Journal of Engineering and Applied Sciences 13 (5): 1057-1063, 2018
ISSN: 1816-949X

Development of Distributed DBMS Based Duplication File Removal Techniques

Jeong-Joon Kim and Jeong-Min Park
Department of Computer Engineering, Korea Polytechnic University,
Gyeonggi-do, 15073 Siheung-si, South Korea

Abstract: By using the server for a long time, various files are created and saved in various folders. Also in the case of a data server, overlapping files become increasing through a large amount of uploading which may result in space shortage. The server PC needs data deduplication technology for effective duplicate file management and file service and various technologies such as MD5, hash, SHA1, etc. are required for this. In addition by using DBMS to store information of files, we try to provide a system that can minimize duplication when new files are created or modified.

Key words: Distributed DBMS, deduplication, recovery, SHA-2, hash, SHA1

INTRODUCTION

As the server is used for a long time, various files are created and stored in various folders. In addition, in the case of a data server, redundant files are increased through a large amount of uploads, resulting in a space shortage (Kim and Kim, 2016; Oh *et al*., 2012).

As the server is used for a long time, various files are created and stored in various folders. Also in the case of a data server, redundant files are increased through a large amount of uploading, resulting in a case where the space is shortened. Server PC needs data de-duplication technology for efficient duplicate file management and file service and various technologies such as MD5, hash and SHA1 are required. Also, it is aimed to provide a system that can minimize duplication when creating new files or modifying files by storing information of files using DBMS (Moon *et al*., 2007).

Literature review

Hash function and CRC (Cycle Redundancy Check): The hash function produces fixed-size results by truncating, substituting or repositioning variable-size data and this result is often referred to as a hash value (Jung and Choi, 2014; Anonymous, 2018a). If the two hash values are different, the original data for that hash value must also be different (not reversed). The quality of the hash function is determined by how little hash collision (when two different hash values of the data are equal) in the expected input area. The more collisions, the more difficult it is to distinguish between different data and the cost of retrieving data increases. CRC is a method of determining a check value for checking whether there is an error in transmitted data when transmitting data through a network or the like. CRC16 and CRC32. Before transmitting the data, the CRC value is calculated according to the value of the given data and the CRC value is added to the data and the CRC value is calculated again as the value of the data received after the data transmission. Then, the two values are compared. If the two values are different from each other, it can be seen that an error is added due to noise or the like during data transmission. CRC is easy to implement in binary-based hardware and is excellent for detecting common errors in data transmission. However, due to the structure of the CRC, it is easy to create other data with a given CRC value intentionally and therefore can not be used to check the integrity of the data failure.

MD5 (Message-Digest Algorithm 5): A 128 bit cryptographic hash function is used to process input variable-size data into padding (data is divided into 512 lengths) and divided into 512 bit message blocks (Anonymous, 2018b, 2017).

Padding first adds the first single bit 1 to the end of the message and then pads it with zeros to a position that is 64 bits less than the length of a multiple of 512. The remaining 64 bits are filled with a 64 bit integer value representing the length of the original (original) message. It is used mainly for checking the integrity of programs or files. In addition, hash collisions can occur and must be implemented in little-endian fashion (Fig. 1).

SHA-1 (Secure Hash Algorithm): There are SHA-1 and SHA-1 and SHA-2 (SHA256/224 and SHA-512/384) but they are based on popular SHA-1 (Lee *et al*., 2010). Using a 160 bit cryptographic hash function, the maximum input

Corresponding Author: Jeong Min Park, Department of Computer Engineering, Korea Polytechnic University,
237 Sangideahak-ro, Siheung-si, Gyeonggi-do, South Korea

