Cloud Storage–Optimization of Initial Phase for Privacy-Preserving Public Auditing

Deepak Kumar Verma, Purnima Gupta and Rajesh Kumar Tyagi

Abstract The integrity of data confirms that an unapproved user cannot temper the outsourced data in cloud storage. The cloud storage needs to be secure from unapproved tempering. For integrity verification process, data owner generates the signature which is sent to the third-party auditor. It is a one-time process, but it increases the overhead of the data owner. We have analyzed the existing data integrity auditing schemes along with their distinctions. The proposed system supports privacy-preserving integrity auditing by using homomorphic linear authentication and employing Boneh–Lynn–Shacham-based signature technique. We extend our research to enable the data owner to speed up the initial phase through detailed experiments and comparisons between single-thread and multi-thread models using different core of CPUs. The proposed scheme demonstrated by using multithreading architecture on multi-core CPU for getting better performance.

Keywords Cloud service provider · Third-party auditor (TPA) · Data integrity · Provable data possession (PDP) · Proof of retrievability (POR)

1 Introduction

Cloud computing provides on-demand access to a shared network resource like storage servers, applications, platforms and many other shared or individual services [10]. It can be speedily administered with minimum effort or the service provider interaction. Lin et al. proposed a method for selecting cloud services with the best

D. K. Verma (✉) · P. Gupta
IEC College of Engineering and Technology, Greater Noida, Uttar Pradesh, India
e-mail: deepak.verma1980@gmail.com

P. Gupta
e-mail: purnimaa018@gmail.com

R. K. Tyagi
Krishna Institute of Engineering and Technology, Ghaziabad, Uttar Pradesh, India
e-mail: profrajeshkumartyagi@gmail.com

Y.-C. Hu et al. (eds.), *Ambient Communications and Computer Systems*,
Advances in Intelligent Systems and Computing 904,
https://doi.org/10.1007/978-981-13-5934-7_32

security and privacy features [8]. In cloud computing, the workload of users can be managed efficiently and economically using virtualization [7]. Data outsourcing reduces the burden of local data management and maintenance. Data owner losses the physical control over this data after uploading it on the cloud server. Hacigumus, Iyer and Mehrotra were first to introduce the concept of data outsourcing [5].

Due to eminent mobility, coherent storage and retrieval of data, users are being attracted towards accessing cloud services [6]. Apart from the benefits of cloud computing, there are three main pillars of security are CIA (Confidentiality, Integrity, and Availability [22]. There are many reasons for cloud service providers to be dishonest. Dishonest CSP may produce the wrong status of outsourced data. Outsourced data, which has been accessed rarely, can be discarded by CSP because of budgetary reason or CSP may hide the data loss information for his reputation [1, 23].

We are focusing on integrity auditing of the outsourced data in cloud storage. There are two categories of auditing schemes, private auditing and public auditing. In private auditing scheme, all computation that is needed for checking integrity is directly performed between data owner and cloud service provider. Second is public auditing, in which the integrity verification process is done by TPA (Third-Party Auditor) [21, 22]. This scheme reduces the computation overhead of users because all computations are done through TPA, and integrity verification results produced by TPAs are commonly accepted by both data owner and CSP [23].

The main contribution of our scheme is as follows:

1. We have analyzed the existing data integrity auditing methods along with their distinctions and proposed an integrity auditing system that supports privacy-preserving integrity auditing using homomorphic linear authentication and employing Boneh–Lynn–Shacham (BLS) based signature technique.
2. We have extended our research to speedup the initial phase of integrity auditing by implementing multithreading architecture on multi-core CPU for better performance. Finally, we justified our proposed approach with the existing methods.

2 Integrity Auditing System

As shown in Fig. 1, the auditing model consists of three main entities like user/data owner (DO), third-party auditor (TPA) and cloud service provider (CSP). DO sets up their data and uploads it to the cloud server. The CSP stores the data in the cloud and allows accessing the data from anywhere and at any time. When the DO sends a request to TPA for checking the integrity of data, the TPA sends a challenge to CSP, and as the answer of that challenge, the CSP sends the proof to the TPA. In this way the TPA ensures the integrity of outsourced data.

