STLDAS: Secure Two Level Deduplication and Auditing of Shared Data in Cloud

Geeta C M[1], Mithila Lakshmi G[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, India. geetacmara@gmail.com, mithu1509@gmail.com, shreyasrajurg@gmail.com, raghush86@gmail.com

[2]Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Australia. raj@csse.unimelb.edu.au

[3]Bangalore University, Bengaluru, India. venugopalkr@gmail.com.

[4]Department of Computer Science and Engineering, Florida International University, USA. iyengar@csc.lsu.edu

[5]INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India. patnaiklm@yahoo.com

***Abstract*—With the cloud repository service furnished by the cloud computing, users can comfortably arrange themselves as a cluster and distribute information effectively. In order to empower public verifier to audit the distributed information, clients in the cluster need to figure out signatures on complete chunks of collaborative information. Every client in the cluster modifies and signs his respective chunks, and deploys in the cloud server. Hence specific chunks of shared information are normally signed by specific clients. If anyone of the customers' is found malicious, he is immediately repudiated from the cluster. The prevailing clients in the cluster are permitted to re-sign the chunks that were earlier signed by this eliminated client. This approach is inefficient due to the massive amount of collaborative information in the cloud. By exploiting the approach of proxy re-signatures, the CSP is acknowledged to re-sign chunks in support of the prevailing clients during customer repudiation. When many clients deploy the same information to the cloud repository, repository space has identical copies, hence deduplication technology is usually utilized to lower the capacity and bandwidth prerequisites of the utilities by removing repetitious information and hoarding only an original replica of them. In order to assimilate both data honesty and deduplication in cloud, we present a novel Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) mechanism. Experimental results show that our mechanism achieves secure deduplication and appreciable improvement in tag generation.**

Keywords*—*Cloud Computing, User Revocation, Deduplication, Public Auditing, Proof of Retrievability, Proof of Ownership.

I. Introduction

Distributed repository is a representative of interconnections of company repository where data are cached in pragmatic pools of depository that are universally managed by arbitrators. Distributed repository grants users with assistance, ranging from minimization in the cost and reduced assistance, to flexibility comforts and expandable facilities. These prominent qualities fascinate the customers to make use of and store individual information in the distributed repository. Although, the distributed depository architecture has been considerably established, it fails to furnish a few imperative emerging requisites such as the ability of examining the sincerity of distributed records and recognizing duplicated documents by cloud servers.

With information repository and collaborative assistance (such as Drop-box and Google Drive) administered by the cloud, users can conveniently perform cooperatively as a cluster by collaborating information with one another. Almost all of the earlier mechanisms [1], [2], target verifying the sincerity of individual information. However, none of these schemes acknowledges the adeptness of customer repudiation while verifying the accuracy of collaborative information in the cloud. With collaborative information, once a customer revises a chunk, he also requires to estimate the latest signature for the revised chunk. As the changes are made from various customers, specific chunks are signed by specific customers.

For reasons of reliability, when a customer's membership of the cluster expires or behaves mischievously, this customer must be renunciated from the cluster. Hence, this renunciated customer shall not be able to fetch and change the distributed data, and the signatures produced by this renunciated client are no longer genuine to the group. By exploiting the concept of agent re-signatures [3], the CSP is empowered to re-sign blocks in support of the current customers while customer renunciation, so that the prevailing customers need not retrieve and re-sign chunks by themselves. Thus, the genuineness of the complete data can yet be verified with the public keys of the prevailing customers only.

The quick acceptance of cloud assistance is associated with the growing size of information cached at distant distributed servers. Between these remote stockpiled documents, almost all are the same: as stated by EMC [4], 75 percent of current

978-1-7281-0297-9/19/$31.00 ©2019 IEEE

digital information is alike documents. This proof boosts a technology namely deduplication, where the distributed servers need to deduplicate by maintaining individual unique replica for each record (or chunk) and produce an interface to the document (or chunk) for every user who holds or solicits to save the identical document (or chunk). We propose Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) mechanism in which the Cloud Service Provider (CSP) performs deduplicate check on the information uploaded by information proprietor as well as checks for deduplication of the existing customers' chunks. Further, the Third Party Auditor (TPA) efficiently performs shared data integrity verification.

