creates the Customer List (CL) that comprises the id's of all the existing customers in their clusters.

Every IP, $d \in \{1, ..., D\}$, encrypts all the chunks of his file $F_d$ and computes signatures for all these chunks. IP sends $(F_d, \phi)$ to the CSP, where $\phi=\{\rho_{k,i}; 1 \leq i \leq n\}$. Now the existing customers of all the clusters retrieve their respective chunks, perform modifications, sign with their secret key ($\tau$) and upload to the server as described in the function $SignatureGen$ [See Algorithm 2, Phase I, Part I]. IP is an authorized person and keeps track of all the customers activities in his cluster. During this process, when anyone of the existing customer is found malicious or the term of his/her membership is expired, then the IP has the right to revoke this customer and withdraw all his credentials.
When a customer is repudiated, the signatures computed by this eliminated client are insignificant to the group, and the chunks that were formerly signed by this repudiated customer should be verified for integrity and re-signed. In the proposed scheme, the IP revokes the malicious customer from the cluster, computes the $1^{\text{Re}} - \text{sign}$ key $(\tau_{\text{Re-key}})$ utilizing the regression method as in Eq. no. 1., and transmits it to the CSP. After obtaining the $\text{Re} - \text{sign}$ key $(\tau_{\text{Re-key}})$ the CSP checks the integrity of the revoked customer chunks and re-signs with the $\tau_{\text{Re-key}}$ as illustrated in the function $\text{Resignature}$ [See Algorithm 2, Phase I, Part II]. The proposed scheme is highly secure, i.e., it is very difficult for the semi-trusted CSP to retrieve the secret keys of the existing customers from the $\text{Re} - \text{sign}$ key $(\tau_{\text{Re-key}})$. By colluding with the revoked customer, the CSP cannot find the secret keys of the existing customers' as the $1^{\text{Re}} - \text{sign}$ key is computed by the IP. Hence, the proposed scheme is collusion resistant, and provides secure integrity auditing of the revoked customer chunks by the CSP.

6.3 PhaseII: Secure Multi-Information Proprietor Cluster Auditing for Shared Information by the Third Party Auditor.

In the proposed system model, the IP's of respective clusters create the auditing request and sends to the TPA. The TPA executes the function $\text{ClusterChal}$ [See Algorithm 2, PhaseII, Part I and Part II] generates challenge=$\{(i, \xi_i)\}_i \in C$ to the respective IP's auditing requests and delivers to the CSP. Upon accepting the challenge from TPA, for every IP, $d (d \in \{1,...,D\})$, the CSP responds to the TPA with the storage proof $\{\rho, \{\chi_d\}_{1 \leq d \leq D}, \{id_d, c_i\}_i \in E\}$. 

![Fig. 2: Adversary model](image-url)
The public verifier executes ClusterVerify [See Algorithm 2, PhaseII, Part III], and validates the accuracy of proof of storage acknowledged by the cloud. The public verifier efficiently performs multi-information proprietor auditing and sends the auditing proof to the respective IP. The multi-information proprietor auditing considerably decreases the transmission cost of the server and the computation cost of the public verifier.

For the public verifier’s challenge request, Challenge:= \{(i, \xi_i)\}, \in N, the CSP utilizes the bilinear aggregate signature [6], and sends one group element \( p \) instead of \( \{\rho_d\}, \leq d \leq D \). Thus, the communication cost on the server side has been greatly reduced. At the same time, combining D auditing equations into one helps to decrease the number of expensive pairing operations from \( 2D \), as individual verification requires \( D+1 \) pairing operations. Hence, reasonable amount of verification time of public verifier is saved.

6.4 Construction of Homomorphic Authenticable Proxy Re-signature Scheme (HAPS) using Regression Method

In the existing scheme, Wang et al., [39], proposed Homomorphic Authenticable Proxy Resignature (HAPS) mechanism. This scheme has five functions: KeyGen, Re – key, Sign, Re – sign and Verify. In the function Re – key of the HAPS mechanism, they have used the Re – key computed by the CSP [39]. They have allowed the semi-trusted CSP to estimate the Re – key employing the secret keys of the existing customers in the cluster. Thus the semi-trusted CSP, who has the knowledge of the secret keys of the existing customers can have access to the information cached in the cloud server. Further, the CSP may collude with the repudiated customer and perform mischievous activity on the data. Hence, the limitation of the scheme is that it is not collusion resistant i.e., CSP and the repudiated customer can find the secret keys of the existing customers.

