International Journal of Multimedia and Ubiquitous Engineering
Vol. 9, No. 12 (2014), pp. 443-452
http://dx.doi.org/10.14257/ijmue.2014.9.12.38

MADR Algorithm to Recover Authenticity from Damage of the Important Data

Seong-Ho An[1], *Kihyo Nam[2], Mun-Kweon Jeong[2] and Yong-Rak Choi[1]

[1]Dept of Computer Engineering, Daejeon University, Yongun-dong, Dong-gu, Daejeon, Republic of Korea
seongho.an83@gmail.com, yrchoi@dju.ac.kr
[2]UMLogics Co., Ltd., 17, Techno 2-ro, Yuseong-gu, Daejeon, Republic of Korea
{nkh, jmk}@umlogics.com

Abstract

In order to preserve important data including a company's intellectual properties which require security or deserve to be archives, we generally do it electronically in disk mirroring, also known as RAID, or external storage devices. Digitized data can be corrupted if the storage devices deteriorate or by external access, and such compromised data undermine their authenticity and usability. This thesis presents a new method to recover damaged electronic data to restore their authenticity and usability.

Keywords: *MADR, Damaged Data, Data Recovery, File Format*

1. Introduction

Key data stored in a digital form can be properties of a company or archives for a country or its industries. To preserve them, they can be encrypted in a perspective of security which only authorized users can access, or we may archive them in disk mirroring or CDs for a recording purpose. However storages like hard disks or CDs to save data have finite durability and are susceptible to outside influences. We can take a precaution using disk mirroring and copying data through various storage devices, currently there is not a reliable safety net to prevent corruption of parts of data in files stored in those devices.

Existing disk recovery methods can be categorized into software-based and hardware-based. Software-based disk recovery methods work using partition attributes and signature information of files [1-2]. But if certain areas are overwritten or damaged, the recovery does not work. Peter Gutmann introduced a hardware-based disk recovery method in 1996 [3]. It inspects hard disks using an atomic microscope that can measure magnetic force and recovers data by the characteristic that data are saved in a hard disk by magnetic feature. If certain areas of data are damaged, the method can recover the corresponding area in hardware level. It is possible when bit information remains in the physical sector area. But it requires a lot of time and efforts and has limitations if the areas have been overwritten seven times or more. In addition RAID technology is used to protect data, RAID5 and RAID6 recover data using a parity disk [4]. However, recovery is limited if there are two faulty disks or more.

We propose a MADR (Multi Array Data Recovery) algorithm that uses a file format to ensure preservation of important data and to technically address partial data damage and that takes advantage of recovery blocks of the corresponding file format. In this thesis we focus on a MADR algorithm with two-array recovery blocks.

[*] *Corresponding author*

ISSN: 1975-0080 IJMUE

2. Data Format for MADR Algorithm

Dollar proposed eight requirements to enhance preservation of electronic archive. Eight requirements are Readable – information system is always accessible and usable, Intelligible – expressing information is possible, Identifiable – able to distinguish among information objects based on unique attributes, Encapsulate – able to combine into composite records, Retrievable – able to search, Understandable, Reconstructable, Authentic – records shall not be altered for a long time [5].

Defining a format to recover electronic records requires metadata. Metadata are values of record attributes in electronic records. To manifest the proof of activity, event information describing not only information of records but also how they have been produced and managed is essential to record management. In other words, in order to prove that records are authentic, have been managed without any damage or any forgery, have been produced while tasks related to them have run, and are always easy to search and clear to understand, metadata tied to the records should be produced and stored [6].

Electronic records marked as key data can be stored encrypted along with metadata, which guarantee authenticity. The Data format shown in Figure 1 supports authenticity for encrypted data and normal data and their data recovery.

