Rapid Detection of Stego Images Based on Identifiable Features

Weiwei Pang*,**, Xiangyang Luo*,***, Jie Ren*,**, Chunfang Yang*,**, Fenlin Liu*,**

*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

**Zhengzhou Science and Technology Institute, Zhengzhou 450001, China

***Science and Technology on Information Assurance Laboratory, Beijing 100072, China

pangweiwei01@126.com, luoxy_ieu@sina.com, renjie@vip.126.com, chunfangyang@126.com, liufenlin@vip.sina.com

***Abstract*—An increasing number of images in the Internet brings forward a higher requirement on the speed of steganalysis. For the problem of real-time detection of stego images, a rapid images steganalysis method based on identifiable features is proposed, where the identifiable features are specific character sequences left in stego images by steganography tools. The stego and cover images are distinguished according to whether the identifiable features are found in the detected images. Meanwhile, for the case of that multiple identifiable features appeared on the same location of an image, the AC (Aho-Corasick) multi-features matching algorithm is applied to improve the detection speed. In experiments, the detection method is used to detect eight steganography tools such as Invisible Secrets, E-Show, BMP Secrets and so on. The results show that the proposed steganalysis method can achieve a nearly perfect detection precision, and the detection speed can be improved significantly comparing with traditional methods (matching bytes one by one).**

***Keyword*—Steganalysis, Identifiable features, Steganography tools, AC(Aho-Corasick) matching algorithm, Stego image detection.**

I. INTRODUCTION

Steganography is a covert communication technique to embed confidential message into the redundancy parts of multimedia files such as digital images, audios and videos, and then transfer the obtained stego objects through public communication channels [1]. Contrarily, steganalysis includes judging detected object is stego or cover, recognizing the steganography algorithm, estimating the length or location of secret message, cracking the embedded key and extracting the secrets message. The stego objects detection is especially important because which is the first step of steganalysis. Generally, steganography has been broken if an attacker can judge the detected object whether contains secret messages with a success better than random guessing. Compare to other media forms (audios, videos, etc), images are the most commonly covers used in steganography. So this paper mainly studies image steganalysis. As the rapid popularization of telephone and camera in recent years, the number of images appearing in the Internet is increasing dramatically. Data shows that about 10 million images are uploaded to social networks each hour [2]. Therefore, the fast and accurate detection of stego images from a large number of images is one of the most urgent practical problems to be resolved.

Currently, researches on detection of stego images can be divided into three classes: sensory detection, statistical feature detection and identifiable feature detection. Sensory detection, as an early detection algorithm, has been obsoleting since it is difficult to implemente automatically. Statistical feature detection is the research hotspot of steganalysis because of that most of steganographic algorithms can be detected reliably by this methodology. For example, Pevnỳ and Fridrich [3] extended the 23 DCT features set [4] to get a 274-dimensional feature vector, then used the new feature to construct a Support Vector Machine multi-classifier capable of assigning stego images to six popular steganographic algorithms: OutGuess [5], F5 [6], MB [7], etc.; Fridrich and Kodovsky extracted the 34671-dimensional SRM (Spatial Rich Model) [8] feature and 22510-dimensional CC-JRM (Cartesian Calibrated-Jpeg Rich Model) [9] feature from spatial images and jpeg images respectively to attack some algorithms successfully, such as HUGO (Highly Undetectable steGO) [10], LSB (Least Significant Bit) [11], EA (Edge Adaptive) [12], MME (Modified Matrix Encoding) [13], nsF5 (no-shrinkage F5) [14], etc. Denemark, Fridrich and Holub [15] put forward the novel concept of content-selective residual to increase the detection precision of S-UNIWARD. These steganalysis methods based on statistical features above can attack many steganographic algorithms reliably, but the dimensions of these statistical features are high, and detection speed is low, it is difficult to meet the requirement of real-time detection for a large number of images. Besides, steganalysis based on statistical features has a high false

Manuscript received June 17, 2015. This work is a follow up of the accepted conference paper as an outstanding paper for the 17th International Conference on Advanced Communication Technology.

This work was financially supported by the National Natural Science Foundation of China (No. 61379151, 61272489, 61302159, 61401512, 61373020 and 61572052), the Excellent Youth Foundation of Henan Province of China (No. 144100510001), and the Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-14-108).

W. PANG is currently a M.S candidate at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China (phone: +86 13027790289; e-mail: pangweiwei01@126.com) .

X. LUO, J. REN, C. YANG (corresponding author, phone: +86 13513891391) and F. LIU are with the State Key Laboratory of Mathematical Engineering and Advanced Computing, and Zhengzhou Science and Technology Institute, China (e-mail: luoxy_ieu@sina.com, e-mail:renjie@vip.126.com, chunfangyang@126.com, liufenlin@vip.sina.com). X. LUO also is a researcher at Science and Technology on Information Assurance Laboratory, Beijing, China.

detecting rate for lower embedding rate images.

