Accelerating Polynomial-Based Image Secret Sharing Using Hadoop

Sonali Patil, Roshani Raut, Chaitrali Sorte, and Gauri Jha

Abstract Polynomial-based secret sharing techniques are most preferred in many applications. Image secret sharing techniques based on polynomial functions become computationally heavy if image size is large. Sequential operations are not that effective for construction and reconstruction algorithms on large size images. Parallel computing works efficiently when load and time reduction on a single machine is required. The concurrent execution of polynomial operations on image pixel values while applying steps of construction of shares and reconstruction of the secret makes the scheme more efficient. Hadoop-based concurrent approach is proposed for polynomial-based image secret sharing scheme. The experimental results show that Hadoop-based cluster speeds up the process up to 18% by one slave and up to 14% by adding two slave nodes. The addition in number of slave nodes accelerates the performance up to 50% for very large size images in the polynomial-based secret sharing algorithms.

Keywords Secret sharing · Image processing · Parallel computing · Hadoop · HDFS

1 Introduction

A polynomial-based secret sharing scheme is first time proposed by Shamir [1] in 1979. Later, Thein and Lin [2] put forward image secret sharing based on Shamir's [1] polynomial-based secret sharing on image pixel values. The approach reduces size of the image shares generated in construction phase, and also the generated secret image is of good quality (same as the original image). The wide ranges of applications are possible using extended capabilities [3] in secret sharing schemes.

S. Patil · R. Raut (✉) · C. Sorte
Department of Information Technology, Pimpri Chinchwad College of Engineering, Nigdi, Pune, India

G. Jha
Consultant, Advanced Delivery Consulting, Sydney, Australia

N. Marriwala et al. (eds.), *Emergent Converging Technologies and Biomedical Systems*,
Lecture Notes in Electrical Engineering 841,
https://doi.org/10.1007/978-981-16-8774-7_6

A polynomial-based image secret sharing algorithms are highly secure but computationally heavy due to the requirement of polynomial operation on each pixel value. In large size of images, construction and reconstruction step computation is done on pixel-by-pixel for complete image. This is computationally meticulous as the database size to be processed is comparatively large. This results into large computational complexities of construction and reconstruction algorithms. In real life, these approaches need acceleration of processing of algorithms for use. The parallel advancements of image-based algorithms are suggested by few researchers [4, 5].

The Hadoop File System (HDFS) [6, 7] is implemented using distributed file system design. Hadoop File System is fault tolerant and can be implemented with minimal cost. It is able to accumulate the huge amount of data and also can provide access to that data at ease. Hadoop enables interface to HDFS through a command interface. It also provides streaming access to file system data. The clusters are formed by the integral servers, Name Node and Data Node [8, 9].

2 Related Work

2.1 Thein and Lin's Secret Sharing [2]

Thein and Lin [2] used polynomial-based Shamir secret sharing [1] for image threshold secret sharing. The secret image pixel values are used as coefficients to construct as polynomial used for constructing share images. The share image pixel values are used for reconstruction of the secret image using Lagrange's interpolation. This is very effective secret sharing method to distribute confidential images secretly. For very large size of images, it becomes inefficient due computationally heavy load.

2.2 Efficient Image Secret Sharing Using Parallel Processing for Row-Wise Encoding and Decoding [4]

A multithreading approach is implemented in [4] where it is observed that the concurrent approach is very effective and efficient to apply practically. In every algorithm, the multithreading logic needs to be implemented to achieve parallelism. The author demonstrated the use of parallel processing for efficient secret construction and reconstruction algorithms. The parallel approach is implemented using Unix platform. The more efficient parallel platforms can enhance the performance of image secret sharing methods.

2.3 Hadoop and HDFS [6–8]

Hadoop is basically an open-source framework which focuses on computations of distributed storage. Hadoop processes massive data on commodity hardware. A dedicated file system called Hadoop Distributed File System (HDFS) stores big data and supports distributed tasks estimations in Hadoop clusters. The conception of HDFS is implemented using Java. The Hadoop services may be characterized in terms of components such as storage component and processing component. Here, the HDFS works as storage component, whereas MapReduce works as processing component.