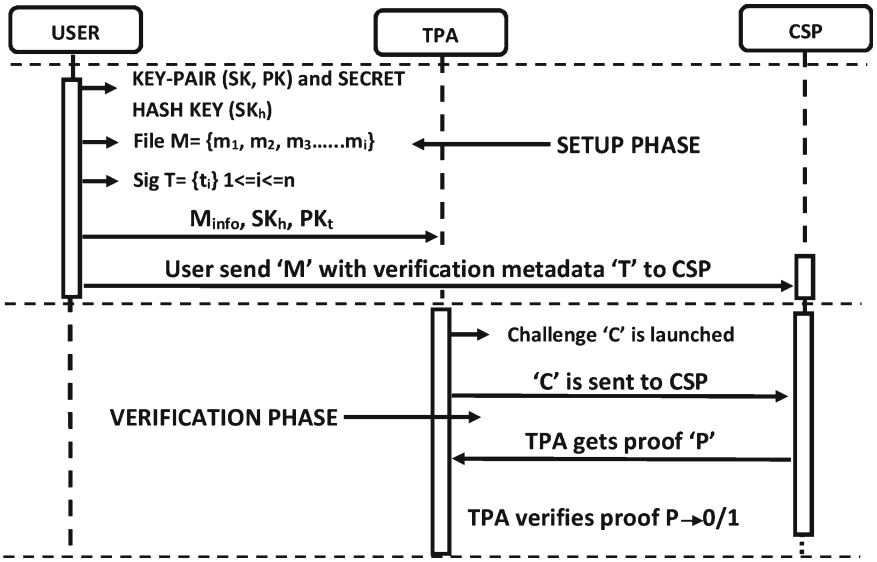

Fig. 1 Auditing model

3 Related Work

Fu et al. proposed a new privacy-aware public auditing system for pooled cloud data by constructing a homomorphic verifiable group signature [4]. Their system ensured that group users can trace data changes through the designated binary tree and can recover the latest correct data block when the current data block is damaged. Public auditing can be achieved through two basic concepts like MAC-Based and HLA-based. Many schemes have been proposed for dynamic data auditing by [2, 11, 15, 16, 18, 21, 25, 26]. These schemes have achieved dynamic auditing through implementing techniques like the indexed hash table [23], Merkle hash tree [19] and dynamic hash table [21]. Zhang and Tu [24] offered a new approach, based on batch-leaves-authenticated Merkle Hash Tree (MHT), to batch-verify multiple leaf nodes and their own indexes altogether, which is more suitable for the dynamic outsourced auditing system than traditional MHT-based dynamism approach that can only verify many leaf nodes one by one. Li et al. [9] concentrated on the multifarious key management in cloud data integrity checking by introducing fuzzy identity-based auditing. They verified the security of the proposed protocol based on the computational Diffie-Hellman postulation and the discrete logarithm hypothesis in the selective-ID security model. Table 1 gives the comparison of the existing integrity auditing schemes.

Table 1 Comparison of existing data integrity auditing schemes

Data integrity auditing scheme	Technique used	Proposed By	Year	Strength	Weakness
PDP First privacy preserving	Integrating Homomorphic authenticator with random masking	Wang et al.	2010	Supports public auditing privacy preserving	Does not support data dynamics
Fully dynamic PDP	Combined BLS based HLA with MHT	Wang et al.	2011	Supports dynamic auditing	No Privacy preserving
CPDP (corporative provable possession)	Hash index hierarchy	Zhu et al.	2012	Supports public auditing Privacy-preserving Batch auditing in multi-cloud	It does not support the dynamic audit. Does not support auditing for multiuser
IHT-PA (Index hash table-public audit)	Index hash table	Zhu et al.	2013	Supports public auditing Privacy-preserving Supports dynamic auditing	Batch auditing is not mentioned
Privacy-preserving public auditing and multi-owner authentication	Homomorphic linear authenticator and random masking	Nandini et al.	2014	Eliminated the trouble of cloud user from the dull and possibly high-priced auditing task	Does not support data dynamics operations
DPDP (Dynamic PDP)	Using ranked based authenticated skip list	Erway et al.	2015	Dynamic data auditing No demand for privacy-preserving	No public auditing Does not support Batch auditing

(continued)

Table 1 (continued)