Nowadays, the kinds of steganography tools are more than one thousand and some of them will leave identifiable features in stego images. The identifiable feature detection method recognizes these stego images through checking whether detected images contain these identifiable features. The missing rate of steganalysis based on identifiable features is 0 and the false detecting rate is low for lower embedding rate images [16]. Besides compare with the steganalysis based on statistical features, this method has a significant speed advantage. Bell and Lee [17] proposed a fast and accurate automatic detection method based on the characterized regularities in output media caused by weak implementations of some steganographic algorithms. Bell and Lee [17] have used the proposed method to detect 6 kinds of steganography tools such as Steganos, Inv.Sevrets, OutGuess, JSteg, STools and MP3Stego. Pevnў and Ker [18] used the length of message as dynamical identifiable feature of OutGuess to crack the stego key. In this method, every key in the key dictionary can be used to get a message length, if the message length extracted (can be seen as a dynamic identification feature) is more than the estimated embedding capacity of the stego image embedded by OutGuess, the key must be wrong and should be dropped from the key dictionary. Because messages length extracted from different stego images (the same stego key is used) by the same wrong key are very possibly different, if the stego images with the same stego key are enough, then keys in the key dictionary can be reduced to one or a few by exhaustion attacks. Because above methods do not consider how to reasonably organize the identifiable features, with the number of steganography tools increasing, the number of identifiable features must increases, and the detection time of the traditional detection (matching bytes one by one) will increase linearly. This would not meet the requirement of detecting stego images generated by many steganography tools from a large number of images.

In [19], we have briefly given a method to rapidly detect the identifiable features in stego images, and experimented with two steganography tools (A Plus the File Protection and 007 Electronic Stego Water). In this paper, above method will be supplied with more details and tested with eight steganography tools. According to the different areas of stego images where identifiable features locate, the proposed method divides identifiable features into head features, data features and tail features. Then, head feature table is constructed to detect the head area of images. Because of the speed advantage of AC multi-pattern matching algorithm [20] in multi-feature matching, a multi-pattern fuzzy matching machine of data features is constructed to detect the data area of images by AC multi-pattern fuzzy matching algorithm, and a multi-pattern exact matching machine of data features is constructed to detect the tail area of images by AC multi-pattern extract matching algorithm. Experimental results demonstrated the effectiveness of the detection method proposed in this paper, which could significantly improve the detection speed on the condition that the missing rate is zero and the false detection rate is very low. The problem that detection time increases linearly with the number of features increasing is relieved effectively.

II. Steganalysis Based on Identifiable Features Classification

Identifiable features are constant marks left in stego images by steganography tools to protect copyright or check whether images have been embedded, they usually present as specific characters sequences appeared in specific bits of stego images. Identifiable features exist in different positions of stego images have different forms. For example, the identifiable features located in the head of images usually are represented as abnormal properties values in head of stego images, and these property values often are less than two characters, so the image which will be detected is a stego or cover will be judged through comparing properties of images which will be detected with "abnormal properties" of stego images. The identifiable features located in data area of images are represented as LSB sequences consisted of pixels' LSBs of stego images usually, the detection speed will be fast if the method of detecting on data area of images based on multi-pattern fuzzy exact matching algorithm is used to detect data area of images. Similarly, the identifiable features located in tail area of images are represented as hexadecimal character sequences consisted of pixels of stego images usually, the detection speed will be fast if the method of detecting on tail area of images based on multi-pattern exact exact matching algorithm is used to detect data tail of images. For this reason, identifiable features are divided into head features, data features and tail features. Different identifiable features recognizing algorithms are used to recognize the three classes of features. Detection methods are described as follows.

A. Identifiable features classification

Head Features: some image properties (such as width, height, resolution, palette, etc.) may be falsified by some steganography tools when embedding. For example, the length of the file head of a BMP image will be increased 1 if the image is embedded by Imagehide [16]. The first reserved value of a BMP image will be changed to the message length if the image is embedded by E-Show. The resolution of a BMP image will be changed to 73*73 if the image is embedded by BMPSecret [16]. The resolution of a BMP image will be changed to 0 if the image is embedded by Invisible Secrets. In addition, there is a characters sequence consisted of 256 characters (1, 2, 3, ⋯ ,255 , 0) will be added in the palette redundancy of a BMP image if the image is embedded by Stegomagic1.0 [16].

TABLE I
BMPSECRET & E-SHOW HEAD FEATURE TABLE ITEMS

Tools	**Format**	**Size**	**Offset**	**Reserved_1**
BMPScerets	*BMP*	*-1*	*-1*	*-1*
E-Show	*BMP*	*-1*	*-1*	*Msg length*
DataSize	**Resolution**	**Width**	**Height**	**……**
-1	*73*73*	*-1*	*-1*	*-1*
-1	*-1*	*-1*	*-1*	*-1*

The head features are often expressed as abnormal properties values in the heads of stego images, and these properties values can be achieved easily through analyzing of image format. So head feature table is constructed for every head feature which existed in feature library. Then whether

the image head contains head features will be judged by comparing head feature table with file head of the detected images. Head feature table items of BMPSecret and E-Show are shown by TABLE I, where the normal properties of image are represented of *-1*.

