IDENTIFICATION AND RESTORATION OF LZ77 COMPRESSED DATA USING A MACHINE LEARNING APPROACH

Beom Kwon, Myongsik Gong, Jungwoo Huh, and Sanghoon Lee
Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea
E-mail: {hsm260, audlr24, gjwjddn9, slee}@yonsei.ac.kr, Tel: +82-2-2123-2767

***Abstract*—Identifying the type of a codec that used to compress data is essential in digital forensics since many trials and errors required to restore data can be reduced. Nevertheless, most compression algorithms have been configured by using several parameters whose values can be different according to each user. Therefore, in order to restore data more effectively, the values of parameters as well as the type of the codec must be identified. In this paper, we present an identification and restoration method for Lempel-Ziv-77 (LZ77) compressed data. In the proposed method, we identify whether a given data is compressed by LZ77 or not. Moreover, we estimate the values of parameters that were used for compression. Using the estimated parameters, we restore the original data from the LZ77 compressed data. The simulation results demonstrate the feasibility and effectiveness of the proposed method with a successful compression identification and parameter estimation accuracies of 100% and 84.41%.**

I. Introduction

Digital forensics is a scientific investigation technique that recovers and analyzes digital evidence found in digital devices. Recently, digital forensics has received increasing attention due to the proliferation of digital crime. Since digital criminals hide their criminal activities in a vast digital data, digital forensic investigators face the task of finding digital evidences from the vast digital sources. Moreover, the criminals sometimes change or delete the header information of data to obstruct the investigations. The header information of a data contains information pertaining to the type of codec used to compress the data. If there is no knowledge of the type of the codec, it is difficult to restore the original data from the compressed data. Therefore, identifying the type of codec may be useful in digital forensics because many trials and errors required to restore the data can be reduced.

However, most compression algorithms have several parameters whose values can be changed according to each user. Therefore, in order to restore data more reliably, the values of the parameters as well as the type of the codec must be estimated. However, most studies for codec identification and parameter estimation have been aimed at audio, speech, image and video files [1], [2]. Moreover, since lossy compression algorithms are used in audio, speech, and video compression, the studies above have been focused on lossy compression algorithms. To the best of our knowledge, little has been studied for codec identification and parameter estimation in text and data files where lossless compression algorithms are used.

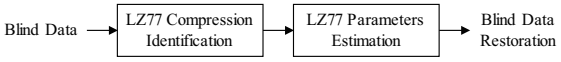


Fig. 1: Block diagram for the identification and restoration of blind data.

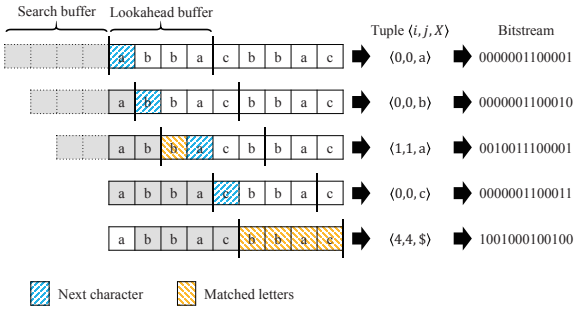


Fig. 2: Compression process of LZ77 when the input stream is "abbacbbac" and the sizes of the search and lookahead buffers are 4 and 4, respectively.

Since the original text and data files must be restorable without any loss of information, lossless compression algorithms are used to compress them. Lempel-Ziv-77 (LZ77) proposed in [3] is one of the most popular algorithms used in lossless compression. In this paper, we present an identification and restoration method for LZ77 compressed data. In the proposed method, via the byte frequency analysis, we identify whether a given data is compressed by LZ77 or not. Then, we estimate the parameters of LZ77 using both the frequency and runs tests. In addition, using the estimated parameters, we restore the original data from the LZ77 compressed data. The simple block diagram of the process above is shown in Fig. 1.

The remainder of this paper is organized as follows. In Section II, we explain LZ77 with a simple example. In Section III, we propose a LZ77 compression identification method. A method for estimation of parameters used in LZ77 is described in Section IV. The results of our experiments are given in Section V. Finally, we conclude our paper in Section VI.

