Journal of Theoretical and Applied Information Technology
31st August 2011. Vol. 30 No.2

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

PERFORMANCE EVALUATION OF HASH ALGORITHMS FOR INTEGRITY IN DATABASE ARCHIVES

[1]AIHAB KHAN, [2]MALIK SIKANDER HAYAT KHIYAL, [3]MUHAMMAD IQBAL, [4]ASMA ARSHAD,
[1,2,4]Department of Software Engineering, Fatima Jinnah Women University, Rawalpindi, Pakistan.
[3]Department of Applied Sciences & Graduate Studies, Bahria University, Karachi Campus, Karachi, Pakistan.

E-mail: [1]aihabkhan@yahoo.com, [2]m.sikandarhayat@yahoo.com, [3]miqbal.bu@gmail.com , [4]super_engr@yahoo.com

ABSTRACT

The concept of digital archives to store long term data has readily increased in past few years. Many social, business and government organizations store their data in digital archives for long time span with many challenges along with its advantages. One of the major and most common problem is to ensure the integrity of data stored in digital archives. Usage of hash algorithms is solution to provide integrity in digital archives. Lots of hash algorithms are available now days but the problem is that usually the organizations are short of resources especially time and memory, so they always look for such algorithm that consumes the least amount of resources providing the security at maximum. So in this work the most commonly used hash algorithms are evaluated for their performance on the basis of time, memory and efficiency with variable data object size. SHA-1, MD-5, RIPEMD-160 and HAVAL-256 are evaluated for performance, being the most commonly used hash algorithms. Results show that MD-5 consume least amount of resources with somewhat questionable security, SHA-1 and RIPEMD-160 are more consistent in behavior with a compromising amount of resource usage and security and HAVAL-256 provides best security with high rate of resource usage.

Keywords: *Hash Algorithms, Integrity, Secure Hash Algorithm (SHA), Message Digest (MD), RACE Integrity Primitives Evaluation Message Digest (RIPEMD), Digital Archives*

1. INTRODUCTION:

The business, cultural and government organizations usually organize their data in form of large databases and store them in archive for long term storage. Along with many of the advantages of data storage in digital archives, a number of security challenges are also raised. One of the critical issues is to ensure the integrity of data objects in digital archives. A variety of hash algorithms are available to determine data integrity in digital archives. Organizations that store large databases in digital archives are usually short of resources especially time and memory so they always search such algorithm that consumes the minimum possible resources with maximum security.

This research work provides a solution to this problem by evaluating the performance of most commonly used hash algorithms on the basis of performance parameters like time, memory usage and efficiency/consistency of the hash algorithm. The best evaluated hash algorithm may be selected by archivist to effectively use it in the system to ensure the integrity of data objects in digital archives.

The main contribution of this research is to recommend the open world the criteria to select the best available hash algorithms depending on their data size and ensure the integrity using that hash algorithm. Also this let them efficient utilization of their resources and makes the integrity auditing process efficient and reliable.

The next coming sections will give the complete insight of the work. Section 2 summarizes related work and section 3 presents the proposed framework overview. Section 4 elaborates preliminaries and section 5 presents the proposed technique. Section 6 includes the experimental results and section 7 gives the conclusion and future work.