Data features: data features are identifiable features left in the data area of image by a part of steganography tools, they often represent as specific character sequences existed in LSB or 2LSB, etc. Because images may be distorted if pixels or DCT coefficients are tampered to specific characters sequence, these sequences often located in LSB or MLSB (commonly under 4LSB). LSBs are the least bits of image's pixels, so a pixel is an odd number if LSB of the pixel is 0 and a pixel is an even number if LSB of the pixel is 1. That means LSBs of image's pixels is equivalent to the odd-even sequence of image's pixels. LSBs represent as characters sequences consisted of 0 and 1 in this paper. Similarly, 2LSBs represent as characters sequences consisted of {0, 1, 2, 3}. Data features are listed as follows, the sequence "0100 0011 0100 100" appear in the LSBs from 37 to 45 pixels in BMP stego images which have been embedded by A Plus File Protection [16], the sequence "0100 0010 0110 1001 0111 0010 0110 0100" appears in the LSBs from 36 to 55 pixels in BMP stego images which have been embedded by 007 Electronic Stego Water [16]. The sequence "1000 1110 1000 0111 1001 1111 1001 0001" appears in the LSBs and the sequence"1010 1000 1111 0100 0001 1010 1001 0000" appears in the 2LSBs from 0x36 to 0x56 pixels in BMP stego images which have been embedded by Inthepicture [16]. The probability that different features have the same prefix will be increasing as the number of identifiable features increasing. Character types of these sequences is no more than four (0 1 2 3), so it is more prone to have the same prefix.

Multiple features detection actually is an issue of multi-pattern matching. As a typical multi-pattern matching algorithm, AC multi-pattern matching algorithm has obvious speed advantage than other algorithms in multi-pattern matching. Therefore, a multi-pattern fuzzy matching algorithm based on AC multi-pattern matching algorithm is proposed and adopted to detect data area of images. The multi-pattern fuzzy matching algorithm is described in section III.

Tail features: some particular characters sequences consisted of pixels will be appended to the tail redundancy area in images by some tools. These features are tail features in this paper. For example, the character sequence "0x07 0x00 0x00 0x00" is appended to the last bit of a BMP or JPEG image if the image is embedded by Bulletproof vest [16]. Character "FF" is appended to the last bit of a BMP or JPEG image if the image is embedded by E-Show. The character sequence "0xCC 0x99 0xFF 0x66" is appended to the last bit of a BMP image if the image is embedded by safe & quick hide file 2002. Besides, the characters sequence "0x5B 0x3B 0x31 0x53 0x00" is appended to the last bit of a JPEG image if the image embedded by Jpegx [16].

Tail features will be represented as specific sequences consisted of hexadecimal character. Hexadecimal character has only 16 kinds of characters. So as similar to data features, a multi-pattern exact matching algorithm based on AC multi-pattern matching algorithm is proposed and adopted to detect the tail redundancy area of images. The algorithm is described in section III.

B. Stego images recognition based on identifiable features classification

As shown in Fig.1 and Fig.2, detection method proposed in this paper is divided into two phases: pre-processing phase and detection phase. Every identifiable feature in the feature library should to be pre-processed before detection. Different processing rules are set to different classes of identifiable features above. Three results of pre-processing (head features table, data features fuzzy matching machine and tail features exact matching machine) are achieved after pre-processing phase. Then the three pre-processing results are used to detect three areas (head area, data area and tail area) of image to get three detection results (R_1, R_2, R_3). The three detection results are used to get the final detection result R by RGUD (Results Judging based on United-Decision) algorithm. The final detection result R will tell you that the detected image is a stego image or cover image, besides, name of the steganography tool will be known by R if the detected image is a stego image.

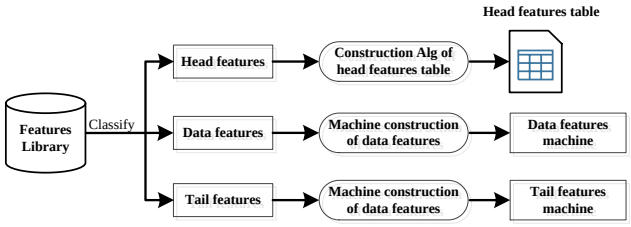

Fig. 1. Identifiable features pre-processing

As shown in Fig. 1, head feature table is constructed for every head feature existed in features library. If other properties (except steganography tools name and image format) have head features, then these head features will be added to head feature tables. The head feature table consisted of identifiable features of Invisible Secrets and E-Show is shown in TABLE I. Head detection is described as follows: properties' values of the detected image are achieved through analyzing the image format first, then decide whether abnormal values (head features) exist in these properties according to checking head features tables. If they exist, the image is a stego image. R_1 is true and tools name is extracted from the head feature table. For example, if the resolution of an image is 73*73, the image may be a stego image and which is embedded by BMPSecrets, if the first reserved value of an image is not 0, the image may be a stego image and which is embedded by E-Show. If all properties of the image have no any abnormal value, head data of the image is normal and R_1 is false. In view of the situation that a property may have more than one features (such as the resolution of BMP images is 0*0 or 73*73 if the image is embedded by Invisible Secrets or BMPSecrets) and the length of head features is short (1 or 2 bit), binary search algorithm is adopted to detect these properties to improve detection efficiency. For data area and tail redundancy area, these areas of images were detected by using matching machines constructed by data features or tail features in the identifiable features pre-processing stage. Then detection results R_2 and R_3 have been achieved. In the end,

united-decision algorithm was adopted to get the final detection result R. Details are introduced as follows.

