Digital Forensics - Lab 1

Class No :	**CH2021221000516**	Slot :	**L49 + L50**
Course Code :	**CSE4004**	Faculty Name :	[redacted]

Aadhitya Swarnesh

[redacted]

- 5 August 2021

Question 1 :

Compute the MD5, SHA-1, SHA-256 hash values of the two strings given below

“The quick brown fox jumps over the lazy dog”

“The quick brown fox jumps over the lazy dogs”

Let us use the tool provided in the link to perform the hashing procedure, in order to do this we open the portal and enter the two lines of text, and we obtain their HASH values through various having algorithms. I have shown the hash values of both the provided sentences :

Sentence 1 : “The quick brown fox jumps over the lazy dog”

Hashing Algorithm	Hashing Algorithm
MD5	9e107d9d372bb6826bd81d3542a419d6
SHA-1	2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
SHA-256	d7a8fbb307d7809469ca9abcb0082e4f8d5651e46 d3cdb762d02d0bf37c9e592

Sentence 2 : "The quick brown fox jumps over the lazy dogs"

Hashing Algorithm	Hashing Algorithm
MD5	3ee6f92b7cddc3f50b7d2ddd145b018b
SHA-1	f8c3c541257a6c31f6fbc697a50f46d9fc8bcc30
SHA-256	1be9a63751d3af7ffa65b21ccc58d2b89eda7011d7fee2bb9229a74085f8eb2e

We can notice from these hash values that a small change in the plain text can **change the HASH value significantly**, which is a property of hashing algorithms in order for them to be effective in real world.

The below diagrams show the HASH values by a few other algorithms for these sentences.

Results	
Original text	The quick brown fox jumps over the lazy dog
Original bytes	54686520717569636b2062726f776e20666f78206a756d7073... (length=43)
Adler32	5bdc0fda
CRC32	414fa339
Haval	713502673d67e5fa557629a71d331945
MD2	03d85a0d629d2c442e987525319fc471
MD4	1bee69a46ba811185c194762abaeae90
MD5	9e107d9d372bb6826bd81d3542a419d6
RipeMD128	3fa9b57f053c053fbe2735b2380db596
RipeMD160	37f332f68db77bd9d7edd4969571ad671cf9dd3b
SHA-1	2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
SHA-256	d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3cdb762d02d0bf37c9e592
SHA-384	ca737f1014a48f4c0b6dd43cb177b0afd9e5169367544c494011e3317dbf9a509cb1e5dc1e85a941bbee3d7f2afbc9b1
SHA-512	07e547d9586f6a73f73fbac0435ed76951218fb7d0c8d788a309d785436bbb642e93a252a954f23912547d1e8a3b5ed6e1bfd7097821233fa0538f3db854fee6
Tiger	6d12a41e72e644f017b6f0e2f7b44c6285f06dd5d2c5b075
Whirlpool-0	4f8f5cb531e3d49a61cf417cd133792ccfa501fd8da53ee368fed20e5fe0248c3a0b64f98a6533cee1da614c3a8ddec791ff05fee6d971d57c1348320f4eb42d null
Whirlpool-T	3ccf8252d8bbb258460d9aa999c06ee38e67cb546cffcf48e91f700f6fc7c183ac8cc3d3096dd30a35b01f4620a1e3a20d79cd5168544d9e1b7cdf49970e87f1
Whirlpool	b97de512e91e3828b40d2b0fdce9ceb3c4a71f9bea8d88e75c4fa854df36725fd2b52eb6544edcacd6f8beddfea403cb55ae31f03ad62a5ef54e42ee82c3fb35