2. RELATED WORK:

Song et. al. [1] focuses on the fundamental requirement of long term integrity of digital archives. In his paper, a new methodology is developed to address the integrity of long term archives using rigorous cryptographic techniques. This approach basically involves generating small integrity token for each object to be archived and the cryptographic summary information (CSI) based on all the objects that are handled within a dynamic time period. It enables the continuous auditing of the holdings depending on the policy set by the archive. The independent auditor is able to verify the integrity of each version of the objects and the link of the current version to the original form of the object as well. The approach involves three basic steps Object registration, Verification and auditing process and Updating integrity information. A complete prototype called ACE (Auditing Control Environment) that implements the presented methods. Stein et. al. [2] presented the technique to ensure file integrity in archive that make usage of re-writeable media that makes the data more vulnerable to accidental or intentional changes, modifications or deletion. So a mechanism is presented to maintain the file integrity that could be integrated into standard PDS (Planetary Data System) procedures. In this mechanism file integrity begins with the data provider and continues through the life of the archive. Moreover, this mechanism uses digital signatures as mean of determining file integrity. The digital signature of each individual file is produced using the cryptographic hash function SHA-1 due to its free availability. These digital signatures are thus used to check the integrity of files. Zhang et.al.[3] presented a technique to provide the integrity protection mechanism on web pages by customized middleware that is embedded in web server. The middleware identifies the web document requested by user in time and recovers the modified one according to its importance, which is efficiency to prevent modified documents and trojan horse scripts from diffusing maliciously. Integrity in this approach is also ensured using the digital signatures that are computed using the hash algorithm. Adeel [4], present a new approach to develop a mathematical search engine for mathematical content retrieval. Math GO system is given in this work to search and present the mathematical information encoded in mathematical expression. The system consists of a modular architecture to organize, query, compare and present math results to the user. The issues that generally occur in math search engine are addressed in this approach. The performance of the search engine is also evaluated however; in this research the integrity of the database is not considered.

3. PRELIMINARIES:

3.1 Data Integrity:

Integrity deals with the concept of consistency of action, values, methods, measure, principles, expectation and outcomes. It is also taken as the quality of having a sense of honesty and truthfulness in regards to the motivation of one’s action.[5]

Data integrity refers to data that has complete or whole structure and that is uncorrupted and un-distorted. Data that has integrity is identically maintained during any operation such as transfer, storage and retrieval so data integrity is the assurance that data is consistent, certified and can be reconciled. [6]

3.2 Cryptographic Hash Algorithm:

A cryptographic hash function is a well-defined, deterministic procedure or mathematical functions that takes an arbitrary block of data and return a fixed size bit string called cryptography hash value. The data to be encoded is called “Message” and the hash value is sometimes called “Message Digest” or simply “Digest”. Any intentional or accidental change to data will change the hash value and so the change will easily be identified. This service simply ensures the integrity of data so hash algorithms are frequently used to verify the integrity of data. [7]

4. FRAMEWORK OVERVIEW:

In this research a new approach is presented to ensure and audit/verify the integrity of digital archives holding the large databases. The approach focuses on use of hash algorithm in combination with asymmetric encryption algorithm ‘RSA’ to ensure and audit the data holdings of archives in a secure way. The commonly used hash algorithms are also evaluated to select the best available hash algorithm for the data object of specific size. This makes the auditing process more fast and efficient. So the technique includes the two processes.

- Object registration into archive

- Integrity auditing of data object

4.1 Object Registration in archive:

Fig.1 elaborates the object registration process in digital archive and shows how the integrity information of a data object is maintained at the time of its registration into archive. Archivist inputs the data object and its hash value is computed using all the four Hash Algorithms SHA-1, MD-5, RIPEMD-160 and HAVAL-160. These hash values are stored in a temporary storage. The performance of each algorithm is evaluated in terms of processing time, memory resource usage for storage and efficiency of the algorithms and data size of input data object. The performance results of each algorithm are stored in different database. Archivist analyses these results and select the hash algorithms with best performance i.e. consuming the least amount of resources. The hash value computed by the selected hash algorithm is then obtained from temporary storage location and stored into archive along with the original object.

On the other hand, the LOG information of object is maintained including the Object ID, Object name, Object size, Selected Hash Algorithm ID and timestamp etc. This information is stored in a separate database. To secure the LOG info of object stored in LOG database, the LOG database is encrypted using the asymmetric encryption scheme RSA. The public key of archivist is used to encrypt the LOG database.