A. Motivation

Distributed depository service is one of the significant facilities provided by the distributed computing, where the customers can easily arrange themselves as the cluster and share the data among themselves. If anyone of the customers' in the cluster is found malicious, he is immediately repudiated from the cluster by the data owner. Now a days, as many customers are sharing the data, cloud storage utility is associated by expanding capacity of information cached at distant servers. Hence, one critical challenge of todays distributed depository utility is to manage the ever-developing capacity of data. Instead of maintaining many information duplicates with the similar content, deduplication deletes repetitious records by maintaining only one physical replica and indicating the other repetitious documents to that copy. This paper focuses on efficient deduplication on the information uploaded by information proprietor as well as checks for deduplication of the existing customers' chunks.

B. Contribution

In this paper, we suggest Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) mechanism that supports secure document level and chunk level deduplication. Our contributions are compiled as follows:

(i) We propose Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) scheme that supports secure document level and chunk level deduplication.

(ii) The algorithm supports secure deduplication and has reduced appreciably the time cost of tag generation.

(iii) Experimental analysis manifests the adeptness and efficacy of Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) mechanism.

C. Organisation

The list of the paper is arranged as follows: We explain the Related works in Section 2 that provides the pros and cons on existing integrity auditing and deduplication schemes. In Section 3, we discuss the earlier models and their drawbacks. In Section 4, we discuss several preliminaries. In Section 5 we explain, Problem statement and System model that illustrates the functioning of the architecture and provides the specifics about the design goals. In Section 6, we explain scheme details of our Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) protocol. In Section 7, we explain the Security analysis. In Section 8, we list out the results of experimental evaluation. Conclusions are given in Section 9.

II. Related Works

As our work is associated with both sincerity verification and assured deduplication, we study the works in both the areas in the following sections.

A. Integrity Auditing

Confirmable information ownership and Proofs of Retrievability (PoR) were originally suggested by Ateniese *et al.,* [5] and Juels *et al.,* [6]. In their techniques, the homomorphic authentication method was incorporated to minimize both the transmission and reckoning cost. Subsequently, numerous alternatives of PDP and PoR strategies are constructed to increase the adeptness and upgrade the performance of fundamental strategies, such as permitting public validation [7] and supporting information update [8].

Mastering C++ can be used to carry out simulations in C++ [9]. Both individuality protection and accountability for batch clients are preserved by an effective public validation convention that is recommended by Yang *et al.,* [7]. The method realizes data reliability while creating a certificate by employing blind signature. The technique has small overhead while realizing both individuality protection and identifiability. The limitation is that trivial certificate production has not been consigned.

B. Secure Deduplication

Deduplication is a method where the server stockpiles only a solitary equivalent of each record, irrespective of the number of how many users demanded to save that document. However, customer side deduplication is accompanied by the leakage of side channel information. Halevi *et al.,* [10] developed the proof of proprietorship mechanism that lets a consumer effectively prove to a server that the particular customer owns this document. Venugopal *et al.,* [11] utilize soft computing methods for data mining applications.

Zheng *et al.,* [12] constructed a safe deduplication model that bolsters protected deduplication with robust video conservation against malevolent users and dishonest cloud. It bolsters protected deduplication with hindrance to restricted data leakage. Reckoning cost is huge in case of regionalized servers. Raghavendra *et al.,* [13] proposed most powerful distinct-keyword ranked inquiry over encrypted cloud information that bolsters adept and authentic search. Limitation is that the scheme does not support multimedia. Geeta *et al.,* [14] have performed extensive review on the latest methods in information auditing and security in cloud computing.

III. Background Work

Li *et al.,* [15] designed two safe mechanisms namely, SecCloud and SecCloud+ that achieves data honesty and deduplication in cloud. SecCloud utilizes a validating entity with a MapReduce cloud that delivers users a few benefits by creating information labels and examines the genuineness of information hoarded in the cloud. SecCloud has small computation cost. SecCloud+ supports sincerity validation and safe deduplication on encrypted information. Wang *et al.,* [3] suggested public validating convention for the reliability of transferred data with adequate client repudiation. By adopting the idea of representative re-signatures, the CSP is granted to re-sign blocks for the prevailing customers during client renunciation. Further, the public examiner checks the forthrightness of collaborative information regularly without retrieving the complete information from the cloud. The mechanism enhances cluster auditing by inspecting various auditing tasks at the same time. Limitation is that the scheme is not collusion resistant.

