INT301: Open-Source Technologies

PROJECT REPORT

(Project Semester January-May 2023)

To Extract the Web Artifacts from Chrome, Edge and Firefox

Submitted by

Ishaan Morani

Registration No: - 11911882

Section: - **KE058**

Course Code: - **INT301**

Under the Guidance of

Navjot Kaur: 20506

Assistant Professor.

The domain of CSE/IT

Lovely School of Computer Science and Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Mr. Morani Ishaan M. bearing Registration no 11911882 has completed the INT301: Open-Source Technologies project "To Calculate MD5 and SHA1 and CRC32 (10 files) for making a comparison of files and check all the files for integrity, internet downloads and data tampering.

Signature and Name of the Supervisor
Designation of the Supervisor
School of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab.

Date: 30/03/2023

DECLARATION

I, **Mr. Morani Ishaan M.**, student of **Open-Source Technologies (INT301)** under CSE/IT Discipline at, Lovely Professional University, Punjab, at this moment declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 30/03/2023 Signature: Ishaan (11911882)

Registration No. 11911882
Name of the student: Mr. Morani Ishaan M

ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to all those who gave me moral support and help to complete this report. Special thanks to my supervisor Mrs. Navjot Kaur whose help, stimulating suggestions, and encouragement always helped me during the Completion process and in writing this report. I also sincerely thank you for the time spent proofreading and correcting my many mistakes.

I would also like to acknowledge with much appreciation the crucial role of the students from my class, who gave me motivation and Guidance while completing my project. Many thanks go to all lecturers and supervisors who have given their full effort in guiding me in achieving the goal as well as their encouragement to maintain our progress track. My profound thanks go to all my classmates, especially to my friends for spending their time helping and giving support whenever I need it in my project.

TABLE OF CONTENT(INDEX)

1. Introduction

GitHub Link:

In the era of digital information, file integrity is of paramount importance. The internet has provided an easy way to share files with others, but it also presents a risk of data tampering and corruption during file transfer. As such, it is crucial to ensure the integrity of files downloaded from the internet or transferred from one device to another.

The term SHA stands for Secure Hash Algorithm , MD5 stands for Message digest Method 5 & and CRC stands for One popular method to check file integrity is to use checksum algorithms such as MD5, SHA1, and CRC32 stands for cyclic redundancy checksum. These algorithms generate unique checksums or hash values for each file, which can be compared to verify if the files have been modified or corrupted in any way.

In this project, we will calculate MD5, SHA1, and CRC32 checksums for a set of ten files and compare them to ensure their integrity. This project aims to demonstrate the usefulness of checksum algorithms in verifying file integrity, detecting data tampering, and ensuring secure data transfer over the internet.

1.1. Objective of the Project: -

The main objective of this project is to demonstrate the effectiveness of checksum algorithms in verifying the integrity of files downloaded from the internet or transferred between devices. Specifically, the project aims to:

- Calculate the MD5, SHA1, and CRC32 checksums for a set of ten files.
- Compare the generated checksums to verify the integrity of each file and detect any data tampering or corruption.
- Demonstrate the usefulness of checksum algorithms in detecting file tampering or corruption during internet downloads or data transfer.
- Provide a practical example of how to use checksum algorithms to ensure secure file transfer and maintain file integrity.

Overall, the project aims to provide a deeper understanding of the importance of file integrity and the role of checksum algorithms in ensuring secure data transfer over the internet.

1.2. Description of the Project.

The goal of this project is to showcase the use of checksum algorithms, specifically MD5, SHA1, and CRC32, in verifying the integrity of files downloaded from the internet or transferred between devices. The project involves generating checksums for ten different files and comparing them to ensure that the files have not been tampered with during internet downloads or data transfer.

To achieve this goal, we will first select ten files of various types and sizes, including documents, images, videos, and audio files. Next, we will generate MD5, SHA1, and CRC32 checksums for each of these files using software tools or programming languages that support these algorithms.

After generating the checksums, we will compare them to ensure that they match the original checksums of the files. If the checksums do not match, this indicates that the file has been tampered with, and its integrity has been compromised. In such cases, we will investigate further to identify the source of the tampering.

We will also demonstrate the use of checksum algorithms in detecting data corruption during internet downloads or data transfer. To do this, we will intentionally corrupt one or two of the files and compare their checksums with the original checksums. We will then show how checksum algorithms can help detect data corruption and ensure that the file is not used in its corrupted state.

1.3. Scope of the Project.

The scope of this project is to calculate MD5, SHA1, and CRC32 checksums for a set of ten files, including documents, images, videos, and audio files, and to compare them to ensure the files' integrity. The project aims to demonstrate the usefulness of checksum algorithms in detecting data tampering or corruption during internet downloads or data transfer.

This project does not include any modifications to the files or their contents, except for the intentional corruption of one or two files to demonstrate the use of checksum algorithms in detecting data corruption. The project also does not cover any specific software tools or programming languages to calculate the checksums, as the project is intended to be adaptable to a variety of tools and languages.

Overall, the project's scope is limited to generating checksums for a set of ten files, comparing them to ensure file integrity, and demonstrating the usefulness of checksum algorithms in detecting data tampering or corruption during internet downloads or data transfer.

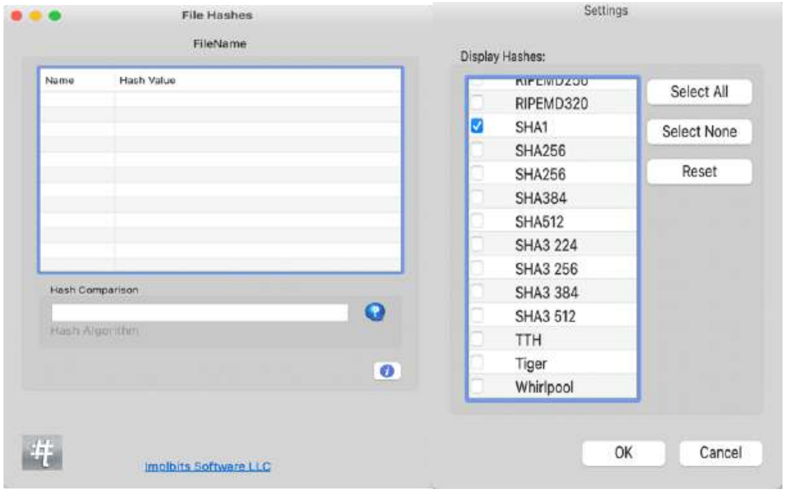
2. System Description.

