Partial Binary Encoding for Slepian-Wolf Based Proof of Retrievability

Tan Choon Beng
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia

Mohd Hanafi Ahmad Hijazi
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Kota Kinabalu, Sabah, Malaysia

Yuto Lim
WiSE Laboratory, School of Information Science
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

***Abstract*— Cloud storage is a storage service offered by Cloud Service Provider (CSP) to data client, where the data is outsourced to distributed servers in cloud, while data client has to pay upon usage. As integrity and availability are the pre-conditions for the existence of a cloud storage system, Proof of Retrievability (PoR) is introduced to cloud storage. Recently, Slepian-Wolf Based Proof of Retrievability (SW-PoR) was introduced that provides cost-efficient and time consistent exact repair mechanism for erroneous outsourced data. However, the encoding process of SW-POR requires a certain amount of computation time, which considerably long compared to conventional storage method involving replication. Hence, this paper proposed an extension work to SW-PoR, named Partial Binary Encoding for SW-PoR (PBE-SW-PoR) to address the limitation. Simulation was conducted to evaluate the performance of the proposed work by means of comparison to the original SW-PoR scheme in term of computation time. In our simulation, result obtained has shown a significant performance improvement in computation time for PBE-SW-PoR compared to original SW-PoR.**

Keywords—cloud computing; cloud storage; Proof of Retrievability; integrity; availability; error recovery

I. INTRODUCTION

The emergence of Internet has contributed to the pollination of several information technologies such as Internet of Things (IoT), big data and cloud services. Although Internet has brought us countless convenience and comfort in our daily lives, however when things become open, security is always a big concern [1]. Thus, security is always a deciding factor of cloud storage adoption [2] [3]. In other meaning, by ensuring security, cloud storage would be a better choice than traditional storage. As integrity and availability are the pre-conditions for the existence of cloud storage [4], cryptographic techniques such as encryption and replication are the conventional methods [5] used to fulfills these pre-conditions.

Nevertheless, the conventional method of data replication throughout distributed servers is no longer applicable due to the exponential growth rate of data. Regarding to this, lots of cloud storage schemes have been proposed by researchers to ensure cloud data availability and integrity. For example, Provable Data Possession (PDP) [6] and Proof of Retrievability (PoR) [7] are protocols established to ensure cloud data availability and integrity. PDP and PoR are similar protocols, but they differ in term of data corruption resiliency. With respect to this, PoR is better than PDP as PoR provides data recovery to repair corrupted cloud data.

Although researchers have been working on PDP and PoR schemes for about a decade, including [8], [9], [10] and [11], but most of these PoR schemes are erasure coding based, which consume a considerably high computation cost for recovery. This is because erasure coding requires reconstruction of the full data before recovery on corruption data blocks can take place. To address this issue, network coding based PoR schemes such as [12], [13] and [14] are designed to provide a more efficient recovery for corrupted data. However, network coding based PoR schemes in general do not provide exact repair on corrupted data block. Regarding to this, a PoR scheme known as SW-PoR [15] has been proposed recently, to provide exact repair for corrupted data with a consistent computation time across data sizes. However, computation time of SW-POR encoding is considerably high. Hence, in this paper, we proposed a scheme named as Partial Binary Encoding for SW-PoR (PBE-SW-PoR) to reduce the SW-PoR encoding time. PBE-SW-PoR works by adapting concepts including data transmission error checking technique as well as conventional replication technique.

Contributions. The key contributions of this paper are listed below:

- Decrease the computation time required by Encode function in SW-PoR via PBE-SW-PoR scheme.
- Simulation to show the performance of the proposed PBE-SW-PoR scheme towards Encode, Retrieve and Repair functions of SW-POR in term of computation time.
- Analysis and discussion on the proposed PBE-SW-PoR scheme with respect to the simulation results.

The remaining of this paper is organized as follows. Section 2 provides preliminary to our work by describing related works on PoR schemes in advance, before proceeds to the detail of SW-POR; Section 3 presents the proposed PBE-SW-POR; Section 4

978-1-5386-2126-4/17/$31.00 ©2017 IEEE

shows, analyzes and discusses the results obtained in our simulation; Section 5 concludes this paper.