Data integrity auditing scheme	Technique used	Proposed By	Year	Strength	Weakness
DHT-PA (Dynamic hash table-public audit)	Dynamic hash table	Tian et al.	2015	Supports public auditing Supports dynamic auditing Supports batch auditing in multi-cloud	Communication cost is greater than DAP and IHT-PA
Integrity checking for outsourced data	Suggested various methods	Kaustubh and Jog	2016	Only be efficient when the verifier of the service maintains a copy of same outsourced data	The original file is the necessity for integrity examine
Secure preserving public auditing	Homomorphic linear authenticator	Poornima and Ponmagal	2016	Supports batch auditing Privacy preserving	Does not support data dynamics
NPP: a new privacy-aware public auditing Scheme	Homomorphic verifiable group signature	Fu et al.	2017	Eliminates misuse of single-authority control Ensures non-frameability	Group users can trace the data changes through the binary tree
Dynamic data operations with deduplication	Markle hash tree (MHT)	Wu et al.	2017	Support dynamic data operations Privacy preserving	Does not support batch auditing

4 Problem Statement

As discussed in literature review, authors have compared existing data integrity verification schemes for cloud storage and noticed that many of them require dividing the data into blocks. All blocks are then required to generate authenticators so that it can be sent to TPA for verifying the integrity of user's data. These authenticators/signatures are highly required to accumulate confidentiality and privacy preser-

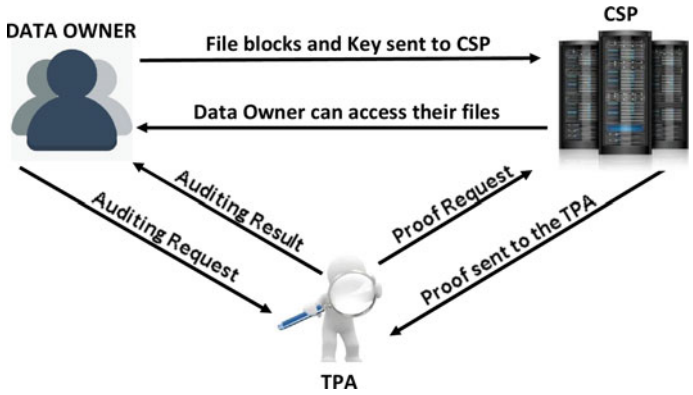

Fig. 2 System model

vation of cloud user's data from TPA. Authenticator/signature generation for the large set of blocks is quite tedious and time-consuming task for the DO. Nandini et al. [12] and [13, 17, 20] have indicated that the data setup happens only once, traditionally it was generated sequentially and expanded the computational overhead at the user side. Initial phase can speed up by implementing it to multi-threading model on latest multi-core CPU system [3, 14, 20].

DO, CSP and TPA are three main entities in the proposed system model shown in Fig. 2. It means that there should not be any discrepancy while retrieving the data by the users. The TPA verifies the data integrity of the outsourced data for the data owner. To verify the outsourced data, the data owner provides the metadata instead of complete data because it is in encrypted form for data privacy. When data owner sends the request to TPA for checking the integrity of data, then TPA challenges the cloud service provider, and then CSP sends the algebraic signature to the TPA for proofing.

5 Proposed System Model

5.1 The Proposed Solution by the Authors for Integrity Auditing Initial Phase

In this paper, HLA (Homomorphic Linear Authenticator) technique has been used to achieve public auditing without downloading heavy count (complete data) of data blocks. To achieve a privacy-preserving public auditing, File "M" is divided into fixed size blocks and then HLA $\phi = \{\sigma_i\} 1 <= i <= n$ is aggregated for dataset $M = \{m_i\}$ where "n" is a total number of blocks. This process of generating authenticators is very tedious and time-consuming operation for the data owner. In the

proposed system, data is first divided into the fixed size of blocks $M = \{m_i\}$. These blocks are further encrypted with AES-128 encryption technique. After encryption of each block, dataset $M = \{m_i\}$ is ready to compute authenticators for each "i". A short signature scheme has been used based on the computational Diffie-Hellman assumptions, i.e., certain elliptic and hyperelliptic curves. Traditional single-threaded models for the generation of authenticators on multi-core CPU systems may be approximate "$p-1$" times slower than our proposed model, which is a multi-threaded model with multi-core CPU systems where "p" is CPU cores available on the system.

