Mara *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2020) 9:47
https://doi.org/10.1186/s13677-020-00188-5

Journal of Cloud Computing:
Advances, Systems and Applications

Research **Open Access**

CRUPA: collusion resistant user revocable public auditing of shared data in cloud

Geeta C. Mara[1*], Usharani Rathod[2], Shreyas Raju R. G.[2], Raghavendra S.[2], Rajkumar Buyya[3], Venugopal K. R.[4], S. S. Iyengar[5] and L. M. Patnaik[6]

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

---

*Correspondence: geetacmara@gmail.com
[1] Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru-560001, Karnataka, India
Full list of author information is available at the end of the article

Introduction

Cloud repository is one of the significant services provided by cloud computing [19]. It empowers the information possessor 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 [7]. 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 [17, 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 case 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. The limitation of the *Knox* scheme is that the customers need to distribute their private value with the public verifier and customer repudiation is expensive.

Wang et al. [39] proposed public verifying mechanism to bolster effective customer repudiation utilizing intermediary re-signatures, that acknowledge the distributed server to transform the signatures estimated by the repudiated 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 repudiated 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. By using regression tools, we permit the *IP* to estimate the $Re - sign$ key and transmit to the distributed server. As the $Re - sign$ key is computed by the *IP*, it is not possible for the malicious cloud to trace out the secret parameters of the existing customers.

Motivation: In the exisiting 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 "Related works" section. Several preliminaries are introduced in "Preliminaries" section. Problem definition, System model are discussed in "Problem statement" section. Mathematical Model using Regression Method, Security Analysis and Adversary Model are explained in "Mathematical model" section. 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 "The algorithm" section. In "Performance evaluation" section, Performance Evaluation results are analysed and "Conclusions" section contains the Conclusions.

Related works

Provable Data Possession [1], authorizes the auditor to publicly validate the integrity of information without fetching the whole information. Improving their earlier work for dynamic operations on data, Ateniese et al. [2] constructed another *PDP* scheme using symmetric keys. This scheme does not support public verification. Erway et al. [11] 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 ($DV-NRPDP$) 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. [20] 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 privacy and identity trackability. The drawback of the mechanism is that it has larger communication cost.

Dong et al. [10] have achieved data confidentiality against the semi-trusted cloud. They designed a protected, adequate and flexible data co-ordinated scheme. The mechanism does not accomplish information consistency. Yaun and Yu [46] have designed an auditing mechanism for distributed data sharing utilities illustrated by multi-user alterations, public auditing, adequate user repudiation and pragmatic reckoning auditing performance. The mechanism overcomes customer impersonation assault. The limitation is that it does not realize dependability and error detection .

Geeta et al. [13] have performed extensive review on the latest methods in information auditing and security in cloud computing. Shen et al. [26] have suggested an effective public verification convention. The proposed convention supports batch verification, blockless verification and lazy update. The limitation of the scheme is that the transmission cost is more in verification phase. Zhu et al. [48] have presented a secure anti-conspiracy information sharing mechanism for dynamic clusters in the cloud. The repudiated customer cannot fetch the original document though he conspires with the CSP. The proposed mechanism bolsters guaranteed key allocation, fine-grained access control and safe customer repudiation. Li et al. [18] have presented a security model and a formal definition for Ciphertext Policy-Attribute-Based Encryption ($CP - ABE$) scheme with effective attribute repudiation. The proposed mechanism is secure against conspiracy attack launched by the prevailing customers and the renunciated 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 public auditing for shared information in distributed repository supporting identity secrecy and trackability. The mechanism achieves data privacy by utilizing blind signature method. The limitation is that the mechanism incurs little overhead to accomplish the identity trackability. Hall et al. [14] 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. [8] introduced two conventions that can authorize protected and effective outsourcing of linear regression 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 to identify practical problems related to computation outsourcing to the cloud.

Verifiable data proprietorship mechanism [29] provides trustworthiness and individuality in an active, multi-user framework. By exploiting trustworthy hardware 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. [16] have introduced the integrity auditing scheme that supports public 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. [9] 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 procedures for distributed computing has been carried out by Garg and Bawa [12]. They have examined that the maximum of the prevailing procedures concentrate on integrity checks to distinctive data depository strategy. Simulations are carried out on C++ platform [33].

Raghavendra et al. [24] have proposed an effective token creation method, that enhances immune and productive label construction phase. A systematic composition is refined to encode the ordered keywords for secure label construction. The method reduces the cost of the information proprietor. Xu et al. [41] have introduced multi-authorization proxy re-encoding method. The scheme greatly reduces 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. [15] have outlined a group signature mechanism supporting the manageable connectivity. The convention supports reliability properties for e.g., confidentiality and connectivity. Privacy is not preserved by global linkability. Yu et al. [45] have suggested a distributed data integrity auditing with identity privacy-conserving convention for mobile cloud repository. The scheme affords anonymity to Third Party Auditor (TPA) and reliable label-updation. The mechanism incurs minimum reckoning, transmission and repository overhead.

Shen et al. [27] outlined a distant information integrity 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 information 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.

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 is does not preserve the privacy of the customers in

Table 1 Comparison of mechanisms for Public Honesty Verification with group customer repudiation

Authors	Concept	Performance	Advantages	Disadvantages
Tian et al. 2019 [31]	Public auditing for distributed cloud data with adept and reliable cluster management	The computational cost is significantly reduced in the verification phase.	Supports individuality privacy, data privacy and individuality trackability.	Low communication cost.
Shen et al. 2019 [27]	Individuality-based integrity auditing and information sharing with sensitive information hiding for reliable cloud repository.	The computation costs of TPA and CSP is higher with the increase of challenged blocks.	Supports information sharing with sensitive information hiding.	Computation cost of TPA in proof verification is high.
Jin et al. 2018 [16]	Dynamic and public auditing with fair negotiation for cloud information.	The scheme introduces additional overhead of data dynamics	Supports public verifiability, efficient data dynamics and fair disputes arbitration.	Reasonable overhead of data dynamics and dispute arbitration.
Tian et al. 2017 [30]	Public auditing mechanism for protected cloud repository based on Dynamic Hash Table (*DHT*).	The scheme has lower costs of storage, communication and computation.	Achieves higher updating efficiency and secure auditing.	Does not support various types of cloud data.
Xu et al. 2016 [41]	Multi-authority inter-mediary re-encryption based on *CPABE* for distributed repository system.	*MPRE* − *CPABE* reduces the estimation cost of the creation of key components.	Small reckoning cost of key allocation.	Additional computational period in *Setup* phase.
Wang et al. 2015 [39]	Public auditing for shared information with adept user renunciation.	No communication overhead to existing customers during customer repudiation, cloud has reduced computation cost.	Secure customer repudiation, public auditing.	Collusion of repudiated customer and cloud.
CRUPA	Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud(*CRUPA*)	Significant improvement in computation cost of *Re* − *sign* key by information proprietor, low processing time in *Setup* phase.	Supports multi-owner batch auditing, efficient customer revocation.	Average auditing time cost is more.

the cluster and is not collusion resistant i.e., the revoked customer colludes with the cloud.