II. Related Works

Cloud storage is always labeled untrusted or semi-trusted by data clients [15], as data clients have no physical access and control over the outsourced data. Therefore, in order to address this issue, Cloud Service Providers (CSP) have to adopt data integrity protocols such as PDP and PoR, where PDP and PoR are protocols used to prove to data clients where outsourced data is properly stored. However, PoR is a better choice for cloud storage due to its recovery mechanism towards data corruption.

The protocol PoR was first proposed by researchers in [7] in which sentinel is used for error checking whereas error-correcting code is used for error recovery for the outsourced data. However, only a limited number of challenge can be conducted in this PoR [7]. A PoR challenge is a request from data client to cloud servers asking for a proof where the outsourced data is properly stored, while cloud servers have to provide the requested proof for verification, as shown in Fig. 1 below. To address the limitation in [7], an erasure coding based PoR is proposed by [16], which then has become a benchmark PoR model for the construction of many PoR schemes in later time.

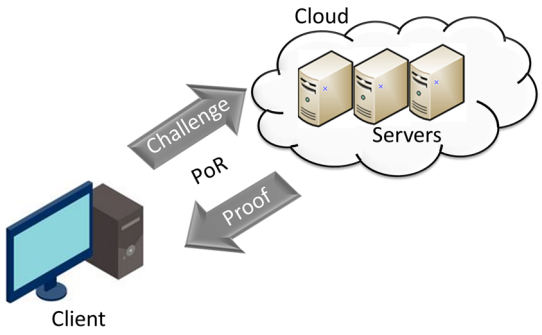

Fig. 1. PoR challenge and proof scenario

Erasure coding [17] is one of the data protection techniques widely used in PoR schemes against data corruption. Generally, erasure coding based PoR breaks data into a number of blocks and redundant codes are added during encoding before these blocks are stored across distributed servers in cloud. Recent work [18] proposed an erasure coding based PoR where data is initially stored separately from erasure coded updates in the form of log; while [19] is another similar work to [18], but differ with a hierarchical log structure.

On the other hand, network coding is an alternative technique adapted in PoR schemes due to its efficient data transmission compared to conventional routing [20]. Regarding to this, researchers have proven efficiency of network coding in cloud storage [12], while providing direct repair mechanism which do not require computing resources from client side. Similarly, researchers have also shown the advantages of the adoption of network coding with a spot-check based PoR, towards cost-effective data repair [13].

Although network coding based PoR schemes provide direct repair to corrupted data blocks stored in cloud servers, the repair operation does not produce the same coded block to replace the corrupted one. This is because network coding repair the corrupted coded blocks by combining linearly a number of healthy coded blocks to construct a new coded block as replacement. In other word, network coding based PoR schemes such as [12] and [13] do not provide exact repair. Regarding to this, researchers have addressed the limitation of network coding based PoR schemes by introducing a PoR scheme with exact repair mechanism, known as Slepian-Wolf based PoR scheme (SW-PoR) [15]. SW-PoR [15] consists of three main functions, namely Encode, Retrieve, and Repair functions, as described in paragraphs below.

Before storing a file into the cloud servers, SW-PoR [15] encodes the file by firstly dividing it into a number of equally sized file blocks. Then, three file blocks are selected to undergo encoding process via exclusive OR (XOR) operation, to form a constructed block which is then converted into a pair of coded block and metadata. The process of XORing file blocks continue until it gets sufficient pairs of coded block and metadata. As three file blocks are chosen out of all file blocks (m) divided from the file, F, the number of possible combination is C(m,3) for the constructed blocks (XORs) formation. Lastly, after all coded block and metadata pairs are constructed, they are stored across distributed servers in cloud.

To retrieve a file stored using SW-PoR, a number of pairs of coded block and metadata are chosen from the storage servers in such a way that the coefficient vectors of these selected pairs of coded block and metadata can construct a binary square matrix which has full rank. Then, a column consists of the constructed block translated from their coded block and metadata pairs is added to the matrix as the last column. Lastly, after undergoing Gaussian Elimination, the resolved matrix is extracted its last column, having each binary of the column joining up in ascending order to form back the raw file.

