CHAPTER 5

FINDINGS

This chapter presents the finding and discussion of experiments conducted for this thesis. Three experiments have been conducted which are performance comparison of encryption and decryption process between using existing AES with enhanced AES geo-key, validating file decryption successfulness at different location, and to validate the integrity of decrypted data file is identical as the original data file. In the next following section will begin by giving a general overview of the proposed method using location-based cryptography and followed with development phase on the experimental setup and its analysis. Each experiment involves the explanation of its procedure, results, and discussion.

5.1 Enhanced AES Geo-Key Method using Location Information to Generate Encryption Key

The main goal of this chapter is to evaluate the enhanced AES geo-key method that introduces the requirement parameter to generate the encryption key by using the location information which are latitude and longitude, device MAC address and user password.

One of the primary methods to compare the performance of both method is by observing the difference on execution time of encryption and decryption process

between the existing AES method and enhanced AES geo-key method. A consistent implementation of both method across the source code was adopted to ensure a fair comparison. Both the existing AES method and enhanced AES geo-key method are implemented using Cipher Block Chaining (CBC) mode of operation in Python language. This experiment has been carried out using an Intel(R) Core i7-9750H CPU @ 2.60GHz machine with Windows 10 as an operating system. For location data retrieval device, a Raspberry Pi 4 module which build in NEO-M8N GPS chip with GY-GPSV3 model antenna were used.

5.2 Experimental Data Set

In order to execute and test the prototype of this enhanced method, difference type of files with variety of file's extension formats are utilized as mentioned in research scope (subchapter 1.5). Most of the files are from personal file and some are taken randomly via Google which were shared publicly as listed in Table 5.1.

Table 5.1: List of experimental data set for encryption and decryption.

Sample File's Name	File's Type	File's Size (kb)	Source
File 1.txt	Document	1.0	Google
File 2.csv	Document	2.3	Google
File 3.png	Photo	7.6	Personal
File 4.xlsx	Document	11.4	Personal
File 5.docx	Document	25.6	Google
File 6.pptx	Document	57.7	Personal
File 7.pdf	Document	194.6	Google
File 8.jpg	Photo	437.9	Personal
File 9.jpg	Photo	985.3	Personal
File 10.wma	Audio	3,325.3	Personal
File 11.jpg	Photo	4,987.9	Personal
File 12.mp3	Audio	11,222.7	Personal
File 13.wav	Audio	25,251.2	Google
File 14.mp4	Video	56,815.1	Personal
File 15.mp4	Video	127,834.0	Personal

File 16.mp4	Video	287,626.6	Personal
File 17.mkv	Video	647,159.8	Personal
File 18.avi	Video	1,456,109.6	Personal

5.3 Experimental Setup

This research has conduct three extensive experiments to evaluate the enhanced AES geo-key method as mentioned in third objective of this research which are (i) performance comparison of encryption and decryption process using the existing AES method and enhanced AES geo-key method, (ii) validating file decryption successfulness at different location, and (iii) data integrity validation testing. There were 18 different types of sample files with different file's size have been used in all Experiment I, Experiment II and Experiment III. Table 5.1 shows the list of files used in the mentioned experiments. The file's type, file's size and sources of the file are also stated.

Experiment I: Time performance comparison of encryption and decryption process between existing AES method and enhanced AES geo-key method.

In this experiment, there were four categories of file's type to be observe which are document, photo, audio, and video. All of the sample file from each type are executed five times for encryption process followed by it decryption process concurrently using existing AES method and repeated using enhanced AES geo-key method. The experiment setup for encryption and decryption process using existing AES method is as in Experiment I(a) and Experiment I(c) of Table 5.2 while Experiment I(b) and Experiment I(d) in the same table configure the setup for encryption and decryption process using enhanced AES geo-key method respectively.

The output from Experiment I(a), Experiment I(b), Experiment I(c) and Experiment I(d) are recorded and shows in Table 5.3, 5.4, 5.5 and 5.6 respectively. The significant differences of execution time for encryption and decryption process using both methods are illustrated accordingly in Figure 5.1, 5.2, 5.3 and 5.4.

Table 5.2: Experiment I configuration set up

Experiment I(a):	**Encryption using existing AES method**
Input:	Plaintext (file – data set in Table 4.1) User password
Key generation:	Using password
Process:	Files will be encrypted using the existing AES method with the used of password as encryption key
Output:	Cipher text (encrypted file) Execution time been recorded
Experiment I(b):	**Encryption using enhanced AES geo-key method**
Input	Plaintext (file – data set in Table 4.1) Latitude & longitude coordinates for decryption User password (user input) MAC address (retrieved by Raspberry Pi) Intended distance threshold for decryption (user input)
Key generation:	Key A: Consist of longitude, latitude, MAC address. Key B: Consist of password. Geo-key: Merging of Key A and Key B
Process:	Files will be encrypted using the enhanced AES geo-key method with the inclusion of geo-key as encryption key.
Output:	Cipher text (encrypted file) Execution time been recorded
Experiment I(c):	**Decryption using existing AES method**
Input:	Cipher text (encrypted file) User password
Key generation:	Using password
Process:	Files will be decrypted using the existing AES method with the used of password as decryption key
Output:	Decrypted file Execution time been recorded
Experiment I(d):	**Decryption using enhanced AES geo-key method**
Input	Cipher text (encrypted file) Latitude & longitude coordinates (retrieved by Raspberry Pi) User password (user input) MAC address (retrieved by Raspberry Pi)

Key generation:	Key A: Consist of longitude, latitude, MAC address. Key B: Consist of password. Geo-key: Merging of Key A and Key B
Process:	Files will be decrypted using geo-key if the retrieved coordinates were inside the intended distance threshold set during initial user input at Experiment I (b)
Output:	Decrypted file Execution time been recorded

