Boyer-Moore String Matching Algorithm and SHA512 Implementation for Jpeg/exif File Fingerprint Compilation in DSA

Rachmad Fitriyanto[1], Anton Yudhana[2], Sunardi[3]

[1]Information System Department, STMIK PPKIA Tarakanita Rahmawati
Jl. Yos Sudarso No.8 Tarakan Barat Tarakan
[2,3]Electrical Departement, Universitas Ahmad Dahlan
Jl.Dr.Soepomo Janturan Yogyakarta

[1]fitriyanto7477@gmail.com
[2]yudhana@ee.uad.ac.id
[3]sunardi@mti.uad.ac.id

Abstract— **the jpeg/exif is file's format for image produced by digital camera such as in the smartphones. The security method for jpeg/exif usages in digital communication currently only full-fill prevention aspect from three aspects of information security, prevention, detection and response. Digital Signature Algorithm (DSA) is a cryptographic method that provide detection aspect of information security by using hash-value as fingerprint of digital documents. The purpose of this research is to compile jpeg/exif file data fingerprint using the hash-value from DSA. The research conducted in four stages. The first stages is the identification of jpeg/exif file structure using Boyer-Moore string matching algorithm to locate the position of file's segments. The second stage is segment's content acquisition. The third stage the image files modification experiments to select the suitable element of jpeg/exif file data fingerprint. The fourth stage is the compilation of hash-values to form data fingerprint. The Obtained result has shown that the jpeg/exif file fingerprint comprises three hash value from the SOI segment, APP1's segment, and the SOF0 segment. The jpeg/exif file fingerprint can use for modified image detection, include six types of image modification there are image resizing, text addition, metadata modification, image resizing, image cropping and file type conversion**

Keywords - **Boyer-Moore, SHA512, Jpeg/exif, Digital Signature Algorithm, Fingerprint**

I. INTRODUCTION

The information security have three stages to secure the communication [1]. Prevention stage have purpose to prevent attack to information. The detection stage have purpose to detect the condition on security parameters on information. The response stage have purposes to provide anticipation based on detection stage's result. Image and video files are two types of files used in many ways communication. The information security must ensure the information that received is in the same condition as when it was sent. Information must also arrive at the right recipient. The recipient of information must also be able to ensure that the received information comes from senders he or she knows. Both image and video files generated from digital camera usage such as in smartphones. Image file from smartphones has specific file format saved as jpeg/exif file. The Jpeg/exif file format, structured from parts that each part store specific information about the image [2].

Wijayanto used image's metadata to protect the copyright of photographic image. The protection methods conducted in three steps. The metadata encrypted then inserted in End Of File segment and the original metadata deleted [2]. Research by Gangwar use metadata to identify the authenticity of digital image [3]. The authenticity of an digital image detected by analysing metadata with several open-source application.

An issue aroused from described previous research is security methods only full-fill the first stage of information security. Previous research cannot provide detection stages yet. Another previous research that provide method for information security detection stage conducted by Refialy [4]. This research use message digest from a digital document. Two hash value generated from pdf document. The first hash value generated before document sent. The second hash value generated after document received. The comparison of two hash value use to determine the authenticity of pdf document.

The information security method proposed in this research for provide detection stage of information security is using Digital Signature Algorithm (DSA). DSA is a cryptographic method that used to provide three of information security parameters, authenticity both information and owner-receiver, data integrity and non-repudiation [5]. DSA uses three elements, data fingerprint, asymmetric key-pair and digital certificate [6]. Data fingerprints used to secure information data integrity. Asymmetric key-pair used to identify the information about sender and receiver of document. The digital certificate used to provide non-repudiation to use before the law [7]. DSA conducted both sides, information sender and receiver as shown in Figure 1.

The First process until third process shows generating process of information fingerprint. These three steps executed both on sender side and receiver side. The Fourth process is information fingerprint encryption with the asymmetric key pair to generate cipher text. The Fifth process is embedding the cipher text from the previous process into information [8]. The seventh step until fifteenth executed on receiver side. The seventh, eighth and ninth step is extraction process to separate between the original information and cipher text. The tenth step is decrypting the cipher text to get data fingerprint from sender side. The eleventh, twelfth and thirteenth is the same process as first, second and third process on sender's side. The fourteenth process is data integrity verification by comparing two data fingerprint. The information in original status if two data fingerprint have the exact value. Otherwise, the received information have been modified along communication process on step sixth..