Whenever a corrupted pair of coded block and metadata is discovered, it is repaired by SW-PoR using coded block and metadata pairs stored in three healthy servers. Firstly, indices of the three operands which make up the XOR of the corrupted pair of coded block and metadata are identified. Then, two numbers are chosen from a list of number ranged from {0, m-1} which are not any of the three operands of the coded block and metadata pair. Then, these five numbers are used in the repair phase of the corruption, where servers storing XORs which made from these five numbers as operands are identified. Servers which are storing the selected three pairs of coded block and metadata are required to provide them to repair the corruption. Then, the obtained three XORs undergo XOR operation to produce a new constructed block (XOR), which is then converted to coded block and metadata pair to replace the corrupted one. The newly constructed pair of coded block and metadata is exactly the same as the corrupted one before corruption happens.

Despite of the efficient exact repair on corruption of a pair of coded block and metadata with consistent repair time across data sizes; considerably linear retrieval time across data sizes; however, encoding process in SW-PoR using C(m,3) setting for

the construction of XORs, coded block and metadata pairs causing the exponential increase in computation time across data sizes, has become its main drawback [15]. As construction of XORs and conversion of these XORs into coded blocks and metadata pairs are the main processes in encoding in SW-PoR, hence the computation time is consumed mostly here. With respect to the exponential increase in the number of XORs required due to the exponential factor C(*m*,3), this impacted directly the computation time of encoding in SW-PoR with an exponential increase.

III. Partial Binary Encoding for SW-PoR (PBE-SW-PoR)

In SW-PoR encoding using C(m,3) setting for construction of XORs, coded blocks and metadata pairs, encoding time increases exponentially as data size increases gradually. Similarly, as the data size decreases gradually, the encoding time decreases exponentially. Hence, the straight forward solution to solve the address problem is to decrease the size of data to be encoded in SW-PoR. Regarding to this, we proposed a solution named as Partial Binary Encoding for SW-PoR (PBE-SW-PoR), where data splitting, Cyclic Redundancy Check (CRC), replication, and lastly SW-PoR scheme are involved.

The main idea of PBE-SW-PoR is to shorten the data to be encoded in SW-PoR, via data splitting. In PBE-SW-PoR, data to be stored in cloud servers is firstly split into two parts; one part is going to be encoded by SW-PoR, whereas its counterpart is stored directly to cloud servers. As SW-PoR ensures encoded data to be fully protected and recoverable if corruption happens, hence the part to be encoded by SW-PoR has no issue on data corruption and recovery. However, the counterpart of the data which is not encoded by SW-PoR required another recovery mechanism in case of data corruption happens. Replication [21] is a good choice in this case, for backup of the counterpart of data which is not encoded by SW-PoR. This is because replication is one of the simplest and fastest method of backup data for disaster recovery purpose; and due to the data counterpart to be replicated is only part of the full data, it reduces the high demands in cloud storage size compared to replicating full data.

The other issue faced by the unencoded part of the data is error checking. To address this, error-detecting code such as Cyclic Redundancy Check (CRC) is used. During data transmission, CRC bits added to transmitting packets are act as error-detecting codes, in order to check whether the data when reaching destination is having any error [22]. Generally, CRC is implemented in such a way that a pre-agreed (client and servers) polynomial is used to construct a divisor; the divisor is then used in XORs division where the raw data as dividend; remainder obtained in the XOR division is then known as CRC bits and it is added to the ending of raw data before transmission take place. An example of CRC operation and error checking is shown in Fig. 2 below.