Results	
Original text	The quick brown fox jumps over the lazy dogs
Original bytes	54686520717569636b2062726f776e20666f78206a756d7073... (length=44)
Adler32	6c29104d
CRC32	444a08a0
Haval	90f0ea27177be192c52b5387f3523d54
MD2	ae742161d556aaa73ad11e4476c06cf2
MD4	6839dd600c6d4c84f8be3932723b97ad
MD5	3ee6f92b7cddc3f50b7d2ddd145b018b
RipeMD128	bea078ae5684e4b9e28514020ea9d699
RipeMD160	af7a2e207b3f7a664363aab46d724e354929d992
SHA-1	f8c3c541257a6c31f6fbc697a50f46d9fc8bcc30
SHA-256	1be9a63751d3af7ffa65b21ccc58d2b89eda7011d7fee2bb9229a74085f8eb2e
SHA-384	42f0aff2ecf3a112a05447ce4cd8b1b84f82b2217272a6746436d48759a1fc479d457ec297057c2dcbb60a0d8f40fbbf
SHA-512	0d52e77f1b76539a224c47af2326b32f5226add715dd7d08b31f1a9c37484f708c9a3ba26fdd2e6857ea43df641553b4fc941b2617b635ecea4003e290ee2236
Tiger	aecc43841d11ab953319aa618cad9e476569bac3b6fe53d8
Whirlpool-0	34071469b0319eb3e8341e7110f17bb5bfc12fcbe6f19ad1c0a358165e54ccd4c2df7bc36fb00e8f3452fc4a809b94db6580e2395d8838664f1bb430f83f2bb5 null
Whirlpool-T	54e414b570fc650266bfde731392225692452ef85ad37cb77bb9040a2b0dd5fa40bde48d4dcffa349424cbb1f346859fe91f7d353b3c21af5335f6cc97f99852
Whirlpool	fce26948c23fefe70c8a94327b081c1cd2c4a33485a8e9bbf0e420a4a2b92a89c80de89b3a53525990c0794e46d88fb78cb50d51f8144b60b5401142d3978eb0

Question 2 :

Perform hash calculations for any TWO files of your choice using the following hash functions: Adler32, CRC32, Haval, MD2, MD4, MD5, RipeMD-128, RipeMD-160, SHA-1, SHA-256, SHA-384, SHA-512, Tiger, and Whirlpool.

For this experiment, we will hash two files using multiple hashing algorithms and see if for files of any size, the hash values are within a limited size.

The First file used is a python code file just of a few lines as shown below :

update_readme.py

```
"""
This is just a test file.
Functionalities are in the Requirement Analysis stage 😁 .
So they will be added here once ready.
"""

import datetime

with open("test.md", "a") as file :
    file.write(str(datetime.datetime.now()))
```

Let us see the hash values of this file :

Results	
Original text	*(binary only)*
Original bytes	2222220a54686973206973206a7573742061207465737420666... (length=233)
Adler32	98964e8b
CRC32	77e63a29
Haval	55831263dd507dc2a8ac5398468f2338
MD2	a7f2f596ff0d04d53060cd062dd99238
MD4	8dd4d34840e888500fec41dd31807fc2
MD5	cf71a58443b47179ae9f7cb337f75671
RipeMD128	d63fbc19784af6a4ace95a8002ac99a7
RipeMD160	b732eee20d061366e8ece52577f0d44681b54817
SHA-1	17493e22bb90a37820e2ee704a9498f9ec665dd8
SHA-256	5abc56dbf85fa2d401e5360452058c7b9f32532ef8add49c59d52a5e888e6533
SHA-384	c852799175a3b259b1b6c6fc2e5e75a2db79dc347b78abc75e521ca07208308717c4a8e31a90ff23e8b162de4341ba83
SHA-512	b46c7e619187f6c5958c6e655e99bb28a116d42eb62eb9c799a6ce18f3c26273a61fb6299817c5709288dd8fe066a6e5e628179438975fb7032eab5b030b11e5
Tiger	1ed9471df29ff0bad21e4feccd0bdc2cd43f68c868ec4f23
Whirlpool-0	e2a812892c0146b5ef3e2a32f445951ccca8f70313a6ffb99d7633bebdfda92a9e358e999ba529b2dc1c6d8bda2328600873ff9634ac2d23c156cf213c1120b9 null
Whirlpool-T	1b17a85612fc46164567273198d0de0db0a24f278af151d5e27a8eda8b5ad4b5a7ad33d1d68f142ab2c39852cbe08518c2680da7dd6122e252e77d0c0bb2fa1f
Whirlpool	d3acf66439d7db56c71cf3bebcecd703e265857565053601e892d76ade03d1f04a54808e78e8041b8cfe8127096993405bb48870ab4edb47fd2c6635f648c2f1