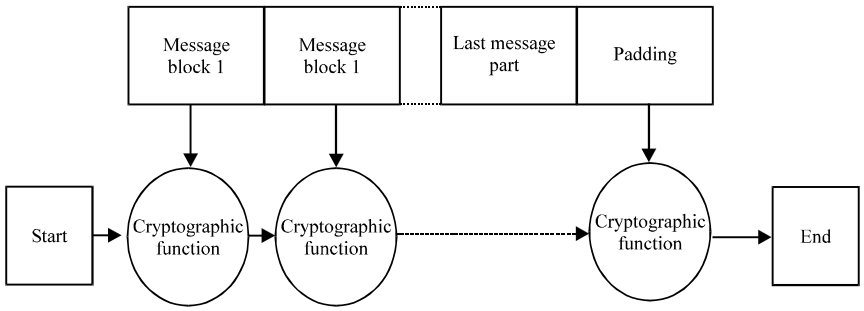

Fig. 1: Message-digest algorithm 5

Table 1: SHA-1 configuration

Algorithms	Hash value size	Inside condition size	Block size	length limit	Word size	Number of courses	Used operation	Crash
SHA-0	160	160	512	64	32	80	+, and, or, xor, rotl	Discovered
SHA-1	160	160	512	64	32	80	+, and, or, xor, rotl	Offense of attack only
SHA-256/224	256/224	256	512	64	32	64	+, and, or, xor, shr, rotr	-
SHA-512/384	512/384	512	1024	128	64	80	+, and, or, xor, shr, rotr	-

message size is limited to 2^{64} bits. The input variable size data is divided into n 512 bit blocks and processed by the US standard encryption algorithm based on MD4. It is somewhat slower than MD5 but it keeps large message summaries safer when subjected to violent collisions and strikes. Finally, it must be implemented in big-endian fashion (Anonymous, 2017; Lee *et al*., 2010) (Table 1).

Considerations and limitations when removing duplicate files: Duplicate and duplicate option setting, duplicate log setting, duplicate condition: file name, time, size, creation time, duplicate file size date, modification date, file contents (checksum/hash value), DB structure (link file name/link path/original file name/original path/(other conditions)). Only file duplication can be removed (except for inter-directory duplication) and the hash code conflict of the file duplication algorithm is specified as a constraint.

MATERIALS AND METHODS

System design

System structure diagram: Figure 2 shows the overall system structure of the duplicate file removal system. Duplicate file removal system consists of file deduplication manager and duplication manager. The file de-duplication manager consists of modules that provide the following functions.

Destination volume to perform deduplication/volume to set directory/directory specification module, the size of the file, MD5 and SHA-2 hash values, a duplicate file check module for comparing the actual data values to check duplication, a duplicate check condition filter module for filtering the check target file according to options such as creation date, MDS is a metadata management module that manages metadata about file duplication, a schedule setting module for scheduling to perform a deduplication operation on a specific schedule are composed.

The redundancy rate manager is composed of modules that provide the following functions. A total redundancy check module for outputting redundancy information for all files, output module for outputting redundancy rate information for each job, a full duplicate file listing module that outputs a list of duplicated files in the past are composed. The main processing steps of the duplicate file removal system consist of request metadata (file name, path, size) of files to be deduplicated to MDS, in the deduplication process, the DS requests the result of comparing the MD5 and SHA-2 checksums with the actual file data, the redundancy rate for each job is displayed and stored in MDS and removal of the actual duplicate file requires the DS to delete the file.

Key features and processes

Main function: Table 2 shows the functions that can be set up for deduplication in the file deduplication system. The deduplication target volume/directory is a mandatory option. The following examples shows an example of performing deduplication from the "default" "volume,/test" path. functions.

Example for Deduplication in./test:

Root@->gfs_dedup-v default-p/test

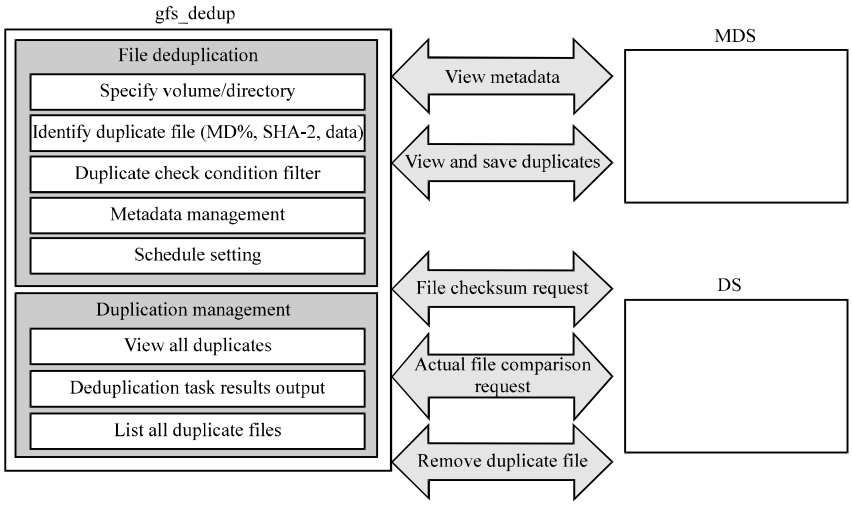

Fig. 2: System structure of duplicate files removal

Table 2: Deduplication function for duplicate files removal

Command	Description
-v <volume_name> --volume <volume_name>	Name the target volume
-p <directory_path> --path <directory_path>	Specify the path to the target directory
--runtime <sec>	Specify the number of seconds to perform deduplication (default: 0) If the specified time is 0, the operation proceeds until all d eduplication is finished
-d<date>	Deduplicate only files created after the specified creation date
--day <date>	Input format "yyyy-mm-dd hh:mm:ss" If you do not specify a date, all files will be deduplicated regardless of the creation date
-r {True|False} -recursive {True|False}	Specifies whether duplicate subdirectories are removed from the specified destination path (default: True)
--checkonly	If this option is specified, only the test results are output
--level {Checksum|Data}	Specify the deduplication level Checksum: file size, SHA-2 Hash value comparison Data: file size, SHA-2 hash value, data actual value comparison

Table 3: Commands to schedule with crontab

Command	Description
-u <user>	Perform a crontab operation on the entered user If omitted do the work for the user who is now performing crontab
-l	Output current cron table
-e	Delete current cron table
-r	Edit current cron table

Table 4: Symbols used in cron table

Symbols	Description
*	Symbol for "always"
-	Specifies a range (ex) "1-3" is 1-3 o'clock
,	Separator role (ex) "1,3" is 1 and 3 o'clock
/	Specify periodic intervals (ex) "3-15/2" is every 2 h between 3:00 and 15:00

Table 5: Redundancy management command in duplicate files removal

Classification	Command	Description
Action	--stat --summary	Outputs the redundancy rate for the entire file
	--stat --job	Group information by each task to output redundancy rate
	--stat --list	Print a list of duplicated files for each job

Schedules can be set up using the crontab utility which supports job scheduling. Table 3 shows the command to set the schedule. Typically, you use the "crontab-e" command to schedule a job and the fields in the cron table are configured as follows.

Configuring fields in the cron table:

<Minute> <Time> <day> <Month>
<Day> <User> <Command>

<Minute> is 0-59, <time> is 0-23, <day> is 0-31, <month> is 0-12 (0 and 12 are December, 1 is January, ...) and <Day of the week> is 0-7 (0 and 7 for Sunday, 1 for Monday, ...). The symbols used in the cron table are as follows (Table 4). The following example shows how to perform deduplication at 5:30 am every Monday.

Example of performing a deduplication operation:

3005** 1 gfs_dedup--runtime

Table 5 shows the functions that can be set for redundancy management in the duplicate file removal system.