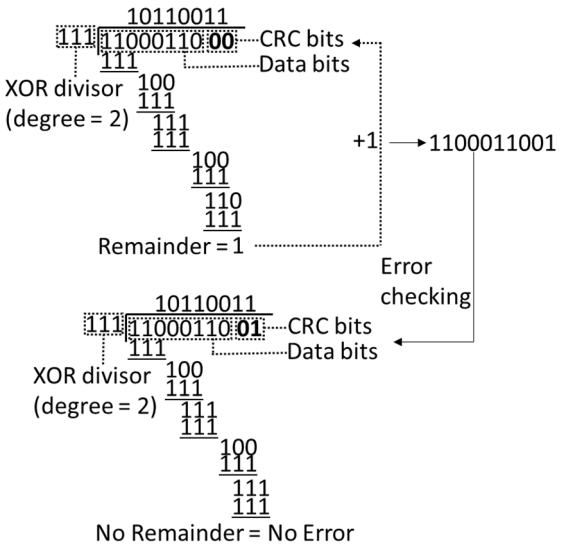

Fig. 2. Example of CRC operation and error checking

In summary, the proposed solution to the drawback of SW-PoR, PBE-SW-PoR is done by firstly split the data into two parts, one part is encoded by SW-PoR, whereas its counterpart is added with CRC bits for error checking, and then replication take place to the CRC bits added data counterpart, before distributed to store in cloud servers. For retrieval of the data, the part which has been encoded by SW-PoR is decoded and joined up with CRC-bits removed counterpart data to form back the original raw data. The concept of the proposed PBE-SW-PoR is illustrated in Fig. 3 below.

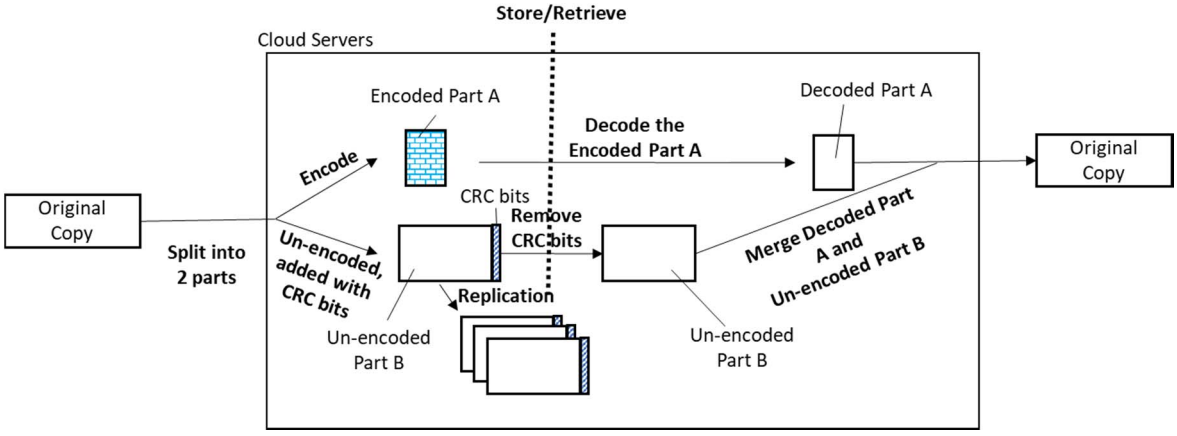

Fig. 3. Overview concept of PBE-SW-PoR

IV. Simulation, Analysis and Discussion

In order to evaluate the performance of the proposed PBE-SW-PoR scheme, simulation is conducted using a machine with specification as follows: Intel i5-3210M processor, 2.50 GHz, 8GB of RAM, Windows 10 (64-bit) OS. In our simulation, both original SW-PoR and PBE-SW-PoR are tested to compare their performances in term of computation time, for their encoding, retrieval, and repairing of outsourced data.

For both the original SW-PoR and PBE-SW-PoR, each file block is set to have a size of 210 or 1024 bits, while each server is storing only one pair of coded block and metadata, which is the same setting as in [15]. To ensure SW-PoR does not lost its role while having the aim of reducing size of data to be encoded in SW-PoR, the setting of 1:1 ratio for file splitting is used instead. For example, before encoding in PBE-SW-PoR, a file of 100 file blocks (102,400 bits) is split into two parts, where each part is made up of 50 file blocks (51,200 bits). Nonetheless, the overall computational performance theoretically would be better when the portion to be encoded by SW-PoR turns smaller as using CRC plus replication is way faster.