Preliminaries

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

Bilinear map:

Consider two cyclic multiplicative groups $\mathbb{G}$ and $\mathbb{G}_{\mathrm{T}}$ of prime order p. $\mathrm{e} : \mathbb{G} * \mathbb{G} \rightarrow \mathbb{G}_{\mathrm{T}}$ is a bilinear map with the subsequent properties [6]:

- *Bilinear*: for all $u, v \in \mathbb{G}_1$ and $a, b \in Z_{\mathrm{p}}$, $\mathrm{e}(u^{\mathrm{a}}, v^{\mathrm{b}})=\mathrm{e}(u, v)^{\mathrm{ab}}$
- *Non* − *degeneracy*: $e(g, g) \neq 1$;
- *Computability*: An effective algorithm prevails for estimating map e.

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

Homomorphic authenticators

Homomorphic authenticators [1], permit a public validator to examine the integrity of information distributed 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_i, s_i), ρ_1 is the signature on chunk $b_1 \in Z_p$, and ρ_2 is the signature on chunk $b_2 \in Z_p$.

- Blockless auditability: Given ρ_1 and ρ_2, two arbitrary values β_1, β_2 in Z_p and a chunk $b'= \beta_1 b_1+\beta_2 b_2 \in Z_p$, 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, ρ_1 and ρ_2, two random values β_1, β_2 in Z_p and a chunk $b'= \beta_1 b_1+\beta_2 b_2 \in Z_p$, a customer without secret key (s_k), is unable to produce a legitimate signature ρ' on chunk b' by joining ρ_1 and ρ_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.

Proxy re-signatures

Proxy re-signatures [4] permit a semi-trusted intermediary to accomplish as an interpreter of signatures amidst two customers. Conventional proxy re-signature mechanisms [3, 4], 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 Authenticable 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*.

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 co-related, 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.

Problem statement

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.

Assumptions

(i) CSP is a semi-trusted entity.
(ii) Private channels (e.g., *SSL*) exist between each pair of entities.

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 prominance 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 renunciated 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.

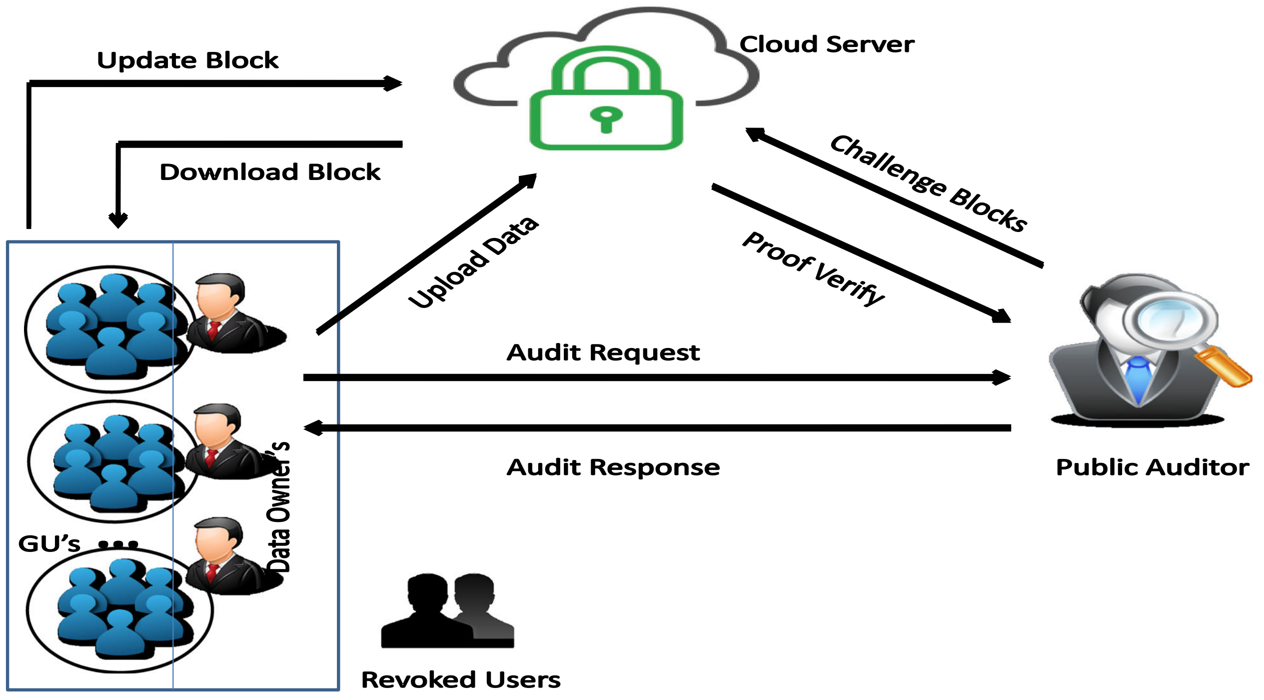

Fig. 1 Cloud storage model

Mathematical model

Computation of re-sign key (τ_{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 (τ_{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 revoked customer chunks with the $Re - sign$ key sent by the *IP*.

The Information Proprietor (*IP*) uses secret key τ_i and public key (pk_j) 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) ∈ [1,c]. H is a hash function with H: $\{0,1\}^* \rightarrow \mathbb{G}_1$. The computation of $Re - sign$ key using regression technique is as follows:

In order to secure secret key and public key, the *IP* substitutes τ_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))τ_i; a_2=(H(id_j)pk_j
By using a_1 and a_2 compute X_1, Y_1 and Z_1 :
$X_1 = 2(a_1)^{a_2}$; $Y_1 = 2(X_1)^{a_2}$; $Z_1 = X_1$-Y_1
The following steps shows the computation of $Re - sign$ key (τ_{Re-key}) using the Regression method:
$X_2 = 2(Y_1)^{a_2}$; $Y_2 = 2(X_1)^{a_2}$; $Z_2 = X_2$-Y_2
S(X) = X_1+X_2; S(Y) = Y_1+Y_2; S=Z_1+Z_2
mX=SX/2; mY=SY/2; mZ=SZ/2
S$X^2 = (X_1)^2+(X_2)^2$; S$Y^2 = (Y_1)^2+(Y_2)^2$;
SZ=$(Z_1)^2+(Z_2)^2$
δX^2 = S(X^2/Z)-(mX)^2
δY^2 = S(Y^2/Z)-(mY)^2
δZ^2 = S(Z^2/Z)-(mZ)^2

$$\tau_{Re-key} = 2\left[\Delta + 4)/\left(2\sqrt{\delta X^2 \delta Y^2} + 4\right)\right] \tag{1}$$

where $\Delta=\delta(X^2)+\delta(Y^2)+\delta(Z^2)$.

