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

Rachmad Fitriyanto¹, Anton Yudhana², Sunardi³

¹Information System Departmen, STMIK PPKIA Tarakanita Rahmawati Tarakan, Indonesia
²Electrical Engineering Department, Universitas Ahmad Dahlan, Indonesia
Email: ¹rachmad7477@gmail.com, ²yudhana@ee.uad.ac.id, ³sunardi@mti.uad.ac.id

Abstract

Data integrity, authenticity and non-repudiation are security parameters provided by Digital Signature Algorithm (DSA). The hash value is an important element inside DSA to identify information data integrity using the hash function to generate a message digest. JPEG/Exif is an image file format that produces by a digital camera as in smartphone. Hardware technology development made the image file has a higher resolution than before. This condition made image file fingerprinting need more time to compile JPEG/Exif fingerprint. This research purpose is to develop a fingerprinting process for JPEG/Exif files using the Boyer-Moore string matching algorithm and SHA512. Research conducted in four stages. The first stage is JPEG/Exif file structure identification, the second stage is segmented content acquisition and hashing, the third stage is image file modification experiments and the last stage is JPEG/Exif file fingerprint compilation. The obtained result showed that the JPEG/Exif file fingerprint comprises three hash values from the SOI segment, APP1 segment, and SOF0 segment. The JPEG/Exif file fingerprint can use to detect 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

1. INTRODUCTION

Image files and videos are two types of files used in many ways communication. Both image and video files produced by digital camera usage such as in smartphones. The image file that produces has a format file as JPEG/Exif. JPEG/Exif file format structured from a few parts that each part store specific information about image [1]. The information security must be provided to ensure that the information received is in the same condition as when it was sent. The information must also arrive at the right recipient. The recipients of information must also be able to ensure that the information received comes from senders he or she knows.

Digital Signature Algorithm (DSA) is a cryptographic method that used to full-fill information security parameters [2]. DSA uses three elements to secure information, data fingerprint, asymmetric key-pair and digital certificate [3]. Data fingerprint used to secure information data integrity. Asymmetric key-pair used to identify the right full information owner and receiver. The digital certificate used to provide non-repudiation before the law [4]. DSA performed on both sides, information sender and receiver as shown in Figure 1.
Figure 1. Digital Signature Algorithm Processes

The first process until third in Figure 1 shows the information fingerprint generating process. The fourth process is information fingerprint encryption with the asymmetric key pair. The fifth process is embedding ciphertext from the previous process into information [5]. DSA processes on the receiver side have similarities with the sender side. Start with generating fingerprint, extracting ciphertext, decrypting ciphertext and verifying two information fingerprints.

The fingerprint of information is the most important element among DSA elements. The fingerprint generating process also called fingerprinting, starting with identity information structure. The identifications result are location data segments representing by segment start index value. Segments index values used as boundary parameters for segment content acquisition in DSA second process and twelfth process.

A segment identifying in DSA conduct by use a string matching algorithm. String matching algorithms categorized into two categories based on comparative methods, exact string matching and approximate string matching [6]. Boyer-Moore (BM) string matching algorithm includes exact string matching categories. BM string matching starts to compare character from the most-right element of the pattern then shift to left. Figure 2 shown the BM searching process.
BM's first step from Figure 2 is comparing the fifth element of pattern to the fifth element of the string. If not match then compare the fourth element of pattern to the fifth element of the string. If comparison has reached to first pattern element and still not match then the pattern will slide as far as pattern length as shown in the second step in Figure 2. Step 2, 3 and 4 have the same process as step 1. Step 5 has a condition called the Good-Character Rule that allows aligning patterns. This happens when the matching character found for the first time after not match condition. Pattern aligning occurred in steps 6 and 8.

Pattern location that has found used as a parameter to generate sub-string for hashing process on DSA third and thirteenth process. The hashing process is a method to generate message digest by executed hash function [7]. The Secure Hash Algorithm (SHA) is a cryptographic hash function developed by the National Institute of Standards and Technology (NIST) [8]. 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 a different hashing process and different output size. SHA512 have 512-bit message digest. SHA512 output size makes this SHA variant have better performance than its predecessor [9].