In the proposed system, authors used a single dimensional array of encrypted data blocks $\{m_i\}$ to generate signature $\{\sigma_i\}$. The single-threaded model may take several minutes to complete its task if the array of data blocks is very large. And that's why authors have implemented a multi-threaded model, which breaks the array up to the size of the threshold. The authors have taken threshold size 50.

5.2 Experimental Setup

We applied Java pairing-based cryptography (JPBC) API with library version 2.0.0, the Fork-Join-Pool framework of JDK 1.8 and "J-Free-Chart" API for generating charts. All these supporting libraries are integrated with Net-Beans 8.0 IDE to make application development easy and bug-free. The Boneh-Lynn-Shacham (BLS) with the short signature scheme is used to generate signatures for blocks. This scheme used a verification pairing function and signatures under some elliptic curves. Authenticator generation on each block is independent, and it is the one-time operation.

5.3 Preliminaries

Bilinear Map: Let G_1, G_2, and G_T be cyclic groups of prime order r. Let g_1 be a generator of G_1 and g_2 is a generator of G_2. A bilinear pairing or bilinear map e is an efficiently computable function e: $G_1 \times G_2 \rightarrow G_T$ such that:

Bilinearity: For all a, b in Z_r (the ring of integers modulo r) it holds that $e(g_1\hat{}a, g_2\hat{}b) = e(g_1, g_2)\hat{}\{ab\}$.

Non-degeneracy: $e(g_1, g_2)! = 1$. Our creation of multi-threaded data setup model includes split, encrypt, setup, keyGen, sigGen, upload, delete, decrypt and join operations as follows:

(i) split(M) = $\{m_1, m_2, m_3 \ldots m_i\}$
(ii) Encrypt (key, m_i, $m_{i\,\mathrm{enc}}$)
(iii) Setup
(iv) Private Key (x) and public key (g^x) generates from selecting a random integer $x \leftarrow Z_r$ and a random "g" $\leftarrow G_2$.
(v) Signature sig = h^x.

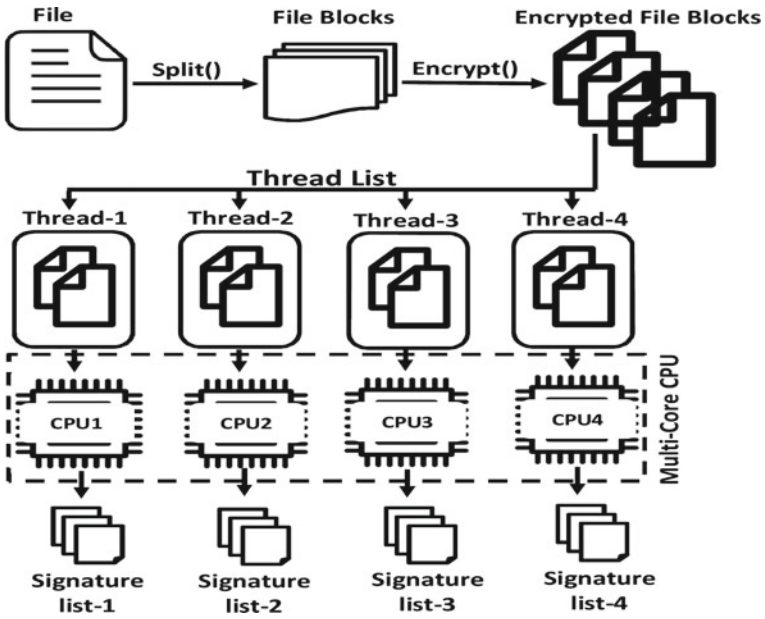

Fig. 3 Multithreading model

Figure 3 represents the architecture of our proposed multithreading model.

6 Result and Discussion

The results of the experiments are shown below in Tables 2, 3 and 4 respectively.

Each file M is splitting into "n" data blocks as $M = (m_1, m_2, m_3 \ldots m_i)$, where $1 <= i <= n$. Due to computation overhead, 50 KB data block size has been taken as in [18]. However, if we get the less size of data blocks, then more blocks will be generated and for each block signature will be produced by the system.

On the basis of the block size (50 KB), 8 MB data component has been divided into 164 blocks which will generate 164 signatures. Data component splitting time and block encryption time have been reduced up to 50% by implementing the proposed system with the multi-threaded model on multi-core CPU system.