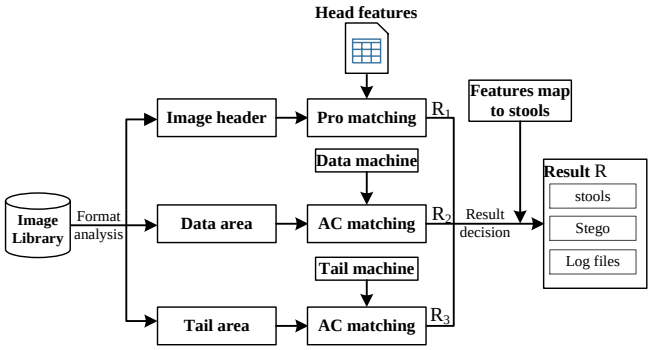

Fig. 2. Identifiable features rapid recognition

If different data features appear in the same location of LSBs or 2LSBs, a multi-pattern fuzzy matching machine consisted of parity sequences should be constructed. Similarly, if different tail features appear in the same location of image, a multi-pattern exact matching machine consisted of characters (0, 1, 2, 3) should be constructed. The algorithm of matching machine construction is described in section III.

Detection method is shown in Fig. 2. Through format analyzing of images, the detected image is divided into three sections: head, data and tail. Different detection algorithms are used to detect three sections of images by pre-processing results (head features tables, fuzzy matching machine and exact matching machine) for three classes of features above. Three detection results R_1, R_2, R_3 would be achieved after detection. United-decision algorithm is proposed to analyze these results, the final detection result R is achieved after decision.

C. Results judging based on united-decision

As shown in Fig. 2, the result set R_1, R_2, R_3 has been achieved from detection of image head area, data area and tail area above, then the three detection results will be judged to get the final result R by RGUD (Results Judging based on United-Decision) algorithm. The pseudo-code of RGUD is shown in TABLE II where T_1, T_2 and T_3 are names of steganography tools. If two or three results from the result set are true and steganography tools which results refers to are different, it is needed to judge the result set. The process is: when the three results are all false, the image detected can be judged as a cover; when three results are true, the image detected can be judged as a stego image, then extracting the name of steganography tool which the true result points to; if two of the results is true, the image is a stego image, if the two true results points to the same steganography tool, then the image is a stego image embedded by the steganography tool, If the two software name are inconsistent , the two types of tools are suspicious, extracting the two tools name and recording into the final result R, respectively; in a similar way, if the three results are true, the image is stego image, when the three results point to the same tool, judging that the image embedded by the tool. If there are two results in three results point to the same tool, the two software are suspicious tool, and should be included in the final detection result, the tool which the two results points to has a higher priority, If there is no any same name of tools which the three results points to, the three types of tools are suspicious and write into the final result R, there is no priority order. At this point, the detection method proposed in this paper is completed and the final result R is gotten.

TABLE II
PSEUDO-CODE OF RGUD ALGORITHM

Name: RGUD
Input: R_1, R_2, R_3
Output: R is stego or cover, stego tool names and probability

```
If R1, R2, R3 all are False
    R is a cover image
End
If one of R1, R2, R3 is True
    R is a stego image
    T1 is the tool with probability 100%
End
If two of R1, R2, R3 are True
    R is a stego image
End
    If two tools are the same
        T1 is the tool with probability 100%
    Else T1, T2 are the two tools with probability 50%, respectively.
    End
If R1, R2, R3 are all True
    R is a stego image
    If three tools are the same
        T1 is the same tool with probability 100%
    End
    If two tools of the three are the same
        T1 is the same tool with probability 67%, T2 is the other tool with
          probability 33%
    End
    If three tools all are different
        T1, T2, T3 are those tools with probability 33%, respectively
    End
End
```

III. IDENTIFIABLE FEATURES DETECTION BASED ON MULTI-PATTERN MATCHING

Multi-pattern matching is an algorithm which can finish detecting in a matching process [20]. AC, WM (Wu-Manber) [21] and SBOM (Set Backward Oracle Match) [22] are typical multi-pattern matching algorithms in the field of intrusion detection. The character type of identifiable feature is less than 16 and there are no bad characters (characters which present in image data but not exist in features) in image data, so the advantage of WM that increasing jump step by bad characters is not reflected. Besides, Chen Xiao-jun argues that memory access time of WM and SBOM is longer than AC in paper [23], and the advantage of memory access time of AC is especially obvious when the number of patterns is large. Taken together, AC algorithm has higher detection efficiency than WM for detecting data area and tail redundancy area, and the efficiency is higher with the number of patterns increasing. In addition, considering the position of identifiable features is relatively fixed, the concept of position verification is added into AC algorithm in order to further reduce the false detecting rate in this paper.

The algorithm includes two parts. The former is

pre-processing phase, a finite state feature matching machine should be constructed of all identifiable features which existed in data area or tail redundancy area; the latter is detection phase, matching machines constructed above will be used to detect image data area or tail redundancy area.