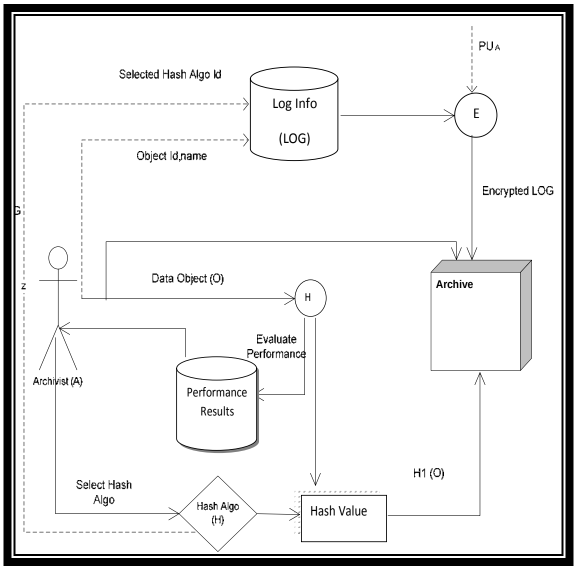

Figure 1: Object Registration in Archive

4.2 Integrity Auditing of Data Object:

The integrity auditing process of data object residing in archive is elaborated in figure 2.The auditor browses the data object from archive, the LOG information is searched by object name or id. The encrypted LOG info is then decrypted using the private key of auditor. The hash algorithms selected for that particular object is extracted from the LOG. The hash value of the browsed object is computed using that algorithm named H2 (O).The initially computed value H1 (O) is browsed from the archive. Both of these values are then compared and auditing result is displayed to user.

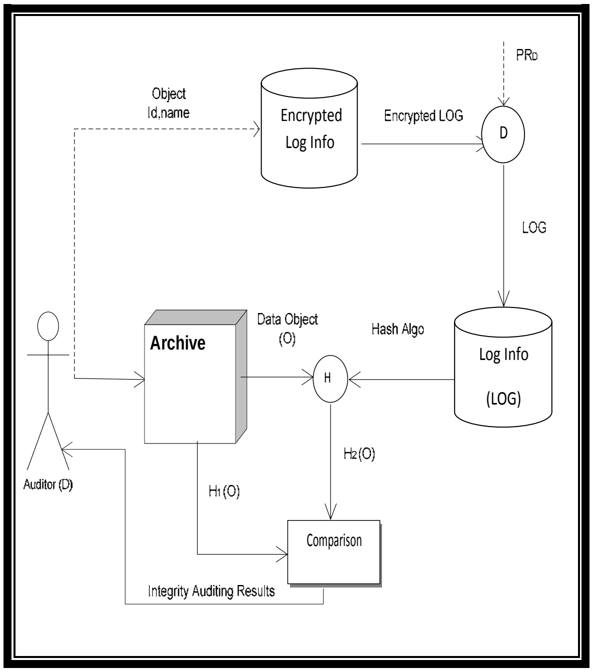

Figure 2: Integrity auditing of data object

This was the framework overview of the proposed approach; the technique in detail is given in the next section.

5. TECHNIQUE:

As the technique involves two phases, the object registration into archive and integrity auditing. In this section both of them are described in detail via their algorithms.

Following is the algorithm for the object registration into archive.

- *The Archivist specifies his unique username and password for the public key. (PUA).*

- *Archivist selects the data object to place it in archive.*

- *System computes the hash value of data object via each hash algorithm.*

- *The computed hash values are stored in a temporary location.*

- *The performance of each hash algorithm is stored in database.*

- *Archivist evaluates the performance and selects the best hash algorithm to store hash value.*

- *Hash value computed through the selected hash algorithm is picked from the temporary location.*

- *LOG information of object is stored in database.*

- *The LOG database is encrypted using public key of archivist.*

- *The original object, hash value H1 (O) and encrypted LOG information is stored in archive.*

Following is the algorithm for the integrity auditing phase.