The comparison has been made on different file which has different size with different count of blocks like 8 MB (164 blocks), 16 MB (328 blocks), 32 MB (656 blocks), and 64 MB (1312 blocks).

Figure 4 shows the performance comparison graph of Table 3. As our paper focuses on the initial phase setup at DO side, it has been observed that up to 50% computation time reduced by using multithreading model as compared to the single-threaded model.

Table 2 List of symbols

Symbol	Meaning	Symbol	Meaning
x	Private key	i	Block/signature sequence number
g^x	Public key	n	Number of blocks
t	Threshold size	p	Number of available CPU cores
M	Data set	σ_i	Signature of a block
Φ	Set of signatures	h	Mapping element of the signature
e	Bilinear map	f	The first index of the block array
m_{i_enc}	Encrypted data block	l	Last index of the block array
m_i	Data block		

Table 3 Computation time comparison on different CPUs (50 KB block size)

		Split time (s)		Encryption time (s)	
File size (MB)	Number of blocks	2 CPU core	4 CPU core	2 CPU core	4 CPU core
8	164	0.56	0.234	3.883	1.645
16	328	0.873	0.506	3.52	2.568
32	656	1.434	0.789	4.49	3.358
64	1312	2.484	1.89	8.215	4.984

Table 4 Computation time on the single-threaded model

File size (MB)	Computation time on 2 CPU cores at 25 KB block size (s)	Computation time on 4 CPU cores at 25 KB block size (s)	Computation time on 2 CPU cores at 50 KB block size (s)	Computation time on 4 CPU cores at 50 KB block size (s)
8	101	23.07	28.46	25.4
16	160	41.2	54.5	47.6
32	286	103.86	104.05	102.1
64	582	234.36	207.8	198.5

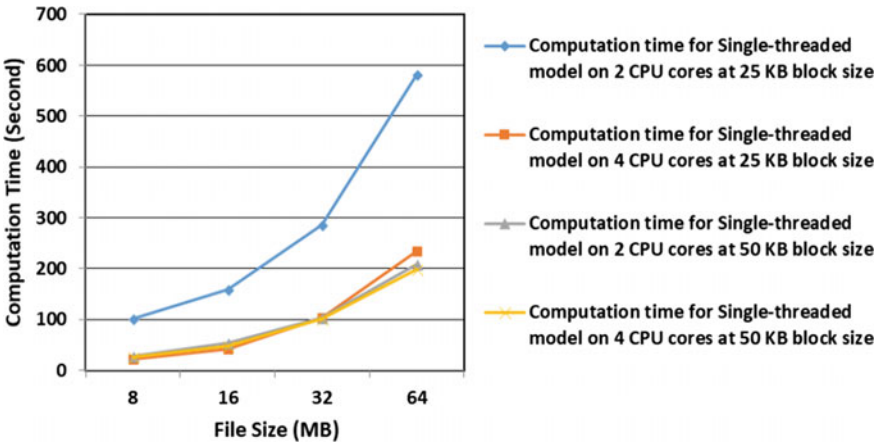

Fig. 4 Computation time on the single-threaded model

Table 5 Computation time on multi-threaded model

File size (MB)	Computation time on 2 CPU cores at 25 KB block size (s)	Computation time on 4 CPU cores at 25 KB block size (s)	Computation time on 2 CPU cores at 50 KB block size (s)	Computation time on 4 CPU cores at 50 KB block size (s)
8	62	10.72	17.71	12.5
16	119	19.52	34.07	23.2
32	246	43.49	69.29	50.5
64	477	81.59	136.43	96.3

Table 5 shows the performance on the multi-threaded model for both 25 KB and 50 KB data block sizes on two parallel CPU cores and four parallel CPU cores.

The performance differences between single-threaded and multi-threaded model have shown in Fig. 5.

Figure 6 shows the difference between throughputs on dual core and core i3 (4 CPU cores) systems with different block sizes.

We achieved the better performance on multi-core CPU system by generating the limited number of threads depends on available CPU cores. Figure 7 represents the throughput analysis (in KB/second) of our proposed system based on different CPU types (multi-core CPUs).