The HDFS grants dependable data storage. It also trails Master–Slave architecture. The Name Node is the master node. It only contains the metadata. The Name Node protects and maintains incoming data. The maintained information, i.e., metadata, is required for data retrieval as the data is distributed over numerous nodes. Data Node behaves as a slave node. It stores the actual data in the HDFS and interconnects with the files which are stored in that particular node along with the Name Node. The slave node is efficient enough to create new blocks of data along with manipulating and removing the blocks. It can also replicate the blocks if Name Node requires it to be done [9].

2.4 Hadoop-HDFS-Map Reduce [10]

The parallel processing of large amount of data is modeled through MapReduce. The main advantages are in effective storing of large images and effective accessing of large size images. Also, filtering of images and processing of images become effective.

2.5 Hadoop Image Processing Interface

A traditional Hadoop MapReduce program strives in presenting input image and output image data in a convenient format. The image library, “Hadoop Image Processing Interface” (HIPI) is built to be exercised with Apache Hadoop. HIPI is a solution to store a huge compilation of images on the dedicated file system of Hadoop. It also makes them available for effective distributed processing with few parallel programming components like MapReduce. A HIPI Image Bundle (HIB) is a collection of images characterized as a single file on HDFS.

3 Proposed Method

The proposed method is implemented for two steps—Formation of image shares from secret images and reformation of original secret from shares. The below section elaborates construction and reconstruction method of images for polynomial image secret sharing using Hadoop.

3.1 *Image Secret Sharing*

Construction and Reconstruction of Master–Slave Approach using Hadoop

The proposed parallel approach is based on master–slave model, as shown in Fig. 1 on Hadoop to achieve the parallel computation. It is implemented on one master and multiple slaves. The used approach is shown below.

As shown in Fig. 1, the intermediate results are produced using input records functional with Mapper. Reducer aggregates these generated results. The local summation is executed by combiners. The shuffling of intermediate data with reducer is performed by partitions.

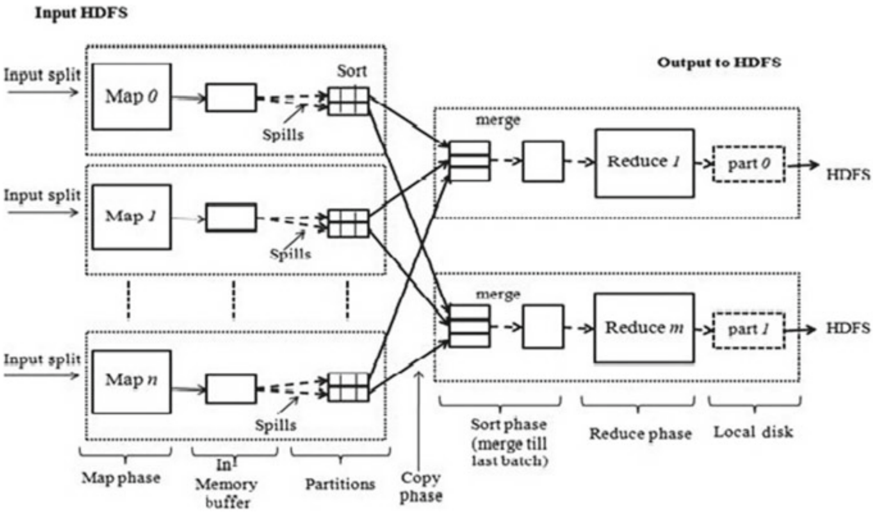

Fig. 1 HDFS framework

3.2 Construction of Shares

Figure 2 indicates construction of shares on Hadoop framework which consist of single Name Node (Master Node) and two Data Nodes (Slave Node). The Name Node distributes partition datasets to Data Nodes. The Data Nodes will compute polynomial computations of their respective pixel values of secret image.