Table 5.3: Records of Experiment I(a) outputs

File No	Time Taken for Encrypt files using Existing AES Method (milliseconds)						Std. dev	Margin of Error
	t_1	t_2	t_3	t_4	t_5	t_{avg}		
File 1.txt	13	8	9	11	9	10	2.00	0.92
File 2.csv	11	10	13	11	12	11.4	1.14	0.53
File 3.png	28	8	9	18	9	14.4	8.62	3.98
File 4.xlsx	12	10	22	11	16	14.2	4.92	2.27
File 5.docx	18	17	9	18	13	15	3.94	1.82
File 6.pptx	22	15	13	19	14	16.6	3.78	1.75
File 7.pdf	32	38	16	35	27	29.6	8.62	3.98
File 8.jpg	22	18	20	20	19	19.8	1.48	0.69
File 9.jpg	132	138	150	135	144	139.8	7.22	3.34
File 10.wma	101	165	131	133	148	135.6	23.68	10.94
File 11.jpg	254	224	228	239	226	234.2	12.50	5.77
File 12.mp3	703	591	649	647	620	642	41.47	19.16
File 13.wav	1498	1504	1535	1501	1520	1511.6	15.60	7.21
File 14.mp4	3345	3370	3050	3358	3210	3266.6	137.21	63.39
File 15.mp4	7488	7326	7443	7407	7385	7409.8	60.96	28.16
File 16.mp4	16865	17187	16559	17026	16873	16902	232.63	107.47
File 17.mkv	24326	25991	25434	25159	25713	25325	638.66	295.05
File 18.avi	40020	37690	37938	38855	37814	38463	983.90	454.54

Table 5.4: Records of Experiment I(b) outputs

File No	Time Taken for Encrypt files using Enhanced AES Geo-key Method (milliseconds)						Std. dev	Margin of Error
	t_1	t_2	t_3	t_4	t_5	t_{avg}		
File 1.txt	15	33	36	24	35	28.6	8.96	4.14
File 2.csv	37	38	37	38	38	37.6	0.55	0.25
File 3.png	32	35	37	34	36	34.8	1.92	0.89
File 4.xlsx	38	35	38	37	37	37	1.22	0.57
File 5.docx	49	45	47	47	46	46.8	1.48	0.69
File 6.pptx	44	47	49	46	48	46.8	1.92	0.89
File 7.pdf	53	59	57	56	58	56.6	2.30	1.06
File 8.jpg	56	57	52	57	55	55.4	2.07	0.96

File 9.jpg	154	158	156	156	157	156.2	1.48	0.69
File 10.wma	173	175	176	174	176	174.8	1.30	0.60
File 11.jpg	279	279	275	279	277	277.8	1.79	0.83
File 12.mp3	866	818	934	842	876	867.2	43.58	20.13
File 13.wav	1545	1561	1518	1850	1539	1602.6	139.16	64.29
File 14.mp4	4311	3303	3419	3807	3361	3640.2	423.79	195.78
File 15.mp4	7779	7521	7769	8315	7645	7805.8	303.45	140.19
File 16.mp4	16286	18458	18566	17372	18512	17838.8	999.31	461.66
File 17.mkv	24511	25026	25226	27768	25126	25531.4	1280.30	591.47
File 18.avi	40654	37301	48634	38978	37986	40710.6	4605.27	2127.53

Table 5.5: Records of Experiment I(c) outputs

File No	**Time Taken for Decrypt files using Existing AES Method (milliseconds)**						**Std. dev**	**Margin of Error**
	t_1	**t_2**	**t_3**	**t_4**	**t_5**	**t_{avg}**		
File 1.txt	13	8	9	11	9	10	2.00	0.92
File 2.csv	11	10	13	11	12	11.4	1.14	0.53
File 3.png	28	8	9	18	9	14.4	8.62	3.98
File 4.xlsx	12	10	22	11	16	14.2	4.92	2.27
File 5.docx	18	17	9	18	13	15	3.94	1.82
File 6.pptx	22	15	13	19	14	16.6	3.78	1.75
File 7.pdf	32	38	16	35	27	29.6	8.62	3.98
File 8.jpg	22	18	20	20	19	19.8	1.48	0.69
File 9.jpg	132	138	150	135	144	139.8	7.22	3.34
File 10.wma	101	165	131	133	148	135.6	23.68	10.94
File 11.jpg	254	224	228	239	226	234.2	12.50	5.77
File 12.mp3	703	591	649	647	620	642	41.47	19.16
File 13.wav	1498	1504	1535	1501	1520	1511.6	15.60	7.21
File 14.mp4	3345	3370	3050	3358	3210	3266.6	137.21	63.39
File 15.mp4	7488	7326	7443	7407	7385	7409.8	60.96	28.16
File 16.mp4	16865	17187	16559	17026	16873	16902	232.63	107.47
File 17.mkv	24326	25991	25434	25159	25713	25325	638.66	295.05
File 18.avi	40020	37690	37938	38855	37814	38463	983.90	454.54

Table 5.6: Records of Experiment I(d) outputs

File No	**Time Taken for Decrypt files using Enhanced AES Geo-key Method (milliseconds)**						**Std. dev**	**Margin of Error**
	t_1	**t_2**	**t_3**	**t_4**	**t_5**	**t_{avg}**		
File 1.txt	15	33	36	24	35	28.6	8.96	4.14
File 2.csv	37	38	37	38	38	37.6	0.55	0.25
File 3.png	32	35	37	34	36	34.8	1.92	0.89
File 4.xlsx	38	35	38	37	37	37	1.22	0.57
File 5.docx	49	45	47	47	46	46.8	1.48	0.69
File 6.pptx	44	47	49	46	48	46.8	1.92	0.89
File 7.pdf	53	59	57	56	58	56.6	2.30	1.06
File 8.jpg	56	57	52	57	55	55.4	2.07	0.96

File 9.jpg	154	158	156	156	157	156.2	1.48	0.69
File 10.wma	173	175	176	174	176	174.8	1.30	0.60
File 11.jpg	279	279	275	279	277	277.8	1.79	0.83
File 12.mp3	866	818	934	842	876	867.2	43.58	20.13
File 13.wav	1545	1561	1518	1850	1539	1602.6	139.16	64.29
File 14.mp4	4311	3303	3419	3807	3361	3640.2	423.79	195.78
File 15.mp4	7779	7521	7769	8315	7645	7805.8	303.45	140.19
File 16.mp4	16286	18458	18566	17372	18512	17838.8	999.31	461.66
File 17.mkv	24511	25026	25226	27768	25126	25531.4	1280.30	591.47
File 18.avi	40654	37301	48634	38978	37986	40710.6	4605.27	2127.53

The total average execution time and the overall difference to encrypt and decrypt the sample files using both existing AES and enhanced AES geo-key method are shown in Table 5.7. The overall execution time takes about 2.73% difference between both methods for executing encryption and decryption process, where the enhanced AES geo-key method takes 2.73% longer time to execute both process compared to existing AES method took.