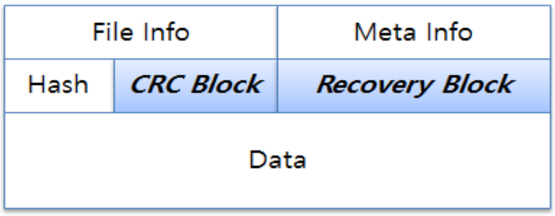


Figure 1. Recovery Data Format Structure

- File Info: basic information of the file including its filename, its size, its format, and so on
- Meta Info: information including creation date, owner, modification events, and hash value.
- Hash: value to guarantee integrity of data
- CRC block: value to confirm an area where data are damaged.
- Recovery block: block used to recover damaged data

The data format has a CRC block to locate a damaged data block and a recovery block to repair damaged data. Additionally it has optional data fields like hash value, creation date, and owner to guarantee authenticity. The crucial components to apply data recovery methods are the CRC block and a data block, and the CRC block is not needed if we can find other ways to correctly locate where data are damaged. To produce CRC blocks from data blocks, the recovery method extracts a CRC value from each data block of a specified size and sorts CRC values to fill the CRC blocks as shown in Figure 2.

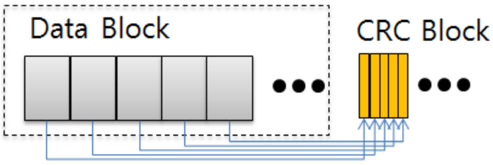


Figure 2. Creation of CRC Block

From data blocks in a two-dimensional matrix with m rows and k columns as shown in Figure 3, k recovery blocks are created with the XOR operation of data blocks in each column, and m recovery blocks are created with the XOR operation of data blocks in each row.

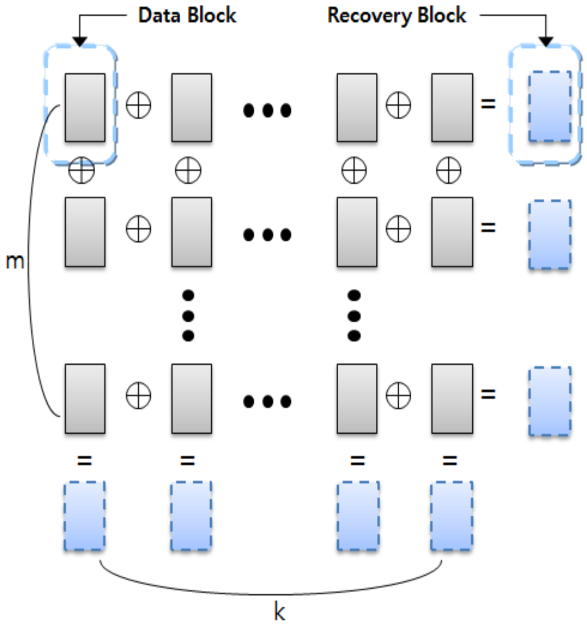


Figure 3. Creation of Recovery Block

Depending on the actual size of data, the recovery method can create multiple blocks of two-dimensional matrixes of recovery block.

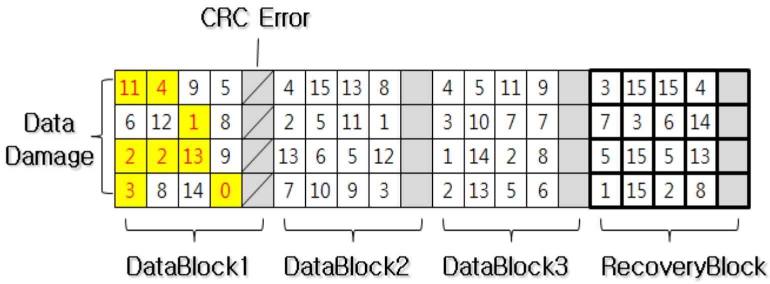
3. tow-array MADR Algorithm

The Data recovery technology is based on characteristics of XOR operation. Let “A”, “B”, and “C” be data blocks in one group and let “R” be a recovery block for that group. The XOR operation allows recovery of one damaged data block using the recovery block and the other two data blocks that are not damaged.