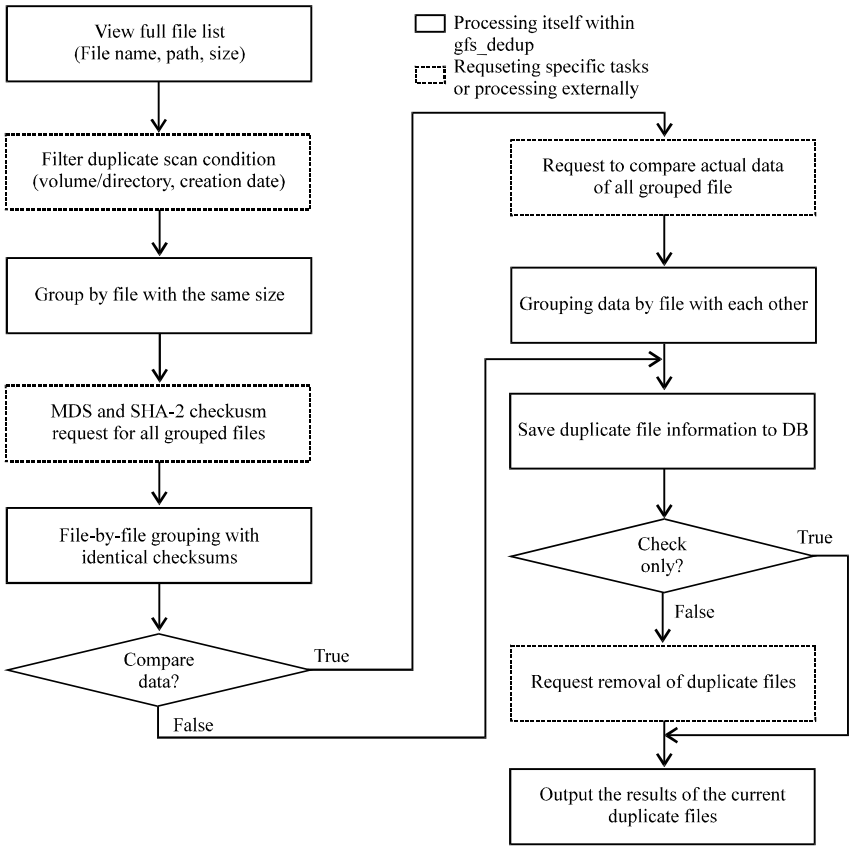

Fig. 3: File deduplication processing in duplicate files removal

Deduplication process: Figure 3 shows the process of de-duplication of files in the duplicate file removal system.

RESULTS AND DISCUSSION

System implementation: It analyzes the degree of file duplication by specific volume/directory and creates a hard link to actually remove duplicate files. The results of the deduplication performed previously are stored in the database and support the duplication rate of the file in various forms. Use the gfs_dedup command to analyze files existing in the target volume/directory, remove the actual duplicate files and query the duplication rate based on the results of the previous deduplication operation (Fig. 4 and Table 6).

Option setting

Examples (with option setting): Specifying a specific volume/directory to check for duplication. Command: gfs_dedup-vdefault-p/D1.1-checkonly. Perform deduplication including subpaths on specific volumes/directories. Command: gfs_dedup-v default-p/D1.1 -r-checkonly. Perform deduplication only on files created on or after a day before a specific volume. Command: gfs_dedup-v default-d 1. Summarize the results of the entire deduplication process. Command: gfs_dedup --stat-summary. Print a list of deduplicated files from a specific volume. Command: gfs_dedup-v default --stat-list (Fig. 5-10).

Table 6: Set options for deduplication

Options	Description	Remarks
-v, ...volume<volname>	The name of the deduplication target volume	New options
-p, ...path <path>	Deduplication target directory path	New options
-d, ...day	Duplicate only files created after that date	New options
-r, ...recursive	Whether subdirectories are searched	New options
...checkonly	Output only test results without actual deduplication	New options
...runtime <sec>	Time to perform deduplication (default: 0)	New options
...stat ...summary	Summarize the overall results and output redundancy	New options
...stat ...job	Output redundancy per deduplication operation	New options
...stat ...list	Output a list of deduplicated files	New options

The -v option of the gfs_dedup command can be used in conjunction with the options used for the redundancy check but at the same time the -p option

