SARRC: Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud Using Regression Method

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

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

***Abstract*—The demand of deploying information has enormously increased within the last decade. Numerous distributed computing service suppliers have emerged (for eg., Microsoft Azure, Dropbox) in order to satisfy the requirements for information repository and high performance computation. The customers using the cloud repository services can conveniently arrange as a cluster and distribute information among themselves. Information proprietor computes the signatures for every chunk and deploys in the distributed server in order to allow the public verifier to perform public integrity verification on the information stored on the cloud server. In Panda scheme [1], by using the proxy re-signatures, Cloud Service Provider (CSP) verifies and re-signs the revoked customer chunks in favor of the existing customers. The malicious CSP might use the Re-sign key deliberately to transform the signature of one customer to another. Apart from this, conspiracy amidst the mischievous cloud server and the repudiated customer reveals the private key information of the customers present in the cluster. We propose Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud Using Regression Method. Re-key computed by the information proprietor using regression method is highly secure and the mischievious cloud cannot detect the private information of the customers in the cluster. Our mechanism is collusion resistant, reduces computation cost of re-sign key by information proprietor and in addition CSP securely performs auditing and re-signing of repudiated customer chunks.**

***Keywords*—Cloud Computing, User Revocation, Public Auditing, Proxy Signatures, Batch Auditing, Regression method.**

I. Introduction

Information repository is one of the greatest elementary assistance afforded by cloud suppliers. With information repository and sharing services, customers are permitted to update and distribute their outsourced information in the cloud server anyplace and at any moment [2]. But one of the worrying factors of the information proprietor is the sincerity of the deployed information in the cloud server. The honesty of deployed information is adulterated because of negligence of people or disruption of hardware/software [3]. Hence, public honesty verification is required to assure the customers that the information is precisely deployed in the cloud.

In the recent past, various schemes [4], [5] have been suggested to authorize not only information proprietor itself but also a public examiner to effectively carry out the sincerity verification without fetching the complete information from the cloud, known as public verification. In these schemes, data is segregated into numerous chunks, where every chunk is individually signed by the information proprietor and arbitrary integration of entire chunks instead of the complete information is retrieved at the time of integrity verification. A public examiner might be a information user (e.g., researcher) who would like to use the proprietors information *via* the cloud or a public examiner who can furnish proficient sincerity verification services.

Distributing information with various customers is one of the most attractive characteristic that inspires cloud repository. Hence, it is also required to assure the honesty of distributed information in the cloud is accurate. Existing mechanisms [6], [1] discuss on how to verify the sincerity of the distributed information. In this scenario, customers can conveniently alter and distribute information as a cluster with the cloud administrations.

Oruta [6], a public verification scheme for distributed information in cloud protects the individuality privacy of cluster customers from the public verifier. The scheme suffers from inadequate customer repudiation. In order to retain the individuality secrecy of customers from the public verifier, Wang *et al.,* [7] suggested a *Knox* mechanism which is implemented by using cluster signatures. The length of the authentication information in *Knox*, and the time it requires to verify the authentication information, are independent from

978-1-5386-7796-4/18/$31.00 ©2018 IEEE

the cluster size. Limitation of *Knox* is that the customers need to distribute a secret value with the public verifier and the customer repudiation is expensive.

Zhu *et al.,* [8] suggested a secure way for distributing the key where the customers can safely obtain their secret keys from the cluster manager. The mechanism also satisfies fragile admission control, and repudiated customers cannot approach the cloud once they are repudiated. The mechanism is safe from collusion attack. By utilizing the polynomial function, the scheme accomplishes a secure customer repudiation. It is efficient as customers need not modify their private keys.

Wang *et al.,* [1] proposed *Panda*, that is designed utilizing intermediary re-signatures, permits the CSP to transform the signatures figured out by repudiated customer into signatures of residing customer in the cluster. The CSP knows in advance the re-signing keys of any two customers in the cluster. This procedure leads to the following two severe safety issues. Initially, a mischevious distributed server may immediately transform signatures between two customers utilizing the re-signing keys. Further, conspiracy amidst the cloud and the repudiated customers might disclose the secret keys of all the current customers in the cluster. Limitation of the mechanism is that the estimation cost of the auditing grows with the size of the cluster.