II. Lempel-Ziv-77

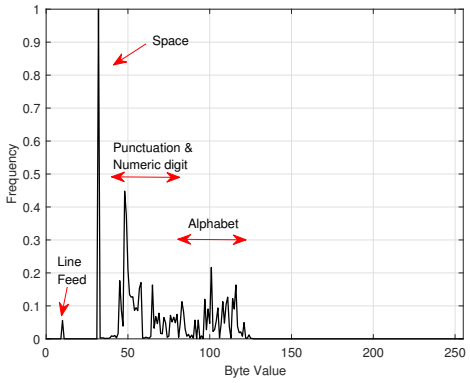
Fig. 2 represents the compression process of LZ77. As illustrated in the figure, LZ77 compresses the input stream using a sliding window that consists of a search buffer and a lookahead buffer. First, LZ77 finds the longest matching length of letters stored in the lookahead and search buffers. The matched letters are encoded as tuple $\langle i, j, X\rangle$, where i signifies the distance between the start of the matched letters in the search buffer and the end of the search buffer, j represents the number of matched letters, and X is the next letter after the matched letters in the lookahead buffer. If there are no matched letters in the search and lookahead buffers, LZ77 outputs $\langle 0, 0, X\rangle$. In this case, X is the first letter in the lookahead buffer. After LZ77 outputs a tuple, the sliding window moves $j+1$ blocks forwards. In the figure, the sliding window moves from left to right.

If there are no further letters to be compressed, LZ77 outputs $\langle i, j, \$\rangle$. Here, $ plays a role in instructing that there are no further letters to be decoded in the decompression process of LZ77. After the compression process is completed, LZ77 converts each tuple into binary form. Let S and L be the sizes of the search and lookahead buffers, respectively. The length of the binary stream for each tuple depends on the value of S. Let B be the length of the binary stream of both i and j. Then, the value of B is determined by the following condition: $2^{B-1} \leq S < 2^B$. On the other hand, X is encoded with 8 bits based on American Standard Code for Information Interchange (ASCII). Therefore, the length of the binary stream for each tuple can be calculated as $(2 \times B + 8)$ bits.

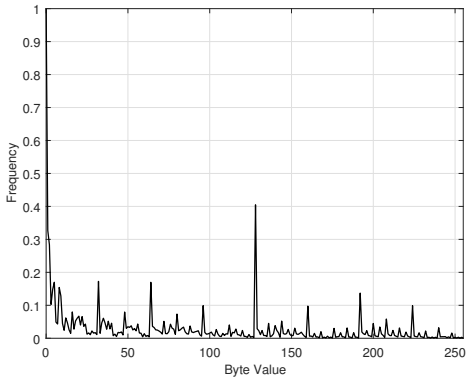
III. LZ77 Compression Identification

In this section, we present the method for LZ77 compression identification. In the proposed method, the identification is performed using a criterion that is designed based on the byte frequency distributions of LZ77 compressed data and uncompressed data. Byte frequency distribution of a bitstream is obtained using the byte frequency analysis (BFA) proposed in [4]. In order to help the understanding for the BFA, we briefly explain it.

Byte Frequency Analysis: the purpose of the BFA is to obtain the byte frequency distribution of an input bitstream. The distribution is obtained by the following procedures. First, an array that is indexed from 0 to 255 is constructed. Then, all elements of the array are initialized with zeros. Next, the byte value is calculated at every 8 bits for the input bitstream. Here, the byte value is defined as the decimal value of 8 bits read from the bitstream. The byte that consists of eight bits is capable of representing the decimal values from 0 to 255. Therefore, the byte value also has a range of 0 to 255. For example, if the byte value of 8 bits read is 17, the 17^{th} element of the array is incremented by one. By incrementing the corresponding element in the array according to the occurrence of the byte value, the byte frequency distribution of the input bitstream can be obtained. After the process above is complete, for normalization, each element in the array is divided by the number of occurrences of the most frequent byte value.



(a) Uncompressed data



(b) LZ77 compressed data ($S = 1024$)

Fig. 3: Byte frequency distributions for uncompressed data and LZ77 compressed data.