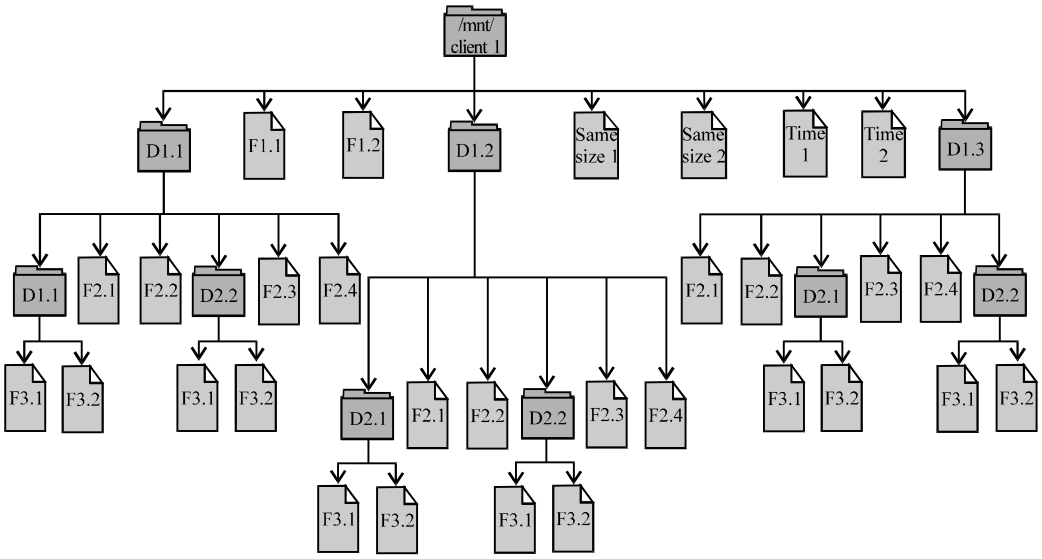

Fig. 4: Configuring deduplication destination files

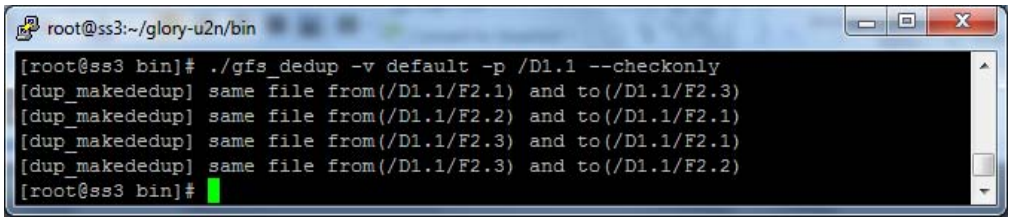

Fig. 5: Checking the amount of duplication in the specified volume/directory

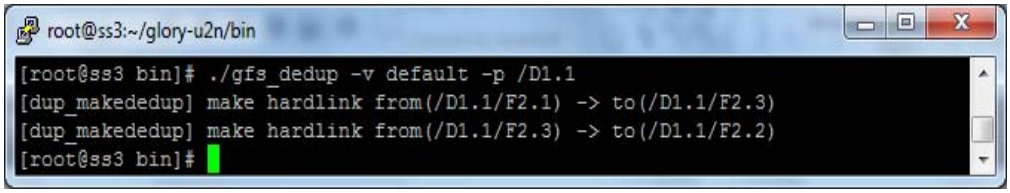

Fig. 6: Checking the amount of duplication in the specified volume/directory

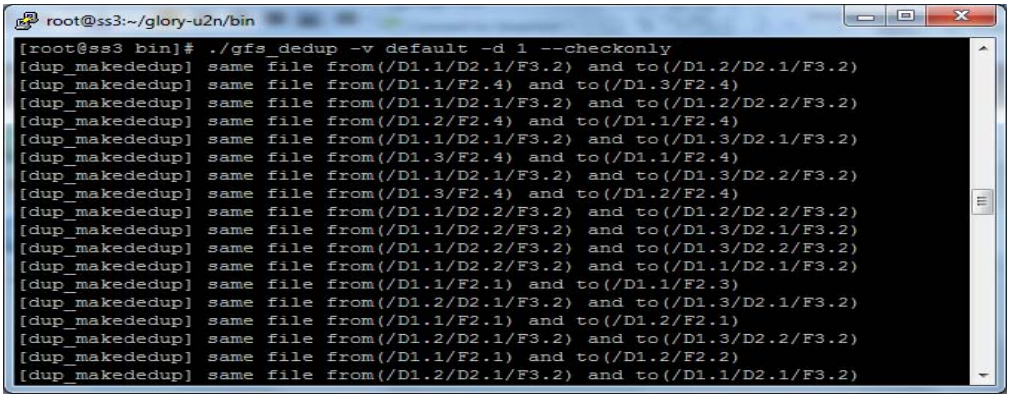

Fig. 7: Perform deduplication only on generated files after a specified date

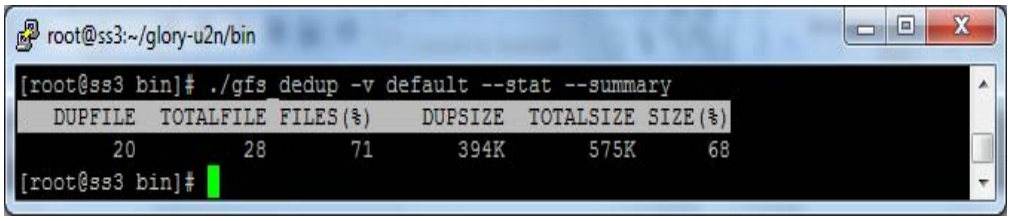

Fig. 8: Deduplication processing result

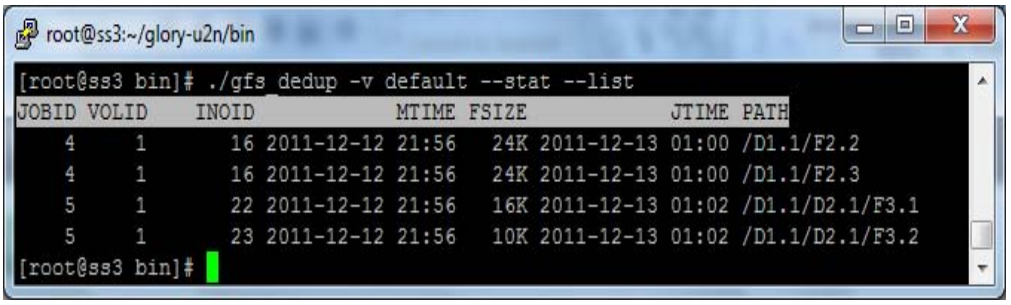

Fig. 9: Output a list of deduplicated files

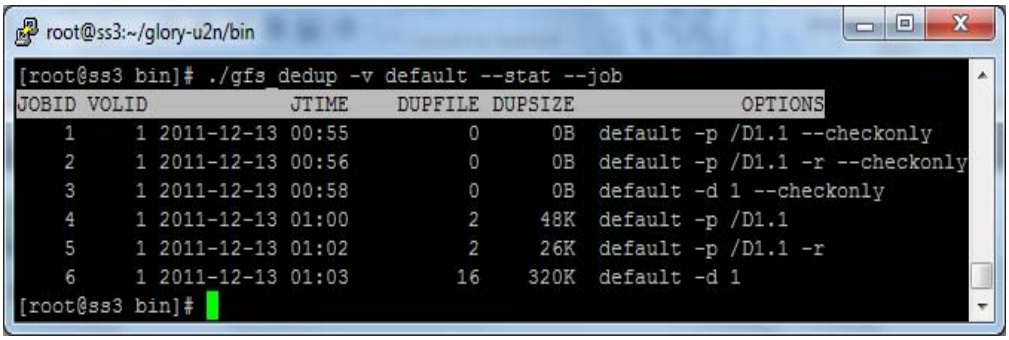

Fig. 10: Deduplication information output

should be excluded from being entered. Output deduplication job information performed on a specific volume. Command: gfs_dedup -v default --stat --job.

CONCLUSION

Server PC needs data de-duplication technology for efficient duplicate file management and file service and various technologies such as MD5, hash and SHA1 are required. Also, it is aimed to provide a system that can minimize duplication when creating new files or modifying files by storing information of files using DBMS.

In this study, we propose a deduplication algorithm that applies encryption algorithms such as MD5 and SHA-1 for efficiency and accuracy of service by providing it as a daemon that performs periodic deduplication by scheduling. Read the configuration file at the start of the service to enable the setting of file duplication option for specific volume/directory and to allow the file duplication rate to be periodically logged.