Considering these two security problems of [1], we propose a novel Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud (SARRC) Using Regression Method mechanism. By using regression tools, we permit the information proprietor to compute the re-signing key and transmit it to the distributed server. As the re-sign key is computed by the information proprietor, it is not possible for the malicious cloud to trace out the secret parameters of the existing customers.

Motivation: In the existing system [1], Re-key is estimated by the Cloud Service Provider (CSP). The malicious CSP may immediately translate signatures of one customer into another customer by utilizing the re-signing keys and hence there is a necessity to secure the Re-key. In the proposed system, after revoking the malicious user, information proprietor computes the Re-key using regression method such that the key computed is highly secure. Then the information proprietor sends the Re-key to the CSP and allows him to audit the revoked customer blocks and re-signs the blocks using the Re-key.

Contribution: We introduce Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud using Regression Method (SARRC) scheme that supports computation of the Re-key using regression method by the information proprietor that is highly secure. Specifically our contributions are outlined as follows:

(i) The computation cost of Re-key using regression method by information proprietor has been significantly reduced.

(ii) We present the state-of-the-art Secure Auditing and Re-signing of the Revoked Customer Chunks by Cloud using Regression Method (SARRC) scheme.

(iii) The algorithm is collusion resistant, supports efficient customer revocation and the CSP efficiently verifies and re-signs the revoked customer chunks.

(iv) Extensive experimental evaluation manifests the efficiency and effectiveness of Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud (SARRC) using Regression Method scheme.

Organisation: This paper is ordered as follows: In Section 2, we review the relevant works that gives the pros and cons on the existing distributed information sincerity verification and efficient customer repudiation mechanisms. In Section 3, earlier models and its drawbacks are discussed and several preliminaries are introduced in Section 4. We specify the Problem statement, System model and design in Section 5. Construction of Homomorphic Authenticable Proxy Re-signature mechanism (HAPS) using Regression Method is explained in Section 6. Scheme details of Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud utilizing Regression Method (SARRC) and Security analysis are discussed in Section 7. Performance analysis is described in Section 8. Conclusions are presented in Section 9.

II. Related Works

Ateniese *et al.,* [9] presented Verifiable Data Possession scheme utilizing symmetric keys that supports dynamic data. The mechanism does not support public auditing. Jiang *et al.,* [4] ensures public trustworthiness with cluster customer repudiation. A conspiracy assault problem is deliberated where a repudiated customer can conspire with a mischievous distributed server to alter the customers information residing in the cluster. Raghavendra *et al.,* [10] suggested a safe multi-owner information distribution for vital cluster in the cloud. The proposed scheme adequately handles repudiation list, key administration, with decreased repository and reckoning cost. The scheme does not support multi-media files.

Yuan and Yu [5] achieves secure batch client elimination along with their vital public forthrightness analyzing mechanism that rely on polynomial confirmation and utilizes intermediary label update procedures which endorse public reviewing and dynamic customer repudiation. This scheme does not support cipher text store. Li *et al.,* [11] presents two confidentiality-conserving public verifying conventions for secure stockpiling in cloud. The scheme supports batch auditing and information dynamics. The disadvantage is that time cost increases continuously as the number of chunks increases at the user side. Raghavendra *et al.,* [12] proposed a compelling token creation method, that improves safe and effective token creation period. The advantage of the mechanism is that it reduces the cost of the information proprietor.

Venugopal *et al.,* [13] use soft computing techniques for various applications. Geeta *et al.,* [14] have performed extensive review on the latest methods in information auditing and security in cloud computing. Boneh and Shacham constructed [15] short group signature mechanism that bolsters Verifier-Local Repudiation (VLR). In this framework, the information of repudiation are only transferred to signature examiners. Hence, it is not required to contact original signers when any customer is repudiated. This framework is desirable for

systems providing verification adequecy. Proposed signatures are as small as standard RSA signatures with equivalent reliability.

Cao *et al.,* [16] designed a safe distributed storage maintenance that addresses the issue of fidelity. The analysis shows that the convention has low capacity cost and speedy information recovery. The scheme does not detect decodability efficiently.