The [1]$Re - sign$ key (τ_{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.

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 (τ_i) and public key (pk_j), the *IP* substitutes τ_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 "Mathematical model" section]. Further in the regression technique, a_1 and a_2 are substituted in X_1 and Y_1. This procedure continues, and the final $^1Re - sign$ (τ_{Re-key}) Eq. 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 $Re - sign$ key and know the secret keys of the existing customers in the cluster. If they succeed then CSP and malicious customer would have achieved their goal of possessing the secret key. But, it is not possible for both CSP and the revoked customer to break the $Re - sign$ key and extract the secret key as the $^1Re - sign$ key is computed by the *IP* using regression technique which is a powerful tool that efficiently secures the secret key and does not allow any adversary by colluding with the CSP to find the secret key from the $Re - sign$ key.

Adversary model

Figure 2 shows the adversary model. The model consists of three entities: Cluster of customers with their respective *IP*, Cloud Service Provider (CSP) and Third Party Auditor (TPA). The *IP*, manager of the cluster monitors all the activities of the customers prevailing in the cluster. From Fig. 2, it is observed that if any one of the customer present in the cluster performs unwanted activity i.e., the malicious customer tries to retrieve the sensitive information or tries to hack the data, these activities are traced out by the *IP*. The *IP* immediately retrieves his credentials and revokes the malicious customer from the cluster. Further, the *IP* computes the $Re-sign$ key using the regression technique and sends to the CSP. After receiving the $Re-sign$ key, the CSP audits and re-signs the revoked customer chunks using the $Re - sign$ key. Next, the revoked customer might collude with the CSP [See Fig. 2], and tries to find the secret keys of the customers present in the cluster. Since, the $Re - sign$ key is computed by the *IP* using regression technique, it is not possible by the CSP or 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.

In the existing scheme [39], the semi-trusted CSP is permitted to compute the $Re - sign$ key. So, the mischievous CSP colludes with the revoked customer and tries to attack or hack the sensitive information cached in the cloud server. Hence, the existing scheme does not preserve the privacy of the customers and is not collusion resistant.

The semi-trusted CSP is permitted to compute 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 and they (CSP and revoked customer) can access the information cached in the cloud server. Hence the existing scheme is not collusion resistant.

The algorithm

System setup

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be multiplicative groups of prime order p, g be a generator of $\mathbb{G}_2$, e: $\mathbb{G}_1$ * $\mathbb{G}_2$ → $\mathbb{G}_T$ be a bilinear map. $H(\cdot)$ is a secure map-to-point hash function:($\{0,1\}^k \rightarrow \mathbb{G}_1$) that map strings consistently to $\mathbb{G}_1$. Another hash function $h(\cdot)$: $\mathbb{G}_1 \rightarrow Z_p$ maps group element of $\mathbb{G}_1$ 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.

PhaseI: 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 the 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

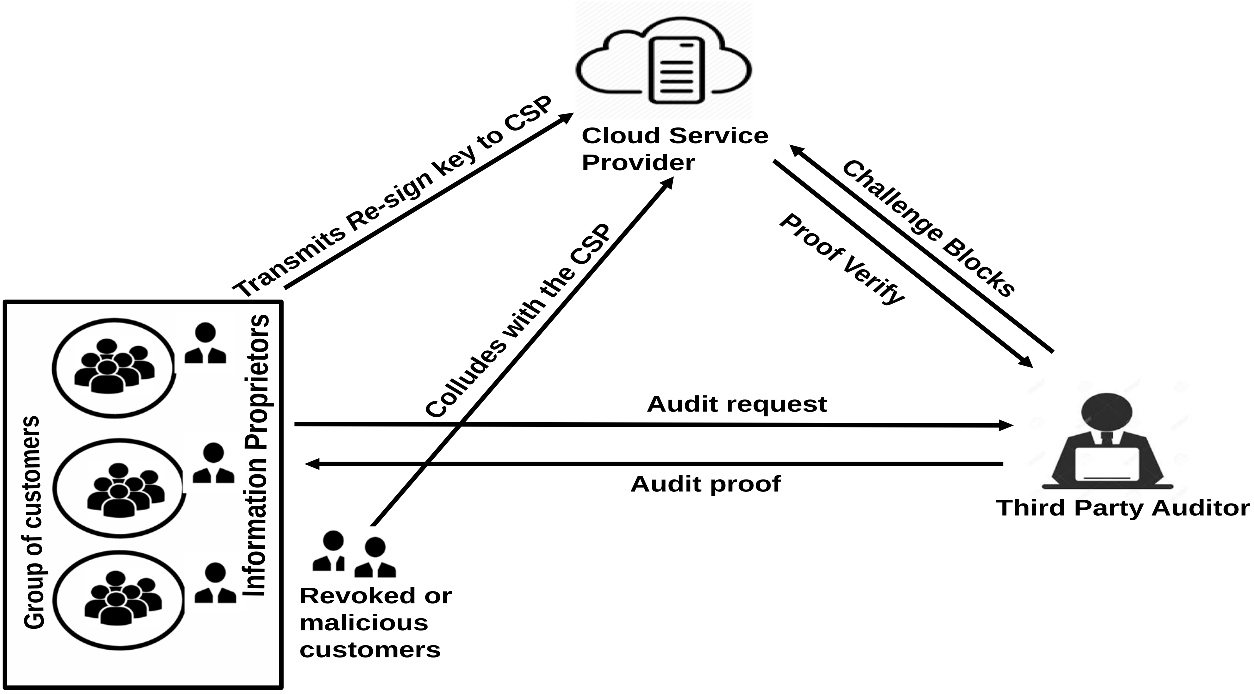

Fig. 2 Adversary model

Information Proprietor d, the private key is $s_k = \tau_d \in Z_p$ 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, ϕ) to the CSP, where $\phi = \{\rho_{k,i}\}_{1 \le i \le n}$. Now the existing customers of all the clusters retrieve their respective chunks, perform modifications, sign with their secret key (τ) and upload to the server as described in the function *SignatureGen* [See Algorithm 2, *PhaseI*, 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 $^1Re-sign$ key (τ_{Re-key}) utilizing the regression method as in Eq. 1, and transmits it to the CSP. After obtaining the $Re-sign$ key (τ_{Re-key}) the CSP checks the integrity of the revoked customer chunks and re-signs with the τ_{Re-key} as illustrated in the function *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 $Re-sign$ key (τ_{Re-key}). By colluding with the revoked customer, the CSP cannot find the secret keys of the existing customers' as the $^1Re-sign$ 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.

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 *ClusterChal* [See Algorithm 2, *PhaseII*, Part I and Part II] generates $challenge = \{(i, \xi_i)\}_{i \in E}$ 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 $\{\rho, \{\chi_d\}_{1 \le d \le D}, \{id_i, e_i\}_{i \in E}\}$.