Table 5.7: Total average times to encrypt and decrypt files and the overall difference between existing AES method and enhanced AES Geo-key method

Sample Files	**Average Execution Times in millisecond (s)**			
	Encryption using AES	**Decryption using AES**	**Encryption using AES geo-key**	**Decryption using AES geo-key**
File 1.txt	10	5	28.6	25.4
File 2.csv	11.4	3	37.6	27.8
File 3.png	14.4	3	34.8	26.4
File 4.xlsx	14.2	11.2	37	28.8
File 5.docx	15	18.2	46.8	31.2
File 6.pptx	16.6	14.2	46.8	37
File 7.pdf	29.6	15.2	56.6	46.2
File 8.jpg	19.8	14	55.4	49.4
File 9.jpg	139.8	23.4	156.2	56
File 10.wma	135.6	51.2	174.8	76
File 11.jpg	234.2	122.8	277.8	164.4
File 12.mp3	642	189.4	867.2	286
File 13.wav	1511.6	609.4	1602.6	764.8
File 14.mp4	3266.6	1468.6	3640.2	1619.6
File 15.mp4	7409.8	3265.6	7805.8	3734.6
File 16.mp4	16902	7555.8	17838.8	7713

File 17.mkv	25324.6	17632.4	25531.4	20463.6
File 18.avi	38463.4	38840.4	40710.6	39107.2
Total average time	4555.65		4811.29	
Differences between two methods (%)		**5.62%**		

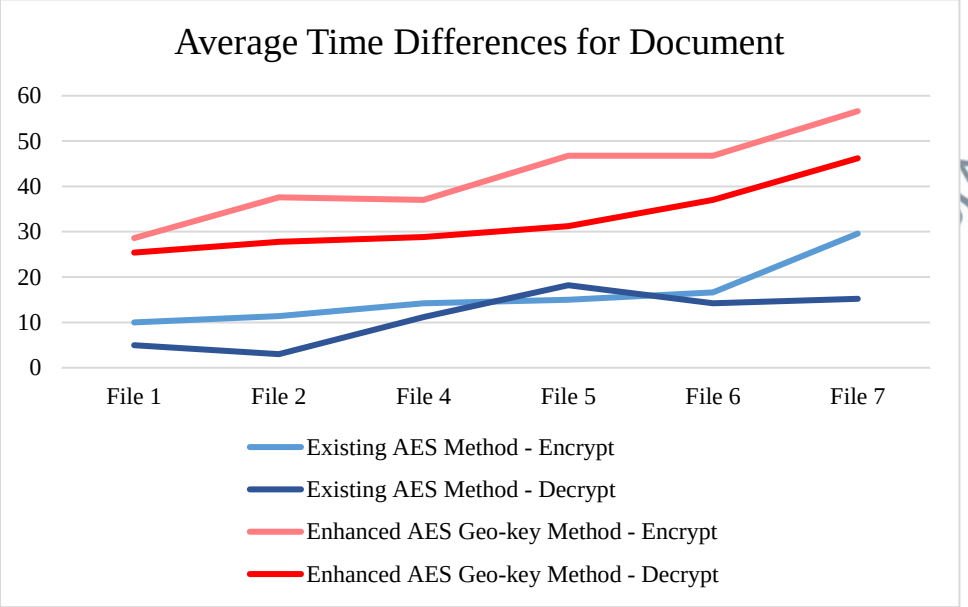


Figure 5.1: Average time difference to encrypt and decrypt document files between using existing AES method with enhanced AES geo-key method.

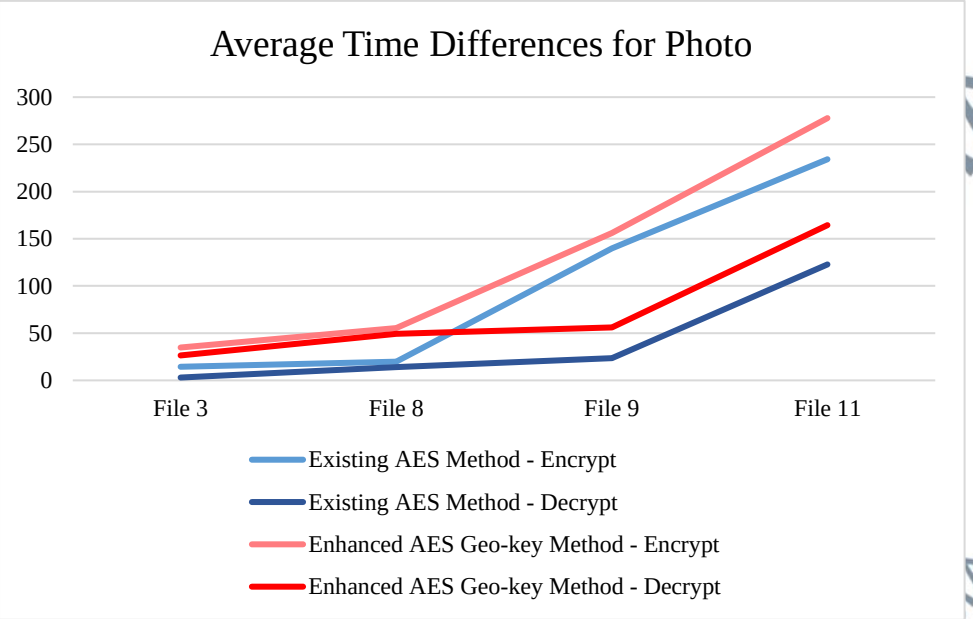


Figure 5.2: Average time difference to encrypt and decrypt photo files between using existing AES method with enhanced AES geo-key method.