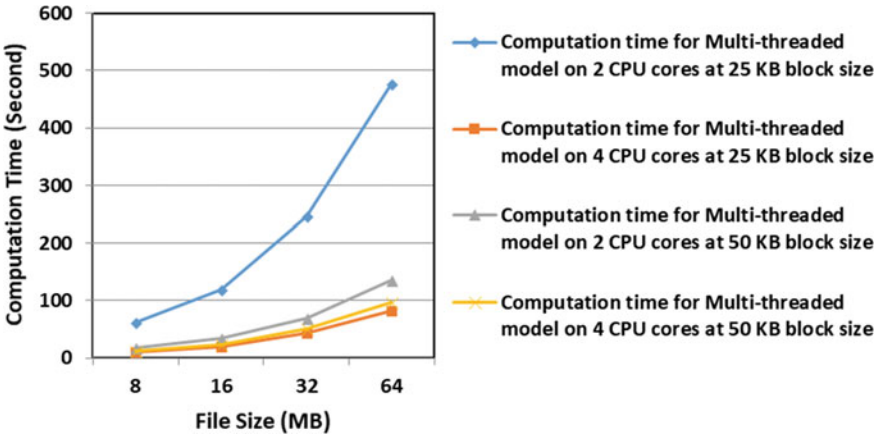

Fig. 5 Computation time on multithreading model

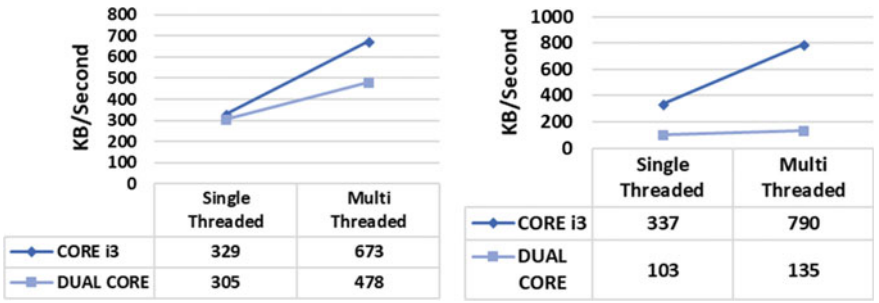

Fig. 6 Throughput analysis based on different block sizes

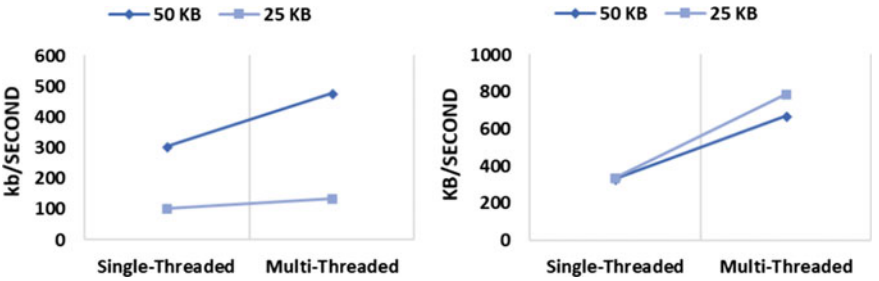

Fig. 7 Throughput analysis based on different CPU types

7 Conclusion

In this paper, a detailed comparative analysis of different types of auditing schemes has been presented. Authors proposed an enhanced scheme for the initial phase setup for data integrity auditing to reduce the computation cost at user side. We used multi-threading parallel execution for initial phase by utilizing all available CPU cores through concurrent execution of tasks. The results show the significant reduction in the computation cost at user side. In future, the proposed scheme can be extended by utilizing more processing units with the help of graphical processing units (GPUs).