SHA512 has two main processes, padding, and hashing. Padding is adding input data bit to form data which has 1024 bit block length. The hashing process comprises three steps. Step one is buffer value assignment, step two is register initiation and step three is message block expanding. Buffer value for SHA512 hash function shown as in Table 1.

| Buffer | Value     |
|--------|-----------|
| H(1)0  | 6a09e667f3bce908 |
| H(2)0  | bb67ae8584caa73b  |
| H(3)0  | 3c6ef372fe94f82b  |
| H(4)0  | a54ff53a5f1d36f1  |
| H(5)0  | 510e527faded82d1  |
| H(6)0  | 9b05688c2b3e6c1f  |
| H(7)0  | 1f83d9ab41bd6b    |
| H(8)0  | 5be0cd19137e2179  |
Register initiation is assigning buffer value into each eight register \(a,b,c,d,e,f,g,h\). Six logic function in equation 1 used to initiate eight registers.

\[
\begin{align*}
    a &\leftarrow H^{(i-1)}_{(1)}; \\
    b &\leftarrow H^{(i-1)}_{(2)}; \\
    c &\leftarrow H^{(i-1)}_{(3)}; \\
    d &\leftarrow H^{(i-1)}_{(4)}; \\
    e &\leftarrow H^{(i-1)}_{(5)}; \\
    f &\leftarrow H^{(i-1)}_{(6)}; \\
    g &\leftarrow H^{(i-1)}_{(7)}; \\
    h &\leftarrow H^{(i-1)}_{(8)}
\end{align*}
\]

After eight registers initiated, message block expanding executed as shown in Figure 3.

![Figure 3. SHA512 message block expanding process](image)

Message block expanding use logic arithmetic operation that shown in equation 2.

\[
f(x,y,z) = \begin{cases} 
    Ch(x,y,z) & = (x \land y) \oplus (\neg x \land z) \\
    Maj(x,y,z) & = (x \land y) \oplus (x \land z) \oplus (y \land z) \\
    \sum 0(x) & = S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x) \\
    \sum 1(x) & = S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x) \\
    \sigma_0(x) & = S^1(x) \oplus S^9(x) \oplus R^7(x) \\
    \sigma_1(x) & = S^{19}(x) \oplus S^{61}(x) \oplus R^6(x)
\end{cases}
\]

The output from message block expanding are eight hash value that store in eight registers. Figure 4 shown eight registers with hash values for string input “abc”.

![Figure 4. Eight registers with hash values example on SHA512 hash function](image)

Eight hash values then add with eight buffer value. 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 5 shown the SHA512 message digest compiling process from eight registers and eight buffer values.
Message digest as the final result of SHA-512 is a combination of the values of H1 to H8 arranged in a series as shown in Figure 6.

The time required for identifying the information segment is linear with information data size. Digital camera technology development has given higher image quality that causes an increase in image file size. The bigger size of the image file becomes a problem to execute the fingerprinting process. The focus of this research is to develop the fingerprinting method for JPEG/Exif files using the Boyer-Moore string matching algorithm and SHA512.

2. METHODS

The research study conducted in four stages shown in Figure 7. The first stage is the image file segment identification. This stage has the purpose of identification JPEG/Exif file structure. Image files as research object acquired from two smartphone types, Asus Z00UD and Samsung Galaxy A5. The result from the first stage is the location index of each file parts. This location index will use on the second stage as a parameter to identify the beginning and end of file parts.

Second stage is segment content acquisition and hashing process. Segment content acquisition conducted by duplicating segment content based location index file parts from first stage. Hashing process conducted by executing of SHA512 with segment content as input.

Research’s third stage is modification experiments that consist six types file modification, recoloring, image resizing, metadata manipulation, file format conversion, text addition and image cropping. This third stage have purpose to identified segments that changed caused by image modification. The last stage is file fingerprint compiling. JPEG/Exif file fingerprint compiling from selected segment hash values.
3. RESULTS AND DISCUSSION

JPEG/Exif file structure consists of seven segments. Each segment has a segment marker that has a function to identify the beginning of the segment and the end of the previous segment. JPEG/Exif file segments and segment markers are shown in Table 2.

| Segment | Marker |
|---------|--------|
| SOI     | ffd8   |
| APP1    | ffe1   |
| DQT     | ff     |
| SOF0    | ffc0   |
| DHT     | ffc4   |
| SOS     | ffd4   |

JPEG/Exif file segment searches using the Boyer-Moore string matching algorithm with a segment marker as an input pattern. Figure 8 shown the Boyer-Moore string matching algorithm flowchart. Boyer-Moore string matching algorithm compares character from the rightmost character on input pattern and shifts to left until reaching the leftmost character from a pattern. The searching process stopped when all characters have found in the sequence.