In our paper, we have used Homomorphic Authenticable Proxy Resignature (HAPS) [39] mechanism. This scheme has five functions: KeyGen, Re – key, Sign, Re – sign and Verify. In the function Re – key [See Algorithm 1], we have used the Re – sign key \( (\tau_{Re-\text{key}}) \) computed by the IP in Eq. no. 1. The Homomorphic Authenticable Proxy Resignature scheme using regression method does not allow the semi-trusted CSP to compute the Re – sign key. Whereas the IP is allowed to estimate the Re – sign key \( (\tau_{Re-\text{key}}) \) as illustrated in Eq. no. 1, utilizing the regression method and then it sends to the CSP. Since the Re – sign key \( (\tau_{Re-\text{key}}) \) is estimated by the IP, it is not possible for the CSP to find the secret keys of the existing customers. Hence the proposed scheme satisfies blockless verifiability, non-flexibility and is also collusion resistant i.e., the semi-trusted CSP cannot collude with the revoked customer.

Table 2 presents the Summary of the Notations used in the Algorithm 2.

### Algorithm 1: Homomorphic Authenticable Proxy Re-signature Scheme (HAPS) using Regression Method

1. Let \( G_1, G_2 \) be two groups of order \( p, g \) be a generator of \( G_1, e : G_1 \times G_1 \rightarrow G_2 \) be a bilinear map, \( w \) be another generator of \( G_1 \). The global parameters are \( (e, p, G_1, G_2, g, w, H) \) where \( H \) is a hash function with \( H: \{0, 1\}^* \rightarrow G_1 \).
2. **Input:** \( \tau_i, b_k \in Z_p \) and \( id_k \) where \( k \in [1, n] \), \( w \), and \( \tau_{Re-\text{key}} \).
3. **Output:** \( \rho_k, \rho_k^{(\tau_{Re-\text{key}})} \).

   - **textbKeyGen:**
     -Customer \( c_i \) selects random number \( \tau_i \) from \( Z_p^* \).
     -Assigns Private key \( sk_i = \tau_i \).
     -Computes Public key \( pk_i = g^{\tau_i} \).

   - **Re-key:**
     -IP computes the Re – sign key \( (\tau_{Re-\text{key}}) \) using regression method [Eq. no. 1]
     \[ (\tau_{Re-\text{key}}) = 2((\Delta + 4)/(2\sqrt{\text{sign} \text{sign} Y^2} + 4)) \]

   - **Sign:**
     -Existing customer \( c_i \) generates the signature \( (\rho_k) \) on block \( b_k \) as:
     \[ \rho_k = (H(id_k) \text{ mod } w^{\tau_k})^\tau \]

   - **Re-sign:**
     -CSP (Proxy) verifies the integrity and re-signs the revoked customer chunks as:
     \[ \text{CSP (Proxy)} \text{ verifies the integrity and re-signs the revoked customer chunks as:} \]

   - **Verify:**
     -The verifier outputs 1, if \( \epsilon(\rho, g) = \epsilon(H(id) \text{ mod } w^{\tau}, pk_1) \) and 0 otherwise.

7 Performance Analysis

To evaluate our proposed mechanism, a prototype system is implemented utilizing Java with Java Pairing-Based Cryptography Library (jPBC) [1] and the experiments are conducted on a PC with windows 7, Intel(R) Core(TM) i5-5200U, CPU @2.20GHz, 8GB RAM. In the following experiments, we assume the size of element in \( G_1 \) or \( Z_p \) is \( |p| = 160 \) bits. The size of an element of \( Z_q \) is \( |q| = 80 \) bits. The size of each chunk is 4kB.
Algorithm 2: CRUPA: Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud

1. **Phase I: Secure Re-signing of Revoked Customer Blocks by CSP**
   
   **Input:** \( \tau_a, \tau_d, b_\ell \in Z_p, i_d, i_k \) where \( k \in \{1, \ldots, m\} \).
   
   **Output:** \( \rho_k, \rho_{d,k}, \nu_k (\tau_{Re-k-key}) \).

2. **Part I: SignatureGen**
   
   Every information proprietor \( d \), divides his file \( F_d \) into \( \{b_{d,1}, \ldots, b_{d,n}\} \) blocks, where \( d \in \{1, \ldots, D\} \).