2.1. Target System Description.

Any device or platform that supports the generation of MD5, SHA1, and CRC32 checksums. This includes desktop and laptop computers running Windows, macOS, or Linux operating systems, as well as mobile devices running Android or iOS.

The project does not require any specific hardware requirements beyond the basic specifications required to run the device or platform. The project does require software tools like HashTab, HashCheck, QuickHash GUI, OpenSSL, and md5Sum command line argument There are several other software options available for generating checksums.

Currently I'm Using HashTab on macOS to calculate MD5, SHA1 and CRC32 , This is an open source software one can download from this website https://implbits.com/hashtab , After downloading and installing , one window of hash tab will open in that you can open and check any file's MD5,SHA1, & CRC32 . You can count certain other values like whirlpool, tiger , TTH , various types of SHA and few others too.



Display Hashes:

□ RIPEMD256
□ RIPEMD320
☑ SHA1
□ SHA256
□ SHA256
□ SHA384
□ SHA512
□ SHA3 224
□ SHA3 256
□ SHA3 384
□ SHA3 512
□ TTH
□ Tiger
□ Whirlpool

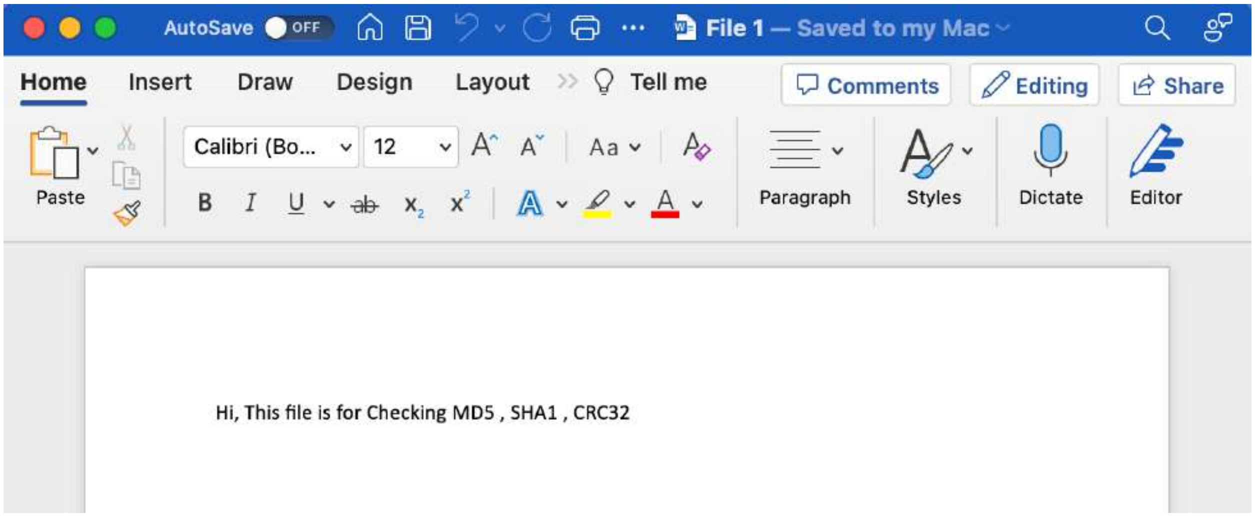
Fig. 1.

3. Analysis Report

3.1. System Snapshots and full Analysis Report.

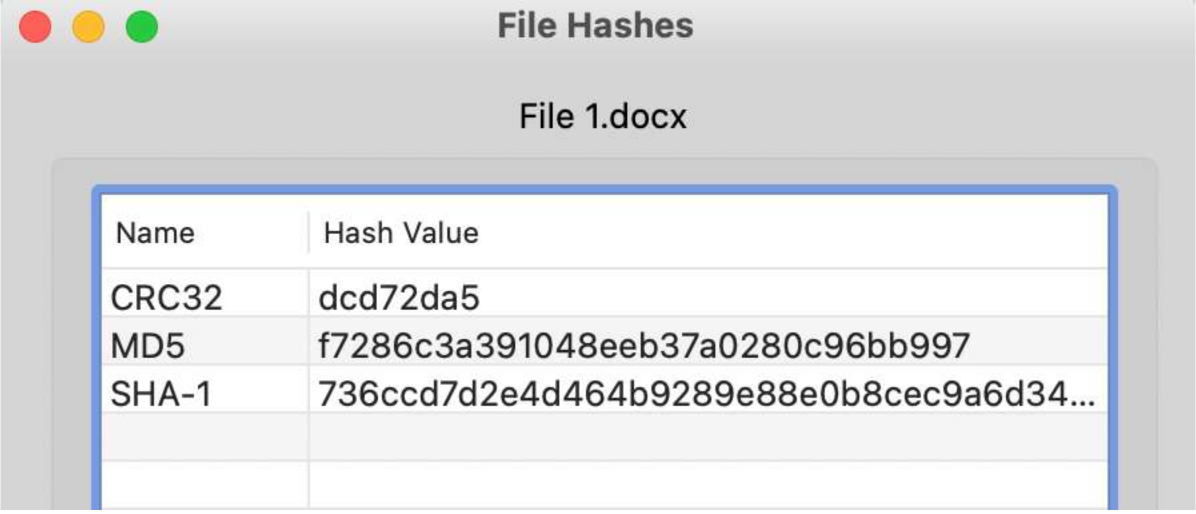
The first task is to Calculate MD5, CRC32, & SHA1 of 10 files.

File 1: So first, I decided to make a word file, added one line and saved it and then calculated MD5, CRC32, SHA1.