Fig. 3 shows the byte frequency distributions for uncompressed data and LZ77 compressed data. As shown in Fig. 3(a), for uncompressed data, the occurrences of the byte values are concentrated in the range of 12 to 129. By the ASCII chart, the byte values in this range include line feed, space, punctuation marks, digits, and Alphabet, which are widely used in text and data files. By contrast, for LZ77 compressed data, the occurrences of byte values are distributed from 0 to 255 as shown in Fig. 3(b). In addition, a large spike at the byte value of 0 is observed. This means that the LZ77 compressed data has many regions that are filled with the byte value of zero.

From the observation between the two distributions, we design the criterion that indentifies whether the input bitstrem is LZ77 compressed data or uncompressed data. Let $d_l(\beta)$ be the number of occurrences of the byte value of l in the bitstream β, $l \in \{0, 1, \cdots, 255\}$, and $D(\beta)$ be the criteria value for LZ77 compression identification. Then, in this paper, $D(\beta)$ is defined as

$$D(\beta) = \frac{\sum_{l=12}^{129} d_l(\beta)}{\sum_{l=0}^{255} d_l(\beta)}. \quad (1)$$

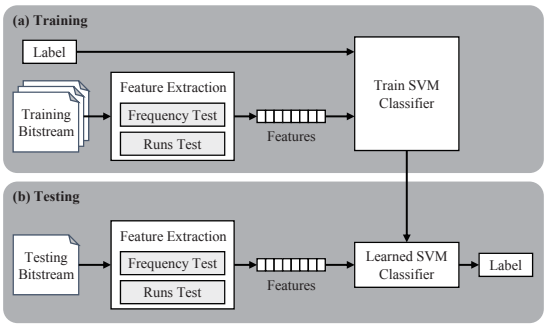


Fig. 4: Block diagram for LZ77 parameter estimation.

TABLE I: Compression results using LZ77 for WikiLeaks.

Type and Parameter		Size (bytes)
Uncompressed data		45,157,523
Compressed data	$S = 2$	54,253,018
	$S = 4$	57,504,782
	$S = 8$	60,829,608
	$S = 16$	64,874,987
	$S = 32$	68,368,800
	$S = 64$	68,156,342
	$S = 128$	60,510,512
	$S = 256$	51,695,414
	$S = 512$	46,734,962
	$S = 1024$	43,747,343
	$S = 2048$	40,782,446
	$S = 4096$	37,426,815
	$S = 8192$	35,201,094

Since the occurrences of the byte values of uncompressed data are concentrated in the range of 12 to 129, $D(\beta)$ for uncompressed data is close to 1. On the other hand, for LZ77 compressed data, $D(\beta)$ is greater than 0 and less than 1. From this observation, we conclude that $D(\beta)$ can be used to the criteria value for LZ77 compression identification. For identification, we adopt the threshold α. In this paper, the value of α is set to 0.9. Finally, if $D(\beta) < \alpha$, the bitstrem β is identified as LZ77 compressed data. If $D(\beta) \geq \alpha$, the bitstrem β is identified as uncompressed data.

IV. LZ77 Parameter Estimation

If a given data is identified as LZ77 compressed data, LZ77 parameter estimation is performed. As we described in Section II, LZ77 has the two parameters: S and L. If there are no knowledge about the values of S and L, it is difficult to restore the original data from the LZ77 compressed data identified. In order to resolve this problem, we propose a method for LZ77 parameter estimation. Towards this goal, we apply two statistical tests proposed in [5] to extract features from the input bitstream. In addition, we use support vector machine (SVM) as a classifier. We train the SVM classifier using the extracted features and the label information for the parameters. Finally, we employ the learned SVM classifier to estimate the parameters that were used to compress the original data. Fig. 4 shows the block diagram for the LZ77 parameter estimation. The two statistical tests used for feature extraction are the frequency and runs tests. In order to help the understanding for the two tests, we briefly review each test.