3. Computes signature \( \rho_{d,k} \) on every block \( b_{d,i} \):

   \[ \rho_{d,k} = \left( H(i_d) + J_b^{k,i} \right)^{\tau_{d,k}} \in G_1 (i = 1, \ldots, n) \]

4. **IP** sends \( (F_d, \rho_{d,k}) \) to the CSP, where \( \text{if} (\rho_{d,k}, i_d) 1 \leq i \leq n \)

5. Existing customers \( c \), in every cluster generates the signature \( \nu_k \) on block \( b_k \):

   \[ \nu_k = \left( H(i_k) + J_b^{k,i} \right)^{\tau_{Re-k-key}} \]

6. for each block with \( i_d \)

7. Compute \( \rho_k = \left( H(i_k) + J_b^{k,i} \right)^{\tau_{Re-k-key}} \).

8. end for

9. **Part II: ReSignature**

10. CSP verifies the integrity and re-signs the revoked customer blocks as:

11. The CSP first verifies that \( c(\rho_k, g) = \left( H(i_k) + J_b^{k,i} \right)^{\nu_k} \).

12. If the auditing result is 0, the CSP outputs \( \perp \)

13. else IP computes \( Re-sign key \) \( \tau_{signature} \) using a regression method and sends to the CSP to re-sign the revoked customer block.

14. CSP re-signs the revoked customer blocks

15. \( \left( (i_k) + J_b^{k,i} \right)^{\nu_k} \in G_1 \) for \( b_k \)

16. The IP performs re-signing, removes customer \( u_c \)’s id from \( CL \), and signs a new \( CL \).

17. **Phase II: Secure Multi-Information Proprietor Cluster Auditing for Shared Information by Third Party Auditor**

18. **Input:** \( d, E, Challenge \)

19. **Output:** Auditing message, verification message

20. **Part I: ClusterChal**

21. The TPA creates verification message as follows: For every cluster’s, \( E \)’s auditing request, the TPA selects a arbitrary \( q \) element subset \( E = \{ e_1, \ldots, e_n \} \) of set \( \{1, 2, \ldots, n\} \).

22. For every element \( i \in E \), the TPA selects an arbitrary value \( \nu_i \). The TPA delivers the challenge \( \{ (\nu_i), e \} \in G \) to the CSP.

23. **Part II: ClusterProof**

24. Upon securing the challenge, for every \( E \)

25. \( \{ e \} \in G \), the CSP computes:

26. \( \chi_{d,e} = \sum_{i=1}^{e} \nu_i b_{d,i} \)

27. \( \rho = \prod_{i=1}^{\nu_e} (\prod_{j=1}^{\nu_e} \rho_{e,j}{i})^{\chi_{d,e}} \).

28. The CSP responses the TPA with \( \{ \rho, \chi_{d,e} \} \) to \( E \), computes:

29. \( \sigma(\rho, e) = \prod_{i=1}^{\nu_e} (\prod_{j=1}^{\nu_e} H(i_d)^{\chi_{d,e}} (J_b)^{e_1}, \nu_1) \)

30. If the output is 1, the TPA considers that the correctness of total chunks in shared information \( S \) is appropriate, else the TPA outputs 0.

### Table 2: Summary of the Notations used in the Algorithm 2

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_1, G_2 )</td>
<td>Multiplicative groups of prime order ( p )</td>
</tr>
<tr>
<td>( q )</td>
<td>Generator polynomial of ( G_1 )</td>
</tr>
<tr>
<td>( H(\cdot) )</td>
<td>Secure multi-point hash function</td>
</tr>
<tr>
<td>( h(\cdot) )</td>
<td>Hash function maps cluster element of ( G_1 ) consistently to ( Z_p )</td>
</tr>
<tr>
<td>( tag_F )</td>
<td>Tag of file ( F )</td>
</tr>
<tr>
<td>( Pk )</td>
<td>Public key</td>
</tr>
<tr>
<td>( SK )</td>
<td>Secret key</td>
</tr>
<tr>
<td>( v_1 )</td>
<td>Signature on block ( b_1 )</td>
</tr>
<tr>
<td>( n )</td>
<td>Total number of chunks in shared data</td>
</tr>
<tr>
<td>( S )</td>
<td>Shared information</td>
</tr>
<tr>
<td>( c )</td>
<td>Total number of customers in a cluster</td>
</tr>
<tr>
<td>( d_{i,k} )</td>
<td>Information proprietor of ( k^{th} ) cluster</td>
</tr>
<tr>
<td>( CL )</td>
<td>Customer List</td>
</tr>
<tr>
<td>( b_k )</td>
<td>( k^{th} ) block</td>
</tr>
<tr>
<td>( i_d )</td>
<td>( k^{th} ) block identifier</td>
</tr>
<tr>
<td>( B )</td>
<td>Subset of ( q ) random blocks</td>
</tr>
<tr>
<td>( \tau_{signature} )</td>
<td>Re-sign key</td>
</tr>
<tr>
<td>( \rho_{k,\tau_{signature}} )</td>
<td>Re-Signature of revoked customer’s ( k^{th} ) block</td>
</tr>
</tbody>
</table>