The processes in DSA shows the importance of data fingerprint for data integrity verification. Information segment identification took the first place in data fingerprint generating processes. The identification step conducted by locating each segments that constructed the information by searching specific pattern with string matching algorithm. Boyer-Moore (BM) string matching algorithm is an exact string matching algorithm. BM string matching algorithm search pattern by comparing pattern's elements to string' elements [9]. BM string matching algorithm applied in many ways, such as for knowledge management system [10], DNA pattern searching [11] and data similarity for digital forensics [12].

Secure Hash Algorithm (SHA) is a cryptographic hash function developed by the National Institute of Standards and Technology (NIST) [13]. SHA has multiple variants, SHA0, SHA1 with two versions of hash functions MD4 and MD5 and SHA2 which have six hash functions include SHA512. Every SHA variants have different hashing process and different output size. SHA512 has 512-bit message digest. SHA512 output size makes this SHA variant have better performance than its predecessor [14]. Hash value that produced from hash function has used for installer file data integrity verification that provided by software or application developer and for data acquisition in digital forensics [15]. The example of the implementation of the hash value indicates that the hash value is an element that can be used to verify data integrity as part of information security parameters. The focus of this research is to compile data fingerprint from the jpeg/exif file using Boyer-Moore string matching algorithm and hash value from SHA512 hash function. Data fingerprint must have function to detect the integrity of image file from common image modification methods.

II. METHOD

The research study conducted in four stages as shown in Figure 2. The First stage is the identification of jpeg/exif file structure. Image files collected from two smartphone types, Asus Z00UD and Samsung Galaxy A5. Image files are output from the usage of smartphone digital camera. Image file taken with default camera application in each smartphones, with mode auto and taken place indoor and outdoor. Image file quantity for this research are 10 images from each smartphones.

Image files data converted into string before identification started. The purpose of first stage is to collect file segments location by searching the location of each segments. A jpeg/exif file constructed from 6 segment, SOI, APP1, DQT, SOF0, DHT and SOS. Each segment have unique data that have function as file signature that located in the beginning of segment called segment marker [16]. Table 1 show segment marker for six segment of jpeg/exif file.

Segment marker arranged in 4-byte hexadecimal format number. Each segment marker used as pattern input parameter for segment location searching by BM string matching algorithm. Searching result only have two condition, match or not match. Match condition occurred if pattern found inside image data, while not match condition occurred when pattern not found inside image data. Figure 3 shows the example of BM searching process.

The example from Figure 3 show data string with 42 characters length and pattern with 5 characters length. The BM string matching first step is comparing the most-right element or the fifth element of pattern (b) to the fifth element of the string (e). If comparing result not match, then comparing process continued to compare the fourth element of pattern (0) to the fifth element of the string (e). If comparison has reached to first element of pattern, and there is still not have match character, searching process will start again by moving pattern to right as far as pattern length as shown in step 2 from Figure X. Same processes and result occurred on step 2 and 3. Step 4 gave different comparing result than three previous steps. Match character found when pattern second character (7) compared to twentieth string character (7). This condition trigger Good-Character Rule. Pattern will moved to right so matching characters aligned. When pattern has move, searching process start from beginning again. Pattern searching from Figure X, stopped at ninth step or the 35th character even string still have 7 characters that has not compared. If pattern from Figure X is a segment marker, then the segment that been search located in 35th character.

Segment identification require two parameters, start index and length of segment. Start index acquired from the last data string index and length of segment acquired from the subtraction result between two start index from adjacent segment. Equation (1) show formula to calculate the length of a segment.

$$L_{(m)} \quad = Startindex(n) - Startindex(m) \qquad (1)$$

Segment start index and length use as parameter to segment's content acquisition in the research's second stage. The acquired content processed with SHA512 hash function to generate segment's hash value. SHA512 has three steps, preprocessing, hash computation and hash value compilation as shown in Figure 4.