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

```
Function 1: GenerateKey
    Function: Generates the system public and secret parameters.
    Input: $c$, $d_1$, global parameter ($g$, $\mathbb{G}_1$, $Z_p{}^*$)
    Output: $pk_i$, $sk_i$, $CL$, ($v_d$, $w_d$, $g_d$, $J_d$)
Assume $d \in (1,....D)$ information proprietors of their respective clusters in the system.
Choose random elements $\tau \in Z_p, J \in \mathbb{G}_1$
Compute $v=g^{\tau}$ and $w=J^{\tau}$
For a particular information proprietor $d$, the secret key, sk=$\tau_d \in Z_p$
Corresponding public parameters are ($v_d$, $w_d$, $g_d$, $J_d$)=($g^{\tau_d}$, ${J_d}^{\tau_d}$, $g$, $J_d$) where $J \in \mathbb{G}_1$
Respective information proprietor of each cluster generates the public and secret parameters for existing customers as:
Start:
for each $i$ upto $c$.
Generate random number $\tau_i$ from $Z_p{}^*$.
Compute Public key $pk_i$=g$^{\tau_i}$.
Assign Private key $sk_i$= $\tau_i$.
End.
$d_1$ of respective clusters creates $CL$.
$CL$ is public and signed by $d_1$.
```

computation cost of the public verifier. For the public verifier's *challenge* request, *Challenge*= $\{(i,\xi_i)\}_{i \in E}$, the CSP utilizes the bilinear aggregate signature [5], and sends one group element ρ instead of $\{\rho_d\}_{1 \leq d \leq D}$. 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.

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 ie., 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 (τ_{Re-key}) computed by the *IP* in Eq. 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 (τ_{Re-key}) as illustrated in Eq. 1, utilizing the regression method and then it sends to the CSP. Since the *Re − sign*

```
Algorithm 1: Homomorphic Authenticable Proxy Re-signature Scheme (HAPS) using Regression method
Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of order $p$, $g$ be a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1{}^*\mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map, $w$ be another generator of $\mathbb{G}_1$. The global parameters are ($e$, $p$, $\mathbb{G}_1$, $\mathbb{G}_2$, $g$, $w$, $H$) where $H$ is a hash function with $H$:(0,1)$^*\rightarrow \mathbb{G}_1$.
    Input: $\tau_i$. $b_k \in Z_p$ and $id_k$ where $k \in$ [1,$n$], $w$, and $\tau_{Re-key}$
    Output: $\rho_k$, ${\rho_k}^{(\tau_{Re-key})}$
KeyGen:
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 ($\tau_{Re-key}$) using regression method [eq. no.1]
    $(\tau_{Re-key})=2[(\Delta+4)/(2\sqrt{sigX^2 sigY^2}+4)]$
Sign:
Existing customer $c_i$ generates the signature ($\rho_k$) on block $b_k$ as:
$\rho_k=(H(id_k)\ w^{b_k})^{\tau_i}$
Re-sign:
CSP (Proxy) verifies the integrity and re-signs the revoked customer chunks as:
The CSP (proxy) first verifies that $e(\rho_k, g)$= $((H(id_\mathrm{k})w^{b_k}), pk_\mathrm{e})$.
If the auditing result is 0, the agent outputs $\perp$
Otherwise, the $IP$ computes $Re-sign$ key ($\tau_{Re-key}$) using regression method and sends to the CSP (proxy) to re-sign the revoked customer chunks.
CSP (proxy) re-signs the revoked customer chunks as ${\rho_k}^{(\tau_{Re-key})}$=$(H(id_k)w^{b_k})^f$
Verify:
The verifier outputs 1, if $e(\rho, g)$= $e(H(id)\ w^b, pk_i)$ and 0 otherwise
```

key (τ_{Re-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 2: CRUPA: Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud

```
Phase I: Secure Re-signing of Revoked Customer Blocks by CSP
    Input: $\tau_i$, $\tau_d$, $b_k \in Z_p$, $id_k$ where $k \in [1, n]$, $J, d, D, F_d, b_{d,i}$, $c_i$ and $\tau_{e \rightarrow f}$
    Output: $\rho_k$, $\rho_{d,i}$, $\rho_k{}^{(\tau_{Re-key})}$
Part I: SignatureGen
Every information proprietor $d$, divides his file $F_d$ into ($b_{d,1}$, ......$b_{d,n}$) blocks, where $d \in$ { 1,....$D$ }.
Computes signature $\rho_{d,i}$ on every block $b_{d,i}$:
$\rho_{d,i}$=$(H(id_i)\ J_d{}^{b_{d,i}})^{\tau_d} \in \mathbb{G}_1$ ($i$=1....$n$)
$IP$ sends ($F_d$, $\phi$) to the CSP, where $\phi$=$\{\rho_{k,i}\}_{1 \le i \le n}$
Existing customers $c_i$ in every cluster generates the signature ($\rho_k$) on block $b_k$ as:
for each $b_k$ with $id_k$.
    Compute $\rho_k$= $(H(id_k)\ J^{b_i})^{\tau_i}$.
end for
Part II: ReSignature
CSP verifies the integrity and re-signs the revoked customer blocks as:
The CSP first verifies that e($\rho_k$, g) $\overset{?}{=}$ (($H(id_k)\ J^{b_k}$), $pk_{\mathrm{e}}$).
If the auditing result is 0, the CSP outputs $\perp$
else $IP$ computes $Re - sign$ key $\tau_{Re-key}$ using regression method and sends to the CSP to re-sign the revoked customer block.
CSP re-signs the revoked customer blocks $\rho_k{}^{(\tau_{Re-key})}$=$(H(id_k)J^{b_k})^{\tau_f}$
The $IP$ performs re-siging, removes customer $u_e$'s $id$ from $CL$, and signs a new $CL$.
Phase II: Secure Multi-Information Proprietor Cluster Auditing for Shared Information by Third Party Auditor
    Input: $d$, $E$, $Challenge$
    Output: Auditing message, verification message
Part I: ClusterChal
The TPA creates verification message as follows: For every cluster's, $IP$ $d$'s auditing request, the TPA selects a arbitrary $q$ element subset $E$=\{ $e_1$, ....$e_q$\} of set \{1, $n$ \}. For every element $i \in E$, the TPA selects arbitrary value $v_i$. The TPA delivers the $challenge$=$\{(i, \xi_i)\}_{i \in E}$ to the CSP.
Part II: ClusterProof
Upon securing the $challenge$, for every $IP$ $d$ ($d \in$ \{1, .....$D$\}), the CSP computes:
$\chi_d$=$\sum_{i=e_1}^{e_q} v_i\ b_{d_i}$ and $\rho$=$\prod_{d=1}^{D}(\prod_{i=e_1}^{e_q} \rho_{d,i}{}^{v_i})$
The CSP responses the TPA with \{$\rho$, $\{\chi_d\}_{1 \le d \le D}$,\{$id_i$, $e_i$\} $_{i \in E}$\}
Part III: ClusterVerify TPA accepts the storage proof from the CSP and approves the response by analyzing the verification equation:
e($\rho$, g) $\overset{?}{=}$ $\prod_{d=1}^{D}$ e($\prod_{i=e_1}^{e_q}$ $[H(id_i)]^{\xi_i}.(J_d)^{\chi_d}$, $v_d$)
If the output is 1, the TPA considers that the sincerity of total chunks in shared information S is appropriate, else the TPA outputs 0.
```