The Second file used here is the question file provided for this experiment, a sample of which is shown below :

Exercise 1

05/08/2021

Hash functions for verifying the integrity of files or messages

One of the well-known applications of hash functions is for verifying the integrity of files or messages. (See https://en.wikipedia.org/wiki/Cryptographic_hash_function under applications)

a.Make use of any online tool such as http://www.fileformat.info/tool/hash.htm to compute the MD5, SHA-1, SHA-256 hash values of the two strings given below

1.The quick brown fox jumps over the lazy dog

2.The quick brown fox jumps over the lazy dogs

Note that the two strings above are slightly different yet their hash values are quite different.

Let us see the hash values of this file :

Results	
Original text	*(binary only)*
Original bytes	7b5c727466315c616465666c616e67313032355c616e73695c... (length=52825)
Adler32	57961632
CRC32	eec70a82
Haval	d0b4e63b828466f3224261f3f42a94c4
MD2	567e2e25c17964ee58645e4d49e85c90
MD4	faf9841f9b15a0ddc5e5d9c154eb2cb0
MD5	7ab3369188b16894e4433d59adb9c920
RipeMD128	0d0ea379d5b3d0f43fb08cc1006633dc
RipeMD160	0241a8be9463f37bf016c7fc8703927586a17aba
SHA-1	b3ea3cc3dcf8e9262d05f40bd1839fea6cb2394a
SHA-256	5c29c94cb4c8062973199d27a8a8c1e9b5122963d022b95361274f83cc5b71bc
SHA-384	c855b0844743976bce5b964114250a1f21f5339312fe6357323ed1cd04dcf55e01adbac393ca54ecf1ec48d45d13a81d
SHA-512	01d63c864ea01ea065f17826979167d728b56fe65f507c9d114e3b9ea33e5dd6a43b02bbbfd542a9560666c520d8f87560866a1dfd78d9095c63f4cd992d4127
Tiger	646e44bbd5be5f14810fca81727dc3c9019af60adfcf8abb
Whirlpool-0	60889bf03125e3400fbd7b4550b0108b53b7e39a139c310adb073977504e81e05a2a90fd48644c011ee0d38717088048ae7c6237caf580f42d709d7fa61a6c11 null
Whirlpool-T	ecb8f43b98daa6f85066fd0f2495d9495c019654f45c51b0902e3fe6a77e8ecd063280c7a4b73b27b65b4bdd302da48afc3a0ebd8cc87bf1b641c662a81cf949
Whirlpool	b9fecc814b3497e3da1727eb17c82f6b341f5f9e1e3258f2393209dfdf955e014fda3981c47a5335fbb33014d5af7a3fd4795ca14a9ec0fb8e824d58e7b91f10

We can notice that the second file is relatively larger than the former, and that both these files have different file formats, so we consider their binary format for hashing, and irrespective of the size of the file or the number of lines in it, the **hash values are of the same size**, which proves the other property of hashing.