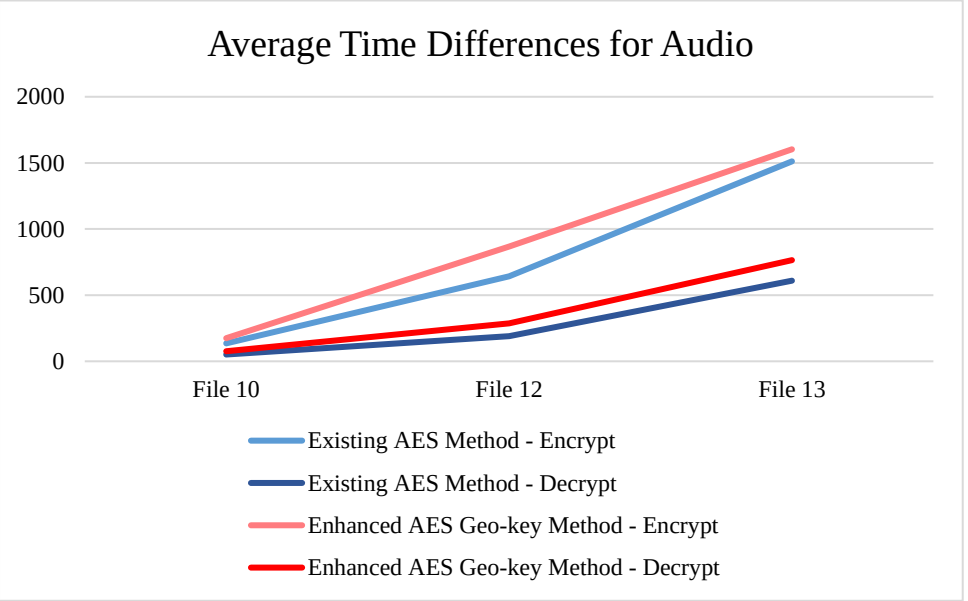


Figure 5.3: Average time difference to encrypt and decrypt audio files between using existing AES method with enhanced AES geo-key method.

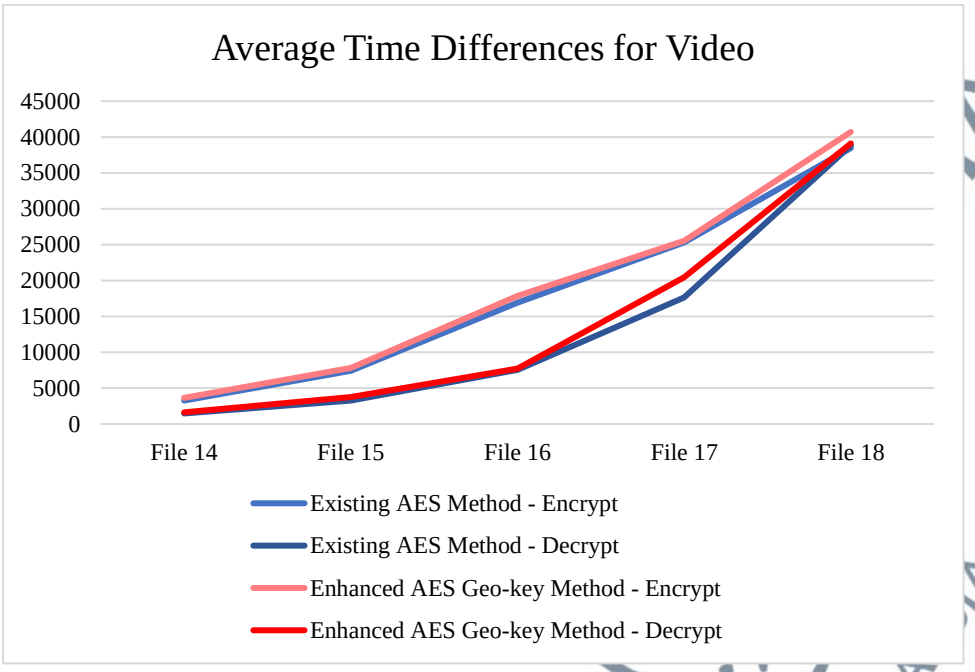


Figure 5.4: Average time difference to encrypt and decrypt video files between existing AES method and enhanced AES geo-key method.

Based on Figure 5.1, 5.2, 5.3, 5.4 and Table 5.7, we can see that both encrypting and decrypting files by using enhanced AES Geo-key method is slower than existing AES method. Average time to encrypt and decrypt document file type using existing AES method shows about 4.19% time-gap differences with enhanced AES Geo-key method (61.80% and 65.99%, respectively).

Aside from that, average time to encrypt and decrypt video using both methods shows the smallest time gap with only 0.91% difference between two methods. The reason is because video file have the largest file's sizes among all the sample files. Therefore, longer time is required to encrypt and decrypt video format files in both existing AES methods and enhanced AES Geo-key method.

Descriptive Analysis

Table 5.8: Descriptive analysis of existing AES method to encrypt and decrypt

Existing AES Method - Encrypt					
	n	**Min**	**Max**	**Mean**	**Std Dev**
Document	6	8	38	16.13	7.81
Photo	4	8	254	102.05	93.98
Audio	3	101	1535	763.07	588.79
Video	5	3050	40020	18273.28	12939.12
Existing AES Method - Decrypt					
	n	**Min**	**Max**	**Mean**	**Std Dev**
Document	6	2	26	11.13	6.25
Photo	4	2	126	40.80	49.19
Audio	3	43	628	283.33	246.09
Video	5	39587	13752.56	14025.78	1436.00

Based on Table 5.8, average time in encrypting and decrypting documents using existing AES method is the fastest (mean: 16.13 and 11.13; standard deviation: 7.81 and 6.25). On the other hand, average time in encrypting and decrypting video using existing AES method is the slowest (mean: 18273.28 and 14025.78; standard deviation: 12939.12 and 1436.0). In short, decrypting files takes shorter average time to done compared to encrypting files by using existing AES method.