Hi, This file is for Checking MD5 , SHA1 , CRC32

Here, you can see the values of CRC32, MD5, SHA-1



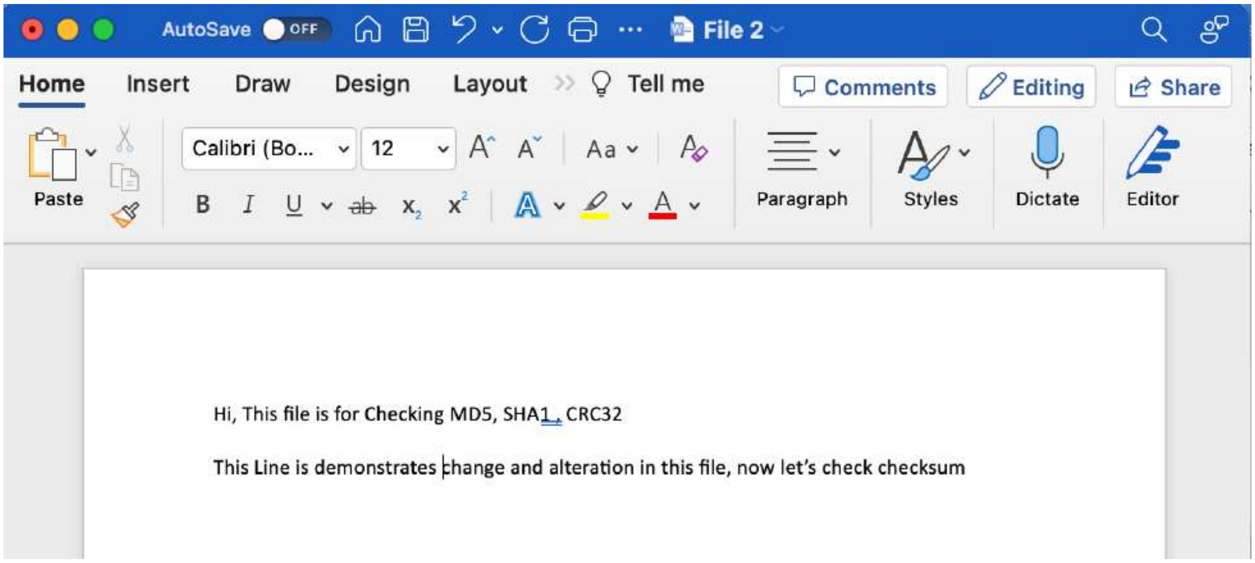
File Hashes

File 1.docx

Name	Hash Value
CRC32	dcd72da5
MD5	f7286c3a391048eeb37a0280c96bb997
SHA-1	736ccd7d2e4d464b9289e88e0b8cec9a6d34...

File 2:

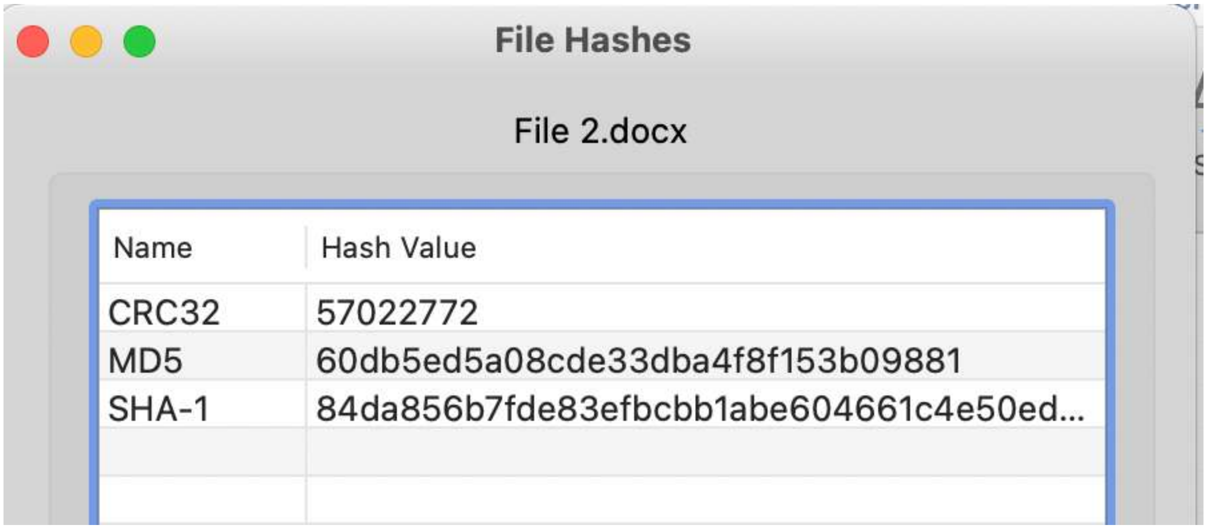
After that, I have made changes to previous file to see if alteration changes the MD5 values or not, I have added one more line as part of change.



Hi, This file is for Checking MD5, SHA1, CRC32

This Line is demonstrates change and alteration in this file, now let's check checksum

So, Adding this second line changes the values of MD5, CRC32 , SHA1

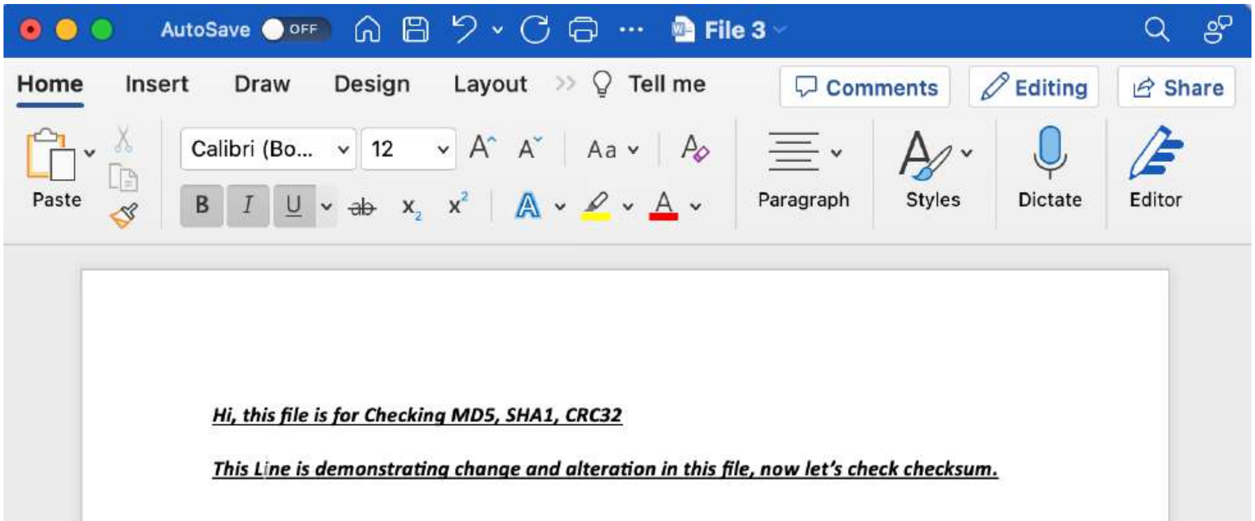


File 2.docx

Name	Hash Value
CRC32	57022772
MD5	60db5ed5a08cde33dba4f8f153b09881
SHA-1	84da856b7fde83efbcbb1abe604661c4e50ed...