Compared to [15] where SW-PoR is only tested in simulation for file size up to 150 file blocks (153,600 bits), our simulation has conducted using much larger file sizes, ranged from 200 file blocks (204,800 bits) to 1,000 file blocks (1,024,000 bits). Nevertheless, file sizes beyond 1000 file blocks (1,024,000 bits) are not simulated in this paper as such simulation would take exponentially longer time for the original SW-PoR to complete the encoding. TABLE(s) I, II, and III; Fig.(s) 4, 5, and 6 below illustrate the results obtained in our simulation.

TABLE I. COMPARISON OF PERFORMANCE OF ENCODE FUNCTION BETWEEN ORIGINAL SW-POR AND PBE-SW-POR

No. of file blocks (*m*)	Data Size (bits)	Computation Time (seconds)	
		Original SW-PoR	*PBE-SW-PoR*
200	204,800	8,314.773	708.124
400	409,600	67,700.826	5,660.672
600	614,400	230,355.710	19,383.427
800	819,200	419,598.908	44,784.311
1000	1,024,000	835,179.698	85,863.045

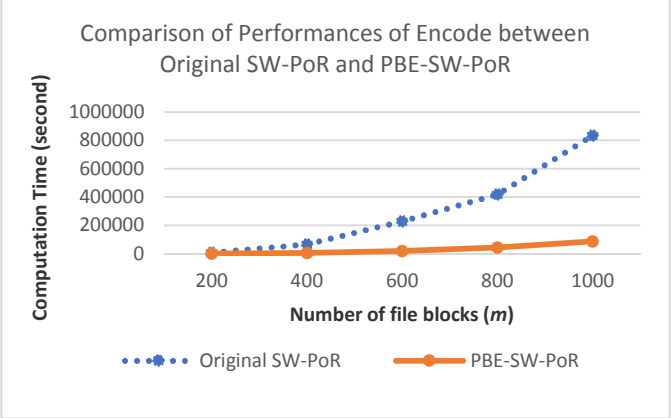

Fig. 4. Comparison of performances of Encode function between original SW-PoR and PBE-SW-PoR

TABLE II. COMPARISON OF PERFORMANCE OF RETRIEVE FUNCTION BETWEEN ORIGINAL SW-POR AND PBE-SW-POR

No. of file blocks (*m*)	Data Size (bits)	Computation Time (seconds)	
		Original SW-PoR	*PBE-SW-PoR*
200	204,800	3.414	1.192
400	409,600	8.736	2.815
600	614,400	12.516	3.924
800	819,200	16.829	5.636
1000	1,024,000	25.732	8.367

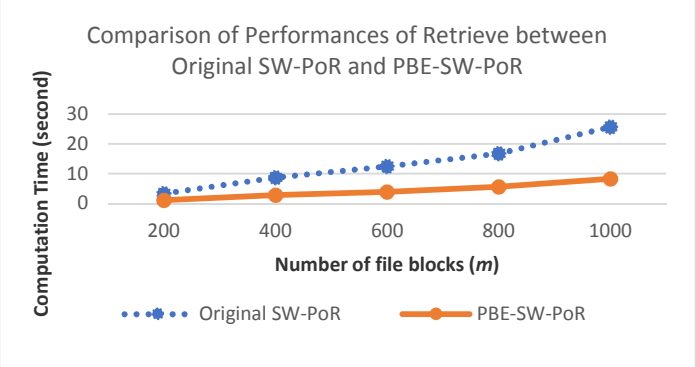

Fig. 5. Comparison of performances of Retrieve function between original SW-PoR and PBE-SW-PoR

TABLE III. COMPARISON OF PERFORMANCE OF REPAIR FUNCTION BETWEEN ORIGINAL SW-POR AND PBE-SW-POR

No. of file blocks (*m*)	Data Size (bits)	Computation Time (seconds)	
		Original SW-PoR	*PBE-SW-PoR*
200	204,800	0.034	0.032
400	409,600	0.032	0.031
600	614,400	0.031	0.031
800	819,200	0.036	0.031
1000	1,024,000	0.031	0.031