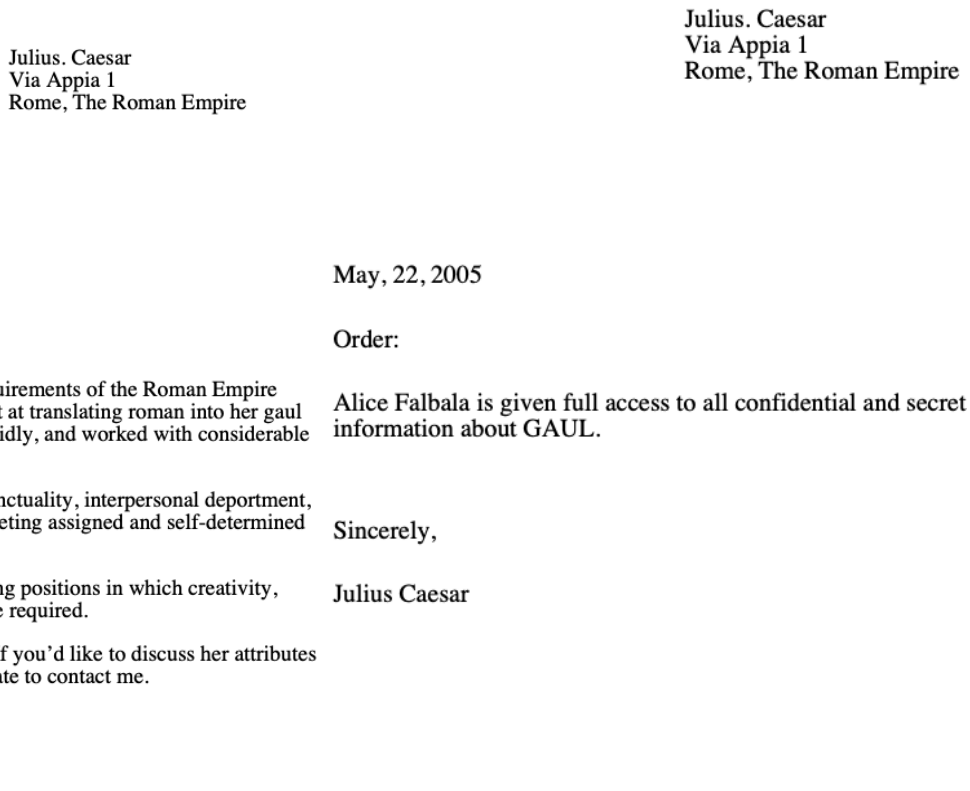
Question 3 :

Consider the two postscript files at

http://web.archive.org/web/20071226014140/http://www.cits.rub.de/MD5Collisions/

Are the two files identical? Now compute the MD5 hash values for each of them. Are they equal? If so why does this happen?

If we use the given links to head over to the website, and have a look at the files, we can notice that they are **not equal.** The first file is a letter of recommendation, while the other is an internal service order to grant permission for certain confidential file access. We can notice from the below pictures that these two files are not same, but let us hash these files.



Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

To Whom it May Concern:

Alice Falbala fulfilled all the requirements of the Roman Empire intern position. She was excellent at translating roman into her gaul native language, learned very rapidly, and worked with considerable independence and confidence.

Her basic work habits such as punctuality, interpersonal deportment, communication skills, and completing assigned and self-determined goals were all excellent.

I recommend Alice for challenging positions in which creativity, reliability, and language skills are required.

I highly recommend hiring her. If you'd like to discuss her attributes in more detail, please don't hesitate to contact me.

Sincerely,

Julius Caesar

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

Order:

Alice Falbala is given full access to all confidential and secret information about GAUL.

Sincerely,

Julius Caesar

Let us use the web portal provided in the previous question to obtain the **MD5 hash values** of these two files. The hash value of the files are :

File 1 - MD5 Hash value : a25f7f0b29ee0b3968c860738533a4b9
File 2 - MD5 Hash value : a25f7f0b29ee0b3968c860738533a4b9

These are the hash values for the letter of recommendation file :