File 3:

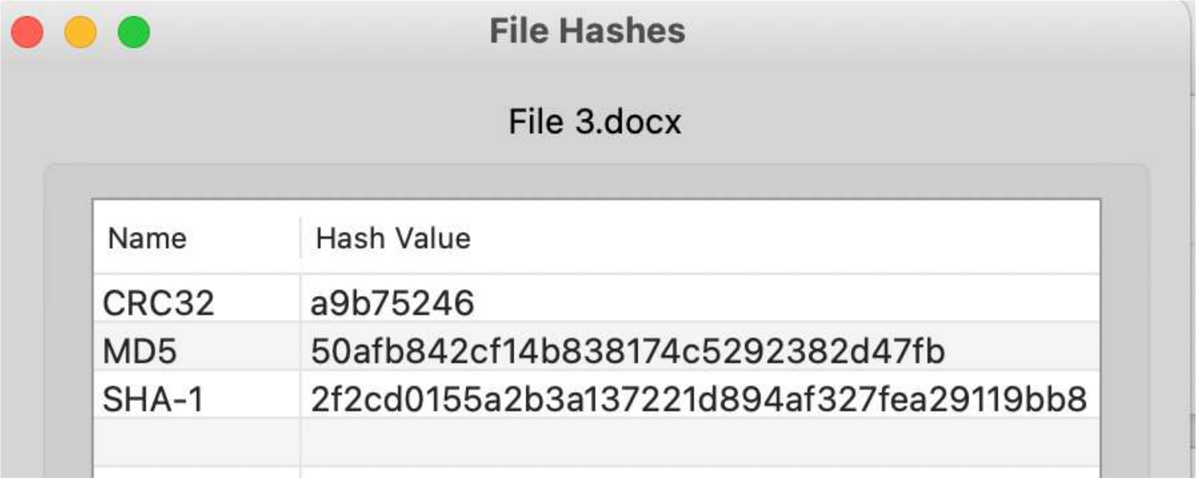
After adding a line, I decided to see if changing font or making them bold or italic changes checksum or not, but it did.



Hi, this file is for Checking MD5, SHA1, CRC32

This Line is demonstrating change and alteration in this file, now let’s check checksum.

Here, We can see the same file but with different checksum because of bold and italic.

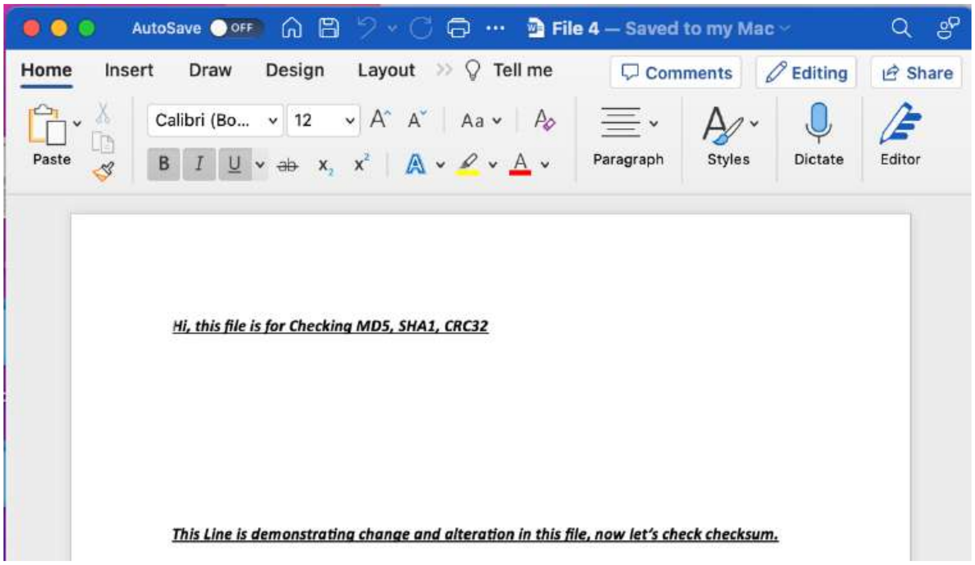

File Hashes

File 3.docx

Name	Hash Value
CRC32	a9b75246
MD5	50afb842cf14b838174c5292382d47fb
SHA-1	2f2cd0155a2b3a137221d894af327fea29119bb8

File 4:

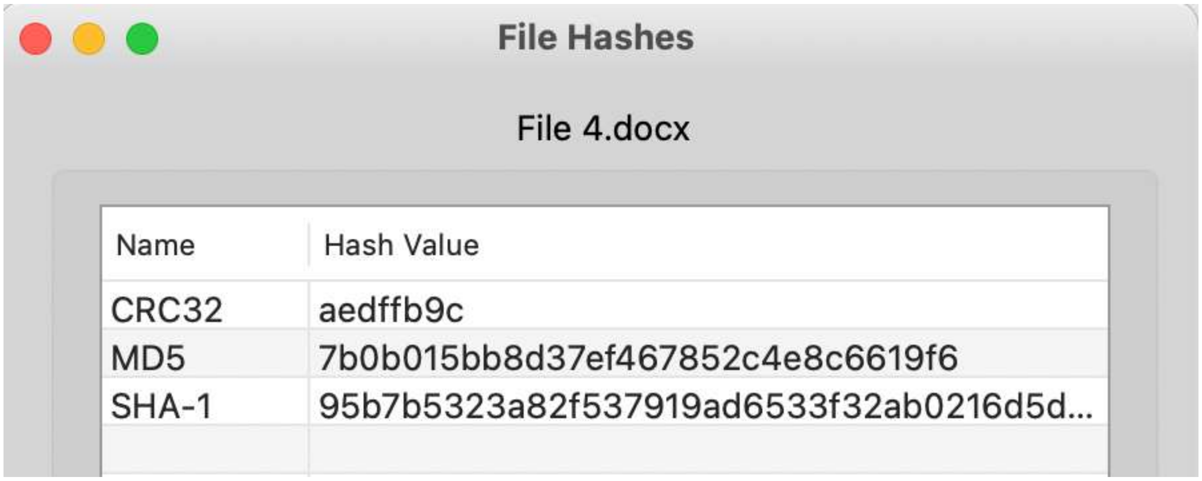
Even Adding Spaces, can alter the checksums of same file



Hi, this file is for Checking MD5, SHA1, CRC32

This Line is demonstrating change and alteration in this file, now let's check checksum.