- *The auditor provides its username and password that is verified by the system.*

- *Auditor selects the data object from archive.*

- *The LOG information of the object is searched either by its id or name.*

- *The LOG information is then decrypted by the Private Key of auditor (PRD)*

- *Hash algorithm selected for the object at time of archiving is retrieved from LOG.*

- *The hash value of data object browsed from archive is computed via this hash algorithm named as H2 (O)*

- *The initially computed hash value of object H1 (O) Is retrieved from the archive.*

- *H1 (O) and H2 (O) are compared to verify integrity. If both the values are same it shows that data object has not been changed and if not then the object has been altered.*

- *The auditing results are displayed to user.*

6. EXPERIMENTAL RESULTS:

The results obtained in this research work are based on the six data object of different size ranging from 3-12 MB. The performance of all four considered hash algorithms is evaluated fro each object in terms of time resource usage, memory resource usage and efficiency of algorithm. The results are conducted on the personal computer of 0.98GB RAM, Intel® Pentium IV @2.18GHz. Following are the results.

6.1 Response Time:

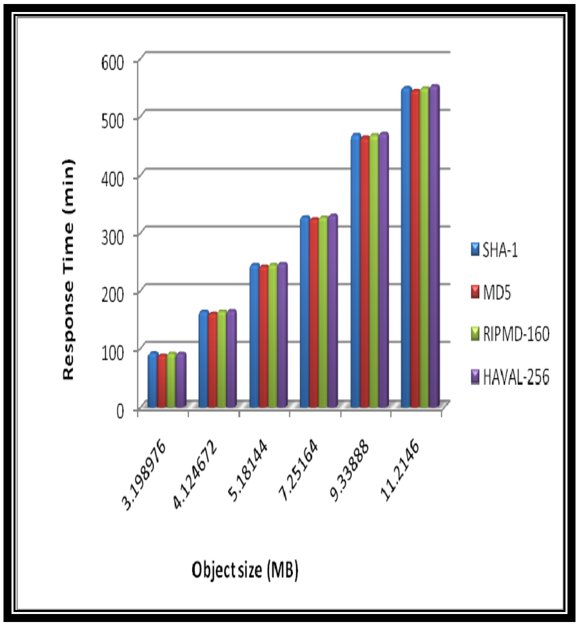

Figure 3: Response Time

The above results show that overall; MD-5 is taking the shorter response time as compared to the other hash algorithms. SHA-1 and HAVAL-256 are almost taking the same response time that is greater than MD-5 and RIPEMD-160 whereas, RIPEMD-160 takes the place in between.

6.2 Resource Usage:

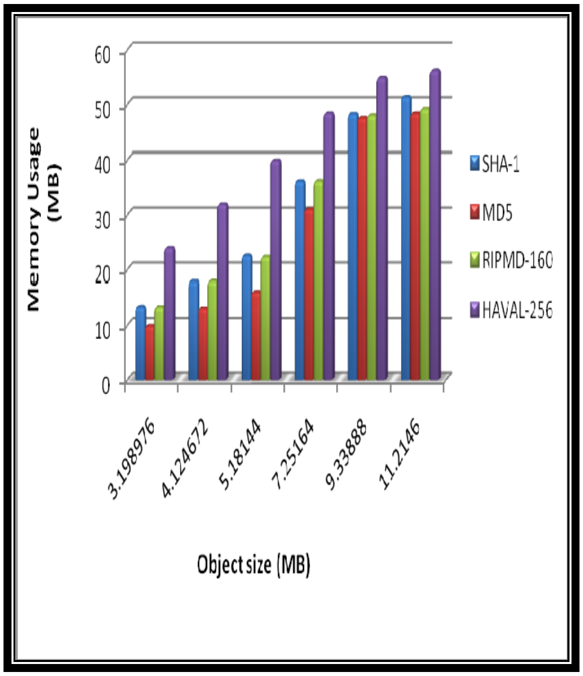

Figure 5 : Resource Usage

The above results show that MD-5 is consuming the minimum memory resource than the rest of three algorithms. Then comes the RIPEMD-160 and SHA-1 which almost consume the same memory resource for an object. Whereas HAVAl-256 consumes the large amount of memory resource as compared to the SHA-1, RIPEMD-160 and MD-5.