Performance evaluation

To evaluate our proposed mechanism, a prototype system is implemented utilizing Java with Java Pairing-Based Cryptography Library (*jPBC*) [21] and the experiments are conducted on a *PC* with windows 7, Intel(*R*) Core(*TM*) *i*5-5200U, CPU @2.20GHz, 8GB RAM. In the following experiments, we assume the size of element in $\mathbb{G}_1$ or Z_p is

Table 2 Summary of the Notations used in the Algorithm 2

Notation	Description
$\mathbb{G}_1$, $\mathbb{G}_2$	Multiplicative groups of prime order p
g	Generator polynomial of $\mathbb{G}_1$
$H(\cdot)$	Secure map-to-point hash function
$h(\cdot)$	hash function maps cluster element of $\mathbb{G}_1$ consistently to Z_p
tag_F	Tag of file F
Pk	Public key
Sk	Secret key
τ_1	Signature on block b_1
n	Total number of chunks in shared data
S	Shared information
c	Total number of customers in a cluster
$d_{1,i}$	Information proprietor of 1^{st} cluster
CL	Customer List
b_k	k^{th} block
id_k	k^{th} block identifier
E	Subset of q random blocks
τ_{Re-key}	Re-sign key
$\rho_k{}^{(\tau_{Re-key})}$	Re-Signature on revoked customer's k^{th} block
Public parameters	(v_d, w_d, g_d, J_d)
$F_{d_1,i}$	File owned by information proprietor (d_1) of i^{th} cluster
ϕ	Set of signatures on entire chunks in distributed information.
$Exp\ \mathbb{G}_1$	One exponentiation in $\mathbb{G}_1$
$Mul\ \mathbb{G}_1$	One multiplication in $\mathbb{G}_1$
$Pair$	Pairing operation on e: $\mathbb{G}_1 * \mathbb{G}_2 \rightarrow \mathbb{G}_T$
$m\text{-}MulExp^t_{\mathbb{G}}$	t m term exponentiations $\sum_{i=1}^{m} g^{a_i}$

$| p |$=160 bits. The size of an element of Z_q is $| q |$=80 bits. The size of each chunk is 4KB.

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 chunk 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 $\{(i, \xi_i)\}_{i \in E}$ is $e.(| n|+| q|)$ bits. The size of the verification proof $\{\rho, \{\chi_d\}_{1 \leq d \leq D}, \{id_i, e_i\}_{i \in E}\}$ is $(2c.|p| + e.(|id|)$ where c is the number of current customers in each cluster, e is the number of challenged chunks, the size of an element in $\mathbb{G}_1$ is $| p|$ and the size of a chunk identifier is $|id|$. The overall transmission cost 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 $2Exp\ \mathbb{G}_1+\ Hash\ \mathbb{G}_1+Mul\ \mathbb{G}_1$. As illustrated in the $Re-Signature$ 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 $Re-sign$ key. The computation cost of the CSP to re-sign a chunk is $Mul\mathbb{G}_1+Hash\mathbb{G}_1+2Exp\ \mathbb{G}_1+2Pair$. 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-$MulExp^1{}_{\mathbb{G}}(|\xi_i|)+Hash^e{}_{\mathbb{G}}+Mul^2{}_{\mathbb{G}}+Exp^2{}_{\mathbb{G}}(|p|)+Pair^2{}_{\mathbb{G},\mathbb{G}}$

The time taken by the *IP* to estimate the $Re-sign$ key (τ_{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 $Re-sign$ key (τ_{Re-key}) [Eq. 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 $Re-sign$ key and send to the CSP. But in the *Panda* mechanism, the CSP estimates the $Re-key$ 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 s while CSP in *Panda* takes 15 s, nearly 30 percent improvement.

The *IP* computes and delivers the $Re-sign$ key (τ_{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 $Re-key$ as well as re-signs the revoked customer chunks. But in our scheme, CSP′s computation cost is completely reduced as CSP receives the $Re-sign$ key (τ_{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*. Figure 6 shows

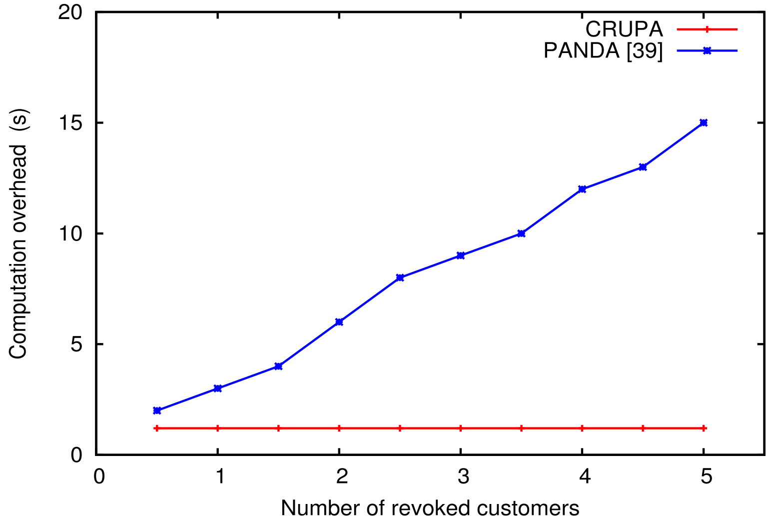

Fig. 3 Computation of $Re-sign$ key by Information Proprietor

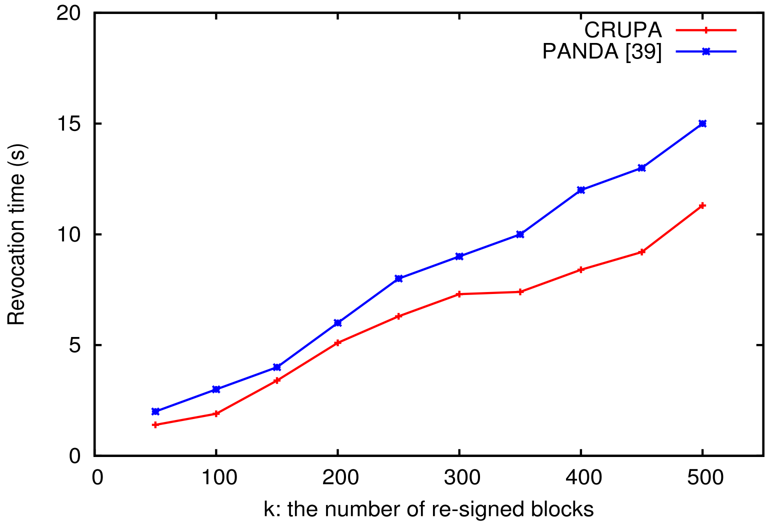

Fig. 4 Revocation time with re-signing of blocks by the CSP

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 Figs. 7, 8 and 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] ie., 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