Pre-processing phase: Matching machine constructed process is shown in Fig. 1. There are three functions Goto function, failure function and output function should be constructed. Goto function stands for turning to the next state, and continuing to match until to the situation that input characters and feature's characters matching successfully, which was indicated by solid arrows in Fig. 3. Node numbers in Fig. 3 were set in the order that feature's ID smaller first between different features and left character is first in the same feature. Failure function stands for which state should be jumped to when input character is not equals to features' character, which was indicated by dotted arrows in Fig. 3. Failure function is a backtracking process and which reduce access times of the same prefix characters from n (the number of features) to one when n features have the same prefix characters. For example, the three features {1011, 1110, 1100} have the same prefix "1", then the first character "1" just need be accessed one time during the whole searching process. Output function stands for outputting an identifiable feature when the feature and image data detected matching successfully, which was indicated by states {4, 7, 9} in Fig. 3. It means the feature exist in image data, then go to the position verification stage. If the position verify successfully, the image is judged as a stego image. Using identifiable features in BMP image data area as an example to illustrate the process, if features exist in least significant bit of BMP image, then they are represented as the sequence consisted of zeros and ones. For example, there are three data features "odd, even, odd, odd", "odd, odd, odd, even", "odd, odd, even, even" in the same pixels, where the "odd" and "even" are mean that the pixel is an odd or even. For the convenience of expressing, "odd" and "even" are expressed with "1" and "0", respectively. That means that the three data features can be expressed with {1011, 1110, 1100}, and LSBs of pixels where these features located is {00101110010}, the process of matching machine construction is shown as step 1) to 3).

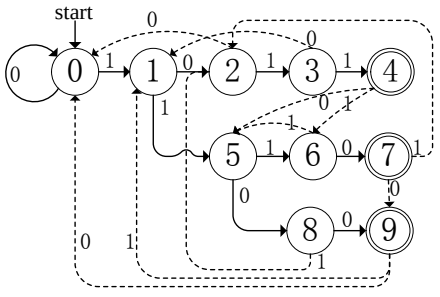

Fig. 3. The fuzzy matching machine constructed of data features

1). Goto function is consisted of characters which in the features set {1011, 1110, 1100}. Prefix relationships between the three features were described by the function consisted of ten directed edges and ten nodes (can be called states). Prefix relationships between the three features has decreased the number of states from 12 to 10. So matching times would be reduced and detection efficiency would be improved. Which are shown as solid arrows in Fig. 3.

2). Failure function maps a state into another [18]. The function's main aim is to look for which state jump to whenever Goto function reports fail. The disadvantage of backtracking in BF (Brute Force) algorithm is eliminated in the failure function. Construction process of failure function was described as follows: first, failure values of all states s which depth is 1 were initialized to 0. Then failure values of other states were computed in order of depth-first. It means failure values of states which depth is d should be concluded by failure values of states r which depth is $d-1$. As shown in formula (1):

$$f(s)=\begin{cases}0 & d=1\\ f(s') & d>1\end{cases} \tag{1}$$

where $f(s')$ is an iterative process of $f(s)$, detailed steps are described as follows: ①Set $state=f(r)$, where r is the direct precursor of s; ②according to the Goto function to calculate the value of $g(state,a)$, where a is an input character, if the value is null, then this step is executed iteratively many times until the value is not null, then the non-null value is the value $f(s)$ maps to. The non-null value must exist in Goto function because $g(0,0)=0$, $g(0,1)=1$ and input character is only 0 or 1; ③certain states are designated as output states which indicate that a set of features.

Example:

Initializing state which depth is 1, set $f(1)=0$;

Calculating the failure values of states {2, 5} which depth are 2 utilizing the failure values of states which depth are 1:

set $state=f(1)=0$, set $f(2)=0$ as $g(0,0)=0$;

set $state=f(1)=0$, set $f(5)=1$ as $g(0,1)=1$;

Calculating the failure values of states {3, 6, 8} which depth are 3 utilizing the failure values of states which depth are 2:

set $state=f(2)=0$, set $f(3)=1$ as $g(0,1)=1$;

set $state=f(5)=1$, set $f(6)=5$ as $g(1,1)=5$;

set $state=f(5)=1$, set $f(8)=2$ as $g(1,0)=2$;

As above, calculating the failure values of states {4, 7, 9} which depth are 4 utilizing the failure values of states which depth are 3;

set $state=f(3)=1$, set $f(4)=5$ as $g(1,1)=5$;

set $state=f(6)=5$, set $f(7)=8$ as $g(5,0)=8$;

set $state=f(8)=2$, set $f(9)=0$ as $g(2,0)=0$,

$state=f(2)=0$ and $f(0,0)=0$.

i	1	2	3	4	5	6	7	8	9
$f(i)$	0	0	1	5	1	5	8	2	0

(2)

3). To improve the detection accuracy, the property *firstCharIndex* (the index of feature's first character in stego images) is added to struct of output features to check whether features is located in the specific location of stego image. The struct of node is shown in TABLE III. Output states are shown as double loop nodes in Fig. 3.

TABLE III
STRUCT OF MATCHING MACHINE OUTPUT NODES

Struct **outpatstruct**; // struct name

Char ***opat****[PATLEN]*; // patterns content
long ***firstCharIndex***; // bit number in stego images where the first character of patterns located.
int ***patternIndex***; // patterns' ID (unique)
struct outpatstruct ****next***; // point to the next pattern

The matching machine consisted of features set {1011, 1110, 1100} has been constructed now, detection phase is beginning. The fuzzy matching machine is constructed to detect the data area of image.

Similarly, the exact matching machine is constructed to detect the tail area of images. The process of exact matching machine constructing is similar to the process of fuzzy matching machine. For example, there are three tail features in the last some bits of tail area and they are {0x00 0x00 0xFF, 0x07 0xEF 0x0A, 0x00 0x00 0x3C}, the last some bits of tail area are {0x0E 0x00 0x00 0x3C 0xEF 0xFF}, then the exact matching machine is show in Fig.4 where Ω is any hexadecimal character.