Results	
Original text	*(binary only)*
Original bytes	252150532d41646f62652d312e300d0a2525426f756e64696e... (length=2029)
Adler32	523daa4c
CRC32	2672366f
Haval	a0df2095bc7357ee57c307e5aee34ef8
MD2	bc53cc5930c273f49ab99feb3e5a1b1f
MD4	9679de1bc1526a891530b2242f305407
MD5	a25f7f0b29ee0b3968c860738533a4b9
RipeMD128	c7d90f8e5e0f9bccfe5c46e64847982e
RipeMD160	90698acc6d676608657b9c26f04759a1dc0e6ca1
SHA-1	07835fdd04c9afd283046bd30a362a6516b7e216
SHA-256	de4e4c6e2b94e95a3c5bd72a9a6af29bc5f83bf759325d9921943a6fc08ea245
SHA-384	00b274f70400f91a7fb041579d9839ea203cdc70db1f8484314a3af8b4f2d4974db536b1ee9346133e970bad62949c47
SHA-512	a6e75027235c689a1887aec0698a8b5be6f78a19fb94f347a6560e47de27404ed1fbf32f8fe73c5a2ab23890b9e241971fbbe574d64b8482e609679892289382
Tiger	a6fde2075a6fb8094821be9a8a2c02448edbe02bb740ea76
Whirlpool-0	d835688b0b3748ce64522654413058cc9962c2ebb97552688693d068773577d0fb861114dd91a4f5250b9f950e404fde94a03f8a0e6700293f71bc09cc901042 null
Whirlpool-T	a4d004e4267b900b8afb121ed1b69ae4b7626740ea94512afccb31980d54681bebab09c256ebcf80b8ba32c2027b4133a99b23866982e6326a1b54244aff9f95
Whirlpool	10d50d582891e41b25779fad0bb73f44f1d1d56be0f37cd272a955593c3b1343d30003a040951b8366d434e6e598a6337c25720533b91025c1970f1d78674243

These are the hash values for the executive order file :

Results	
Original text	*(binary only)*
Original bytes	252150532d41646f62652d312e300d0a2525426f756e64696e... (length=2029)
Adler32	d144a94d
CRC32	4e21c5bf
Haval	b334d95c36e23af9d6d5ba88b7526639
MD2	a8b4256a215dd86585b1f3dc2be5036f
MD4	47559a9efd3205bb2fa26f31b012803a
MD5	a25f7f0b29ee0b3968c860738533a4b9
RipeMD128	084a47c85d9d37eb482323a057521ec0
RipeMD160	c1bbde12b312eaaddd3dd3b84ca1cb1bba47dd13
SHA-1	3548db4d0af8fd2f1dbe02288575e8f9f539bfa6
SHA-256	077046dd66015e05c3e03a43a6e4de129038e0701de5a4103fc7ed91c3782d06
SHA-384	b198f3c5588f105182cf66e77e42ffcc93321dd0ff904a3b35c2e376f0053f0f5f6055d6bc41488d54905707a338c75c
SHA-512	d0bb7b8a0765d1f761cd3f7d41890884a5f3b114e21d4232500b3b0f2a614d8c19eefc77c70fa3b1f89eb835892d1dc6b789932f7d61543ec01468ee36f72cb0
Tiger	c090b8aac36249f6ddec625d499e4990a63d0b2d2362dad7
Whirlpool-0	54c80eed18ee020239a3d16f99fec3ac4e656b96b59a4cc8cabfade2b24d07ce668947ae9078c7d0cda2d6a213f29f2eab5a0513592a686c8e7bc82eccadfa16 null
Whirlpool-T	c51a5f2087fb4965cbadf289dba2a211ed3611ddcb70a21559eb59fac2495b5cb963e508adc004df990c34f083a2ede2ab8f276906a5620b75102ab3dc09633f
Whirlpool	10d529d13c6a760ee0eadd72564742cf7121eb761e23f4b9d9178579d60cd4ee122097199631c7f3edba8639d6e22db36a0a84225d601137cf60049abe9b4f3e

We can notice by this that the **MD5 hash values** of both these files are **Equal**.

This situation is called as HASH Collision. This occurs because the hashing algorithms map a file or text of any arbitrary size into a small fixed length string, and so there will be many files or text with the same hash value and different plain text.
Though Hash collisions are possible, we can be assured that they are extremely rare, furthermore it is a property of a good hash function that they should be collision resistant i.e hash collisions must be extremely rare. By this we can further say that the MD5 hash function has been broken and is thus no more a strong hashing algorithm.

CONCLUSION

In this set of lab experiments, we have dealt with the concept of hashing text, files and the different hashing algorithms, their properties, and also the general properties and necessary qualifications of a hash function, and also we have seen what makes a hash function have collisions. (We have proved this concept with an example where collisions occur in MD5 hash function).