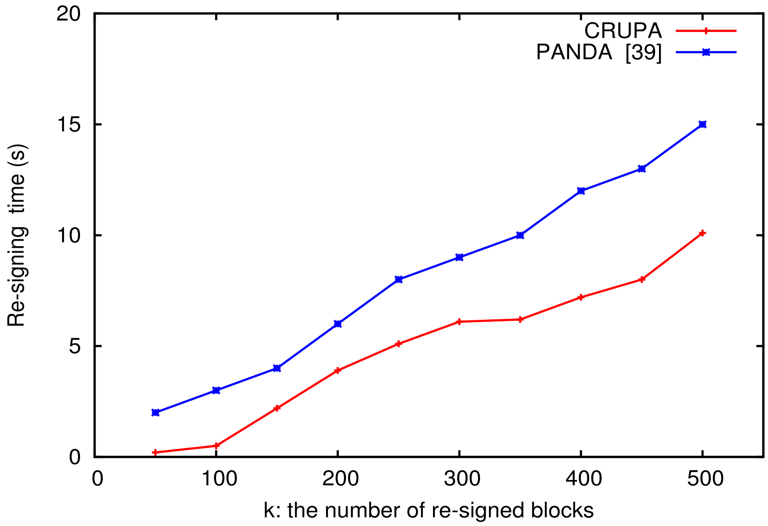

Fig. 5 Re-signing time of the blocks by Cloud Service Provider

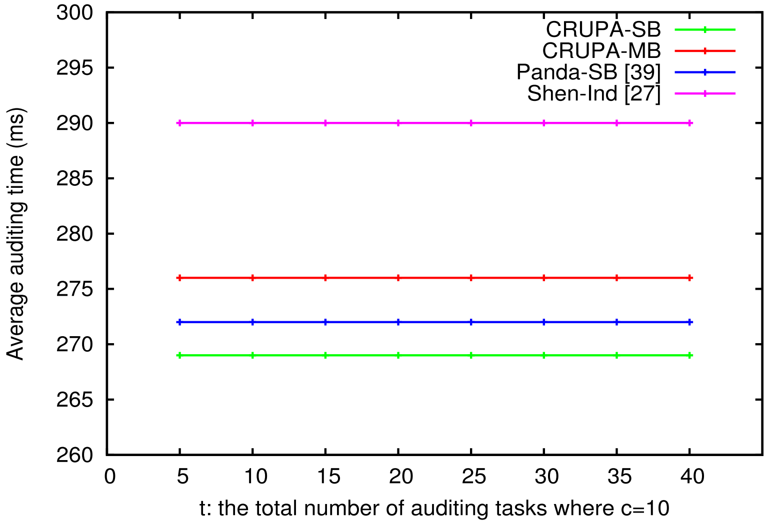

Fig. 6 Impact of t on average auditing time (ms) per task where c=10

taken by the TPA to verify the proof in batch auditing (single and multiple clusters) is less compared to individual auditing.

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 form 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]

Figure 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

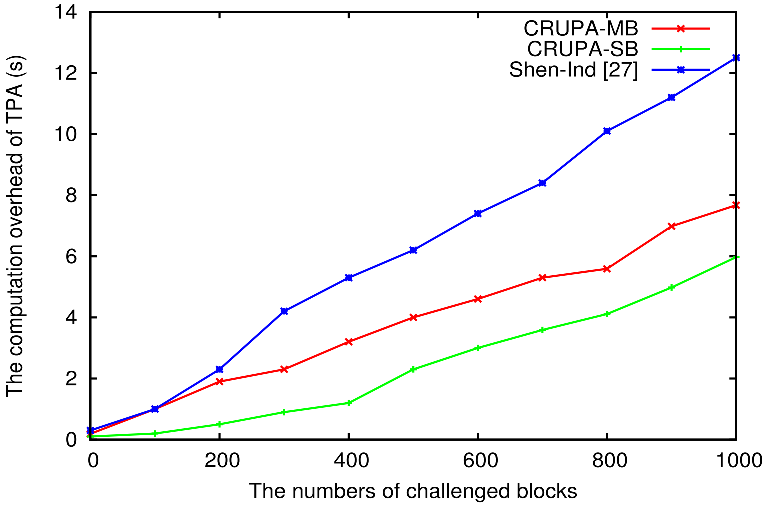

Fig. 7 Computation overhead of TPA during proof verification

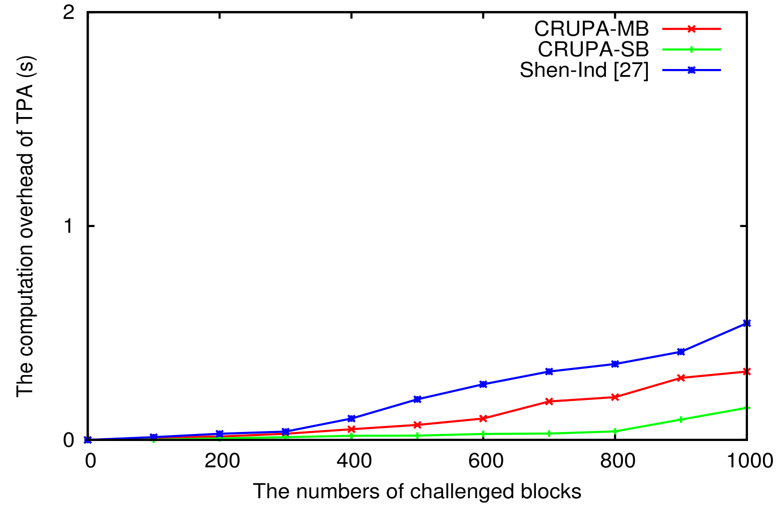

Fig. 8 Computation overhead of TPA during challenge generation

processing time for different block numbers in *CRUPA* is less compared to $DHT-PA$ scheme.

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 the *integrity auditing* phase. The limitation of the mechanism is that it has a slightly more auditing cost for multi-information proprietor batch auditing.