6.3 Efficiency of Hash Algorithms:

Efficiency is evaluated by observing the consistency in behavior of each hash algorithm with respect to response time and resource usage in relation to the size of the object. The more the algorithm is consistent in behavior, the more efficient it is. The following subsections present the two measures of hash algorithms efficiency.

6.3.1 Response Time to Original Size Ratio:

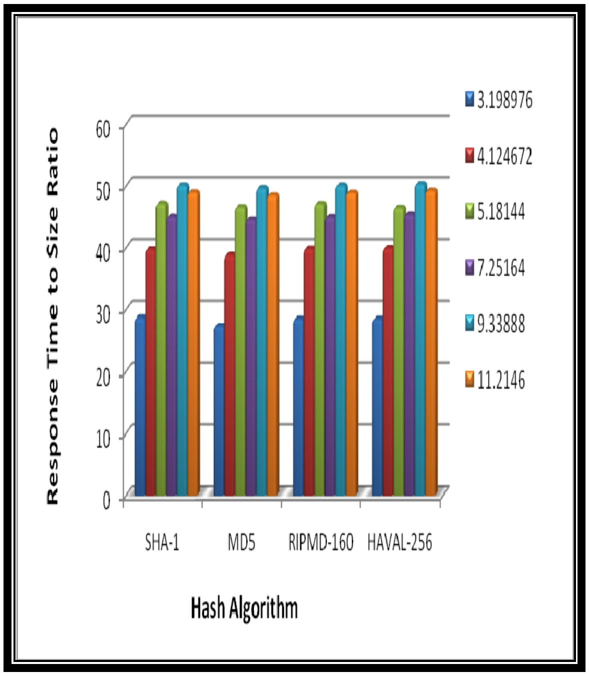

Figure 6: Response Time to Original Size Ratio

The above results show that almost all the algorithms are behaving efficiently with respect to the time resource usage. But RIPEMD-160 is exhibiting a bit more efficient behavior than the other three algorithms.

6.3.2 Resource Usage to Original Size Ratio:

The following graph is plotted to analyze the efficiency of hash algorithms with respect to their memory resource usage in dependence to the size of the object. Memory usage to original size ratio is taken for this purpose.

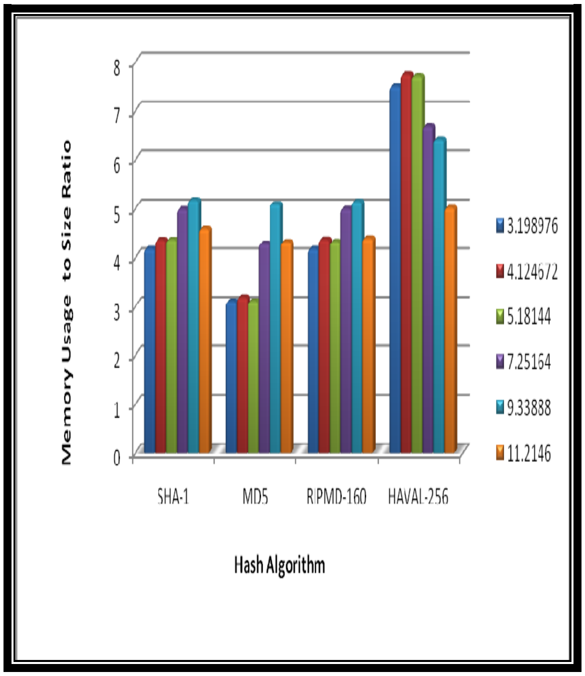

Figure 7: Resource Usage to Original Size Ratio

From the above result it is analyzed that RIPEMD-160 and SHA-1 are exhibiting more efficient and consistent behavior in memory resource usage with the change in the size of the data object.MD-5 and HAVAL-256 are not exhibiting that much efficient behavior.