### Communication Cost

- The proposed mechanism is a secure and efficient customer revocation mechanism.
- The existing customers in every cluster are relieved from the burden of verifying the revoked customer chunks and hence the communication cost of all the existing customers in every cluster is reduced. While performing auditing, the TPA retrieves only the combination of all the chunks (Challenge) instead of the complete information, therefore the communication cost of the TPA is saved. The size of the verification message is \( \{ h(\cdot) \} i \in G \) is \( c.(|n| + q) \) bits. The size of the verification proof \( \{ \rho, \chi_{d,e} \} i \leq d < e, \{ d_{i,k}, e_i \} i \in G \) is \( (2c + |p| + c(|id|) \) where \( c \) is the number of current customers in each cluster, \( e \) is the number of challenged chunks, the size of an element in \( G_1 \) is \(|p|\) and the size of a chunk identifier is \(|id|\). The overall transmission cost
Fig. 3: Computation of $\text{Re} - \text{sign}$ key by Information Proprietor

of a verifying task is $d(2c, |p| + e, \{|id| + |n| + |q|\})$ bits where $d$ is the number of information proprietors, $|n|$ is the size of element of set $[1, n]$.

Computation Cost:
The computation cost of an individual signature of a chunk is about $2\text{Exp} \cdot G_{1} + \text{Hash} \cdot G_{1} + \text{Mul} \cdot G_{1}$. As illustrated in the $\text{Re} - \text{Signature}$ function [See Algorithm 2, Phase 1, Part II] of the proposed scheme, the CSP initially checks the accuracy of the initial signature on a chunk and a fresh signature is estimated on the same chunk using $\text{Re} - \text{sign}$ key. The computation cost of the CSP to re-sign a chunk is $\text{Mul} \cdot G_{1} + \text{Hash} \cdot G_{1} + 2\text{Exp} \cdot G_{1} + 2\text{Pair}$. The proof of storage response generated by the CSP consists of the aggregated signatures and linear combination of sampled chunks. After receiving the proof of storage from the CSP, the computation cost for verification by an auditor is $e\cdot\text{Mul}\cdot G_{c}(\{\xi_{i}\}) + \text{Hash}^{2} \cdot G_{c} + \text{Mul}^{2} \cdot G_{c} + \text{Exp}^{2} \cdot G_{c}(\{p\}) + \text{Pair}^{2} \cdot G_{c}$.

The time taken by the IP to estimate the $\text{Re} - \text{sign}$ key ($\tau_{\text{Re-key}}$) is as shown in Fig. 3. The computation time is independent of the size of the cluster. The IP takes the keys from two existing customers and computes the $\text{Re} - \text{sign}$ key ($\tau_{\text{Re-key}}$) [Eq. no. 1]. Hence the time cost remains the same throughout. In comparison to the Panda scheme, the computation cost is reduced as we have allowed IP of the respective clusters to compute the $\text{Re} - \text{sign}$ key and send to the CSP. But in the Panda mechanism, the CSP estimates the $\text{Re} - \text{key}$ and re-signs the revoked user blocks, hence the computation cost increases.

The performance comparison between CRUPA and Panda schemes during customer revocation is shown in Fig. 4. In the proposed mechanism, the CSP securely and efficiently re-signs the respective cluster's revoked customer chunks and also saves the prevailing customer’s reckoning and correspondence resources. As depicted in Fig. 4, the CSP in CRUPA re-signs 500 chunks in 11 seconds while CSP in Panda takes 15 seconds, nearly 30 percent improvement.