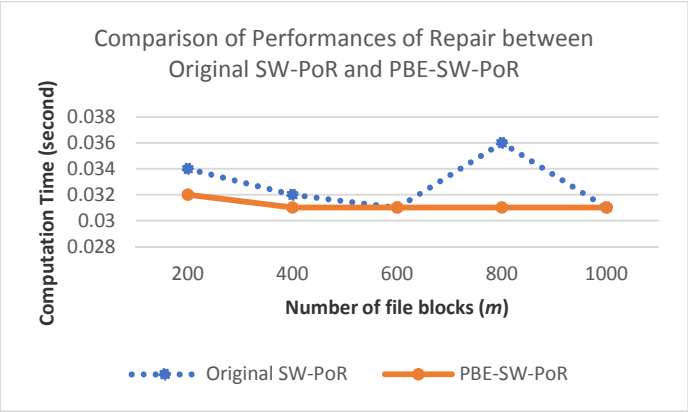

Fig. 6. Comparison of performances of Repair function between original SW-PoR and PBE-SW-PoR

The results of simulation conducted show that the encoding time of PBE-SW-PoR is always about one-tenth of that of original SW-PoR across data sizes. It is clearly shown that the reduction in the size of data (in term of number of file blocks) to be encoded in SW-PoR by using PBE-SW-PoR had greatly reduces the number of XORs, coded block and metadata pairs to be constructed due to the factor of exponential variable C(m,3), thus showing a direct impact towards the significant reduction in the encoding time.

Using PBE-SW-PoR, the retrieval of file consists of two phases; firstly, the decoding of required coded block and metadata pairs into its original form before encoding while removing CRC bits from the data counterpart; and secondly, joining of these two data parts into one, as the original data before the split took place. Although both PBE-SW-PoR and original SW-PoR are required to solve a single large sized square matrix for file retrieval [15], but the square matrix (m/2 x m/2) to be solved in PBE-SW-PoR is much smaller than (m x m) of the original SW-PoR, precisely one-quarter. Generally, solving a matrix with larger size will consume more time than that of smaller sized matrix [23]. Hence, the retrieval time of PBE-SW-PoR should equals to one-quarter of that of original SW-PoR. However, the removal of CRC of data counterpart also consumed some computational power and time in PBE-SW-PoR, hence the computation time for PBE-SW-PoR is about one-third instead of one-quarter of that of original SW-PoR, as shown in simulation.

For data repairing in our simulation, both PBE-SW-PoR and original SW-PoR have shown similar and consistent performances in term of computation time, about 0.03 to 0.04 seconds across data sizes. This is because the data repairing concept in SW-PoR is preserved in PBE-SW-PoR as well, where only three pairs of coded block and metadata are required in only one XOR operation for the reconstruct of the coded block and metadata pair to replace the corrupted pair.

V. CONCLUSION

In this paper, PBE-SW-PoR has been proposed by applying data splitting, CRC and replication, to decrease the encoding time of SW-PoR which is extensively high in particular for large file. Simulation is conducted by using larger file sizes compared to [15], to evaluate the performance of PBE-SW-PoR with setting 1:1 of split ratio. From the result obtained from simulation, PBE-SW-PoR has shown its efficiency by decreasing the encoding time to around one-tenth of that of original SW-PoR; whereas reduction of retrieval time to about one-third of that of original SW-PoR; while having similar and consistent file retrieval time compared to original SW-PoR across data sizes. Although PBE-SW-PoR needs only about one-tenth of time used in original SW-PoR for encoding, the encoding time for PBE-SW-PoR is still exponential due to the setting of C(m,3). Hence, this left us with a future work to further reduce the encoding time of SW-PoR to linear, instead of exponential.

REFERENCES

[1] T. Watson, "Security vs openness: The challenge for universities", *Telegraph.co.uk*, 2013. [Online]. Available: http://www.telegraph.co.uk/technology/internet-security/10021700/Security-vs-openness-The-challenge-for-universities.html. [Accessed: 20- Apr- 2017].

[2] Hanifah Abdul Hamid and Mokhtar Mohd Yusof, "State-of-the-Art of Cloud Computing Adoption in Malaysia: A Review", *Journal Technology (Sciences & Engineering)*, vol. 77, no. 18, pp. 131-136, Aug. 2015.