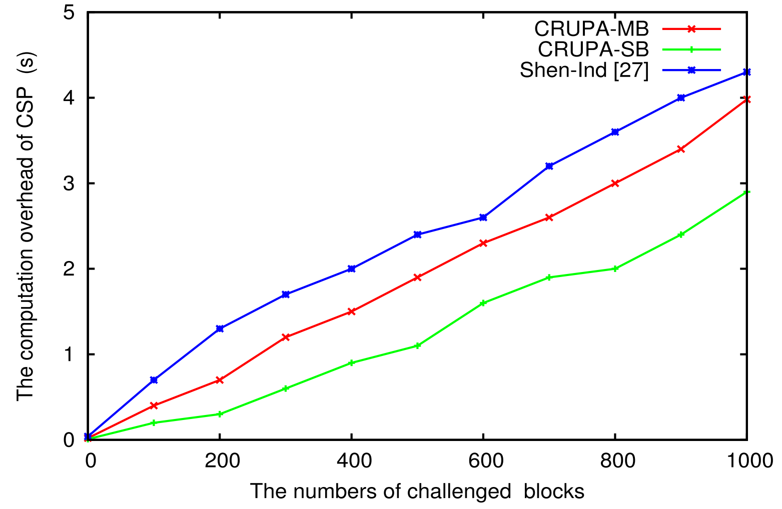

Fig. 9 Computation overhead of CSP during proof generation

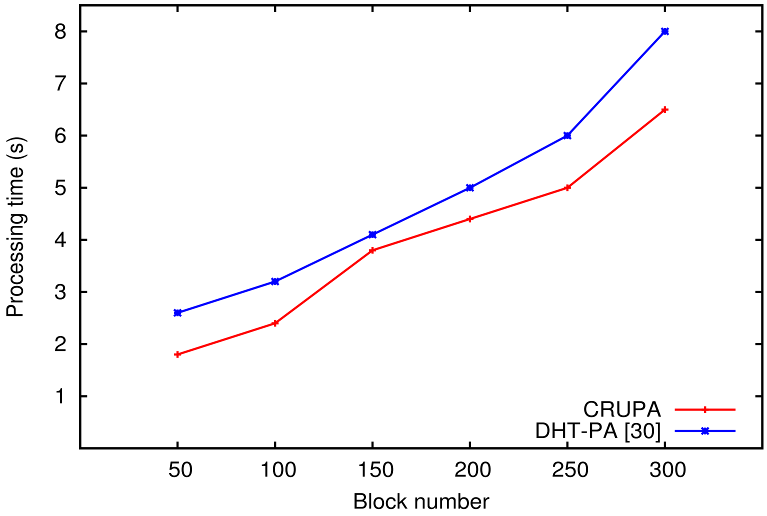

Fig. 10 Processing time for different block numbers in the *Setup* phase

Footnote

The $^{1}Re - sign$ 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.

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

Acknowledgments

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

Authors' contributions

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.

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, BigData 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*).

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, KJIP. He is a member of the IEEE.

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=136, 98,500+ 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, Madr as. 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. During 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.

About the Authors

S S Iyengar is currently Ryder Professor, Florida International University, USA. He was Roy Paul Daniels Professor and Chairman of the Computer Science Department of Louisiana State University. He heads the Wireless Sensor Networks Laboratory and the Robotics Research Laboratory at USA. He has been involved with research in High Performance Algorithms, Data Structures, Sensor Fusion and Intelligent Systems, since receiving his Ph.D degree in 1974 from MSU, USA. He is Fellow of IEEE and ACM. He has directed over 40 Ph.D students and 100 post graduate students, many of whom are faculty of Major Universities worldwide or Scientists or Engineers at National Labs/Industries around the world. He has published more than 800 research papers and has authored/co-authored 6 books and edited 7 books. His books are published by John Wiley and Sons, CRC Press, Prentice Hall, Springer Verlang, IEEE Computer Society Press etc.. One of his books titled Introduction to Parallel Algorithms has been translated to Chinese. He is a Fellow of IEEE and a Fellow of ACM.
L M Patnaik is currently Senior Scientist, Consciousness Studies Program, National Institute of Advanced Studies, Indian Institute of Science, India. He was a Vice Chancellor, Defence Institute of Advanced Technology, Pune, India and was a Professor since 1986 with the Department of CSA, Indian Institute of Science, Bengaluru. During the past 35 years of his service at the Institute he has over 1150 research publications in refereed International Journals and Conference Proceedings. He has received twenty national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to High Performance Computing and Soft Computing. His areas of research interest have been Parallel and Distributed Computing, Mobile Computing, CAD, Soft Computing and Computational Neuroscience. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow the Academy of Science for the Developing World and a Fellow of IEEE.

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.

Availability of data and materials

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

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru-560001, Karnataka, India. [2]Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bengaluru, India. [3]Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia. [4]Bangalore University, Bengaluru, India. [5]Department of Computer Science and Engineering, Florida International University, Miami, USA. [6]INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bengaluru, India.