Here, we can see the different MD5 , SHA1, and CRC32


File Hashes

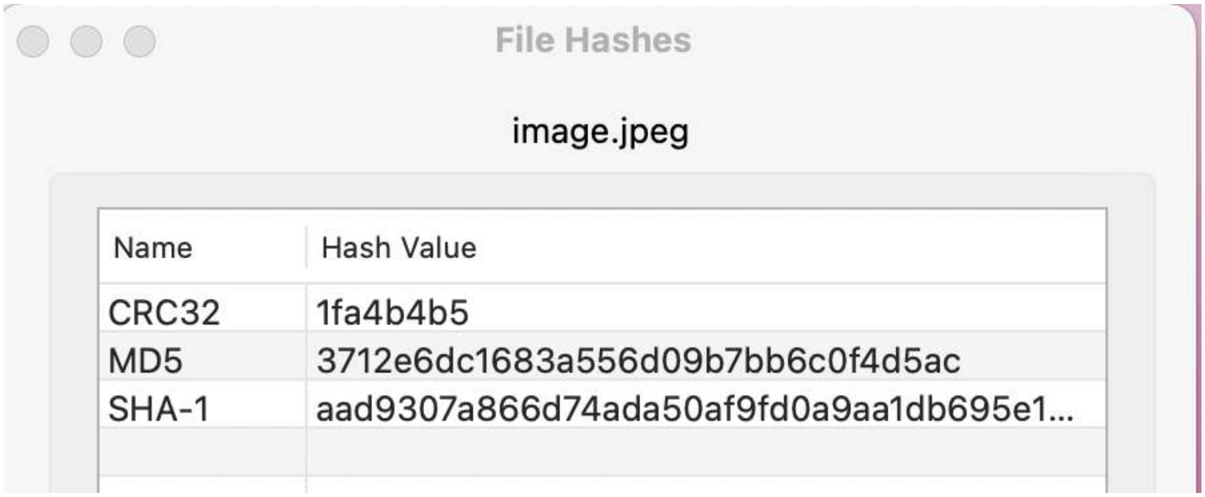
File 4.docx

Name	Hash Value
CRC32	aedffb9c
MD5	7b0b015bb8d37ef467852c4e8c6619f6
SHA-1	95b7b5323a82f537919ad6533f32ab0216d5d...

File 5:

I downloaded, image from safari (web browser) on 30th March and calculated the MD5, CRC32 and SHA 1

Here, are the value of CRC32, MD5 , and SHA-1



File Hashes

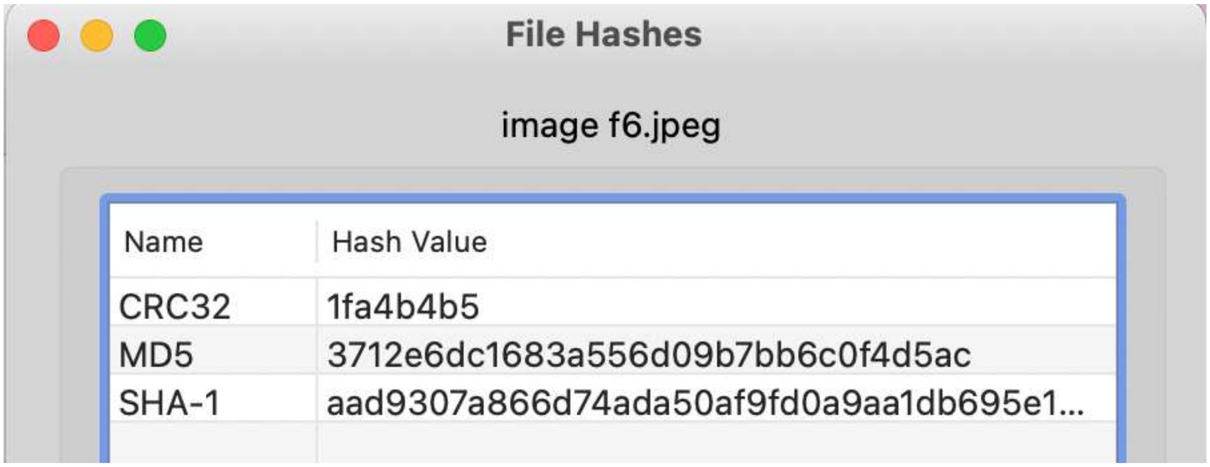
image.jpeg

Name	Hash Value
CRC32	1fa4b4b5
MD5	3712e6dc1683a556d09b7bb6c0f4d5ac
SHA-1	aad9307a866d74ada50af9fd0a9aa1db695e1...

File 6:

After few days, on 5th April I downloaded same image from Google Chrome in hope that it will give me Different Checksum as time has passed and so browser is different, but results were surprising.

Here, are the values which are exactly same from past image's checksum values!!