Frequency Test: this test focuses on the proportion of zeros and ones in the input bitstream. This test is based on the hypothesis that the proportion of zeros and ones may vary according to the values of S and L. The output (feature) of the frequency test is obtained by the following procedures. First, the zeros in the input bitstream are converted into the values of -1. In addition, the ones in the bitstream are converted into the values of $+1$. After the conversion is complete, these values are summed. Let T_F be the sum of the values. Then, T_F is divided by the square root of the length of the input bitstream. Finally, the result value is outputted as a feature. For example, if the bitstream "1110" is inputted, "1110" is converted as "$(+1)(+1)(+1)(-1)$" and T_F is calculated as $T_F = 1+1+1-1 = 2$. Then, $T_F = 2$ is divided by $\sqrt{4} = 2$. Finally, the result value of 1 is outputted.

Runs Test: this test focuses on the total number of runs in the input bitstream. Here, a run means an uniterrupted sequence of identical bits. This test is based on the hypothesis that the total number of runs may vary according to the values of S and L. For ease of explanation, let n be the length of the input bitstream and $\varepsilon = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n]$ be a vector representation of the input bitstream, where ε_i is the i^{th} bit of the input bitstream. In addition, let $R(\varepsilon)$ be the total number of runs for ε. Then, $R(\varepsilon)$ is given by

$$R(\varepsilon) = \sum_{k=1}^{n-1} r(k) + 1, \tag{2}$$

where $r(k) = 0, 1$ for $\varepsilon_k = \varepsilon_{k+1}$ and $\varepsilon_k \neq \varepsilon_{k+1}$, respectivley. The output (feature) of the runs test is obtained by the following procedures. First, $R(\varepsilon)$ for the input stream ε consisting of n bits is calculated. Then, $R(\varepsilon)$ is divided by $\sqrt{n}$, and then the result value is outputted as a feature. For example, if $\varepsilon = 111010$, then $n = 6$. $R(\varepsilon)$ is calculated as $R(\varepsilon) = 0+0+1+1+1+1 = 4$, and then $R(\varepsilon)$ is divided by $\sqrt{n} = \sqrt{6}$. Finally, the result value of $4/\sqrt{6}$ is outputted.

V. Simulation Results

In this section, we describe the results of our experiments. To validate the proposed method, we use text files from the publicly available database of WikiLeaks[1]. This database contains 1,440 text files. By referring to [6], each text file is compressed individually with LZ77. During the compression, the value of L is set to the value of S (i.e, L=S). The compression results are presented in Table I. As shown in the table, the LZ77 compression efficiency is not good for $S = 2, 4, 8, 16, 32, 64, 128, 256, 512$. It means that the sizes of the lookahead and search buffers are not sufficient to compress the files. Therefore, in this paper, we utilize the compression results with $S = 1024, 2048, 4096, 8192$.

[1]https://911.wikileaks.org/files/

TABLE II: Compression identification accuracy.

S	1024	2048	4096	8192
True Positive	100	100	100	100
False Positive	100	100	100	100

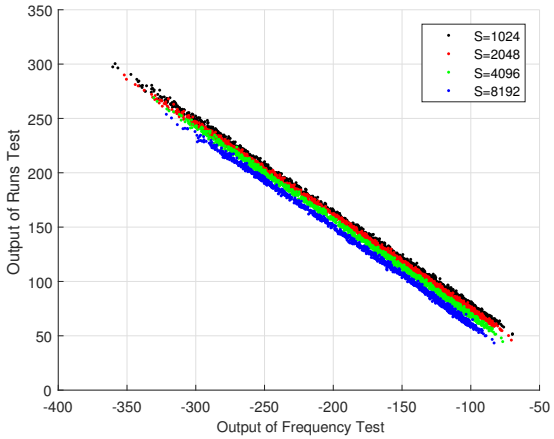


Fig. 5: Scatter plot of $S = 1024, 2048, 4096, 8192$ classes in a two dimensional feature space.

Table II shows the LZ77 compression identification accuracy of the proposed method. As shown in the table, the identification accuracy of the proposed method for uncompressed data and LZ77 compressed data are always 100%.

Fig. 5 shows the two dimensional feature space resulting from the frequency and runs tests. As shown in the figure, each class is well separated from each other. Table III shows the confusion matrix for the LZ77 parameter estimation of our method. The diagonal elements in the confusion matrix represent the correctly estimated rates for LZ77 parameters. The average estimation accuracy of the proposed method was approximately 84.41%.