IV. Preliminary

The preliminary concepts that will form the basis of our strategy are analysed below.

A. Bilinear Map:

Consider two cyclic multiplicative clusters $\mathcal{G}$ and $\mathcal{G}_T$ of large prime order p. $e : \mathcal{G} * \mathcal{G} \rightarrow \mathcal{G}_T$ [16] is a bilinear pairing map with the subsequent properties:

- Bilinear: $e(g_1{}^i, g_2{}^j) = e(g_1, g_2)^{ij}$ and i, j $\in_R$ Z_p ;
- Non-degenerate: There exists $g_1, g_2 \in \mathcal{G}$ such that $e(g_1, g_2) \neq 1$;
- Computable: An effective algorithm prevails that estimates $e(g_1, g_2)$ for all $(g_1, g_2) \in_R \mathcal{G}$.

Computational Diffie-Hellman (CDH) Problem: The Computational Diffie-Hellman (CDH) problem is that, given g, g^m, $g^n \in \mathcal{G}$ for unknown m, $n \in Z_p$, to estimate g^{mn}.

B. Convergent Encryption

Convergent encryption [17] grants confidentiality of information in deduplication. A customer (or original customer) obtains a concurrent key produced by taking a part of the document and encodes the information duplicate with the concurrent key. In addition, the customer determines a label for the information duplicate, such that the label is utilized to find the same copies.

V. Problem Definition and System Model

A. Problem Definition

Given that the data owner encrypts and outsources the document to the distributed server, cluster of customers distributes this document the main objectives are:
(i) To reduce the time cost of document label construction of the document.
(ii) To perform secure document level and chunk level deduplication

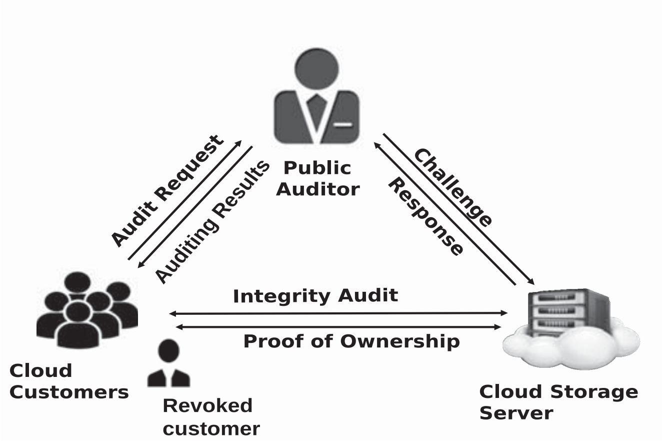

Fig. 1. Cloud Storage Model

B. System Model

Aiming at allowing verifiable and deduplication of shared data repository, we propose Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) scheme. The cloud repository framework (as shown in Fig. 1.) consists of three objects: Original customer with group of customers, Cloud Server, and the Auditor.

Original customer encodes the document with the convergent key and uploads to the CSP. The CSP performs deduplication, if the file exists in its storage, the CSP intimates the original customer that the file already exists and runs the PoW protocol. The original customer is allowed to retrieve the file. If the file is not a duplicate then the CSP saves the file. Further, the group of customers headed by the original customer shares the data uploaded by the original customer. Shared data are divided into chunks and the existing customers perform changes, sign with the secret key τ_k and upload to the CSP. During this process, the original customer keeps on watching every activity of the existing customers. If he finds any one of the existing customers performing malicious activity or expiry of membership in the cluster, he immediately revokes him from the cluster withdrawing all his credentials and informs the CSP.