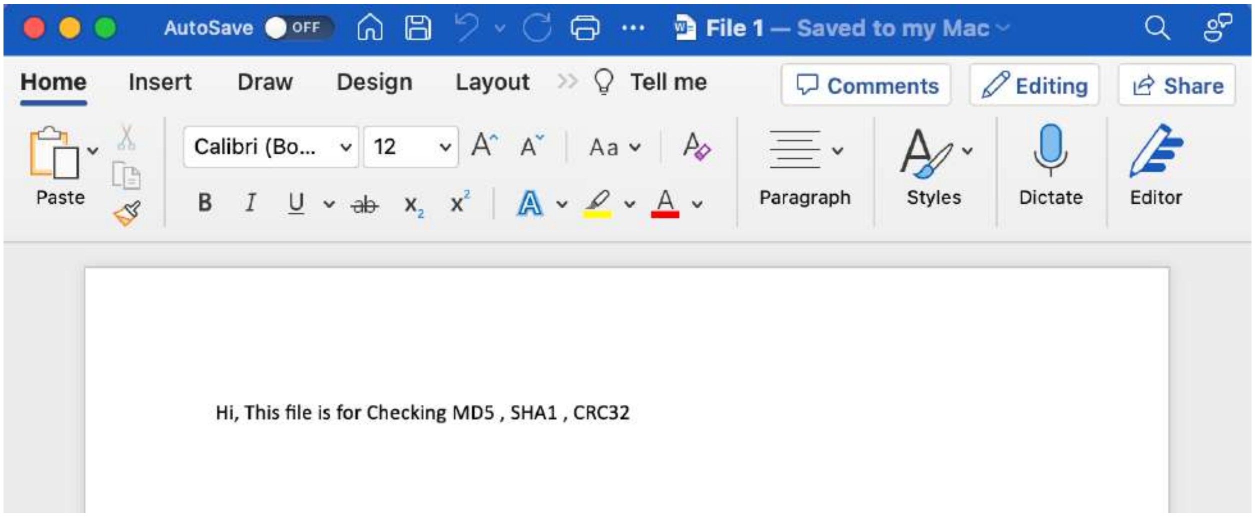
File Hashes

image f6.jpeg

Name	Hash Value
CRC32	1fa4b4b5
MD5	3712e6dc1683a556d09b7bb6c0f4d5ac
SHA-1	aad9307a866d74ada50af9fd0a9aa1db695e1...

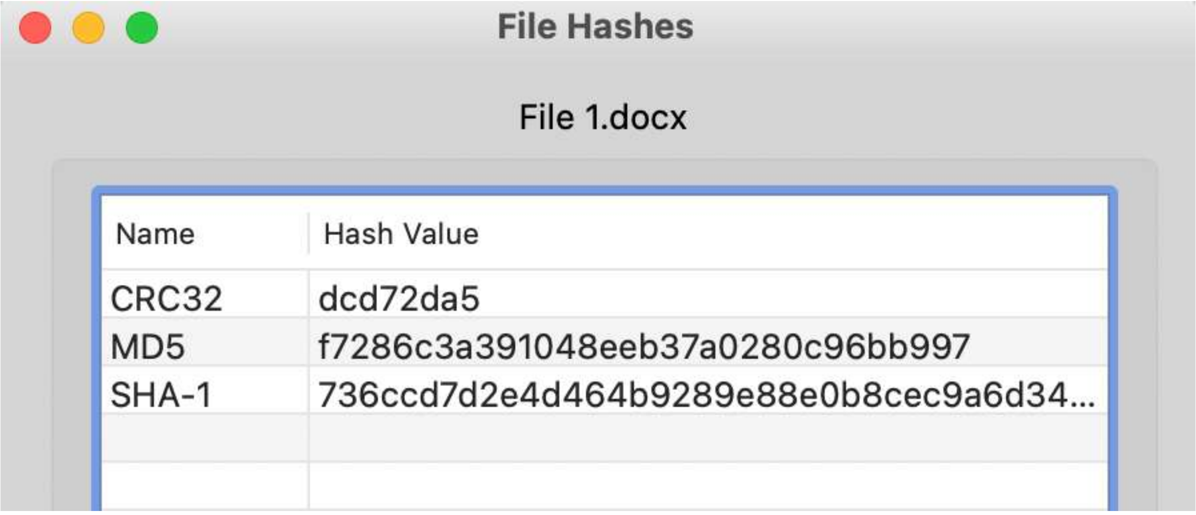
File 7:

Comparing file 1 again after 14 days to see if there is change is checksum or not!!



Hi, This file is for Checking MD5 , SHA1 , CRC32

This is the same file which I created first time , I again checked it to see if time variable can change the MD5 , CRC32 and SHA1 values but to my surprise they were same.

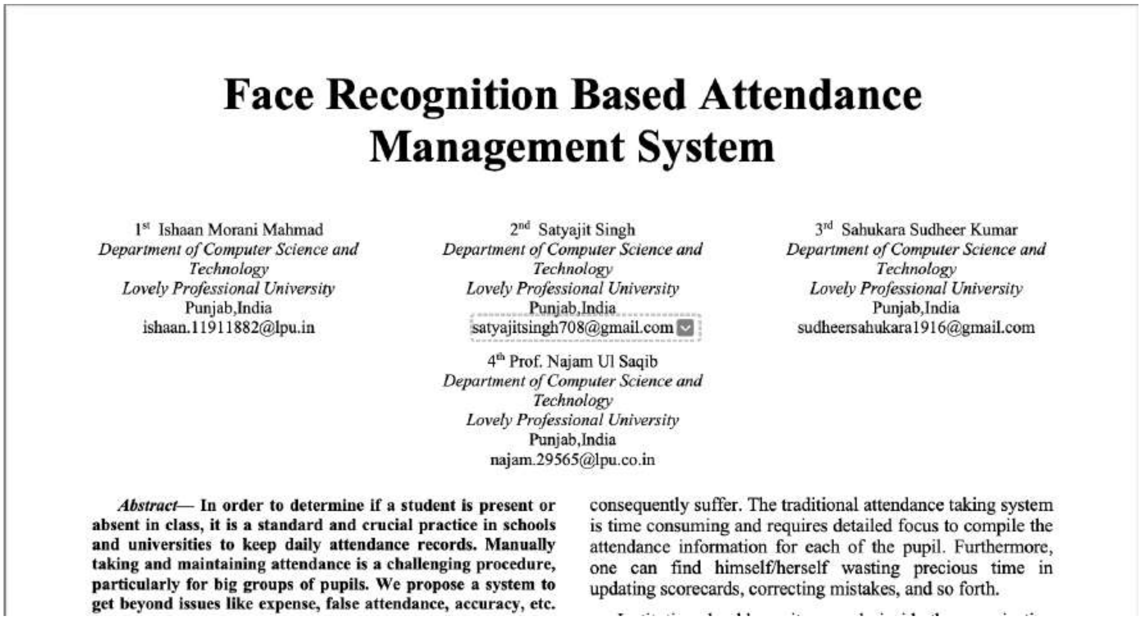


File 1.docx

Name	Hash Value
CRC32	dcd72da5
MD5	f7286c3a391048eeb37a0280c96bb997
SHA-1	736ccd7d2e4d464b9289e88e0b8cec9a6d34...

File 8:

I decided, to calculate MD5, CRC32, and SHA1 of pdf of Capstone paper !!