The IP computes and delivers the $\text{Re} - \text{sign}$ key ($\tau_{\text{Re-key}}$) to the CSP. The time taken by the CSP to re-sign the revoked user chunks in CRUPA is less as compared to the Panda scheme [see Fig. 5]. In Panda scheme, the CSP computes the $\text{Re} - \text{key}$ as well as re-signs the revoked customer chunks. But in our scheme, CSP’s computation cost is completely reduced as CSP receives the $\text{Re} - \text{sign}$ key ($\tau_{\text{Re-key}}$) by the IP and only re-signs the revoked customer chunks. Hence our mechanism is secure and effective.

The system model that we have proposed consists of multiple clusters with their respective IP. Fig. 6.
shows the batch auditing for single cluster and multiple clusters compared with the existing schemes. When TPA receives individual customer’s auditing requests, the average auditing time taken by the TPA is more i.e., 290ms [27] [see Fig. 6]. By allowing the TPA to carry out the verification for cluster of customers auditing requests simultaneously i.e., single cluster auditing, then the average auditing time taken by the TPA in CRUPA is less (269ms) compared to Panda scheme [39] (272ms). In CRUPA, the TPA’s average auditing time cost is slightly more for multi-information proprietor cluster auditing.

Considering the TPA generates the different number of challenged information chunks, we respectively show the computation cost of the TPA and that of the CSP in integrity auditing phase in Fig. 7, Fig. 8, and Fig. 9. The computation overhead of the TPA during proof verification is as shown in Fig. 7. The computation overhead of TPA during proof verification in Shen scheme [27] i.e., for individual customers proof of possession sent by CSP to TPA varies from 0.3s to 12.5s while in CRUPA (single batch), it varies from 0.1s to 5.97s and for multiple batch, it varies from 0.19s to 7.67s. TPA takes more time to provide the verification proof for the individual customers proof of possession sent by the CSP. When multiple cluster dataowners sends auditing requests, the TPA randomly chooses a set of chunks i.e., generates challenge set and sends it to the CSP. Now, the CSP sends a single proof of possession for the received challenge set to the TPA. Hence, the time taken by the TPA to verify the proof in batch auditing (single and multiple clusters) is less compared to individual auditing.
Fig. 10: Processing time for different block numbers in the Setup phase

Compared with the time of proof verification, the time of challenge generation increases slowly [see Fig. 8.], just varying from 0.013s to 0.546s in [27] while in CRUPA (multiple clusters) it varies from 0.011s to 0.32s and for single cluster it varies from 0.001s to 0.15s. The time of challenge generation by the TPA in CRUPA is less compared to Shen scheme [27]

Fig. 9. shows the computation cost of CSP during proof generation. The computation cost of CSP is more in Shen scheme [27], as CSP provides proof for the individual customer’s challenged chunks. In the proposed scheme, TPA performs batch auditing. The TPA sends the challenge set for single batch or multiple batch auditing to the CSP. Now, the CSP provides proof of possession of the challenged blocks present in the challenge set i.e., the CSP takes less time to provide proof of possession for batch auditing as compared to the individual auditing.

The processing time for different block numbers [see Fig. 10.] in the Setup phase [30] is more compared to the CRUPA scheme. In the Setup phase of DHT – PA scheme, the CSP computes the tag for each uploaded blocks (i.e., TagGeneration phase) that includes the communication cost and computation cost while in CRUPA, the IP performs processing of all the blocks. Thus, the processing time for different block numbers in CRUPA is less compared to DHT – PA scheme.

8 Conclusions

In this paper, we have introduced a Collusion Resistant User Revocable Public Auditing (CRUPA) of distributed information in the cloud. The IP of the respective revoked customer cluster computes the Re – sign key (τ_{Re – key}) using regression method and transmits it to the cloud server. The computation cost of Re – sign key (τ_{Re – key}) using regression method by the IP has been significantly reduced. The algorithm supports effective and secure customer repudiation. Once the IP of the respective clusters revokes the customer, the CSP verifies the revoked customer chunks and securely re-signs with the Re – sign key (τ_{Re – key}) that allows the proposed scheme to be collusion resistant. Further, the algorithm supports multi-information proprietor batch auditing. The TPA in CRUPA takes less time to perform single batch auditing compared to the existing scheme. The proposed scheme is scalable as cloud information is effectively distributed among the existing customers of multiple clusters. Extensive experimental results demonstrate the efficiency and effectiveness of Collusion Resistant User Revocable Public Auditing (CRUPA) scheme. The processing time taken by the IP in the Setup phase is low. The computation cost of TPA and CSP is low in an integrity auditing phase. The limitation of the mechanism is that it has a slightly more auditing cost for multi-information proprietor batch auditing.