Table 5.9: Descriptive analysis of enhanced AES Geo-key method to encrypt and decrypt

Enhanced AES Geo-key Method - Encrypt					
	n	**Min**	**Max**	**Mean**	**Std Dev**
Document	6	15	59	42.23	9.80
Photo	4	32	279	131.05	98.90
Audio	3	173	1850	881.53	608.46
Video	5	3303	48634	19105.36	13661.31
Enhanced AES Geo-key Method - Decrypt					
	n	**Min**	**Max**	**Mean**	**Std Dev**
Document	6	23	51	32.73	7.45
Photo	4	23	175	74.05	55.13
Audio	3	74	785	375.60	298.57
Video	5	1331	40654	14527.60	14266.62

Based on Table 5.9, average time in encrypting and decrypting documents using enhanced AES Geo-key method is the fastest (mean: 42.23 and 32.73; standard deviation: 9.80 and 7.45). On the other hand, average time in encrypting and decrypting video using enhanced AES Geo-key method is the slowest (mean: 19105.36 and 142527.60; standard deviation: 13661.31 and 14266.62). In short, decrypting files takes shorter average time to done compared to encrypting files by using enhanced AES Geo-key method.

In summary, it is shows that enhanced AES Geo-key method has a compromisable time differences to execute the encryption and decryption process with overall time average about 5.62% compared to existing AES method as shows in Table 5.7. This value is not showing a huge-time gap differences between both methods in term of it time performance to execute the encryption and decryption process.

Experiment II: Validating file decryption successfulness at different location.

This experiment was conducted at outdoor environment to validate either the encrypted files are able to be decrypted at different location or not based on the initial distance threshold limit which has been set before the file been encrypted. As we want to test the enhanced AES geo-key method could manage the issue of file access restriction at storage based on location, this experiment will determine whether the decrypted file could still also be access out of restricted location. The encrypted file is only valid to be decrypt is when the variant distance between original location of encryption and request decryption location are less or equal to the distance threshold limit. The calculation to determine the validity of decryption request is as the following:

Let D_T , the distance threshold limit z

(x, y) , set as coordinate location (lat, lon)

A , original encrypt location

B , requested decrypt location

ΔT, variant distance between two point of A and B

R, is the approximate radius of earth 6,378

V, Validity (when, 1 = valid, 0 = invalid)

So,
$$x = R * \cos(lat) * \cos(lon)$$
$$y = R * \cos(lat) * \sin(lon)$$
$$\Delta T = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

Thus,
$$V = \begin{cases} 1, & if\ \Delta T \leq D_T \\ 0, & if\ \Delta T > D_T \end{cases}$$

The result of this experiment is as the following Table 4.10

Table 5.10: Results of decryption successfulness validation at different variant distance

Sample Files	Distance threshold limit, D_T (m)	Validity for decryption, V at		
		$\Delta T = 3m$	$\Delta T = 25m$	$\Delta T = 60m$
File 1.txt	5	valid	invalid	invalid
File 2.csv	5	valid	invalid	invalid
File 3.png	5	valid	invalid	invalid
File 4.xlsx	10	valid	invalid	invalid
File 5.docx	10	valid	invalid	invalid
File 6.pptx	10	valid	invalid	invalid
File 7.pdf	15	valid	invalid	invalid
File 8.jpg	15	valid	invalid	invalid
File 9.jpg	15	valid	invalid	invalid
File 10.wma	20	valid	invalid	invalid
File 11.jpg	20	valid	invalid	invalid
File 12.mp3	20	valid	invalid	invalid
File 13.wav	25	valid	valid	invalid
File 14.mp4	25	valid	valid	invalid
File 15.mp4	25	valid	valid	invalid
File 16.mp4	50	valid	valid	invalid
File 17.mkv	50	valid	valid	invalid
File 18.avi	50	valid	valid	invalid

All file were encrypted at one location called as point A with several different value of the distance threshold limit, D_T from 5 meters to 50 meters has been set at the initial process of the encryption. To validate successfulness of the enhanced AES geo-key method, the decryption request of file was carried out at different location than the initial encryption process which called as point B. There are three variant distances between two point of A and B, ΔT were tested in this experiment which are 3 meters, 25 meters and 60 meters as per assumption of a common terrace residential area range. The results of this experiment were recorded as stated in Table 4.x above. Based on the results, it is shown that the decryption process is only valid to be process if the ΔT value is less than the D_T while all the request to decrypt the encrypted file outside of the D_T range was unsuccessful. Therefore, it is proven that enhanced AES geo-key method can manage the issue on lack of location-based access restrictions at cloud storage.

Experiment III: Data Integrity Evaluation

This experiment was conducted to evaluate the data integrity of decrypted files. Data integrity is where the content of a file is well preserved, and its originality is maintained. Every created file has its own hash value and the hash value is always unique. Whenever a created file been modified or even one letter is been changed, the hash value of the file will also change. Therefore, one of the methods to evaluate the integrity of file is by comparing the hash value of original file with the hash value of decrypted file.

In this experiment, a hash function tools called *HashMyFile* was used to compute the hash value of the original sample file and its decrypted version of file by using SHA-1 and SHA 256. Those two functions were selected as both have high collision resistance compared to other hash function such as MD5. The following Figure 5.5 and 5.6 shows the hash value of the original files and the decrypted files using SHA 1 respectively.

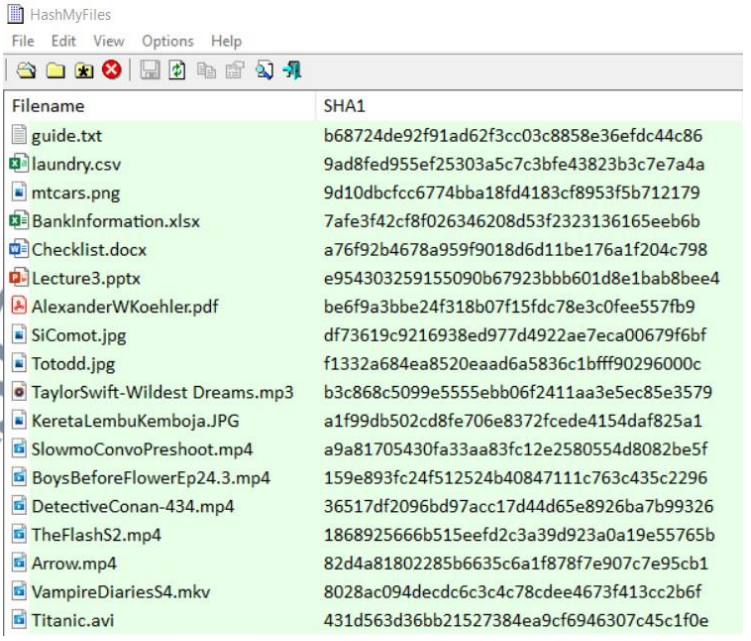


Figure 5.5: Hash values of original sample files using SHA1 before been encrypted.