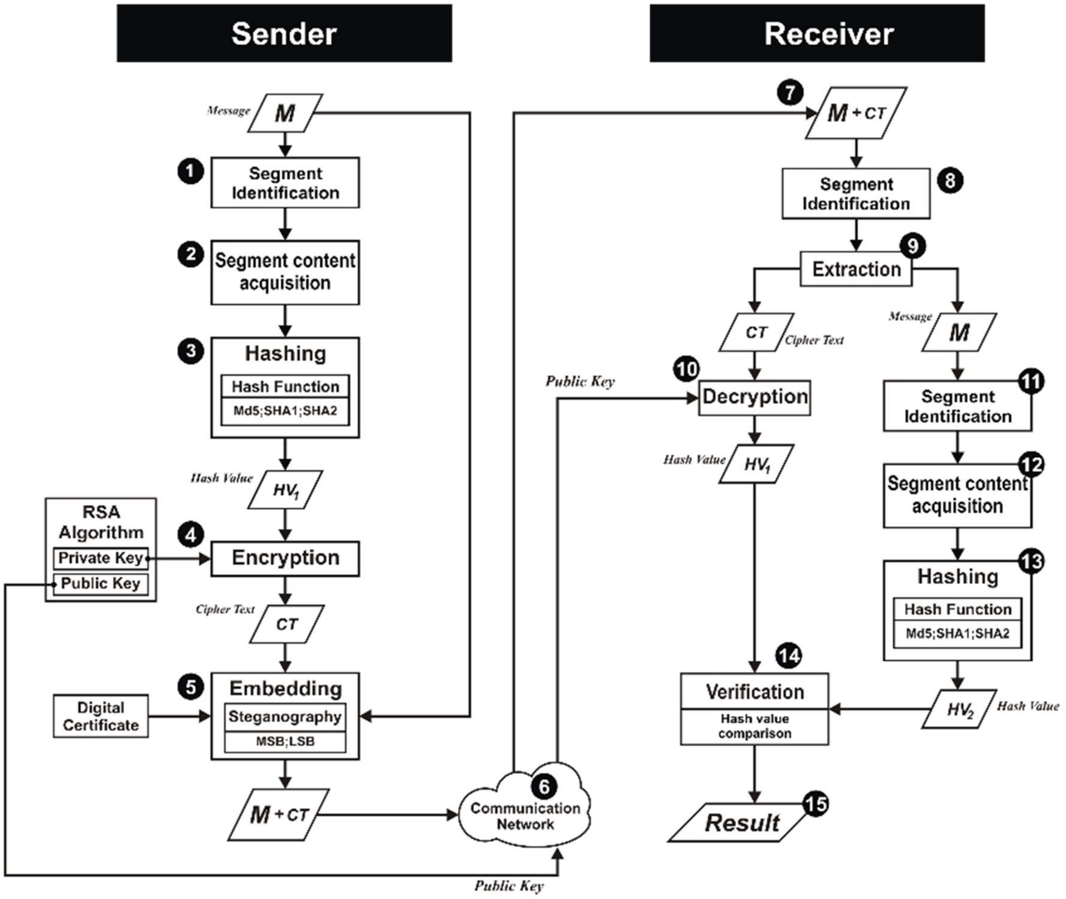


Fig. 1 Digital Signature Algorithm Processes

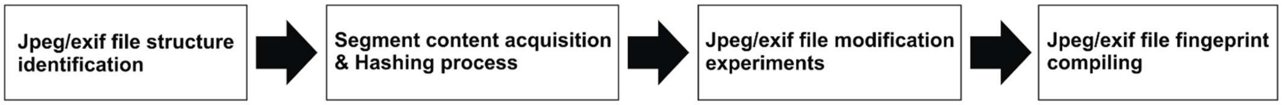

Jpeg/exif file structure identification
Segment content acquisition & Hashing process
Jpeg/exif file modification experiments
Jpeg/exif file fingerprint compiling

Fig. 2 Research stages

TABLE I
FONT SIZES FOR PAPERS

Jpeg/exif	
segment	**marker**
SOI (Start of Image)	ffd8
APP1 (Application-1)	ffe1
DQT (Define Quantization Tables)	ffdb
SOF0 (Start of Frame-0)	ffc0
DHT (Define Huffman Tables)	ffc4
SOS (Start of Scan)	ffda

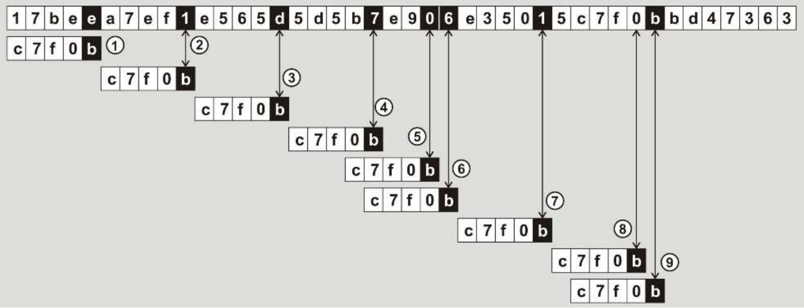

1 7 b e e a 7 e f 1 e 5 6 5 d 5 d 5 b 7 e 9 0 6 e 3 5 0 1 5 c 7 f 0 b b d 4 7 3 6 3
c 7 f 0 b
①
②
③
④
⑤
⑥
⑦
⑧
⑨