Face Recognition Based Attendance Management System

1st Ishaan Morani Mahmad
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
ishaan.11911882@lpu.in

2nd Satyajit Singh
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
satyajitsingh708@gmail.com

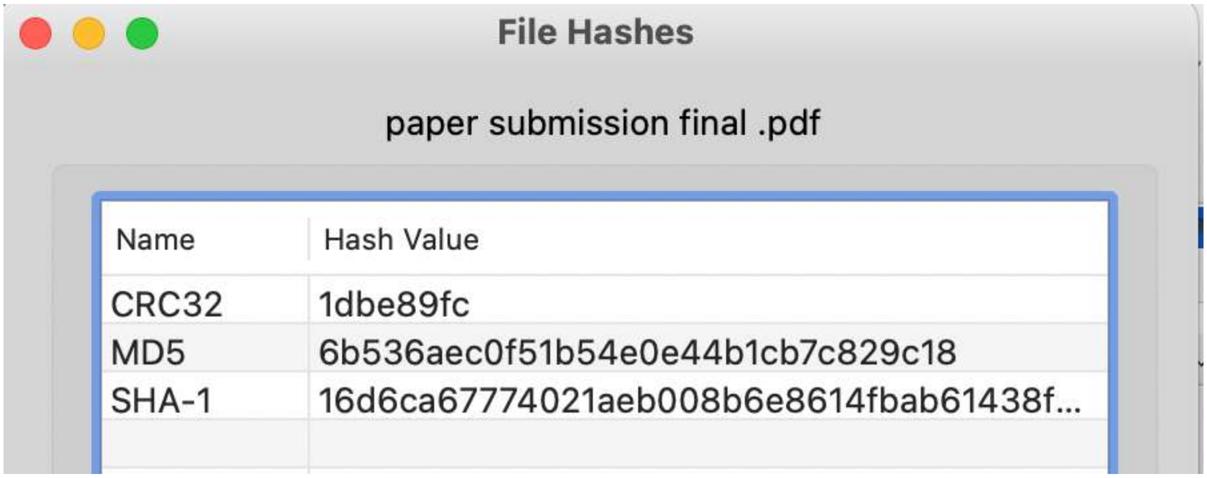
3rd Sahukara Sudheer Kumar
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
sudheersahukara1916@gmail.com

4th Prof. Najam Ul Saqib
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
najam.29565@lpu.co.in

***Abstract*— In order to determine if a student is present or absent in class, it is a standard and crucial practice in schools and universities to keep daily attendance records. Manually taking and maintaining attendance is a challenging procedure, particularly for big groups of pupils. We propose a system to get beyond issues like expense, false attendance, accuracy, etc.**

consequently suffer. The traditional attendance taking system is time consuming and requires detailed focus to compile the attendance information for each of the pupil. Furthermore, one can find himself/herself wasting precious time in updating scorecards, correcting mistakes, and so forth.

Here, are the values of the given pdf



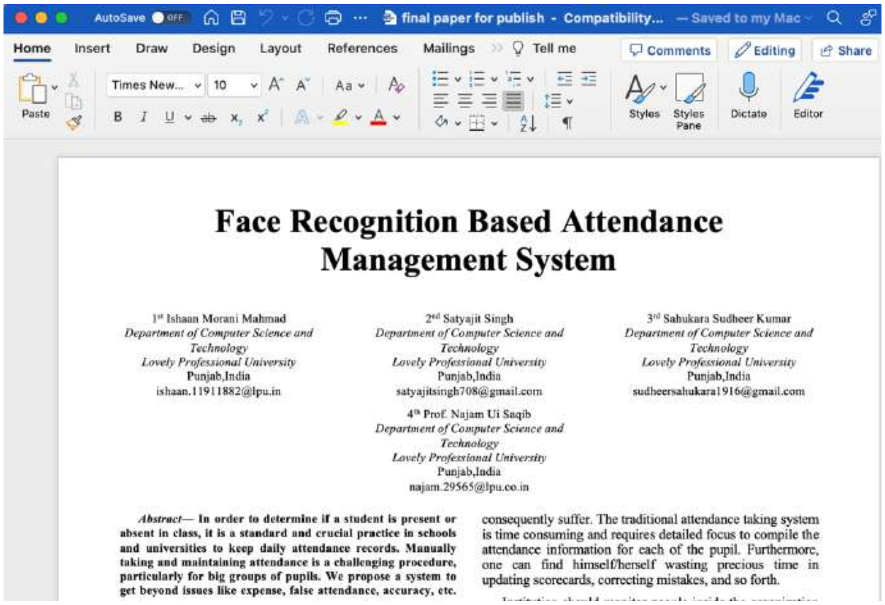
File Hashes

paper submission final .pdf

Name	Hash Value
CRC32	1dbe89fc
MD5	6b536aec0f51b54e0e44b1cb7c829c18
SHA-1	16d6ca67774021aeb008b6e8614fbab61438f...

File 9:

Same Content, but altering file type can also change the checksum values .



Face Recognition Based Attendance Management System

1st Ishaan Morani Mahmad
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
ishaan.11911882@lpu.in

2nd Satyajit Singh
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
satyajitsingh708@gmail.com

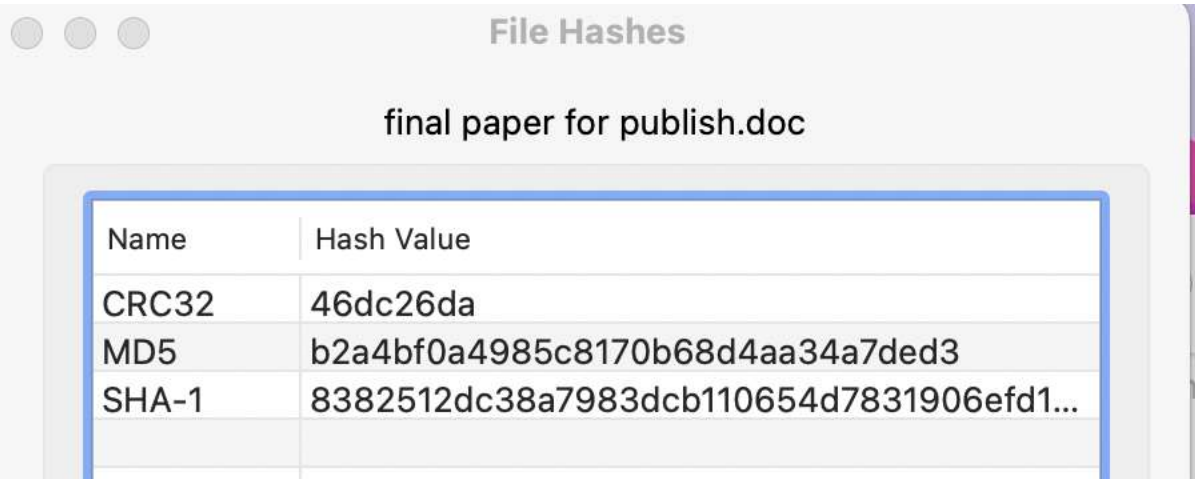
3rd Sahukara Sudheer Kumar
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
sudheersahukara1916@gmail.com