Li *et al.,* [17] proposed a key-revising and authorization-developing scheme with void-intelligence protection of the hoarded documents for safe cloud information examining, which includes void learning evidence frameworks, intermediary re-signature and homomorphic direct endorser. The scheme has low communication and calculation cost while maintaining attractive security. The time cost of key-updating is linear with the updating times.

III. Background Work

Wang *et al.,* [1], suggested public verification scheme for the sincerity of combined information with accomplished customer renunciation. By adopting the notion of intermediary re-signatures, the CSP is allowed to re-sign the revoked customer chunks in favor of current customers at the time of customer renunciation, to avoid current customers to retrieve and re-sign revoked customer chunks by themselves. In addition, the public verifier inspects the sincerity of combined information without downloading the complete information from the cloud server, though few chunks of distributed information are re-signed by the CSP. The mechanism also bolsters batch verification. The drawback of the mechanism is that it is not collusion resistant i.e., the repudiated customer conspires with the cloud. Mischievious CSP might transform signatures between two customers utilizing the re-signing keys.

IV. Preliminaries

This section considers the foundations of Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud Using Regression Method (SARRC) scheme and are defined below: Bilinear Map: Consider two multiplicative cyclic groups G_1 and G_2 of prime order p, g be a generator of G_1. e : G_1 * G_1 $\rightarrow$ G_2 is a bilinear map [18] with the subsequent properties :

- Bilinear: for all $p, q \in G_1$ and $m, n \in Z_p$, e(p^m, q^n)=e(p, q)^{mn}
- Non-degeneracy: e(g, g) $\neq$ 1;
- Computability: An effective algorithm exists for computing map e.

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

A. Homomorphic Authenticators

Homomorphic authenticators [19], also termed as homomorphic provable labels, allows a public examiner to examine the honesty of information distributed in the cloud without retrieving the complete information. A homomorphic authenticable signature mechanism, supports the properties of Blockless verifiability and Non-elasticity. Let the customer's public/secret key pair be (p_i,s_i), ρ_1 is the signature on chunk $v_1 \in Z_p$, and ρ_2 is the signature on chunk $v_2 \in Z_p$.

• *Blockless verifiability*: Given ρ_1 and ρ_2, two random values δ_1, δ_2 in Z_p and a chunk v'= $\delta_1 v_1$+$\delta_2 v_2$ $\in Z_p$, a verifier examine the exactness of chunk v' without the knowledge of v_1 and v_2.

• *Non-malleability*: Given v_1 and v_2, ρ_1 and ρ_2, two random values δ_1, δ_2 in Z_p and a chunk v'= $\delta_1 v_1$+$\delta_2 v_2 \in$ Z_p, a customer without secret key (s_k), is not able to generate an appropriate signature ρ' on chunk v' by joining ρ_1 and ρ_2.

Blockless analysis permits an examiner to examine the reliability of the information hosted on the remote server by producing the linear collection of entire chunks *via* a challenge-and-response convention. Hence, the examiner need not download the complete information from the cloud. Non-elasticity demonstrates that different entities who do not hold suitable private keys are not able to make valid signatures on aggregate of chunks by utilizing the signatures that they hold.

B. Proxy Re-Signatures

Blaze *et al.,* [20] suggested intermediary re-signatures mechanism, that allows an intermediary to accomplish as a delegator of signatures among two customers. The cloud is allowed to perform as intermediary and interpret signatures for customers at the time of customer renunciation. Conventional intermediary re-signature mechanisms [21], [20] does not support blockless verifiability. If we utilize these intermediary re-signature mechanisms in the public verification schemes, then the auditor has to retrieve the complete information to check the honesty that certainly decreases the effectiveness of verification. Hence, we utilize Homomorphic Authorizable Proxy Re-signature (HAPS) [1] mechanism, that satisfies blockless auditability and non-flexibility. The information proprietor computes the Re-key using regression method and issues to the CSP. The CSP checks the sincerity of the repudiated customer chunks and signs these chunks using the Re-key sent by the information proprietor.

C. Regression Co-efficient

Regression co-efficient is computation of 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 reasonably a good relation between p_k and s_k. Similarly, in our scenario, the regression co-efficient secures the p_k and s_k of re-sign key.