Fig. 3 Example of Boyer-Moore string matching algorithm

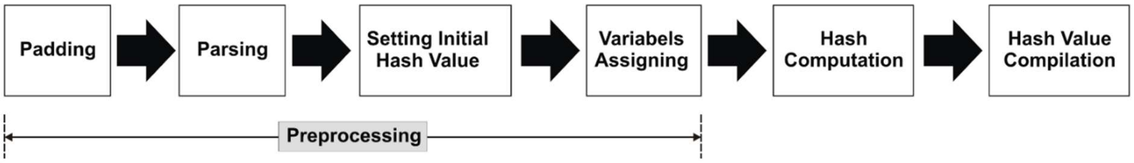


Fig. 4 SHA512 steps computation

The pre-processing steps consist of four processes, padding, parsing, initial hash value setting and variables assigning. The purpose padding is to ensure that the padded message length or input data length is a multiple or 1024 bits. Parsing processes will parse padded data into N blocks of data. If input data have 1024-bits length after padding, then the parsing process will make 1024/64 blocks of data or 16 blocks data [13]. The setting of initial hash value is a process to initialize 8 variables ($H_{(1)}^{0}$, $H_{(2)}^{0}$,… $H_{(8)}^{0}$) and set their values with 8 hash values as shown in Table 2.

The variables assignment is assigning the 8 variables (a,b,c,d,e,f,g,h) with 8 initial hash values as shown in Table 2. Six logic function in equation (2) used to initiate eight registers.

$$a \leftarrow H_{(1)}^{(i-1)} ;\ b \leftarrow H_{(2)}^{(i-1)} ;\ c \leftarrow H_{(3)}^{(i-1)} ;\ d \leftarrow H_{(4)}^{(i-1)} ;\ e \leftarrow H_{(5)}^{(i-1)} ;\ f \leftarrow H_{(6)}^{(i-1)} ;\ g \leftarrow H_{(7)}^{(i-1)} ;\ h \leftarrow H_{(8)}^{(i-1)} \qquad (2)$$

The hash computation start with message block expanding as shown in Figure 5.

The computation from Figure 5, use logic arithmetic operation that shown in equation (3).

$$f(x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \sum 0(x) = S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x) \\ \sum 1(x) = S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x) \\ \sigma_0(x) = S^{1}(x) \oplus S^{8}(x) \oplus R^{7}(x) \\ \sigma_1(x) = S^{19}(x) \oplus S^{61}(x) \oplus R^{6}(x) \end{cases} \quad ..\ (3)$$

The output from hash computation are eight hash values that store in eight registers. Figure 6 shows the eight registers with hash values.

Eight hash values then add with eight initial hash-value from pre-processing's step. The "a" register with H1, the "b" register with H2 and so on. Message digest from the input as final hash value compiles from eight hash value that arranged as one string. Figure X shown SHA512 message digest compiling process from eight registers and eight buffer values.

The final result of SHA512 hash function is a combination of the values of H1 to H8 arranged in a sequence as shown in Figure 8.

Research's third stage is modification experiments that consist of six types file modification, recoloring, image resizing, metadata manipulation, file format conversion, text addition, and image cropping. This third stage has a purpose to identify segments that changed caused by image modification. The last stage is file fingerprint compiling. Jpeg/exif file fingerprint compiling from selected segment hash values.

TABLE II
SHA512 BUFFER VALUE

Buffer	Value
$H_{(1)}^{0}$	6a09e667f3bcc908
$H_{(2)}^{0}$	bb67ae8584caa73b
$H_{(3)}^{0}$	3c6ef372fe94f82b
$H_{(4)}^{0}$	a54ff53a5f1d36f1
$H_{(5)}^{0}$	510e527fade682d1
$H_{(6)}^{0}$	9b05688c2b3e6c1f
$H_{(7)}^{0}$	1f83d9abfb41bd6b
$H_{(8)}^{0}$	5be0cd19137e2179