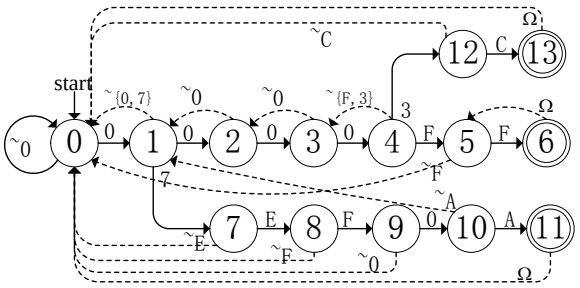

Fig. 4. The exact matching machine constructed of tail features

Detection phase: According to a window sliding on a characters sequence consisted of image data bytes to searching for features by the matching machine constructed above. An identifiable feature will be outputted when the feature is found in images. Then check whether the value of *firstCharIndex* is the index of location where the feature located. If so, detection results R_2 or R_3 is true and the feature code will be outputted. If not, then R_2 or R_3 is false. Detection of data area and tail area completed now. The detection result set R_1, R_2, R_3 are achieved when detection of the whole image has been completed, then united-decision algorithm is adopted to get the final detection result R.

IV. EXPERIMENTS

Detection precision includes two sides: undetected rate and false rate. As AC, WM and BF are precise matching algorithms, so the undetected rate is determined by identifiable features are accurate or not. If identifiable feature is correct, then undetected rates of three algorithms are zero. If identifiable features are can't distinguish stego images from cover images, then undetected rates are not reliable yet. Besides, the false rate is determined by identifiable features are integrity or not. The false rate will be increase if identifiable features are not integrity. Besides, the false rate is affected by the length of identifiable feature. For example, characters "0xFF" are appended to the last bit of BMP or JPEG images by E-Show, and the probability of that the cover images have the same last bytes as stego images is $1/32$. So the false rate of E-Show is $1/32$ under the condition that the detected image has tail redundancy bytes. For the steganography tool "007 Electronic Stego Water", the length of identifiable feature is 32, and the probability that images have the same LSBs of the identifiable feature is $1/2^{32}$, so the false rate is $1/2^{32}$.

To validate the accuracy and rapidity of the detection method proposed in this paper, experiments are designed as follows. They include three parts: detection of image head, detection of image data area and detection of image tail redundancy.

Hardware environment: the experimental environment is 64-bit Windows 7 Operating System, Pentium(R) P6200 (2.13 GHz) CPU, 3.67 GB RAM, and development environment is Microsoft Visual Studio 2010. Note that the level of hardware performance and the busy degree of CPU all might make the detection time float on a small range, therefore, in order to ensure the precision of the detection time, the same PC is used to test AC, WM and BF algorithms at the same time in detection.

Construction of images library: 10000 PGM gray-scale images of BOSSBase-1.01 database [24] are transformed into BMP images, the resolution and size of these images are 512*512 and 257 KB uniformly. 80 BMP images selected randomly are used to generate stego images. First, they are divided into 8 groups randomly, every group include ten images. Then eight groups of images are embedded by 8 steganography tools separately. Eight steganography tools are Invisible Secrets, E-Show, BMPSecret, A Plus File Protection, 007 Electronic Stego Water, Encryption Excellent Soldier, Small Encryption Lock and Bulletproof Vest. Secret message is a random length (less than embedding capacity of cover images) of text document (txt). These 80 stego images were put into other 9920 cover images. Then the test images library has been constructed.

TABLE IV
TABLE OF STEGANOGRAPHY TOOLS INDENTIFIABLE FEATURES

Tools	Area	Location	Features
Invisible Secrets	Head	Resolution	0*0
E-Show	Head & Tail	Reserved _1 Last character	Msg length "0xFF"
BMPSecret	Head	Resolution	73*73
A Plus File Protection	Data	0x36~0x45 LSBs	0100 0011 0100 100
007 Electronic Stego Water	Data	0x36~0x55 LSBs	0100 ... 0100
Encry Excellent Soldier	Tail	First 24 characters of tail redundancy	0x21 0x3F ... 0x3F 0x21
Small Encryption Lock	Tail	Last of tail redundancy	0x3C 0x3C ... 0x3C 0x3C
Bulletproof Vest	Tail	Last 4 characters of tail redundancy	0x07 0x00 0x00 0x00

A. Detection of image head area

As shown in TABLE IV, three steganography tools which identifiable features in image head are Invisible Secrets, E-Show and BMPSecret. If the detected image is a BMP image, then the resolution and reserved values are extracted by format analysis. When the resolution value is 0*0 or 73*73,

the image may be a stego image and is embedded by Invisible Secrets or BMPSeret. When the resolution value is not zero and less than embedding capacity (because steganographic algorithm of E-Show is LSB replacement, the embedding capacity can be replaced of image size/8) of the image, the image may be a stego image and is embedded by E-Show. Experimental result is described as TABLE V.

TABLE V
TABLE OF HEAD DETECTION RESULT

	Invisible Secrets	E-Show	BMPSecret
Stego number	9990	10	10
Undetected rate	0%	0%	0%
False rate	99.8%	0%	0%

Detection precision includes two sides: undetected rate and false rate. From detection result above, it can be seen that undetected rates of three tools are zero. But the false rate of Invisible Secrets is 99.8%, because of identifiable feature of the tool is can't distinguish stego images from cover images, then undetected rates are not reliable yet. No correlation with the detection method proposed in this paper. It can be proven that the false rate of BMPSecret is 0%. The time of image head image detection is about 496ms.