V. Problem Definition And System Model

A. Problem Definition

Given a Cloud Storage Auditing model, the objectives are:

1) Secure Re-key generation: After revoking the malicious customer from the cluster, the information proprietor estimates the Re-key using Regression method and transmits the Re-key to the cloud server.

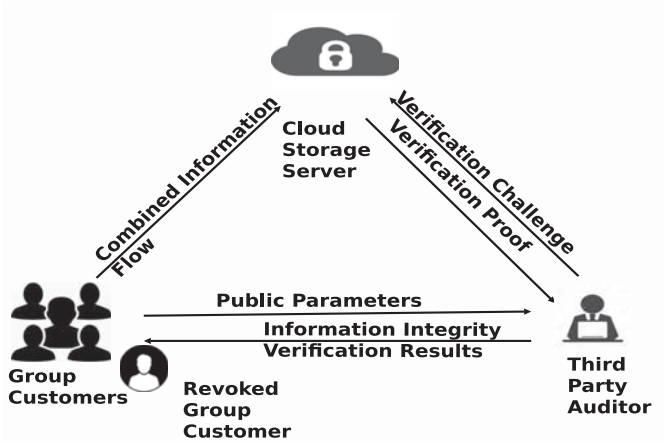

Fig. 1. Cloud Storage Model

2) Effective and safe customer repudiation: Once the information proprietor revokes the customer outside the cluster, the cloud server verifies the revoked customer chunks and securely re-signs with the Re-key.

3) Scalability: Cloud information is effectively distributed among existing customers of the cluster.

B. System Model

As depicted in Fig. 1., the system framework has three entities: the Cloud Service Provider (CSP), a cluster of customers and a public verifier. The cluster of customers consists of the information proprietor and numerous customers in the cluster. The information proprietor originally generates combined information in the cloud, and distributes it with customers in the cluster. The Third-Party Auditor (TPA) carries out the information honesty verification of the combined information saved in the distributed server. The information proprietor revokes the malicious customer from the cluster and computes the re-signing key (Re-key) and transmits to the Cloud Service Provider (CSP). The CSP audits the revoked customer chunks and re-signs with the Re-key sent by the information proprietor.

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

Wang *et al.,* [1] proposed a Homomorphic Authenticable Proxy Re-signature mechanism that satisfies blockless verifiability and non-flexibility. But their mechanism is not collusion resistant. We propose Homomorphic Authenticable Proxy Re-signature (HAPS) mechanism using Regression method that consists of five functions: *KeyGen*, *Re-key*, *Sign*, *Re-sign* and *Verify*.

VII. The Algorithm

The proposed mechanism (Algorithm 1), Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud Using Regression Method (SARRC) consists of four functions: *KeyGeneration*, *Re-key*, *SignatureGen*, *ReSignature*. In *KeyGeneration*, the Information proprietor generates the secret key and public key for all the customers of the cluster. In *Re-key*, the information proprietor computes the *Re-key* utilizing the regression method and sends it to the CSP. Either the information proprietor or the existing customer estimates a signature on each chunk as in *Signature*. In *ReSignature*, when a customer is repudiated outside the cluster, the CSP verifies and re-signs the chunks, that were signed earlier by the repudiated customer. 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. Since the Re-key is estimated by the Information proprietor, it is not possible for the CSP to find the private keys of the existing customers. Hence the proposed mechanism is protected and collusion resistant.

A. System setup

Two groups G_1, G_2 are of order p, g be a generator of G_1, e: G_1 * $G_1 \rightarrow G_2$ 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)$^* \rightarrow G_1$. The overall number of chunks in distributed information is n and distributed information is specified as V=(v_1, v_2,.....v_n). The overall number of customers in the group is k.

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

B. Security Analysis

Theorm 1: For the cloud, by colluding with the revoked customer, it is not possible to find the secret keys of the existing customers from the re-sign key (Re-Key).

Information proprietor takes the secret key s_k of customer u_i, public key p_k of customer u_j and computes the Re-key by using the regression method. Regression co-efficient is an estimation of independent variable in terms of the other. If s_k and p_k are co-related, the best fitting straight line in the least square sense gives reasonably a good relation between s_k and p_k. The regression coefficient secures the secret key of the Re-key. When a customer is revoked from the cluster, the information proprietor sends this Re-key to the semi-trusted CSP to verify the integrity of the revoked customer chunks and re-signs these chunks using Re-key. As the re-sign key (Re-key) is highly secure and is computed by information proprietor, the semi-trusted CSP by colluding with the revoked customer cannot detect the secret key of the existing customer. Hence the proposed mechanism is safe and collusion resistant.