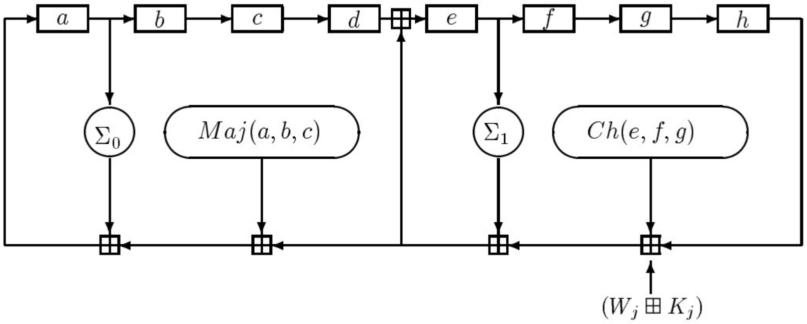


Fig. 5 SHA512 hash computation

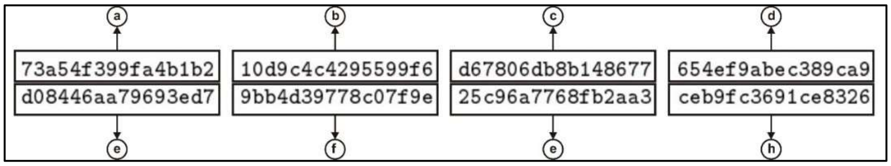
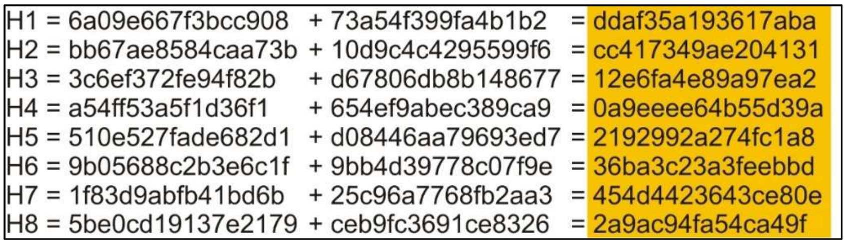
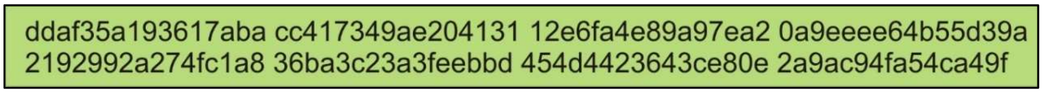


Fig. 6 Eight register with hash values example on SHA512 hash function



```
H1 = 6a09e667f3bcc908  + 73a54f399fa4b1b2  = ddaf35a193617aba
H2 = bb67ae8584caa73b  + 10d9c4c4295599f6  = cc417349ae204131
H3 = 3c6ef372fe94f82b  + d67806db8b148677  = 12e6fa4e89a97ea2
H4 = a54ff53a5f1d36f1  + 654ef9abec389ca9  = 0a9eeee64b55d39a
H5 = 510e527fade682d1  + d08446aa79693ed7  = 2192992a274fc1a8
H6 = 9b05688c2b3e6c1f  + 9bb4d39778c07f9e  = 36ba3c23a3feebbd
H7 = 1f83d9abfb41bd6b  + 25c96a7768fb2aa3  = 454d4423643ce80e
H8 = 5be0cd19137e2179  + ceb9fc3691ce8326  = 2a9ac94fa54ca49f
```

Fig. 7 Buffer values and register values arithmetic adding operation



```
ddaf35a193617aba cc417349ae204131 12e6fa4e89a97ea2 0a9eeee64b55d39a
2192992a274fc1a8 36ba3c23a3feebbd 454d4423643ce80e 2a9ac94fa54ca49f
```

Fig. 8 SHA512 hash value

III. RESULT AND DISCUSSION

A. *Image file structure identification*

Image file structure identification stage conducted by search jpeg/exif segment location. Table 3 show the searching result for jpeg/exif file from smartphone Asus Z00UD.

The segment location data from Table 3 shows that each files have different segment location except for SOI and APP1. The SOI segment location have value as 0 because this segment located in the beginning of image file. The reason of location of APP1 segment for every image files have same value because its SOI segment only contain it segment marker “ffd8” that consist of 4-byte. The location difference between DQT, SOF0, DHT and SOS is caused by the size of data stored in the preceded segment. The DQT location from first file have smaller segment index than second file. This condition shows that APP1 from the first file stored smaller data than APP1 in the second file. The same reason can apply for SOF0, DHT and SOS segments. Table 4 shows the searching result for the jpeg/exif file from the smartphone Samsung Galaxy A5.

The segment location from Table 4 have different result than Table 3. The same segment from each files has the same index. This condition shows the smartphones and digital camera manufacturers have designed each segment to have a fixed size. The result from Table 4 shows that segments location in one file can be used to find out the location of segments in other files.

B. *Segment content acquisition and hashing process*

The acquisition of each segment required two parameters, segment index and segment length. Segment index acquired from previous stage while segment length determined by two segment indexes from two segment as shown in equation (1). Table 5 shows sample of segments length.