The detailed illustration formation of shares of secret image using polynomial method is shown in Fig. 2.

Master node

i. Master will split the image row-wise into k-number of blocks (partitions) of fixed size.
ii. Master will transfer each partition to respective slaves.
iii. Each partition will contain parameters with row and column numbers.

Slave node

i. Each slave will apply Thein and Lin's method on every row of received partition to construct a $(k - 1)$ degree polynomial.
ii. Polynomials must be computed for each participant, starting from 1…n.
iii. No slaves will quit until the task is finished.
iv. All the computed values will be sent to master node.

Master node

i. Master node will collect all the values from all slaves and will create shares.
ii. These shares will be distributed to all the participants.

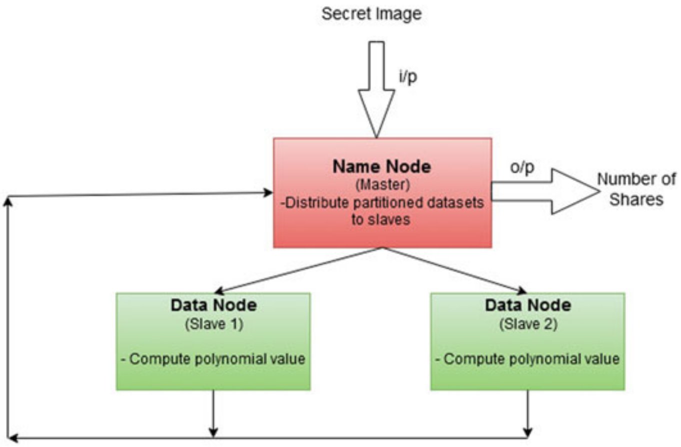
Secret Image
i/p
Name Node
(Master)
-Distribute partitioned datasets to slaves
o/p
Number of Shares
Data Node
(Slave 1)
- Compute polynomial value
Data Node
(Slave 2)
- Compute polynomial value

Fig. 2 Construction of shares

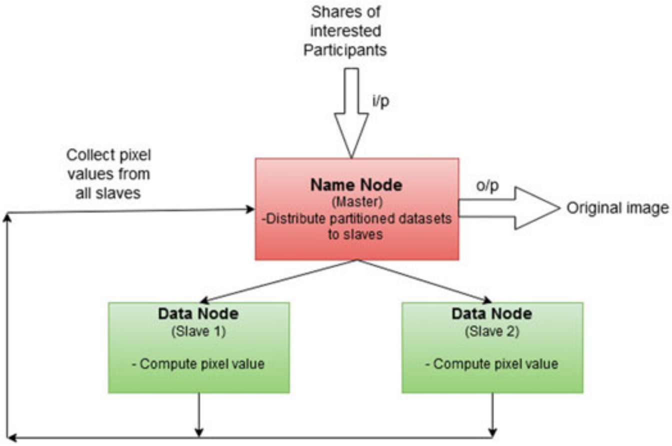

Fig. 3 Reconstruction of original secret

3.3 Reconstruction of Original Secret from Shares

Figure 3 indicates reconstruction of original secret on Hadoop framework with single Name Node (Master) and two Data Nodes (Slave).

Master node

i. *k*-shares will be given as input to master node collected from interested participants.
ii. Master node will decide *k*-partitions for slaves.
iii. Each partition will be passed containing parameters with row and column numbers of all shares.
iv. The partitions will be distributed to all the slaves.

Slave node

i. Each slave will choose first pixel from every share which is chosen.
ii. Every single slave will utilize Lagrange's interpolation formula to create an equation from *k*-selected pixel values of *k*-shares.
iii. Entire coefficient of derived equations will be used as pixel values for resultant image.
iv. Each slave will imitate the steps (ii) and (iii) for each and every specified row.