B. Detection of image data area

Construction of features set: characters sequences "0100 0011 0100 100" and "0100 0010 0110 1001 0100 0010 0110 0100" which generated by A Plus the File Protection and 007 Electronic Stego Water, respectively. Besides, since the current identification features which we have are deficiency, to test the influence on detection speed with the number of features increasing, we have constructed a generator to generate virtual identification features code. About 500 virtual identification features consisted of random characters {0, 1} were constructed by the generator, the length of these virtual identification features is 17 bits.

Detection objects: the LSBs of pixels which before the 56 (hex) bytes in BMP images, a total of 87 bits characters sequences consisted of zeros or ones.

Detection algorithms: AC, WM and BF.

Accuracy verification: experimental results show that 20 stego images can be accurately identified by three detection algorithms, undetected rate and false rate are 0. The recognition algorithm of identifiable features proposed in this paper is a precise matching algorithm. There is no undetected case if identifiable features are correct and complete. So undetected probability can keep zero for any embedding rate. For pixels' LSB in data area, the probability of cover images and stego images having the same characters is $1/2^n$, where n is the number of features bits. So false rate is less than $1/2^n$. In the experiment, n is more than 15, so the false rate is less than 0.003% and close to 0. Besides, the results show that the undetected probability and the false rate of three detection algorithms (AC, WM and BF) are 0. On the contrary, steganalysis based on statistical features is difficult to get the higher detection precision when the embedding rate is less than 1%. So steganalysis based on identifiable features is reliable for lower embedding rate images.

Rapidity verification: the algorithm of identifiable features recognized based on AC proposed in this paper has a speed advantage to other algorithms, and result is shown as Fig. 5 and TABLE VI.

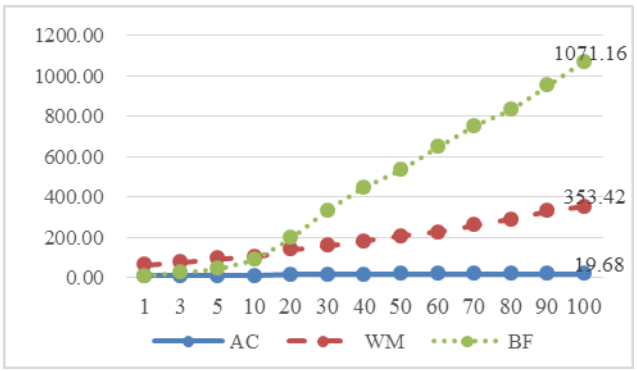

Fig. 5. Net detection time for three algorithms

Fig. 5 abscissa is the number of identifiable features and which ordinate is net detection time (ms). The net detection time only refers to the matching time, does not include the time of reading image and outputting result. Namely, it is just refers to the time of matching stage, does not include pre-processing for AC and WM. The time of pre-processing can be ignored when a large of detected images. In a word, the detection time is only impacted by the number of identifiable features and the size of detected images.

The first column of TABLE VI is the number of features. Detection result of three algorithms (AC, WM, BF) shows that the undetected probability and the false rate are 0, the detection accuracy of the detection method proposed in this paper is still keep 100% for the lower embedding rate images. As can be seen from the Fig. 5 and TABLE VI, detection time of the detection algorithm based on AC keeps steady with identifiable features increasing. While detection time of the other two algorithms is increasing linearly. So detection algorithm based on AC proposed in this paper has a higher detection speed.

TABLE VI
TABLE OF NET DETECTION TIME

Features number	Detection net time (ms)			Time rate	
	AC	WM	BF	AC/WM	AC/BF
100	19.68	353.42	1071.16	5.57%	1.84%
200	21.40	680.37	2075.72	3.15%	1.03%
300	23.72	1032.57	3199.38	2.30%	0.74%
400	24.15	1389.35	4289.33	1.74%	0.56%
500	24.44	1692.50	5488.26	0.91%	0.45%

C. Detection of image tail area

TABLE VII
TABLE OF FINAL DETECTION RESULT

	E-Show	Soldier	Lock	Vest
Stego number	10	10	10	10
Undetected rate	0%	0%	0%	0%
False rate	0%	0%	0%	0%

As shown in TABLE IV, four steganography tools which identifiable features in image tail redundancy are E-Show, Encryption Excellent Soldier, Small Encryption Lock and Bulletproof Vest. If the detected image has tail redundancy bytes, then detection of image tail redundancy is started. If tail redundancy bytes of images have one identifiable feature,

then the image may be stego image. Then combine the detection result with head and data area detection results, the final detection result is achieved by decision of detection result. The final detection result is shown as TABLE VII.

Detection results show that there are forty images have tail redundancy bytes. It means that cover images and stego images can be divided just by checking images have tail redundancy bytes or not. Possibly because cover images used are transformed from BOSSBase-1.01 database and they are unified. Which steganography tool used for every stego image can be obtained by recognizing identifiable features.

V. Conclusions

A method of steganalysis of image based on identifiable features left by steganography tools is proposed in this paper. The method can detect reliably the lower embedding rate images. To solve the problem that many identifiable features appearing in the same place, an algorithm of identifiable features recognized rapidly based on AC is proposed. Experimental results show that the algorithm improve effectively the detection speed. However, this algorithm applies to some steganography tools which identifiable features have been achieved and can't work well in the case of tools which identifiable features have not been achieved. So, extraction of identifiable features of steganography tools will be studied in further research.