TABLE III
SEGMENT INDEX OF ASUS Z00UD JPEG/EXIF FILE

No.	JPEG/Exif file	Segment Index					
		SOI	APP1	DQT	SOF0	DHT	SOS
1	P_20180723_141211	0	4	26400	26844	26726	27630
2	P_20190324_100013	0	4	34212	25544	25584	26490
3	P_20180823_124724	0	4	26378	26664	26704	27610
4	P_20180905_085850	0	4	25350	25636	25676	26580
5	P_20190110_100735	0	4	26318	26602	26642	27548
6	P_20190324_100013	0	4	34212	25544	25584	26490
7	P_20190324_100040	0	4	25378	25662	25704	26608
8	P_20190324_114023	0	4	25278	25769	25809	26713
9	P_20190324_115302	0	4	25348	25789	25792	26713
10	P_20190324_121005	0	4	25405	25663	25564	26580

TABLE IV
SEGMENT INDEX OF SAMSUNG GALAXY A5 JPEG/EXIF FILE

No.	JPEG/Exif file	Segment Index					
		SOI	APP1	DQT	SOF0	DHT	SOS
1	01_20180825_131753	0	4	2000	2286	2326	3218
2	02_20171225_111236	0	4	2000	2286	2326	3218
3	03_20171201_124906	0	4	2000	2286	2326	3218
4	04_20171201_130420	0	4	2000	2286	2326	3218
5	05_20180825_134942	0	4	2000	2286	2326	3218
6	06_20181213_172235	0	4	2000	2286	2326	3218
7	07_20190114_154205	0	4	2000	2286	2326	3218
8	08_20190114_154209	0	4	2000	2286	2326	3218
9	09_20190114_154215	0	4	2000	2286	2326	3218
10	10_20190114_154220	0	4	2000	2286	2326	3218

TABLE V
JPEG/EXIF SEGMENTS LENGTH

No.	File name	Segment Length (bit)					
		SOI	APP1	DQT	SOF0	DHT	SOS
1	P_20180723_141211	4	26396	444	118	904	9181254
2	P_20180823_124724	4	26374	286	40	906	5386020
3	P_20180905_085850	4	25346	286	40	904	5319634
4	P_20190324_100013	4	34208	8668	40	906	3197654
5	P_20190324_100202	4	63584	38180	40	906	5009834
6	01_20180825_131753	4	1996	286	40	892	10805918
7	02_20171225_111236	4	1996	286	40	892	7275684
8	03_20171201_124906	4	1996	286	40	892	6686880
9	04_20171201_130420	4	1996	286	40	892	6598378
10	05_20180825_134942	4	1996	286	40	892	4788372

The SOI segment has a length as 4-bit for each files in Table 5, counted from SOI start index (0) until one index before the APP1 segment index (4-1). APP1 segment for first file in Table 5, has a length as a 26396-bit, counted from APP1 segment index (4) until one index before DQT segment index (26400-4). Those two index values used as a boundary parameter to generate substring for SHA512 hash function input in the third stage. Figure 9 shows segment content acquisition and hashing result from Asus Z00UD jpeg/exif file.

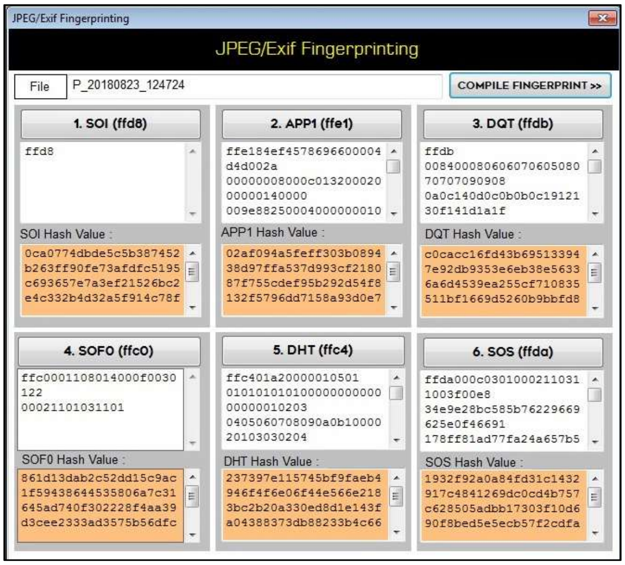
C. *Jpeg/exif file modification experiments*

Six hash values from six jpeg/segments form Figure 9 have the possibility to use as file fingerprint on research fourth stage. File fingerprint selection process purpose is to determined segments that affected if image file altered. Image file modification experiments conduct in six type experiments. Recoloring, resizing and cropping experiments as shows in Figure 10, conducted using ACDSee pro.8 application. Metadata modification conducted using Neo Hexeditor application as shows in Figure 11. The file type conversation from jpeg/exif to png conducted using FormatFactory application and the text addition experiment conducted using Corel Photo Paint application. Table 6 shows the affected segments for each image file modification experiment.