References

1. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM conference on Computer and communications security, pp. 598–609 (2007)
2. Ateniese, G., Di Pietro, R., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, p. 9 (2008)
3. Chen, K.Y., Chang, J.M., Hou, T.W.: Multithreading in Java: performance and scalability on multicore systems. IEEE Trans. Comput. **60**(11), 1521–1534 (2011)
4. Fu, A., Yu, S., Zhang, Y., Wang, H., Huang, C.: NPP: a new privacy-aware public auditing scheme for cloud data sharing with group users. IEEE Trans. Big Data (2017)
5. Hacigumus, H., Iyer, B., Mehrotra, S.: Providing database as a service. In: Proceedings of 18th International Conference on Data Engineering, pp. 29–38 IEEE (2002)
6. Kaustubh, S., Jog, V.V.: Article: A Survey on Integrity Checking for Outsourced Data in Cloud using TPA. IJCA Proceedings on International Conference on Internet of Things, Next Generation Networks and Cloud Computing ICINC **2016**(1), 6–9 (2016)
7. Kumar, N.S., Lakshmi, G.R., Balamurugan, B.: Enhanced attribute based encryption for cloud computing. Procedia Comput. Sci. **46**, 689–696 (2015)
8. Lin, L., Liu, T., Hu, J., Ni, J.: PQsel: combining privacy with quality of service in cloud service selection. Int. J. Big Data Intell. **3**(3), 202–214 (2016)
9. Li, Y., Yu, Y., Min, G., Susilo, W., Ni, J., Choo, K.K.R.: Fuzzy identity-based data integrity auditing for reliable cloud storage systems. IEEE Trans. Dependable Secure Comput. (2017)
10. Mell, P., Grance, T.: The NIST definition of cloud computing. Nat. Inst. Stand. Technol. **53**(6), 50 (2009)
11. Mutyalanna, C., Srinivasulu, P., Kiran, M.: Dynamic audit service outsourcing for data integrity in clouds. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) **1**(2), 2482–2486 (2013)
12. Nandini, J., Sugapriya, N.P., Vinmathi, M.S.: Secure multi-owner data storage with enhanced TPA auditing scheme in cloud computing. Int. J. Adv. Comput. Sci. Cloud Comput. **2**, 2321–4058 (2014)
13. Poornima, S.N., Ponmagal, R.S.: Secure preserving public auditing for regenerating code based on cloud storage. Networking Commun. Eng. **8**(5), 200–204 (2016)
14. Rinku, D.R., Rani, M.A.: Analysis of multi-threading time metric on single and multi-core CPUs with Matrix Multiplication. In: 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), pp. 152–155. IEEE (2017)
15. Rizvi, S., Razaque, A., Cover, K.: Cloud data integrity using a designated public verifier. In: High Performance Computing and Communications (HPCC). 2015 IEEE 7th International

Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on, pp. 1361–1366. IEEE (2015)
16. Ryoo, J., Rizvi, S., Aiken, W., Kissell, J.: Cloud security auditing: challenges and emerging approaches. IEEE Secur. Priv. **12**(6), 68–74 (2014)
17. Shirahatti, A.P., Khanagoudar, P.S.: Preserving integrity of data and public auditing for data storage security in cloud computing. Int. Mag. Adv. Comput. Sci. Telecommun. **3**(3), 161 (2012)
18. Tian, H., Chen, Y., Chang, C.C., Jiang, H., Huang, Y., Chen, Y., Liu, J.: Dynamic-hash-table based public auditing for secure cloud storage. IEEE Trans. Serv. Comput. (2015)
19. Wang, C., Chow, S.S., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for secure cloud storage. IEEE Trans. Comput. **62**(2), 362–375 (2013)
20. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: Infocom, 2010 Proceedings IEEE, pp. 1–9. IEEE (2010)
21. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst. **22**(5), 847–859 (2011)
22. Wu, Y., Jiang, Z.L., Wang, X., Yiu, S.M., Zhang, P.: Dynamic data operations with deduplication in privacy-preserving public auditing for secure cloud storage. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 562–567 (2017)
23. Yang, K., Jia, X.: An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. Parallel Distrib. Syst. **24**(9), 1717–1726 (2013)
24. Zhang, H., Tu, T.: Dynamic Outsourced Auditing Services for Cloud Storage Based on Batch-Leaves-Authenticated Merkle Hash Tree. IEEE Trans. Serv. Comput. (2017)
25. Zhu, Y., Ahn, G.J., Hu, H., Yau, S.S., An, H.G., Hu, C.J.: Dynamic audit services for outsourced storages in clouds. IEEE Trans. Serv. Comput. **6**(2), 227–238 (2013)
26. Zhu, Y., Hu, H., Ahn, G.J., Yu, M.: Cooperative provable data possession for integrity verification in multicloud storage. IEEE Trans. Parallel Distrib. Syst. **23**(12), 2231–2244 (2012)