7. CONCLUSION AND FUTURE WORK:

In this research work four well known and commonly used hash algorithms, SHA-1, MD-5, RIPEMD-160 and HAVAL-256 are analyzed for performance on the basis of time resource usage, memory resource usage and algorithm efficiency.

From the analysis of performance results, it is concluded that amongst the four selected hash algorithms SHA-1, MD-5, and RIPEMD-160 and HAVAL-256, MD-5 is consuming the minimum amount of resources and HAVAL-256 uses the large amount of resources where SHA-1 and RIPEMD-160 takes place in between. They almost use the same amount of resources, more than MD-5 but less than HAVAL-256.

The analyses of efficiency graphs shows that SHA-1 and RIPEMD-160 are more efficient and consistent in behavior .They behave more consistently with the change in the size of data object.MD-5 and HAVAL-256 are not that much efficient especially in case of memory resource usage. They exhibit larger change in behavior with change in size of data object.

In general, MD-5 is considered secure to an extent but there are security weakness identified in MD-5 [8][9].Same in the case of SHA-1 which minimizes its security[9]. Although both of the mentioned algorithms have security loop holes but still they are the most commonly used algorithms in this time. Organizations prefer to use them because they are not much complex. RIPEMMD-160 is making its place in the market because it makes a balance between security and complexity [9][10].HAVAl-256 is however considered more secured against the security attack on data but it is very much complex[9][11].

This conclusion made on the basis of results provides an efficient way to Archivist to choose the hash algorithm according to its available resource and security needs. If the organization can compromise on security but has limited resources than it must select MD-5 to ensure the data integrity. If security is the main concern of the organization then it must go with HAVAL-256 and if there is a possibility to find a middle way then RIPEMD-160 and SHA-1 can be a good choice.

This research was conducted on a limited number of hash algorithms. In future other algorithms available should also be analyzed for performance. In addition, the medium selected was mainly databases. For further work other media like images etc. should also be utilized.

REFERENCES:

[1] Sangchul Song, Joseph JaJa, "*New Techniques for Ensuring the Long Term Integrity of Digital Archives*", The Proceedings of the 8th Annual International Digital Government Research Conference,2007, pp 57-65

[2] Thomas C. Stein, Edward A. Guinness, Susan H. Slavney, "*Establishing a Mechanism for Maintaining File Integrity within Data Archives***"**, Ensuring Long-term Preservation and Adding Value to Scientific and Technical data (PV 2005)21- 23 November 2005,Royal Society, Edinburgh, UK

[3] Jianhua Zhang, Nan Zhang, Xianze Yang, Chunyan Yan, "*Security Mechanism to Protect the Integrity of Web Documents*," *Management of e-Commerce and e-Government, International Conference on*, pp. 395-398, 2008 International Conference on Management of e-Commerce and e-Government, October 17-19 2008, Nanchang, China

[4] Muhammad Adeel, Hui Siu Cheung, Sikandar Hayat Khiyal **"***Math Go! Prototype of A Content Based Mathematical Formula Search Engine***"**, Journal of Theoretical and Applied Information Technology, Vol 4 No 10, pp-1002-1012, October 2008.

[5] John Louis Lucaites, Celeste Michelle Condit, Sally Caudill.*Contemporary rhetorical theory: a reader*. Guilford Press, 1999, pp. 92. ISBN 1572304014.

[6] Beynon-Davies. *Database Systems* 3rd Edition. Palgrave, Basingstoke, UK.2004,ISBN 1-4039-1601-2.

[7] http://www.searchsecurity.com

[8] http://www.net-seurity.org

[9] http://www.wikipedia.org

[10] http://www.kremlinencrypt.com

[11] http://www.search.cpan.org