Data fingerprint from jpeg /exif image file determined based on the result from Table 6. The Result categorized into three groups based on the affected segments. The first group shows metadata modification only affected APP1 segment. The second group shows the file convertion and text addition alter the SOI segment. The third group shows recoloring, resizing and cropping affecting others segments except SOI. Based on three group, the element of jpeg/exif data fingerprint consist of three hash values. The first element is SOI hash value and the second element is APP1 hash value. Third component selected from four segments (DQT, SOF0, DHT and SOS). Based on segment's length, the third element is SOF0 hash value. The SOF0 segment have smaller content size than DQT, DHT and SOS. The smaller content's size made the content acquisition and hash computation work faster. Figure 10 shows a jpeg/exif file data fingerprint compiled from three hash values, SOI, APP1 and SOF0

TABLE VI
AFFECTED SEGMENT ON IMAGE MODIFICATION

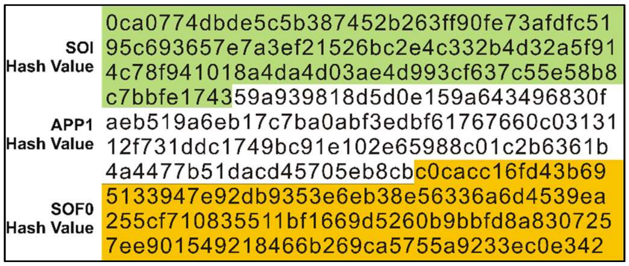
No.	Modification Experiments	Affected Segment					
		SOI	APP1	DQT	SOF0	DHT	SOS
1	Recoloring	-	✓	✓	✓	✓	✓
2	Metadata Modification	-	✓	-	-	-	-
3	Resizing	-	✓	✓	✓	✓	✓
4	Convert to PNG	✓	✓	✓	✓	✓	✓
5	Text addition	✓	✓	✓	✓	✓	✓
6	Cropping	-	✓	✓	✓	✓	✓



File: P_20180823_124724

```
1. SOI (ffd8)
ffd8
SOI Hash Value :
0ca0774dbde5c5b387452
b263ff90fe73afdfc5195
c693657e7a3ef21526bc2
e4c332b4d32a5f914c78f

2. APP1 (ffe1)
ffe184ef4578696600004
d4d002a
00000008000c013200020
00000140000
009e88250004000000010
APP1 Hash Value :
02af094a5feff303b0894
38d97ffa537d993cf2180
87f755cdef95b292d54f8
132f5796dd7158a93d0e7

3. DQT (ffdb)
ffdb
008400080606070605080
70707090908
0a0c140d0c0b0b0c19121
30f141d1a1f
DQT Hash Value :
c0cacc16fd43b69513394
7e92db9353e6eb38e5633
6a6d4539ea255cf710835
511bf1669d5260b9bbfd8

4. SOFO (ffc0)
ffc0001108014000f0030
122
00021101031101
SOF0 Hash Value :
861d13dab2c52dd15c9ac
1f59438644535806a7c31
645ad740f302228f4aa39
d3cee2333ad3575b56dfc

5. DHT (ffc4)
ffc401a20000010501
010101010100000000000
00000010203
0405060708090a0b10000
20103030204
DHT Hash Value :
237397e115745bf9faeb4
946f4f6e06f44e566e218
3bc2b20a330ed8d1e143f
a04388373db88233b4c66

6. SOS (ffda)
ffda000c0301000211031
1003f00e8
34e9e28bc585b76229669
625e0f46691
178ff81ad77fa24a657b5
SOS Hash Value :
1932f92a0a84fd31c1432
917c4841269dc0cd4b757
c628505adbb17303f10d6
90f8bed5e5ecb57f2cdfa
```

Fig. 9 Segment content and hash values from jpeg/exif segments



```
                 0ca0774dbde5c5b387452b263ff90fe73afdfc51
            SOI  95c693657e7a3ef21526bc2e4c332b4d32a5f91
     Hash Value  4c78f941018a4da4d03ae4d993cf637c55e58b8
                 c7bbfe17435 9a939818d5d0e159a643496830f
           APP1  aeb519a6eb17c7ba0abf3edbf61767660c03131
     Hash Value  12f731ddc1749bc91e102e65988c01c2b6361b
                 4a4477b51dacd45705eb8cbc0cacc16fd43b69
           SOF0  5133947e92db9353e6eb38e56336a6d4539ea
     Hash Value  255cf710835511bf1669d5260b9bbfd8a830725
                 7ee901549218466b269ca5755a9233ec0e342
```