Figure 8. Boyer-Moore string matching algorithm flowchart

Segment searching result is shown in Table 2 for JPEG/Exif file from smartphone Asus Z00UD and Table 3 from Samsung Galaxy A5.
Table 3. 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_20190905_085850 | 0    | 4     | 25350 | 25636 | 25676 | 26580 |
| 5   | P_20190110_100735 | 0    | 4     | 26318 | 26602 | 26642 | 27548 |

Table 4. Segment Index of Samsung Galaxy A5 JPEG/Exif File 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  |

Segment content length determined by two indexes from the two-segment start index. SOI content has the length as 4 bit from SOI start index (0) until one index before the APP1 segment start index (4-1). APP1 content has the length as 1997 bit from APP1 start index (4) until one index before the DQT segment start index (2000-1). Those two index values used as a parameter to generate substring for SHA512 hash function input in the third stage. Figure 9 shown hashing result from Asus Z00UD JPEG/Exif file.

Figure 9. Hash values from JPEG/Exif segments
Six hash values from six jpeg/segments form Figure 9 can use as file fingerprint on the research fourth stage. File fingerprint selection process purposes are to determine segments that affected if the image file altered. Image file modification experiments conduct in six experiments as describe in section 2 (methods). Table 5 shown affected segments for each experiment.

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

The result from Table 3 categorized into three groups. The first group showed the metadata modification experiment affected the APP1 segment only. The second group showed that the SOI segment altered for image file conversion and text/object addition experiments. The third group shown all segments altered for image display modification. Each result group used as file fingerprint components. The first component is SOI hash value and the second component is APP1 hash value. Third component selected from four segments (DQT, SOF0, DHT, SOS) based on segment content and segment length. Table 6 shown segments length from JPEG/Exif image file that counted use segment index from research first stage result.

| 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 40 906 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   |

SOS segment has the most length content from other segments. This condition made the segment marker searching process need more time because the Boyer-Moore string matching algorithm has O(mn) [10] time complexity for the worst-case condition. SOF0 has the smallest size than others except for the SOI segment. SOF0 store image information such as image dimension and the number of color components that always change if image altered as in 1st, 2nd, 5th and 6th experiments. Figure 10 shown the JPEG/Exif file fingerprint that arranged from three hash values, SOI, APP1, and SOF0.
4. CONCLUSION

The compilation of fingerprint files for JPEG/Exif files is conducted by identifying three segments location (SOI, APP1, SOFO) using the Boyer-Moore string matching algorithm. Content from all three segments is used as input for hash function SHA512. The selection of the three hash values determined based on the information inside segments and the size of the segment. Segment size affects the length of time needed to identify the location of the segment and compile the hash value.

5. REFERENCES

[1] Wijayanto, H., Riadi, I., & Pryudi, Y. (2016). Encryption EXIF metadata for protection photographic image of copyright piracy. International Journal of Research in Computer and Communication Technology, 5(5).
[2] Stallings, W., & Brown, L. (2015). Computer Security Principles and Practice (Third Edit). Pearson Education Inc.
[3] Chaudhary, H. A. (2017). Process, application and authenticity of digital signature. International Journal of Scientific Research Engineering & Technology (IJSRET), 6(8), 882–888.
[4] Refialy, L., Sedyono, E., & Setiawan, A. (2015). Pengamanan sertifikat tanah digital menggunakan digital signature SHA-512 dan RSA. Jurnal Teknik Informatika dan Sistem Informasi, 1(3).
[5] Bansal, D., Sharma, M., & Mishra, A. (2017). Analysis of digital signature based algorithm for authentication and privacy in digital data. International Journal of Computer Applications, 161(5), 43–45.
[6] Jiji, N., & Mahalaksmi, T. (2018). An efficient string matching algorithm for detecting pattern using forward and backward searching approach. International Journal of Computer Science, 6(2), 16–26.
[7] Almazrooie, M., Samsudin, A., Gutub, A. A. A., Salleh, M. S., Omar, M. A., & Hassan, S. A. (2020). Integrity verification for digital Holy Quran verses using cryptographic hash function and compression. Journal of King Saud University-Computer and Information Sciences, 32(1), 24-34.
[8] NIST. (2015). FIPS PUB 180-4 Secure Hash Standard (SHS) (Issue August).
[9] Riadi, I., & Sumagita, M. (2018). Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application. International Journal of Cyber-Security and Digital Forensics, 7(4).
[10] Al-Khamaisheh, K., & ALShtagrin, S. (2014). A survey of string matching algorithms. International Journal of Engineering Research and Applications, 4(7), 144-156.

Figure 10. JPEG/Exif file fingerprint