$$A \oplus B \oplus C = R, \qquad A \oplus B \oplus R = C$$

$$A \oplus R \oplus C = B, \qquad R \oplus B \oplus C = A$$

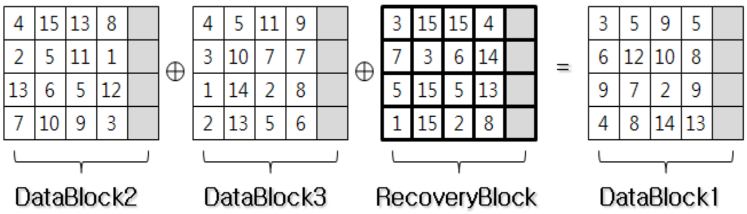
We can check a damaged data block with the CRC value. Suppose that a data block has 4x4 bytes along with a CRC byte for each row and that a recovery block is created from such three data blocks as depicted in Figure 4. If data in DataBlock1 are damaged as in Figure 4, a CRC error is detected in DataBlock1. We can recover the damaged data using DataBlock 2, DataBlock 3, and the recovery block.



11	4	9	5		4	15	13	8		4	5	11	9		3	15	15	4	
6	12	1	8		2	5	11	1		3	10	7	7		7	3	6	14	
2	2	13	9		13	6	5	12		1	14	2	8		5	15	5	13	
3	8	14	0		7	10	9	3		2	13	5	6		1	15	2	8	

Figure 4. Damaged Data Block by CRC Error

After the recovery method detects the CRC error and finds the damaged data block, it recovers the data block using the recovery block as shown in Figure 5.



4	15	13	8	
2	5	11	1	
13	6	5	12	
7	10	9	3	

DataBlock2

⊕

4	5	11	9	
3	10	7	7	
1	14	2	8	
2	13	5	6	

DataBlock3

⊕

3	15	15	4	
7	3	6	14	
5	15	5	13	
1	15	2	8	

RecoveryBlock

=

3	5	9	5	
6	12	10	8	
9	7	2	9	
4	8	14	13	

DataBlock1

Figure 5. Recovery of Damaged Data Block using the Recovery Block

The XOR operation of DataBlock2, DataBlock3, and the recovery block is performed to recover the damaged DataBlock1. If there are two damaged data blocks or more in the group, recovery is not possible. The probability of recovery may increase through adjustment of the group size and the interval of the recovery block.

As an example, we created samples of a 4x4 data block and its recovery blocks and analyzed the types in which data blocks can be recovered. Let “L1” and “L2” be lines for rows and “L3” and “L4” be lines for columns, and we can recover damaged blocks using the XOR operation.