VI. Conclusion and Future Work

In this paper, we proposed the method for identification and restoration of LZ77 compressed data. When a data is given, the method firstly identifies whether the data is LZ77 compressed data or uncompressed data. This identification process is performed based on the two statistical tests. If the given data is identified as LZ77 compressed data, the method estimates the parameters that were used to compress the original data. This estimation process is performed based on the byte frequency analysis. Although the experimental results were promising, there are several issues that need to be solved in future work. First, the feasibility of the LZ77 compression identification method needs to be validated including additional compression algorithms, such as Lempel-Ziv-Storer-Szymanski (LZSS) [7]. The current results were obtained under conditions that there are LZ77 compressed data and uncompressed data only. Second, the current version

TABLE III: Confusion matrix for the parameter estimation.

Type	ID	1	2	3	4
$S = 1024$	1	**81.25**	18.47	0.28	0.00
$S = 2048$	2	20.28	**72.64**	7.08	0.00
$S = 4096$	3	0.97	10.42	**86.25**	2.36
$S = 8192$	4	0.28	0.00	2.22	**97.50**

of the LZ77 parameter estimation is performed under the assumption that the values of S and L are the same. However, for generalization, the current method needs to be extended to cover the cases where the values of S and L are different. Third, the current method is performed under the assumption that there are no bit errors in the LZ77 compressed data. However, in practical application, bit errors can occur in LZ77 compressed data for a varierty of reasons, such as bad communication channel environment [8]–[14].

Acknowledgment

This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

References

[1] S. H. Lee, J. Kim, and S. Lee, "An identification framework for print-scan books in a large database," *Information Sciences*, vol. 396, pp. 33–54, Aug. 2017.
[2] D. Kim *et al*., "Robust fingerprinting method for webtoon identification in large-scale databases," *IEEE Access*, vol. 6, pp. 37932–37946, 2018.
[3] J. Ziv, and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Information Theory,* vol. 23, no. 3, pp. 337–343, May 1997.
[4] M. McDaniel and M. H. Heydari, "Content based file type detection algorithms," in *Proc. IEEE Hawaii Int. Conf. Syst. Sci.*, Jan. 2003, pp. 1–10.
[5] A. Rukhin *et al*., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology Special Publication 800-22, pp. 1–131, Apr. 2001.
[6] T. Bell, "Better OPM/L text compression," *IEEE Trans. Commun.*, vol. 34, no. 12, pp. 1176–1182, Dec. 1986.
[7] B. Kwon, M. Gong, and S. Lee, "Novel error detection algorithm for LZSS compressed data," *IEEE Access,* vol. 5, pp. 8940–8947, May 2017.
[8] B. Kwon, J. Park, and S. Lee, "Virtual MIMO broadcasting transceiver design for multi-hop relay networks," *Digit. Signal Process.*, vol. 46, pp. 97–107, Nov. 2015.
[9] B. Kwon *et al*., "A downlink power control algorithm for long-term energy efficiency of small cell network," *Wireless Netw.*, vol. 21, no. 7, pp. 2223–2236, Oct. 2015.
[10] B. Kwon, J. Park, and S. Lee, "A target position decision algorithm based on analysis of path departure for an autonomous path keeping system," *Wireless Personal Commun.*, vol. 83, no. 3, pp. 1843–1865, Aug. 2015.
[11] B. Kwon *et al*., "Iterative interference cancellation and channel estimation in evolved multimedia broadcast multicast system using filter-bank multicarrier-quadrature amplitude modulation," *IEEE Trans. Broadcast.*, vol. 62, no. 4, pp. 864–875, Dec. 2016.
[12] B. Kwon and S. Lee, "Cross-antenna interference cancellation and channel estimation for MISO-FBMC/QAM-based eMBMS," *Wireless Netw.*, pp. 1–13, 2017.
[13] B. Kwon and S. Lee, "Effective interference nulling virtual MIMO broadcasting transceiver for multiple relaying," *IEEE Access*, vol. 5, pp. 20695–20706, Oct. 2017.
[14] B. Kwon, S. Kim, and S. Lee, "Scattered Reference Symbol-Based Channel Estimation and Equalization for FBMC-QAM Systems," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3522–3537, Aug. 2017.