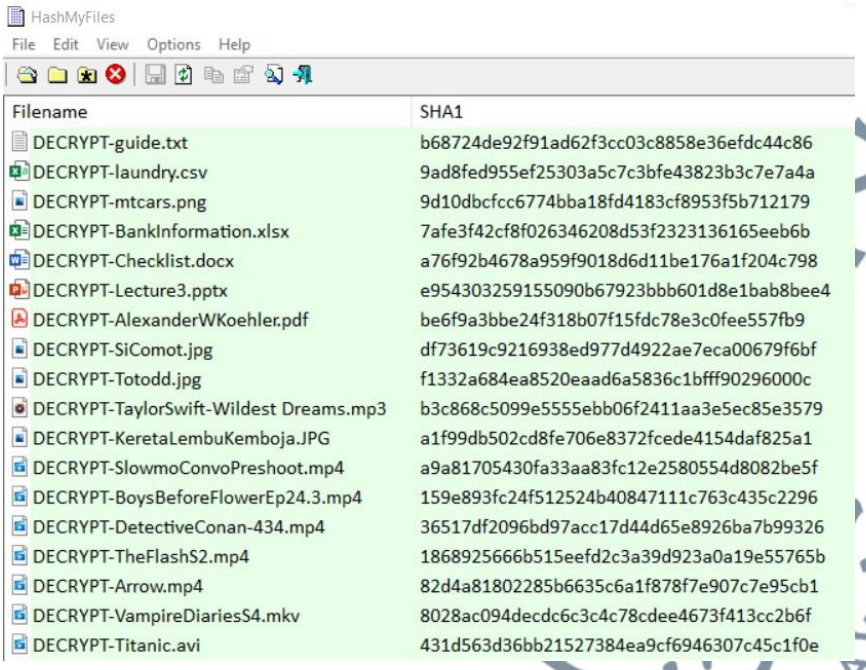
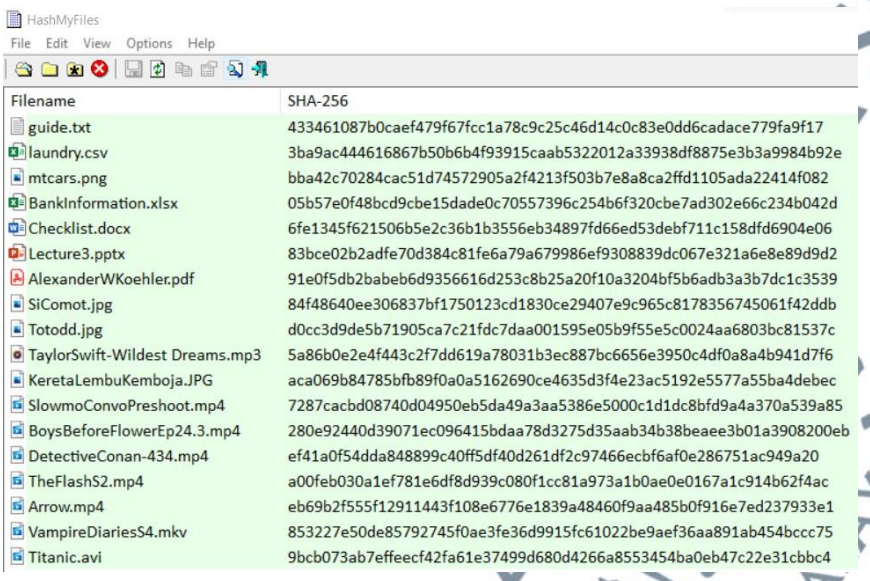


Figure 5.6: Hash values of decrypted files using SHA1 after been decrypted.

Table 5.11: Comparison between hash values of the original file with the decrypted file using SHA 1 hash function.

Sample File	Original File SHA 1	Decrypted File SHA 1
File 1	b68724de92f91ad62f3cc03c8858e36efdc44c86	b68724de92f91ad62f3cc03c8858e36efdc44c86
File 2	9ad8fed955ef25303a5c7c3bfe43823b3c7e7a4a	9ad8fed955ef25303a5c7c3bfe43823b3c7e7a4a
File 3	9d10dbcfcc6774bba18fd4183cf8953f5b712179	9d10dbcfcc6774bba18fd4183cf8953f5b712179
File 4	7afe3f42cf8f026346208d53f2323136165eeb6b	7afe3f42cf8f026346208d53f2323136165eeb6b
File 5	a76f92b4678a959f9018d6d11be176a1f204c798	a76f92b4678a959f9018d6d11be176a1f204c798
File 6	e954303259155090b67923bbb601d8e1bab8bee4	e954303259155090b67923bbb601d8e1bab8bee4
File 7	be6f9a3bbe24f318b07f15fdc78e3c0fee557fb9	be6f9a3bbe24f318b07f15fdc78e3c0fee557fb9
File 8	df73619c9216938ed977d4922ae7eca00679f6bf	df73619c9216938ed977d4922ae7eca00679f6bf
File 9	f1332a684ea8520eaad6a5836c1bfff90296000c	f1332a684ea8520eaad6a5836c1bfff90296000c
File 10	b3c868c5099e5555ebb06f2411aa3e5ec85e3579	b3c868c5099e5555ebb06f2411aa3e5ec85e3579
File 11	a1f99db502cd8fe706e8372fcede4154daf825a1	a1f99db502cd8fe706e8372fcede4154daf825a1
File 12	a9a81705430fa33aa83fc12e2580554d8082be5f	a9a81705430fa33aa83fc12e2580554d8082be5f
File 13	159e893fc24f512524b40847111c763c435c2296	159e893fc24f512524b40847111c763c435c2296
File 14	36517df2096bd97acc17d44d65e8926ba7b99326	36517df2096bd97acc17d44d65e8926ba7b99326
File 15	1868925666b515eefd2c3a39d923a0a19e55765b	1868925666b515eefd2c3a39d923a0a19e55765b
File 16	82d4a81802285b6635c6a1f878f7e907c7e95cb1	82d4a81802285b6635c6a1f878f7e907c7e95cb1
File 17	8028ac094decdc6c3c4c78cdee4673f413cc2b6f	8028ac094decdc6c3c4c78cdee4673f413cc2b6f
File 18	431d563d36bb21527384ea9cf6946307c45c1f0e	431d563d36bb21527384ea9cf6946307c45c1f0e