VIII. Performance Analysis

Communication Cost: The proposed mechanism is a safe and effective customer revocation mechanism and hence the existing customers in the cluster are relieved from the burden of verifying the revoked customer chunks and hence the communication cost of all the existing customers in the cluster is totally reduced.

Algorithm 1: SARRC: Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud Using Regression Method

Input: γ_i, $v_t \in Z_p$, id_t where t $\in$ [1,n], k, k_1
Output: pk_i, ρ_t, $\gamma_{(i\to j)}$, ${\rho_t}^{\gamma_{(i\to j)}}$

KeyGeneration
Information proprietor k_1 creates a random $\gamma \in {Z_p}^*$
for each i upto k
Assign Private key sk_i= γ
Compute Public key pk_i=g$^{\gamma}$
k_1 creates the KL, that includes *id's* of all customers in the cluster.
The KL is public and signed by k_1.
Re-Key
After repudiating the malicious customer from the cluster, Information proprietor estimates the Re-key ($\gamma_{(i\to j)}$) utilizing Regression method and sends to the CSP.
$\gamma_{(i\to j)}$=2[($\delta(X^2)$+$\delta(Y^2)$+$\delta(Z^2)$+4)/($2\sqrt{sigX^2 sigY^2}+4$)]
SignatureGen
Customers k_i in the cluster generates the signature ρ_t on block v_t as:
ρ_t= ($H(id_t)$, w^{v_t})$^{\gamma_i}$
ReSignature
CSP verifies the integrity and re-signs the revoked customer blocks as:
The CSP first verifies that $e(\rho_t, g) \stackrel{?}{=} ((H(id_t)\ w^{v_t}), pk_i)$.
If the auditing outcome is 0, the cloud outputs $\perp$
or else CSP re-signs the repudiated customer blocks with the Re-key $\gamma_{(i\to j)}$ sent by the Information proprietor as: ${\rho_t}^{\gamma_{(i\to j)}}$ = $(H(id_t)w^{v_t})^{\gamma_j}$
After re-signing, the Information proprietor removes customer k_i's *id* from KL and signs the new KL.

TABLE I. NOTATIONS

Notation	Description
G_1 , G_2	Groups of order p
g, w	Generator polynomial of G_1
H	Hash function with H:(0,1)$^* \to G_1$
P_k	Public key
S_k	Secret key
ρ_t	Signature on t^{th} block
n	Total number of chunks in shared data
V	Shared information
k	Total number of customers in cluster
k_1	Information proprietor
KL	customer list
v_t	t^{th} block
id_t	t^{th} block identifier
$\gamma_{(i\to j)}$	Re-key used by the CSP

Computation Cost:
The estimation cost of respective signature of a chunk is about $2Exp\ \mathbb{G}_1$ + $Mul\ \mathbb{G}_1$ + $Hash\ \mathbb{G}_1$. As illustrated in the Re-Sign algorithm of the proposed scheme, the CSP initially checks the accuracy of the initial signature on a chunk and then computes the fresh signature on the corresponding chunk with the Re-key. The estimation cost of the CSP to re-sign a chunk is $2Exp\ \mathbb{G}_1$ + $Mul\ \mathbb{G}_1$ + $Hash\ \mathbb{G}_1$+ $2Pair$.

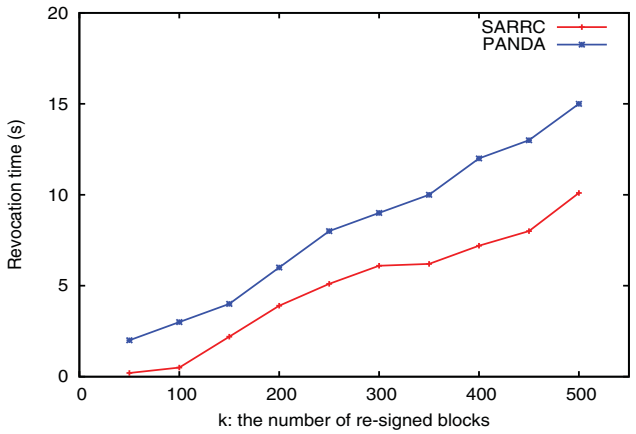

Fig. 2. Re-signing time of the blocks by Cloud Service Provider.