Received: 28 May 2019 Accepted: 2 July 2020

References

1. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, Song D (2007) Provable Data Possession at Untrusted Stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. pp 598–609. https://doi.org/10.1145/1315245.1315318
2. Ateniese G, Di Pietro R, Mancini LV, Tsudik G (2008) Scalable and Efficient Provable Data Possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks. ACM. pp 1–9. https://doi.org/10.1145/1460877.1460889
3. Ateniese G, Hohenberger S (2005) Proxy Re-signatures: New Definitions, Algorithms, and Applications. Proc 12th ACM Conf Comput Commun Secur:310–319
4. Blaze M, Bleumer G, Strauss M (1998) Divertible Protocols and Atomic Proxy Cryptography. Int Conf Theory Appl Cryptographic Tech:127–144. https://doi.org/10.1007/bfb0054122
5. Boneh D, Gentry C, Lynn B, Shacham H (2003) Aggregate and Verifiably Encrypted Signatures from Bilinear Maps:416–432. https://doi.org/10.1007/3-540-39200-9_26
6. Boneh D, Lynn B, Shacham H (2004) Short Signatures from the Weil Pairing. J Cryptol 17(4):297–319
7. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Futur Gener Comput Syst 25(6):599–616
8. Chen F, Xiang T, Lei X, Chen J (2014) Highly Efficient Linear Regression Outsourcing to a Cloud. IEEE Trans Cloud Comput 2(4):499–508
9. Dong X, Yu J, Luo Y, Chen Y, Xue G, Li M (2014) Achieving an Effective, Scalable and Privacy-Preserving Data Sharing Service in Cloud Computing. Comput Secur 42:151–164
10. Dong X, Yu J, Zhu Y, Chen Y, Luo Y, Li M (2015) SECO: Secure and Scalable Data Collaboration Services in Cloud Computing. Comput Secur 50:91–105
11. Erway CC, Küpçü A, Papamanthou C, Tamassia R (2015) Dynamic Provable Data Possession. ACM Trans Inf Syst Secur (TISSEC) 17(4):213–222
12. Garg N, Bawa S (2016) Comparative Analysis of Cloud Data Integrity Auditing Protocols. J Netw Comput Appl 66:17–32
13. Geeta CM, Raghavendra S, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM (2018) Data Auditing and Security in Cloud Computing: Issues, Challenges and Future Directions. Int J Comput (IJC) 28(1):8–57
14. Hall R, Fienberg SE, Nardi Y (2011) Secure Multiple Linear Regression based on Homomorphic Encryption. J Off Stat 27(4):669
15. Hwang JY, Chen L, Cho HS, Nyang D (2015) Short Dynamic Group Signature Scheme Supporting Controllable Linkability. IEEE Trans Inf Forensic Secur 10(6):1109–1124
16. Jin H, Jiang H, Zhou K (2018) Dynamic and Public Auditing with Fair Arbitration for Cloud Data. IEEE Trans Cloud Comput 6(3):680–693
17. Li J, Yan H, Zhang Y (2018) Certificateless Public Integrity Checking of Group Shared Data on Cloud Storage. IEEE Trans Serv Comput. https://doi.org/10.1109/tsc.2018.2789893
18. Li J, Yao W, Han J, Zhang Y, Shen J (2017) User Collusion Avoidance CP-ABE with Efficient Attribute Revocation for Cloud Storage. IEEE Syst J 12(2):1767–1777
19. Liu F, Tong J, Mao J, Bohn R, Messina J, Badger L, Leaf D (2011) NIST Cloud Computing Reference Architecture. NIST Spec Publ 500(2011):1–28
20. Luo Y, Xu M, Fu S, Wang D, Deng J (2015) Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation. In: Trustcom/BigDataSE/ISPA, IEEE, vol. 1. pp 434–442. https://doi.org/10.1109/trustcom.2015.404
21. Pairing Based Cryptography (PBC) Library. http://crypto.stanford.edu/pbc/,2014..
22. Pattar S, Buyya R, Venugopal KR, Iyengar S, Patnaik L (2018) Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions. IEEE Commun Surv Tutor 20(3):2101–2132
23. Raghavendra S, Doddabasappa PA, Geeta CM, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM (2016) Secure Multi-Keyword Search and Multi-User Access Control over an Encrypted Cloud Data. Int J Inf Process 10(2):51–61
24. Raghavendra S, Geeta CM, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM (2015) MSIGT: Most Significant Index Generation Technique for Cloud Environment. In: Proceedings of the Annual IEEE India Conference (INDICON). pp 1–6. https://doi.org/10.1109/indicon.2015.7443531
25. Ren K, Wang C, Wang Q (2012) Security Challenges for the Public Cloud. IEEE Internet Comput 16(1):69–73
26. Shen J, Shen J, Chen X, Huang X, Susilo W (2017) An Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data. IEEE Trans Inf Forensic Secur 12(10):2402–2415

27. Shen W, Qin J, Yu J, Hao R, Hu J (2019) Enabling Identity-Based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage. IEEE Trans Inf Forensic Secur 14(2):331–346
28. Tarannum S, Aravinda B, Nalini L, Venugopal KR, Patnaik LM (2006) Routing Protocol for Lifetime Maximization of Wireless Sensor Networks. In: International Conference on Advanced Computing and Communications. IEEE. pp 401–406. https://doi.org/10.1109/adcom.2006.4289925
29. Tate SR, Vishwanathan R, Everhart L (2013) Multi-User Dynamic Proofs of Data Possession using Trusted Hardware. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy. pp 353–364. https://doi.org/10.1145/2435349.2435400
30. Tian H, Chen Y, Chang CC, Jiang H, Huang Y, Chen Y, Liu J (2017) Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage. IEEE Trans Serv Comput 10(5):701–714
31. Tian H, Nan F, Jiang H, Chang CC, Ning J, Huang Y (2019) Public Auditing for Shared Cloud Data with Efficient and Secure Group Management. Inf Sci 472:107–125
32. Tian JF, Guo RF, Jing X (2019) Stern-Brocot-based Non-Repudiation Dynamic Provable Data Possession. IEEE Access. https://doi.org/10.1109/access.2019.2916173
33. Venugopal KR, Buyya R (2013) Mastering C++. McGraw-Hill Education, New Delhi
34. Venugopal KR, Rajan EE, Kumar PS (1998) Performance Analysis of Wavelength Converters in WDM Wavelength Routed Optical Networks. In: Proceedings. Fifth International Conference on High Performance Computing (Cat. No. 98EX238). IEEE. pp 239–246. https://doi.org/10.1109/hipc.1998.737994
35. Venugopal KR, Rajan EE, Kumar PS (1999) Impact of Wavelength Converters in Wavelength Routed All-Optical Networks. Comput Commun 22(3):244–257
36. Venugopal KR, Srinivasa KG, Patnaik LM (2009) Soft Computing for Data Mining Applications. Springer. https://doi.org/10.1007/978-3-642-00193-2
37. Wang B, Li B, Li H (2012) Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud. Int Conf Appl Crypt Netw Secur:507–525. https://doi.org/10.1007/978-3-642-31284-7_30
38. Wang B, Li B, Li H (2014) Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud. IEEE Trans Cloud Comput 2(1):43–56
39. Wang B, Li B, Li H (2015) Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud. IEEE Trans Serv Comput 8(1):92–106
40. Wu TY, Tseng YM, Huang SS, Lai YC (2017) Non-Repudiable Provable Data Possession Scheme with Designated Verifier in Cloud Storage Systems. IEEE Access 5:19333–19341
41. Xu X, Zhou J, Wang X, Zhang Y (2016) Multi-Authority Proxy Re-encryption Based on CPABE for Cloud Storage Systems. J Syst Eng Electron 27(1):211–223
42. Yan H, Li J, Han J, Zhang Y (2016) A Novel Efficient Remote Data Possession Checking Protocol in Cloud Storage. IEEE Trans Inf Forensic Secur 12(1):78–88
43. Yan H, Li J, Zhang Y (2019) Remote Data Checking with a Designated Verifier in Cloud Storage. IEEE Syst J. https://doi.org/10.1109/jsyst.2019.2918022
44. Yang G, Yu J, Shen W, Su Q, Fu Z, Hao R (2016) Enabling Public Auditing for Shared Data in Cloud Storage Supporting Identity Privacy and Traceability. J Syst Softw 113:130–139
45. Yu Y, Ni J, Xia Q, Wang X, Yang H, Zhang X (2016) SDIVIP2: Shared Data Integrity Verification with Identity Privacy Preserving in Mobile Clouds. Concurr Comput Pract Experience 28(10):2877–2888
46. Yuan J, Yu S (2015) Public Integrity Auditing for Dynamic Data Sharing with Multi-User Modification. IEEE Trans Inf Forensic Secur 10(8):1717–1726
47. Zhu Y, Wang H, Hu Z, Ahn GJ, Hu H, Yau SS (2011) Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds. In: Proceedings of the 2011 ACM Symposium on Applied Computing. pp 1550–1557. https://doi.org/10.1145/1982185.1982514
48. Zhu Z, Jiang R (2015) A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud. IEEE Trans Parallel Distrib Syst 27(1):40–50

Publisher's Note