ACKNOWLEDGEMENT

This researcher was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2017 R1A2B4011243).

REFERENCES

Anonymous, 2017. Secure hash algorithms. Wikimedia Foundation, Inc, San Francisco, California. https://en.wikipedia.org/wiki/Secure_Hash_Algorithms.

Anonymous, 2018b. Cyclic redundancy check. Wikimedia Foundation, Inc, San Francisco, California. https://en.wikipedia.org/wiki/Cyclic_redundancy_check.

Anonymous, 2018a. Hash function. Wikimedia Foundation, Inc., San Francisco, California. https://en.wikipedia. org/wiki/Hash_function.

Jung, S.O. and H. Choi, 2014. Performance analysis of open source based distributed deduplication file system. KIISE. Trans. Comput. Practices, 20: 623-631.

Kim, D.W. and J.S. Kim, 2016. Analysis of the efficiency for some selected double-block-length Hash functions based on AES/LEA. J. Korea Inst. Inf. Secur. Cryptol., 26: 1353-1360.

Lee, E.H., J.H. Lee, Y.J. Jang and K.R. Cho, 2010. Implementation of high-throughput SHA-1 Hash algorithm using multiple unfolding technique. J. Inst. Electron. Eng. Korea SD., 47: 41-49.

Moon, C.J., M.Y. Choi, S.M. Kim and J.H. Jung, 2007. A synchronization algorithm for mobile database using message digest. J. KIISE. Databases, 34: 357-368.

Oh, S.J., T.J. Yun, C.H. Lee, J.K. Lee and K.H. Chung *et al*., 2012. Improved a mutual authentication protocol in RFID based on hash function and CRC code. J. Korean Inst. Commun. Inf. Sci., 37: 132-139.