In the proposed scheme, CSP performs deduplication and integrity verification for the revoked customer chunks [3]. After revoking the customer, the original customer informs CSP to verify the revoked customer chunks. The CSP performs deduplication and integrity verification for the revoked customer chunks and re-signs with $rk_{e\rightarrow f}$. In ReSign, we presume that the CSP transforms consistently the signatures of a renunciated client into signatures of the information proprietor. After re-signing, the information proprietor removes the customer's id from Customer List (UL) and signs the new UL. The confirmation on truthfulness of shared information is carried out via a challenge-and-response convention amidst the CSP and a public examiner.

VI. The Algorithm

In this section, we define two conventions including file uploading convention and Proof of Ownership (PoW) convention. At first, we present the framework setup phase of our scheme.

A. System setup

Consider two clusters $\mathcal{G}_1$, $\mathcal{G}_2$ of order p, g be a generator of $\mathcal{G}_1$, $e : \mathcal{G}_1 * \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a bilinear map, w be another generator of $\mathcal{G}_1$. The global specifications are (e, p, $\mathcal{G}_1$, $\mathcal{G}_2$, g, w, H) where H is a hash function with H: $(0,1)^* \rightarrow \mathcal{G}_1$. The overall number of chunks in collaborative information is n and collaborative information is represented as $\mathcal{S}$ = ($\mathfrak{b}_1$, $\mathfrak{b}_2$,.....$\mathfrak{b}_n$). Let u be the number of customers in the cluster.

Function: Key generation

```
1) Generates the system public and secret parameters.
2) Input: $u$, $u_1$, global parameter (g, $Z_p{}^*$)
3) Output: $pk_i$, $sk_i$
4) for each $i$ upto $u$
5) Generate random number $\delta_i$ from $Z_p{}^*$
6) Assign Private key $sk_i$= $\delta_i$
7) Compute Public key $pk_i$=g$^{\delta_i}$
8) $u_1$ creates the $UL$ that contains id's of all customers in
   the cluster.
9) The $UL$ is public and signed by $u_1$.
10) End
```

B. File Uploading Protocol

Customer u_1 is considered as the information proprietor of the cluster. The information proprietor produces private key sk_i and public key pk_i for all the existing customers in the cluster as shown in the $Function\ KeyGeneration$. In addition, the customers' list (UL) that has the $id's$ of all the existing customers of the cluster is generated and publishes it as public. The information proprietor executes the deduplication test by transmitting hash value of the document Hash F_1 to the distributed server (see Algorithm 1, Phase 1). If there is an identical document, the cloud user executes proof of proprietorship convention with the distributed server. If it is passed, the client is certified to retrieve this cached document without uploading the document. Otherwise, the CSP divides the file F_1 into chunks, creates a tag for each chunk generated dynamically using Pairing Based Cryptography, where the tags are represented in the form of $\mathfrak{b}(x,\ y)$ where $\mathfrak{b}$ is block and (x, y) is vector. Then the information of tags are sent to the Information proprietor u_1.

The existing customers retrieve their respective chunks, perform modifications, sign with their respective secret key and then upload to the distributed server. The CSP verifies for the deduplication of the chunk with the respective customers. If it is the modified chunk then CSP allows to upload otherwise CSP executes the proof of ownership convention; if it is a duplicate then CSP allows the respective customers to retrieve the chunk as illustrated in Algorithm 1, Phase 2.