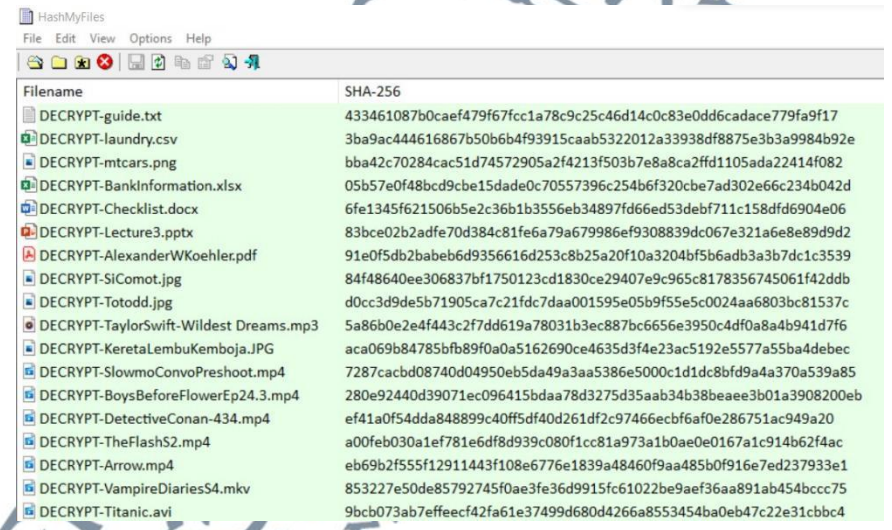
The following Figure 5.7 and 5.8 shows the hash value of the original files and the decrypted files using SHA 256 respectively.


HashMyFiles

File Edit View Options Help

Filename	SHA-256
guide.txt	433461087b0caef479f67fcc1a78c9c25c46d14c0c83e0dd6cadace779fa9f17
laundry.csv	3ba9ac444616867b50b6b4f93915caab5322012a33938df8875e3b3a9984b92e
mtcars.png	bba42c70284cac51d74572905a2f4213f503b7e8a8ca2ffd1105ada22414f082
BankInformation.xlsx	05b57e0f48bcd9cbe15dade0c70557396c254b6f320cbe7ad302e66c234b042d
Checklist.docx	6fe1345f621506b5e2c36b1b3556eb34897fd66ed53debf711c158dfd6904e06
Lecture3.pptx	83bce02b2adfe70d384c81fe6a79a679986ef9308839dc067e321a6e8e89d9d2
AlexanderWKoehler.pdf	91e0f5db2babeb6d9356616d253c8b25a20f10a3204bf5b6adb3a3b7dc1c3539
SiComot.jpg	84f48640ee306837bf1750123cd1830ce29407e9c965c8178356745061f42ddb
Totodd.jpg	d0cc3d9de5b71905ca7c21fdc7daa001595e05b9f55e5c0024aa6803bc81537c
TaylorSwift-Wildest Dreams.mp3	5a86b0e2e4f443c2f7dd619a78031b3ec887bc6656e3950c4df0a8a4b941d7f6
KeretaLembuKemboja.JPG	aca069b84785bfb89f0a0a5162690ce4635d3f4e23ac5192e5577a55ba4debec
SlowmoConvoPreshoot.mp4	7287cacbd08740d04950eb5da49a3aa5386e5000c1d1dc8bfd9a4a370a539a85
BoysBeforeFlowerEp24.3.mp4	280e92440d39071ec096415bdaa78d3275d35aab34b38beaee3b01a3908200eb
DetectiveConan-434.mp4	ef41a0f54dda848899c40ff5df40d261df2c97466ecbf6af0e286751ac949a20
TheFlashS2.mp4	a00feb030a1ef781e6df8d939c080f1cc81a973a1b0ae0e0167a1c914b62f4ac
Arrow.mp4	eb69b2f555f12911443f108e6776e1839a48460f9aa485b0f916e7ed237933e1
VampireDiariesS4.mkv	853227e50de85792745f0ae3fe36d9915fc61022be9aef36aa891ab454bccc75
Titanic.avi	9bcb073ab7effeecf42fa61e37499d680d4266a8553454ba0eb47c22e31cbbc4

Figure 5.7: Hash values of original sample files using SHA 256 before been encrypted.