4th Prof. Najam Ui Saqib
Department of Computer Science and Technology
Lovely Professional University
Punjab,India
najam.29565@lpu.co.in

***Abstract*— In order to determine if a student is present or absent in class, it is a standard and crucial practice in schools and universities to keep daily attendance records. Manually taking and maintaining attendance is a challenging procedure, particularly for big groups of pupils. We propose a system to get beyond issues like expense, false attendance, accuracy, etc.**

consequently suffer. The traditional attendance taking system is time consuming and requires detailed focus to compile the attendance information for each of the pupil. Furthermore, one can find himself/herself wasting precious time in updating scorecards, correcting mistakes, and so forth.

Here are the

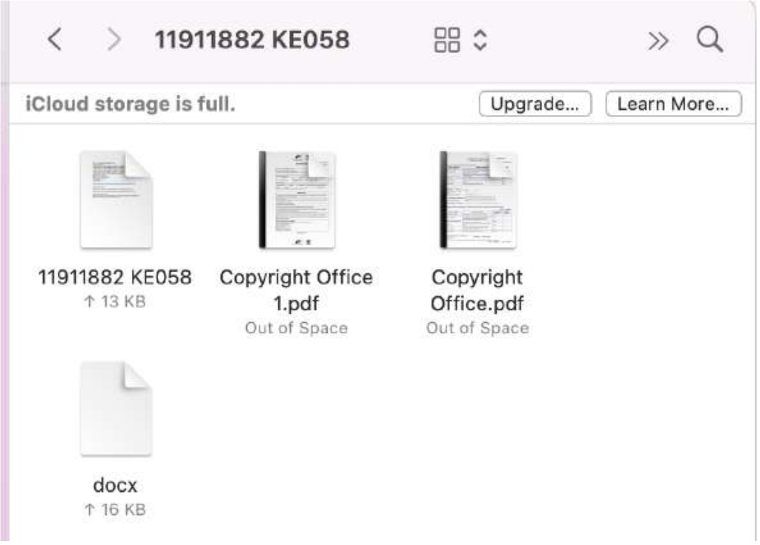


final paper for publish.doc

Name	Hash Value
CRC32	46dc26da
MD5	b2a4bf0a4985c8170b68d4aa34a7ded3
SHA-1	8382512dc38a7983dcb110654d7831906efd1...

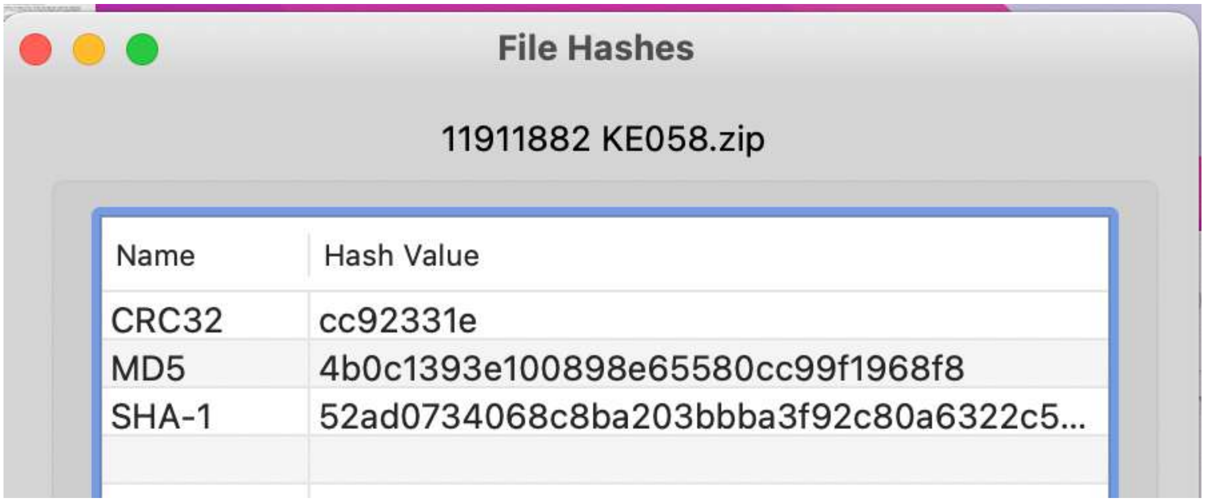
File 10:

All this files, I merged into folder and then made a zip file and calculated checksum values



iCloud storage is full.

Here are they:



11911882 KE058.zip

Name	Hash Value
CRC32	cc92331e
MD5	4b0c1393e100898e65580cc99f1968f8
SHA-1	52ad0734068c8ba203bbba3f92c80a6322c5...

4. Conclusion:

In conclusion, calculating the MD5, SHA1, and CRC32 checksums for files is a useful way to compare them and verify their integrity, especially for internet downloads and data tampering detection. By comparing the checksums of two files, it is possible to determine if they are identical or if one of them has been tampered with. Additionally, by comparing the checksum of a downloaded file with the provided checksum by the author or website, it is possible to ensure that the file has not been corrupted during the download process. It is highly recommended to calculate the checksums for each file, especially when dealing with sensitive data or software, to prevent any potential data tampering or loss of data integrity.

5. References/ Bibliography

https://www.autodesk.com/support/technical/article/caas/sfdcarticles/sfdcarticles/Checking-the-MD5-checksum-of-a-Downloaded-File.html

https://youtu.be/Wgehyj0CEdY

https://youtu.be/YM2CE6zKvoo

6. GitHub Link:

https://github.com/ishaanmorani26/int_301

THANKYOU