Algorithm 1: STLDAS: Secure Two Level Deduplication and Auditing of Shared Data in Cloud mechanism

```
Input: $F1$ = ($\mathfrak{b}_1$, $\mathfrak{b}_2$,.....$\mathfrak{b}_n$), $u$, $u_1$, $\delta_e$, $\mathfrak{b}_k \in$ Z$_p$, $id_k$
       where $k \in$ [1, $n$], $pk_e$
Output: $\eta_k$
(1)  Phase 1: Document Level Deduplication
(2)  For every outsourcing document $F_1$ by $u_1$ the following
     tasks are implemented:
(3)  CSP examines for the deduplication of the document. If
     it is a current document then it moves to step 4. If the
     document exists then PoW convention is performed
     between CSP and $u_1$.
(4)  After the validation that there is no duplicate copy of the
     document that $u_1$ has tried to deploy, $u_1$ divides the
     document into chunks $F1$ = ($\mathfrak{b}_1$, $\mathfrak{b}_2$,.....$\mathfrak{b}_n$) and encodes
     the entire distributed information and outsources to the
     CSP.
(5)  CSP produces a label for every chunk that is created
     actively utilizing Pairing Based Cryptography, where the
     labels are represented in the form of $\mathfrak{b}(x,\ y)$ where $\mathfrak{b}$ is
     block and ($x$, $y$) is vector.
(6)  Once the label is constructed for respective chunks, the
     key of every chunk is transmitted to $u_1$.
(7)  Phase 2: Chunk Level Deduplication
(8)  The prevailing user downloads his respective chunks from
     the cloud server, carries out updations and then signs
     with his secret key $\delta$ and transmits to the CSP.
(9)  for each $\mathfrak{b}_k$ with $id_k$
(10) Estimate $\eta_k$ = ($H(id_k)$, $w^{\mathfrak{b}_k})^{\delta_e}$
(11)  end for
(12) This process consists of the following steps:
(13) CSP validates for the deduplication of the block. If it is
     an update block then it moves to step 14. If the block is
     present then PoW convention is executed between CSP
     and the prevailing user.
(14) If the block does not exist in the cloud then the
     prevailing user uploads the modified block to cloud.
```

A summary of the Notations used in the Algorithm is as shown in Table I.

C. Proof of Ownership Protocol

The PoW convention aims at achieving secure deduplication at the distributed server. The cloud server picks a set of chunk identifiers randomly for challenge. Upon acquiring the challenge set, the original customer searches in the customers' list for the corresponding tags of blocks. If the respective tags are retrieved then the original customer sends the tags as response to cloud to prove his ownership.

VII. Security Analysis

In this section the security analysis of the Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) scheme is performed. Let us consider a game in which an Adversary and a Challenger are the two players. The

TABLE I. SUMMARY OF THE NOTATIONS USED IN THE ALGORITHM

Notation	Description
$\mathcal{G}_1$, $\mathcal{G}_2$	Groups of order p
g_1, w	Generator polynomial of $\mathcal{G}_1$
H	Hash function with H:(0,1)* $\rightarrow \mathcal{G}_1$
tag_F	Tag of file F
P_k	Public key
S_k	Secret key
η_k	Signature on block k
n	Total number of chunks in shared data
$\mathcal{S}$	Shared information
u	Total number of customers in cluster
u_1	Information proprietor
UL	Customer list
$\mathfrak{b}_\mathrm{k}$	k^{th} block
id_k	k^{th} block identifier

adversary is aiming to gain the goal condition as said in the game.

1) Secure File-level Deduplication: Let us assume that a mischievous consumer attempts to demand it has a challenge document F through colluding with the consumers in the cluster who do not own this document. A challenge file F is randomly selected and sent to the challenger. The challenger executes the summary principle and generates an abstract of the document F. The Adversary colludes with the other clients and provokes them to communicate with the distributed server to try to prove the proprietorship of document F. Here the distributed server acts as a sincere validator and executes the proof of proprietorship convention. The adversary outputs a challenge for this file F to the distributed server. If the distributed server accepts the file F, then we say the adversary succeeds. But the distributed server, by running the proof of ownership protocol will verify securely that the challenger for this file F is an unauthorised person and hence our proposed mechanism satisfies secure file level deduplication.

2) Secure Block Level Deduplication: Let us assume that an adversary tries to upload his chunks of the record F to the server by colluding with the existing clients in the cluster. He sends these chunks as challenge to the CSP. After receiving these chunks, CSP runs the proof of ownership protocol and identifies that the challenger is an attacker and informs the information proprietor. Thus, the CSP performs block level deduplication securely and will protect the shared data from the adversaries efficiently.

VIII. PERFORMANCE ANALYSIS

In this section, we present an experimental analysis of our scheme. We exploit Pairing Based Cryptography (PBC) Library [18] to perform cryptographic operations in our convention. We have used Intel(R) Core(TM) i5-5200U, CPU @2.20GHz, 8GB RAM. In order to accomplish λ = 80 bit security, the prime order p of the bilinear cluster $\mathcal{G}$ and $\mathcal{G}_\mathrm{T}$ are respectively chosen as 160 and 512 bits in length. We also set the chunk size as 4 KB.