HashMyFiles

File Edit View Options Help

Filename	SHA-256
DECRYPT-guide.txt	433461087b0caef479f67fcc1a78c9c25c46d14c0c83e0dd6cadace779fa9f17
DECRYPT-laundry.csv	3ba9ac444616867b50b6b4f93915caab5322012a33938df8875e3b3a9984b92e
DECRYPT-mtcars.png	bba42c70284cac51d74572905a2f4213f503b7e8a8ca2ffd1105ada22414f082
DECRYPT-BankInformation.xlsx	05b57e0f48bcd9cbe15dade0c70557396c254b6f320cbe7ad302e66c234b042d
DECRYPT-Checklist.docx	6fe1345f621506b5e2c36b1b3556eb34897fd66ed53debf711c158dfd6904e06
DECRYPT-Lecture3.pptx	83bce02b2adfe70d384c81fe6a79a679986ef9308839dc067e321a6e8e89d9d2
DECRYPT-AlexanderWKoehler.pdf	91e0f5db2babeb6d9356616d253c8b25a20f10a3204bf5b6adb3a3b7dc1c3539
DECRYPT-SiComot.jpg	84f48640ee306837bf1750123cd1830ce29407e9c965c8178356745061f42ddb
DECRYPT-Totodd.jpg	d0cc3d9de5b71905ca7c21fdc7daa001595e05b9f55e5c0024aa6803bc81537c
DECRYPT-TaylorSwift-Wildest Dreams.mp3	5a86b0e2e4f443c2f7dd619a78031b3ec887bc6656e3950c4df0a8a4b941d7f6
DECRYPT-KeretaLembuKemboja.JPG	aca069b84785bfb89f0a0a5162690ce4635d3f4e23ac5192e5577a55ba4debec
DECRYPT-SlowmoConvoPreshoot.mp4	7287cacbd08740d04950eb5da49a3aa5386e5000c1d1dc8bfd9a4a370a539a85
DECRYPT-BoysBeforeFlowerEp24.3.mp4	280e92440d39071ec096415bdaa78d3275d35aab34b38beaee3b01a3908200eb
DECRYPT-DetectiveConan-434.mp4	ef41a0f54dda848899c40ff5df40d261df2c97466ecbf6af0e286751ac949a20
DECRYPT-TheFlashS2.mp4	a00feb030a1ef781e6df8d939c080f1cc81a973a1b0ae0e0167a1c914b62f4ac
DECRYPT-Arrow.mp4	eb69b2f555f12911443f108e6776e1839a48460f9aa485b0f916e7ed237933e1
DECRYPT-VampireDiariesS4.mkv	853227e50de85792745f0ae3fe36d9915fc61022be9aef36aa891ab454bccc75
DECRYPT-Titanic.avi	9bcb073ab7effeecf42fa61e37499d680d4266a8553454ba0eb47c22e31cbbc4

Figure 5.8: Hash values of decrypted files using SHA 256 after been decrypted.

Table 5.12: Comparison between hash values of the original file with the decrypted file using SHA 256 hash function.

Sample File	Original File SHA 256	Decrypted File SHA 256
File 1	433461087b0caef479f67fcc1a78c9c25c46d14c0c83e0dd6cadace779fa9f17	433461087b0caef479f67fcc1a78c9c25c46d14c0c83e0dd6cadace779fa9f17
File 2	3ba9ac444616867b50b6b4f93915caab5322012a33938df8875e3b3a9984b92e	3ba9ac444616867b50b6b4f93915caab5322012a33938df8875e3b3a9984b92e
File 3	bba42c70284cac51d74572905a2f4213f503b7e8a8ca2ffd1105ada22414f082	bba42c70284cac51d74572905a2f4213f503b7e8a8ca2ffd1105ada22414f082
File 4	05b57e0f48bcd9cbe15dade0c70557396c254b6f320cbe7ad302e66c234b042d	05b57e0f48bcd9cbe15dade0c70557396c254b6f320cbe7ad302e66c234b042d
File 5	6fe1345f621506b5e2c36b1b3556eb34897fd66ed53debf711c158dfd6904e06	6fe1345f621506b5e2c36b1b3556eb34897fd66ed53debf711c158dfd6904e06
File 6	83bce02b2adfe70d384c81fe6a79a679986ef9308839dc067e321a6e8e89d9d2	83bce02b2adfe70d384c81fe6a79a679986ef9308839dc067e321a6e8e89d9d2
File 7	91e0f5db2babeb6d9356616d253c8b25a20f10a3204bf5b6adb3a3b7dc1c3539	91e0f5db2babeb6d9356616d253c8b25a20f10a3204bf5b6adb3a3b7dc1c3539
File 8	84f48640ee306837bf1750123cd1830ce29407e9c965c8178356745061f42ddb	84f48640ee306837bf1750123cd1830ce29407e9c965c8178356745061f42ddb
File 9	d0cc3d9de5b71905ca7c21fdc7daa001595e05b9f55e5c0024aa6803bc81537c	d0cc3d9de5b71905ca7c21fdc7daa001595e05b9f55e5c0024aa6803bc81537c
File 10	5a86b0e2e4f443c2f7dd619a78031b3ec887bc6656e3950c4df0a8a4b941d7f6	5a86b0e2e4f443c2f7dd619a78031b3ec887bc6656e3950c4df0a8a4b941d7f6
File 11	aca069b84785bfb89f0a0a5162690ce4635d3f4e23ac5192e5577a55ba4debec	aca069b84785bfb89f0a0a5162690ce4635d3f4e23ac5192e5577a55ba4debec
File 12	7287cacbd08740d04950eb5da49a3aa5386e5000c1d1dc8bfd9a4a370a539a85	7287cacbd08740d04950eb5da49a3aa5386e5000c1d1dc8bfd9a4a370a539a85
File 13	280e92440d39071ec096415bdaa78d3275d35aab34b38beaee3b01a3908200eb	280e92440d39071ec096415bdaa78d3275d35aab34b38beaee3b01a3908200eb
File 14	ef41a0f54dda848899c40ff5df40d261df2c97466ecbf6af0e286751ac949a20	ef41a0f54dda848899c40ff5df40d261df2c97466ecbf6af0e286751ac949a20
File 15	a00feb030a1ef781e6df8d939c080f1cc81a973a1b0ae0e0167a1c914b62f4ac	a00feb030a1ef781e6df8d939c080f1cc81a973a1b0ae0e0167a1c914b62f4ac
File 16	eb69b2f555f12911443f108e6776e1839a48460f9aa485b0f916e7ed237933e1	eb69b2f555f12911443f108e6776e1839a48460f9aa485b0f916e7ed237933e1
File 17	853227e50de85792745f0ae3fe36d9915fc61022be9aef36aa891ab454bccc75	853227e50de85792745f0ae3fe36d9915fc61022be9aef36aa891ab454bccc75
File 18	9bcb073ab7effeecf42fa61e37499d680d4266a8553454ba0eb47c22e31cbbc4	9bcb073ab7effeecf42fa61e37499d680d4266a8553454ba0eb47c22e31cbbc4

Based on the Table 5.11 and 5.12, it is shown that all the hash values for the original files before been encrypted were having the same hash values for the decrypted file in both SHA 1 and SHA 256 function. These results have proven the enhanced AES geo-key method that been used to encrypt the files was not changing the content of the encrypted file. The same exact hash value also confirmed the enhanced AES geo-key method does not even compressing the original file's size for the purpose of encryption process. Therefore, it can be concluded that the enhanced AES geo-key method can be used to secure file in cloud storage without compromising the integrity of the stored data.