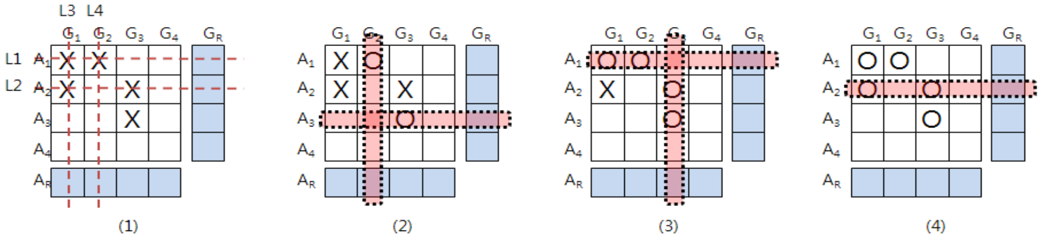


Figure 6. Sample 1 of Data Recovery of Randomly Damaged Data Blocks

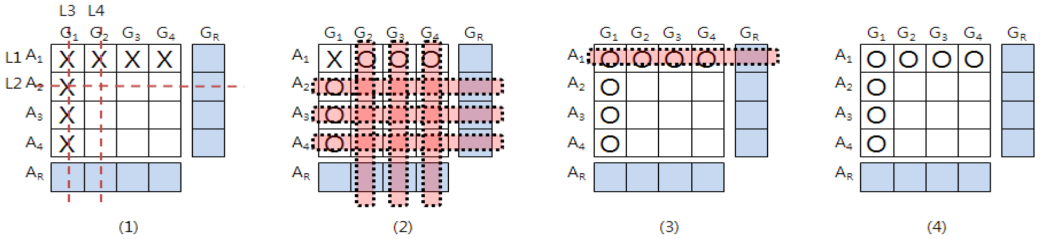


Figure 7. Sample 2 of Data Recovery of Randomly Damaged Data Blocks

Data recovery is possible if not all four data blocks intersected by the four red dashed lines are damaged as in Figure 6, Figure 7. Damaged data blocks are recovered with recovery blocks through steps (2), (3), and (4). On the other hand, if all four data blocks intersected by the four red dashed lines are damaged, partial recovery is possible, but the complete recovery is not possible as in Figure 8. The method can recover damaged data blocks at (A4,G2) and (A2,G4) using a pair of data blocks at (AR,G2) and (A4,GR) and a pair of data blocks at (AR,G4) and (A2,GR), respectively. Recovery blocks to recover data blocks at (A1,G1), (A1,G3), (A3,G1), and (A3,G3) are not available.

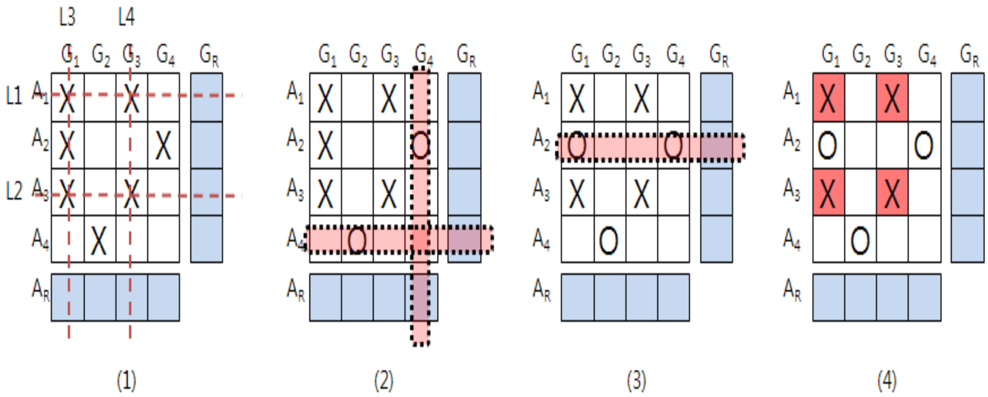


Figure 8. Partial Recovery of Damaged Data Blocks

We can see that data recovery is not possible if all four data blocks intersected by the four red dashed lines are damaged as in Figure 9.

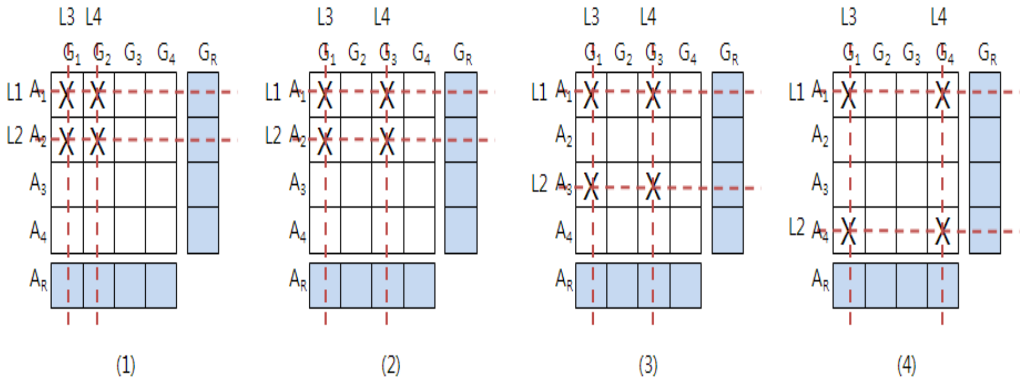


Figure 9. Unrecoverable Case

4. Percentage of Recovery Blocks

Two arrays of recovery block require more recovery blocks than one array. We measured the percentage of recovery blocks based on the following criteria:

- Set the size of the original image to 1 Gbyte(1,048,576 Kbyte)
- Set the size of one data block to 32 Kbyte
- Set the group size of data blocks to 1
- Measure the area size in the A axis by 128, 256, 512, and 1024
- Measure the two-array area size in the G axis by 128, 256, 512, 1024
- Assume the group size of data blocks to 1

We can get the number of total data blocks and total recovery block using the following formulae below.

$$DB\_quantity = Disk\_Size \; / \; DB\_size$$

$$RB\_quantity = T + k(T \; / \; m)$$

$$(T = Disk\_Size \; /(DB\_Size \times k))$$

For example we set both DB and RB to 128, 256, 512, and 1024 and can get the total number and percentage of data blocks and recovery blocks using the formulae as in Table I. The number of data blocks is 32,768 because the group size is fixed.