Master node

i. Master node will collect all the pixel values from all slaves and will present the reconstructed secret image.

The proposed method is implemented in HDFS with varying the number of slave nodes which results in better performance as compared to the traditional approach.

4 Result and Analysis

The proposed approach is implemented using Apache Hadoop on one master node and multiple slave nodes. The secret image of Leena, Baboon, Barbara, and Pepper of size 512 × 512 pixel are used from standard image dataset.

Tables 1 and 2 show the experimental results obtained with both approaches. The readings are observed for sequential approach (standalone machine) and using Hadoop cluster having one slave and Hadoop cluster having two slaves.

As it is observed from Table 1, the Hadoop cluster speeds up the process as we increase the number of slaves in the system for processing of large images as compared to sequential method.

The construction time is more as compared to reconstruction time as the number of shares, from which original image is to be generated are less than the shares needs to be created in the construction phase.

5 Conclusion

It is observed that the time required by the system on a Hadoop-based distributed platform is much less than that is required on a standalone machine for sequential approach. Compared to standalone machine, time requirement of system with one slave is 18% and the same with two slaves is 14%. This demonstrates

Table 1 Time comparison table for construction using sequential verses Hadoop approach

Secret image	Required time		
	Standalone machine (in s)	Single slave machine (in s)	Two-slave machine (in s)
Lena	1.012	0.182	0.142
Baboon	0.984	0.177	0.138
Barbara	0.962	0.173	0.135
Pepper	0.912	0.164	0.128
Average	0.9675	0.174	0.135

Table 2 Time comparison table for reconstruction using sequential verses Hadoop approach

Secret image	Required time		
	Standalone machine (in s)	Single slave machine (in s)	Two-slave machine (in s)
Lena	0.0683	0.0122	0.010
Baboon	0.0652	0.0117	0.0091
Barbara	0.0644	0.0115	0.0089
Pepper	0.0612	0.0110	0.00854
Average	0.0647	0.0116	0.0091

that on increasing number of slaves, the system will be more efficient and will construct/reconstruct in even less time, thus reducing load on single machine. Also for image of smaller sizes, sequential approach is proved to be better but as the size of image goes on increasing, time efficiency is observed. Hadoop accelerated the construction and reconstruction time for bigger images, for which computations are extra compared to smaller images by distributing the task to various slaves.

References

1. Shamir (1979) How to share a secret. Commun ACM 22(11):612–613
2. Thein CC, Lin JC (2002) Secret image sharing. Comput Graph 26:765–770
3. Patil S, Deshmuth P (2012) An explication of multifarious secret sharing schemes. Int J Comput Appl 46(19):0975–8887
4. Sonali P (2018) Efficient image secret sharing using parallel processing for row-wise encoding and decoding. In: International conference on intelligent computing and applications, advances in intelligent systems and computing. Springer Nature Singapore Pvt Ltd, Singapore, p 632
5. Sachdeva K, Kaur J, Singh G (2017) Image processing on multi node hadoop cluster. In: International conference on electrical, electronics, communication, computer and optimization techniques (ICEECCOT)
6. Premchaiswadi W, Romsaiyud W (2012) Optimizing and tuning map reduce jobs to improve the large-scale data analysis process. Int J Intell Syst
7. Arkan A, Al-Hamodi G, Songfeng LU (2016) MRFP: discovery frequent patterns using mapreduce frequent pattern growth. In: International conference on network and information systems for computer
8. Chen H, Lin TY, Zhibing Z, Zhong J (2013) Parallel mining frequent patterns over big transactional data in extended mapreduce. In: IEEE international conference on granular computing (GrC)
9. Archita B, Deipali G (2015) CFI mining using DAG based on MapReduce. Int J Emerg Trends Technol Comput Sci (IJETTCS) 4(3)
10. Mohamed HA (2012) Hadoop mapreduce for remote sensing image analysis. Int J Emerg Technol Adv Eng (IJETAE) 2(4)