Fig. 2. shows the time cost for creating the file tags. When compared to the Mapreduce algorithm [15] (SecCloud) the time cost of tag generation by using AES and MD5 hash function is reduced. In this implementation some part of data in the file is selected and the key is computed using AES and we input the output of the AES to MD5, the output of MD5 is the final tag generated for each file whereas mapreduce is a lengthy process which has a complicated multiplication over slave node. So the time taken to generate a tag in STLDAS is reduced compared to the time taken by Mapreduce (SecCloud). So we have reduced this complexity by replacing mapreduce to AES and MD5 for generating tags.

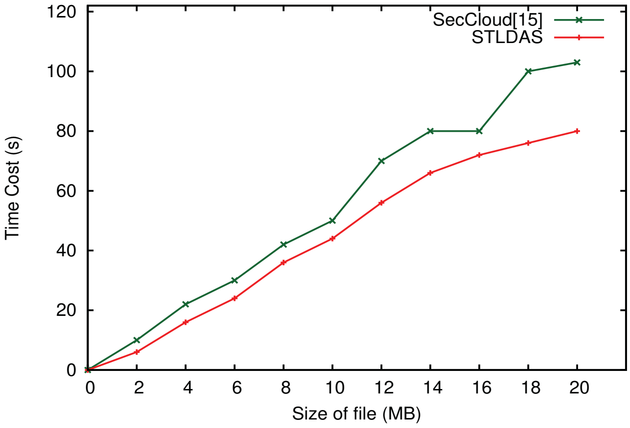

Fig. 2. Tag generation

IX. CONCLUSIONS

With the objective of accomplishing both information sincerity and deduplication in cloud, we introduce Secure Two Level Deduplication and Auditing of Shared Data in Cloud (STLDAS) mechanism. In the proposed scheme, CSP performs secure deduplication and generates data tags for the revoked user blocks and audits the integrity of the revoked user blocks efficiently that has been stored in cloud. The time cost for tag generation by CSP has been improved appreciably. Third Party Auditor examines collaborative information cached in the cloud efficiently and supports cluster verification. The experimental results show that our mechanism is effective and protected.

REFERENCES

[1] H. Shacham and B. Waters, “Compact proofs of retrievability,” *Journal of Cryptology*, vol. 26, no. 3, pp. 442–483, 2013.

[2] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” *Computer Security–ESORICS*, pp. 355–370, 2009.

[3] 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.

[4] J. Yuan and S. Yu, “Secure and constant cost public cloud storage auditing with deduplication,” in *IEEE Conference on Communications and Network Security (CNS)*. IEEE, pp. 145–153, 2013.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 598–609, 2007.

[6] A. Juels and B. S. Kaliski Jr, "PORs: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 584–597, 2007.

[7] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *Journal of Systems and Software*, vol. 113, pp. 130–139, 2016.

[8] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, pp. 213–222, 2015.

[9] K. R. Venugopal and R. Buyya, "Mastering C++," McGraw-Hill Education, 2013.

[10] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*. ACM, pp. 491–500, 2011.

[11] M. Sookhak, A. Akhunzada, A. Gani, M. Khurram Khan, and N. B. Anuar, "Towards dynamic remote data auditing in computational clouds," *The Scientific World Journal*, vol. 2014, pp. 1–12, 2014.

[12] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Towards encrypted cloud media centre with secure deduplication," *IEEE Transactions on Multimedia*, pp. 1–16, 2016.

[13] S. Raghavendra, C. M. Geeta, K. Shaila, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "MSSS: Most significant single-keyword search over encrypted cloud data," in *Proceedings of the 6th Annual International Conference on ICT: BigData, Cloud and Security*, 2015.

[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 (IJC)*, vol. 28, no. 1, pp. 8–57, 2018.

[15] J. Li, J. Li, D. Xie, and Z. Cai, "Secure auditing and deduplicating data in cloud," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2386–2396, 2016.

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

[17] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, pp. 296–312, 2013.

[18] "Pairing based cryptography (PBC) library." [Online]. Available: http://crypto.stanford.edu/pbc/, 2014..