Table I. Total Number and Percentage of Data Blocks and Recovery Blocks

k / m		64		128		256		512		1024	
		count	per	count	per	count	per	count	per	count	per
64	DB	32768	3.13%	32768	2.34%	32768	1.95%	32768	1.76%	32768	1.66%
	RB	1024		768		640		576		544	
128	DB	32768	2.34%	32768	1.56%	32768	1.17%	32768	0.98%	32768	0.88%
	RB	768		512		384		320		288	
256	DB	32768	1.95%	32768	1.17%	32768	0.78%	32768	0.59%	32768	0.49%
	RB	640		384		256		192		160	
512	DB	32768	1.76%	32768	0.98%	32768	0.59%	32768	0.39%	32768	0.29%
	RB	576		320		192		128		96	
1024	DB	32768	1.66%	32768	0.88%	32768	0.49%	32768	0.29%	32768	0.20%
	RB	544		288		160		96		64	

We can see that the percentage of recovery block is decreasing as the numbers of k and/or m are increasing. As k and m increase, the percentage of recovery block is decreasing under 1%, which results in decrease of additional disk space required. But if k and m are too high, the number of data blocks for one recovery block is increasing so much that it may cause an adverse effect of decreasing the recovery percentage. To optimize recovery performance, we need to set the value of k and m accordingly while checking to see if there is enough free space in disk.

5. Evaluating the Recovery Percentage based on the Number of Recovery Blocks

5.1. Standards for the Performance Evaluation

We have set the following standards to measure the recovery percentage.

- Set the size of one data block to 1 Byte
- Set the size of the whole data blocks to 1 Mbyte(1,048,576 Byte) and 10 Mbyte(10,485,760 Byte).
- Set a normal data block to 0 and a damaged data block to 1
- Set the group size of data blocks to 1
- Measure the area size by 128, 256, 512, and 1024
- Measure the two-array area size with the same ratio of the area (ex: 128x128, 256,256...)
- Iterate 100 times at randomly generated 0.1% ~ 0.9% damaged areas.

Since the total disk size is 1Mbyte or 10Mbyte with each data block of 1Byte, assuming the actual data block size is 32 Kbyte in the key-data format, the total disk size of 1Mbyte corresponds to the total data block size of 32Gbyte, and the total disk size of 10Mbyte corresponds to the total data block size of 320Gbyte.

We will measure the percentage of recovery in a virtual disk (k x m x j) using a two-array recovery block. We will test with virtual disks of 64x64x256, 128x128x64, 256x256x16, 512x512x4, and 1024x1024x1 Byte. The experiment is conducted with the following procedures.

① Randomly make 0.1% of all the data blocks in a 64x64x256 disk damaged.
② Keep attempting to recover a data block of 64x64 until unable to do so.
③ From step ②, attempt to recover the damaged data block if it is the only one corresponding to one recovery block.

④ The disk recovery is successful if there is no damaged data block in the whole disk.
⑤ Repeat steps ①~④ 100 times.
⑥ Calculate the number of successful disk recoveries in the 100 iterations into a percentage
⑦ Repeat steps ①~⑥ in different damage rates
⑧ Repeat steps ①~⑦ in different area sizes
⑨ Repeat steps ①~⑧ in different disk sizes

We come up with probabilities and percentages of successful recovery with respect to the entire disk through the proposed algorithm. In other words, the probability of recovery refers to that of the entire disk recovery, and the percentage of recovery refers to how much the disk is recovered. The percentage of recovery block in the test results is the number of recovery block divided by the total number of data block.

5.2. Performance Test and Results

We first tested at damaged areas of 0.001%~0.009% and got a recovery percentage of 100% in all cases. Then we increased the damaged areas tenfold of 0.01%~ 0.09% and came up with the following results.