In this section, the performance of our scheme is estimated in experiments. We implement a prototype system of our scheme utilizing Java with Java Pairing-Based Cryptography Library (jPBC) [22] and the experiments are carried out on a PC with windows 7, Intel(R) Core(TM) i5-5200U, CPU @2.20GHz, 8GB RAM. We consider the size of an element in $\mathbb{G}_1$ or Z_p is $\mid p \mid$=160 bits. The size of an element of Z_q is $\mid q \mid$=80 bits. The size of every chunk is 4KB.

As shown in Fig., 2, the time taken by the Cloud Service Provider to re-sign the revoked customer chunks in SARRC scheme is reduced compared to the Panda scheme. In Panda scheme, the time taken by CSP is more, as CSP computes the re-sign key and re-signs the revoked customer chunks. But in our scheme, CSP′s computation cost is totally reduced as the CSP receives the re-sign key by the information proprietor and only re-signs the revoked customer chunks. Hence our mechanism is protected and effective.

IX. CONCLUSIONS

In this paper, we have proposed a procedure that provides Secure Auditing and Re-signing of Revoked Customer Chunks by Cloud using Regression Method (SARRC). The algorithm supports efficient customer revocation, CSP efficiently audits and re-signs the revoked customer chunks. The computation cost of Re-key, using regression method by information proprietor has been significantly reduced. Experimental results shows that the CSP securely and efficiently re-signs the revoked customer chunks and also saves existing customers computation and communication resources.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 92–106, 2015.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[4] T. Jiang, X. Chen, and J. Ma, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363–2373, 2016.

[5] J. Yuan and S. Yu, "Efficient Public Integrity Checking for Cloud Data Sharing with Multi-User Modification," in *INFOCOM Proceedings*, pp. 2121–2129, IEEE, 2014.

[6] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[7] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," *International Conference on Applied Cryptography and Network Security*, pp. 507–525, 2012.

[8] Z. Zhu and R. Jiang, "A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 40–50, 2016.

[9] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–9, ACM, 2008.

[10] S. Raghavendra, P. A. Doddabasappa, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Secure Multi-Keyword Search and Multi-User Access Control over an Encrypted Cloud Data," *International Journal of Information Processing*, vol. 10, no. 2, pp. 51–61, 2016.

[11] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-Preserving Public Auditing Protocol for Low-Performance End Devices in Cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2572–2583, 2016.

[12] S. Raghavendra, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "MSIGT: Most Significant Index Generation Technique for Cloud Environment," in *Proceedings of the Annual IEEE India Conference (INDICON)*, pp. 1–6, 2015.

[13] K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik, "Soft Computing for Data Mining Applications," Springer, 2009.

[14] C. M. Geeta, S. Raghavendra, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Data Auditing and Security in Cloud Computing: Issues, Challenges and Future Directions," *International Journal of Computer uyya(IJC)*, vol. 28, no. 1, pp. 8–57, 2018.

[15] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation," pp. 168–177, 2004.

[16] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "Lt Codes-Based Secure and Reliable Cloud Storage Service," in *INFOCOM Proceedings IEEE*, pp. 693–701, 2012.

[17] Y. Li, Y. Yu, B. Yang, G. Min, and H. Wu, "Privacy Preserving Cloud Data Auditing with Efficient Key Update," *Future Generation Computer Systems*, 2016.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[19] H. Shacham and B. Waters, "Compact Proofs of Retrievability," *Journal of Cryptology*, vol. 26, no. 3, pp. 442–483, 2013.

[20] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144, 1998.

[21] G. Ateniese and S. Hohenberger, "Proxy Re-signatures: New Definitions, Algorithms, and Applications," *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 310–319, 2005.

[22] "Pairing Based Cryptography (PBC) Library." [Online]. Available: http://crypto.stanford.edu/pbc/, 2014..