Fig. 10 Jpeg/exif file fingerprint

IV. CONCLUSION

The aspect of data integrity detection for jpeg/exif file accomplished by using data fingerprint in form of hash value. BM string matching algorithm have role to identifying segment marker for each jpeg/exif segments. BM string matching have two usage method. First method, BM string matching used to search segment marker for each files such as image files from smartphone Asus Z00UD. Second method, BM string matching used once to search segment marker for image files that have fixed size of segment, such as image files from smartphone Samsung Galaxy A5. The acquisition and hashing computation using SHA512 hash function required time that linear with content sized. The bigger content size will require longer time to collect segment content and compute the hash value. The Jpeg/exif data fingerprint compiled from three hash value. SOI hash value used for detect file type conversion and text addition modification. The APP1 hash value used for detect metadata modification and the SOF0 hash value used for detect modified images by recoloring, resizing and cropping.

REFERENCES

[1] S. Park, A. B. Ruighaver, S. B. Maynard, and A. Ahmad, “Towards understanding deterrence: Information security managers’ perspective,” *Lect. Notes Electr. Eng.*, vol. 120 LNEE, no. January, pp. 21–37, 2012.

[2] H. Wijayanto, I. Riadi, and Y. Prayudi, “Encryption EXIF Metadata for Protection Photographic Image of Copyright Piracy,” *Int. J. Res. Comput. Commun. Technol.*, vol. 5, no. 5, 2016.

[3] D. P. Gangwar and A. Pathania, “Authentication of Digital Image using Exif Metadata and Decoding Properties,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3, no. January, pp. 335–341, 2019.

[4] L. Refialy, E. Sediyono, and A. Setiawan, “Pengamanan Sertifikat Tanah Digital Menggunakan Digital Signature SHA-512 dan,” *JUTISI*, vol. 1, pp. 229–234, 2015.

[5] W. Stallings, *Cryptography and Network Security Principles and Practice*, 6th ed. New Jersey: Pearson Education Inc., 2014.

[6] H. A. Chaudhary, “Process , Application and Authenticity of Digital Signature,” *Int. J. Sci. Res. Eng. Technol.*, vol. 6, no. 8, pp. 882–888, 2017.

[7] U. Abubakar Idris, J. Awwalu, and B. kamil, “User authentication in securing communication using Digital Certificate and public key infrastructure,” *Int. J. Comput. Trends Technol.*, vol. 37, no. 1, 2016.

[8] D. Bansal, M. Sharma, and A. Mishra, “Analysis of Digital Signature based Algorithm for Authentication and Privacy in Digital Data,” *Int. J. Comput. Appl.*, vol. 161, no. 5, pp. 43–45, 2017.

[9] K. Al-Khamaiseh and S. Al-Shagarin, “A Survey of String Matching Algorithms,” *Int. J. Eng. Res. Appl.*, vol. 4, no. June 2015, pp. 144–156, 2014.

[10] D. R. Candra and K. D. Tania, “Application of Knowledge Sharing Features Using the algorithm Boyer-moore On Knowledge Management System (KMS),” *J. Sist. Inf.*, vol. 9, no. 1, pp. 1216–1221, 2017.

[11] Y. D. Prabowo, “Pencocokan DNA NR_108049 dan DNA DI203322 Menggunakan Algoritma Boyer Moore,” *Pros. Semin. Nas. Teknol. Inf. dan Komun.*, no. 40, pp. 18–19, 2016.

[12] V. Roussev, “Hashing and Data Fingerprinting in Digital Forensics,” *IEE Secur. Priv.*, no. April, pp. 49–55, 2009.

[13] NIST, *FIPS PUB 180-4 Secure Hash Standard (SHS)*, no. August. Gaithersburg: National Institute of Standards and Technology, 2015.

[14] I. Riadi and M. Sumagita, “Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application,” *Int. J. Cyber-Security Digit.*

Forensics, vol. 7, no. 4, 2018.

[15] V. Roussev, “An Evaluation of Forensic Similarity Hashes Vassil Roussev An evaluation of forensic similarity hashes,” *Proc. Digit. Forencsic Res. Conf.*, pp. s34–s41, 2011.

[16] A. L. . Sandoval, D. M. . Gonzales, L. J. . Villaba, and J. Hernandez-Castro, “Analysis of errors in exif metadata on mobile devices,” *Multimed Tools Appl*, no. 74, pp. 4735–4763, 2015.