9 Declarations

9.1 List of Abbreviations

CRUPA : Collusion Resistant User Revocable Public Auditing
HAPS : Homomorphic Authenticable Proxy Re-signature
CSP : Cloud Service Provider
TPA : Third Party Auditor
jPBC : Java Pairing-Based Cryptography Library.

9.2 Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

9.3 Competing Interest

The authors declare that they have no competing interests.

9.4 Funding

This work is funded by Technical Education Quality Improvement Programme (TEQIP)(World Bank) University Visvesvaraya College of Engineering, Bengaluru and the University of Melbourne, Australia.
9.5 Authors' Contribution

GM carried out the experimental design, data analysis, interpretation, mathematical model design, and drafted the manuscript. UR carried out the mathematical model design. SR participated in the experimental design, RS participated in design of the study and performed the experimental analysis, RB participated in the design of the study and approved the final manuscript, VK participated in its design and coordination and approved the final manuscript, SS participated in design, conceptualization and approved the final manuscript and LM participated in conceptualization, implementation and approved the final manuscript. RagHAVendra S is a Associate Professor in the department of Computer Science and Engineering, Vivekanand a College of Engineering and Technology, Puttur. He received his Bachelor degree in Computer Science and Engineering from BMS Institute of Technology, Visvesvaraya Technological University, Bengaluru and Masters degree from R V College of Engineering, Visvesvaraya Technological University, Bengaluru. Dr. Raghavendra S has authored over 20 publications and his research interests include Cloud Computing, Applied Cryptography and Internet of Things. He is serving as editorial board member and Guest editor for a number of prestigious journals, like Elsevier, Springer, KIJP. He is a member of the IEEE.

9.6 Acknowledgements

We sincerely thank the Reviewers and the Editor for their valuable suggestions.

9.7 Authors' Information

GeetaCM, is a research scholar in the department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. She has received B.E. degree in Electronics and Communication from Basaveshwara College of Engineering, Bagalkot, Karnataka, India and M.E degree in Information Technology, from Bangalore University, Bengaluru, Karnataka, India. Her areas of interest are Cloud Computing, Cloud Security and Wireless Sensor networks. She is a student member of the IEEE.

UshaRani is a PG student in the department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. She has received B.E. degree in Computer Science and Engineering from Rao Bahadur Y Mahaballeshwarappa Engineering College, Karnataka, India and M.E degree in Computer Science and Engineering, from Bangalore University, Bengaluru, Karnataka, India. Her areas of interest are Cloud Computing, Big Data Analytics and IT Security.

Shreyas Raju R G is a UG student in the department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. He is currently pursuing B.E. degree in Information Science and Engineering. He has received diploma in Computer Science and Engineering, from Govt. Polytechnic, Chintamani, Karnataka, India. His areas of interest are Cloud Computing, Internet of Things, Cloud Security and E-Commerce. He is member of International Association of Engineers (IAENG).

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He has authored over 625 publications and seven text books including Mastering Cloud Computing published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He is one of the highly cited authors in Computer Science and Software Engineering worldwide (h-index=132, 92,500+ citations). Software technologies for Cloud Computing developed under his leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya is recognized as a Web of Science Highly Cited Researcher in 2016, 2017 and 2018 by Thomson Reuters, a Fellow of IEEE, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud Computing.

Venugopal K R is currently the Vice Chancellor, Bangalore University, Bengaluru. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science, Bangalore. He was awarded Ph.D in Economics from Bangalore University and Ph. D in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 72 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ and Digital Circuits and Systems etc., He has filed 101 patents. Dur-
ing his three decades of service at UVCE he has over 900 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems. He is a Fellow of IEEE and ACM Distinguished Educator.

9.8 Endnotes

1 The Re – sign key computed by the information proprietor using the regression technique mentioned in this paper must be considered as an indication. Since, the Re – sign key is computed by the IP using regression technique, it is not possible by the CSP by colluding with the revoked customer to break the Re – sign key and find the secret keys of the customers. Hence the proposed scheme preserves the privacy of the customers and is collusion resistant. Further, the proposed scheme supports effective and secure customer repudiation, multi-information proprietor batch auditing and is scalable.

References