[3] B. Trudel and M. Lim, "2016 Cloud Readiness Index", 2017. [Online]. Available: http://www.asiacloudcomputing.org/research/cri2016. [Accessed: 20- Apr- 2017].

[4] *ISACA® Glossary of Terms*, 1st ed. Information Systems Audit and Control Association (ISACA), 2015, p. 49.

[5] "Amazon Simple Storage Service (S3) — Cloud Storage — AWS", *Amazon Web Services, Inc.*, 2017. [Online]. Available: https://aws.amazon.com/s3/faqs/. [Accessed: 06- Apr- 2017].

[6] G. Ateniese, R. Burns, and J. Herring, "Provable Data Possession at Untrusted Stores," *Proc. 14th ...*, no. 1, pp. 598–610, 2007.

[7] A. Juels and B. S. Kaliski Jr., "Pors: Proofs of retrievability for large files," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 584–597, 2007.

[8] A. Juels and B. S. Kaliski Jr., "Pors: Proofs of retrievability for large files," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 584–597, 2007.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, 2008.

[10] J. Xu, F. Zhou, Z. Jiang, and R. Xue, "Dynamic proofs of retrievability with square-root oblivious RAM," *J. Ambient Intell. Humaniz. Comput.*, vol. 7, no. 5, pp. 611–621, 2016.

[11] M. H. Au, Y. Mu, and H. Cui, "Proof of retrievability with public verifiability resilient against related-key attacks," *IET Inf. Secur.*, vol. 9, no. 1, pp. 43–49, 2015.

[12] K. Omote and T. P. Thao, "MD-POR: Multisource and Direct Repair for Network Coding-Based Proof of Retrievability.," *Int. J. Distrib. Sens. Networks*, vol. 2015, pp. 1–14, 2015.

[13] T. P. Thao and K. Omote, "ELAR: Extremely Lightweight Auditing and Repairing for Cloud Security," *ACM Int. Conf. Proceeding Ser.*, vol. 5, pp. 40–51, 2016.

[14] K. Omote and P. Tran, "D2-POR : Direct Repair and Dynamic Operations in Network Coding-Based Proof of Retrievability," *IEICE Trans. Inf. Syst.*, no. 4, pp. 816–829, 2016.

[15] T. P. Thao, L. C. Kho, and A. O. Lim, "SW-POR: A Novel POR Scheme Using Slepian-Wolf Coding for Cloud Storage," 2014 IEEE 11th Intl Conf Ubiquitous Intell. Comput. 2014 IEEE 11th Intl Conf Auton. Trust. Comput. 2014 IEEE 14th Intl Conf Scalable Comput. Commun. Its Assoc. Work., pp. 464–472, 2014.

[16] H. Shacham and B. Waters, “Compact proofs of retrievability,” *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, 2008.

[17] S. Lin, T. Al-Naffouri, Y. Han and W. Chung, "Novel Polynomial Basis With Fast Fourier Transform and Its Application to Reed–Solomon Erasure Codes", *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6284-6299, 2016.

[18] M. Etemad and A. Küpçü, “Generic Efficient Dynamic Proofs of Retrievability,” *Cryptol. ePrint Arch.*, pp. 85–96, 2015.

[19] E. Shi, E. Stefanov, and C. Papamanthou, “Practical Dynamic Proofs of Retrievability,” *CCS ’13 Proc. 2013 ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 325–336, 2013.

[20] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright and K. Ramchandran, "Network Coding for Distributed Storage Systems", *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539-4551, 2010.

[21] Manisha Kalkal and Sona Malhotra, “Replication for Improving Availability & Balancing Load in Cloud Data Centres”, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 4, pp 108-110, Apr. 2015.

[22] S. Sang, "Implementation of Cyclic Redundancy Check in Data Communication", in *International Conference on Computational Intelligence and Communication Networks (CICN)*, Jabalpur, India, 2015.

[23] V. Pan, "Complexity of Computations with Matrices and Polynomials", *SIAM Review*, vol. 34, no. 2, pp. 225-262, 1992.