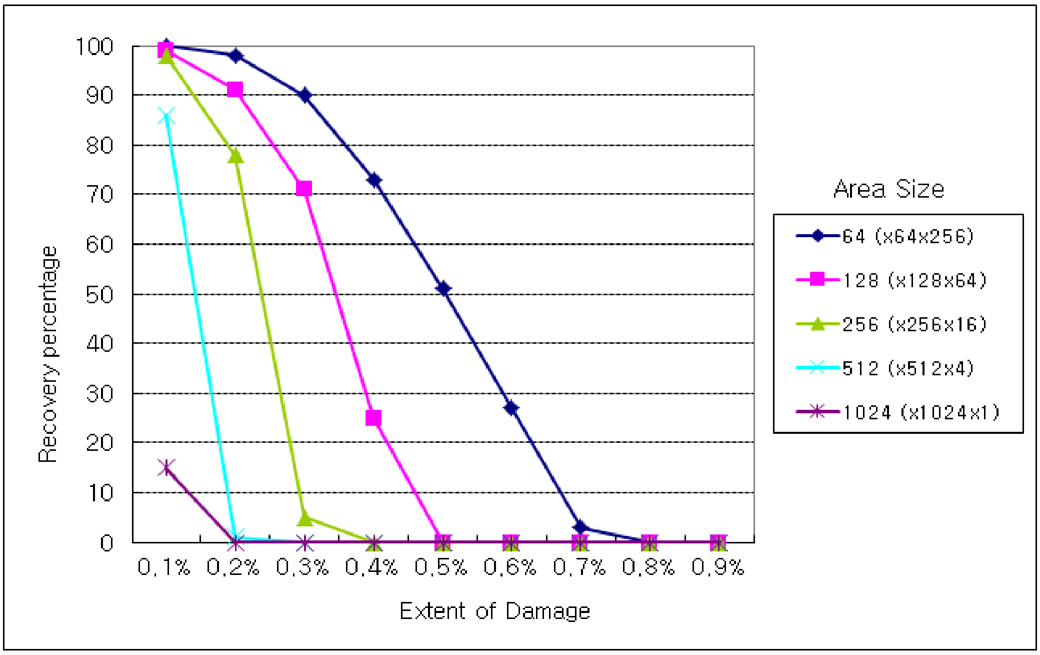


Figure 10. Recovery Percentage of the Entire Disk

The recovery percentage in one recovery block for each 128x128 data block unit is around 90% when the 1Mbyte virtual disk has 0.2% damaged area. When there is one recovery block for each 256x256 data block unit under the aforementioned conditions, the percentage is down to 78%. The recovery percentage in one recovery block for each 128x128 data block unit is around 55% when the 10Mbyte virtual disk has 0.2% damaged area. When there is one recovery block for each 256x256 data block unit under the aforementioned conditions, the percentage is down to 4%.

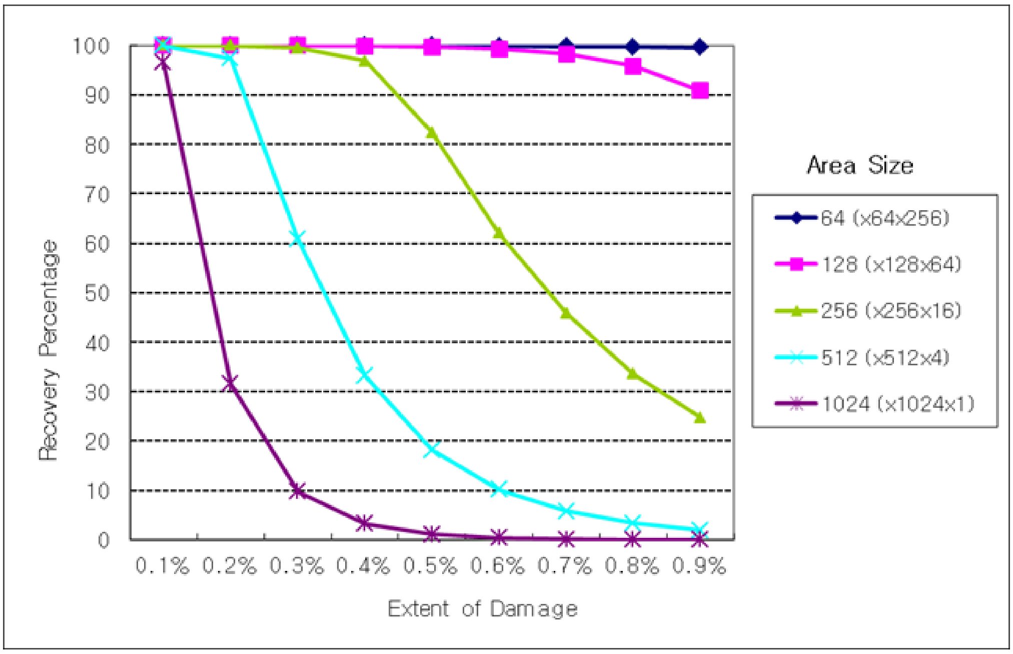


Figure 11. Successful Recovery Percentage of Damaged Data Blocks

The successful recovery percentage at 0.2% damaged area is 100% when the disk contains units of 128 x 128 data blocks and 99.93% when 256 x 256 data blocks.

The summary of test results indicates that a high percentage of recovery blocks in the two-array format maximizes the recovery probability.

6. Conclusion

The thesis has presented the two-array MADR algorithm with the associated data format structure to brace damage of key data. We have introduced the enhanced data format that includes a CRC block and a recovery block that are required to apply the recovery technology and a method to recovery damaged data blocks with recovery blocks. Because the higher proportion of recovery block results in the higher recovery probability, we can manage how many recovery blocks to deploy and how to place them, depending on the amount of free disk space. The number and size of recovery blocks are stored in metadata. The data format and the recovery method guarantee authenticity and integrity of recovered data.

We plan to further continue this study using MADR algorithms with recovery blocks of three or more arrays to possibly improve recovery performance.

Acknowledgements

The present study was supported by the Ministry of Knowledge Economy of Korea, Global Professional Technology Development Business. (Project No.10044104)

References

[1]. E. Huebner, D. Bem and C. K. Wee, “Data hiding in the NTFS file system”, Digital Investigation, vol. 3, Issue 4, **(2006)** December.

[2]. X. Guo-tian, “The Research of File Recovery Method on EXT3 File System”, Netinfo Security, **(2012)** March.

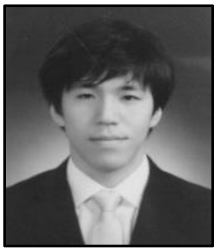
[3]. P. Gutmann, "Secure Deletion of Data from Magnetic and Solid-State Memory", 6th USENIX Security Symposium Proceedings, **(1996)** July.

[4]. M. Gilroy and J. Irvine, “RAID 6 Hardware Acceleration”, Field Programmable Logic and Applications, 2006. FPL '06, International Conference on, **(2006)** August.

[5]. C. M. Dollar, "Authentic Electronic Records: Strategies for Long-Term Access", Chicago: Cohasset Associates Inc., **(1998)** June.

[6]. In Requirements for Electronic Records management System, "Metadata Standard", The National Archives, **(2004).**

Authors

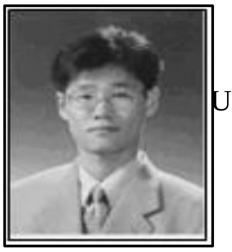


Seong-Ho An

Master of Science in Computer Engineering
Daejeon University
Republic of Korea

Kihyo Nam

Ph.D of Industrial Engineering(Korea University)
CISA(Certified Information Systems Auditor)
CISSP(Certified Information Systems Security Professional)
Adjunct Professor(Konkuk University)
Republic of Korea



Mun-Kweon Jeong

Master of Information Industrial Engineering(Chungbuk University)
Director-General of UMLogics Co., Ltd.
Republic of Korea

Yong-Rak Choi

Professor of Computer Engineering(Daejeon University)
Ph.D of Computer Science(Chung-Ang University)
Republic of Korea