References

[1] J. Lu, F. Liu, and X. Luo, "Recognizing F5-like stego images from multi-class JPEG stego images," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 11, pp. 153-169, 2014.

[2] http:// www.computer.org/portal/web/tetc/

[3] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis." in *Proceedings of SPIE Electronic Imaging*, vol. 6505, pp. 3 1-3 14, 2007.

[4] J Fridrich. "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes". in *Proceedings of 6th International Workshop on Information Hiding*, 2005: 67-81.

[5] N. Provos, "Defending Against Statistical Steganalysis." *Usenix Security Symposium*, vol. 10, pp. 323-336, 2001.

[6] A. Westfeld, "High capacity despite better steganalysis (F5–a steganographic algorithm)." in *Proceedings of 4th International Workshop on Information Hiding*, vol. 2137, pp. 289-302, 2001.

[7] P. Sallee, "Model-based steganography." in *Proceedings of 2nd International workshop on Digital water-marking*, vol. 2939, pp.154-167, 2004.

[8] J. Fridrich, and J. Kodovsky, "Rich models for steganalysis of digital images." *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, 2012.

[9] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models." in *Proceedings of SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics*, vol. 8303, pp. 0A 1-13, 2012.

[10] T. Pevný and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography." in *Proceedings of 12th International Workshop on Information Hiding*, vol. 6387, pp. 161-177, 2010.

[11] C. Kurak and J. McHugh, "A cautionary note on image downgrading." in *Proceedings of the Computer Security Applications*, pp. 153-159, 1992.

[12] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited." *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 201-214, 2010.

[13] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography." in *Proceedings of 8th International Workshop on Information Hiding*, vol. 4437, pp. 314-327, 2007.

[14] J. Fridrich, T. Pevný, and J. Kodovský. "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities." in *Proceedings of the ACM 9th workshop on Multimedia & security*, pp. 3-14, 2007.

[15] T. Denemark, J. Fridrich, and V. Holub, "Further study on the security of S-UNIWARD." in *Proceedings of SPIE, Electronic Imaging, Media watermarking, Security, and Forensics*, vol. 9028, pp. 04 1-12, 2014.

[16] G. Ren, "Analysis & Attack of the popular network steganography software." *Zhengzhou Information Science and Technology Institute in Chinese*, 2009.

[17] G. Bell and Y. Lee, "A method for automatic identification of signatures of steganography software." *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp.354-358, 2010.

[18] T. Pevný, A. Ker. "Steganographic key leakage through payload metadata." in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pp.109-114, 2014.

[19] W. Pang, X. Luo, J. Ren, C. Yang, & F. Liu. Rapid detection of stego images based on identifiable features. in *Proceeding of the IEEE 17th International Conference on Advanced Communication Technology, ICACT 2015*, pp. 472-477, 2015.

[20] N. P. Tran and M. Lee, "High performance string matching for security applications." in *Proceedings of the International Conference on ICT for Smart Society (ICISS)*, pp. 1-5, 2013.

[21] D. Agrawal and A. EI Abbadi, "An efficient and fault-tolerant solution for distributed mutual exclusion." *ACM Transactions on Computer Systems (TOCS)*, vol. 9, no. 1, pp. 1-20, 1991.

[22] V. Rana, G. Singh, "MBSOM: An agent based semantic ontology matching technique." in *Proceedings of Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp.267-271, 2015.

[23] X. Chen, Z. Zhang, and Y. Liu, "Charactering memory access behavior of large scale multi-string matching algorithms." *Computer Engineering and Applications*, vol. 43, no. 26, pp. 106-109, 2007.

[24] http://boss.gipsa-lab.grenobleinp.fr/

Weiwei Pang was born in Henan Province, China, in 1989. Pang got the B.S. degree from Zhengzhou University, Zhengzhou, China, in 2013. He is currently a M.S candidate in the State Key Laboratory of Mathematical Engineering and Advanced Computing at Zhengzhou Science and Technology Institute. His current research interest is in image steganography and steganalysis technique.

Xiangyang Luo was born in Hubei Province, China, 1978. Luo received the B.S. degree, the M.S. degree and the Ph.D. degree from Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2001, 2004 and 2010, respectively. He is now a researcher at Science and Technology on Information Assurance Laboratory. He is the author or co-author of more than 70 refereed international journal and conference papers. He is also a guest editor for "International Journal of Internet" and "Multimedia Tools and Applications". His current research interests include Networking and Information Security.

Rana V, Singh G. MBSOM: An agent based semantic ontology matching technique[C]//Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on. IEEE, 2015: 267-271.

Jie Ren was born in Anhui Province, China, 1977. He received the B.S.and M.S degrees from the Zhengzhou Science and Technology Institute in 1999 and 2007, respectively. Currently, he is now a researcher at Science and Technology on Information Assurance Laboratory. His current research interest is Information Security.

Chunfang Yang was born in Fujian Province, China, 1983. He received the B.S., M.S., and Ph.D. degrees from the Zhengzhou Science and Technology Institute in 2005, 2008, and 2012, respectively. Currently, he is now a researcher at Science and Technology on Information Assurance Laboratory. His current research interests include image steganography and steganalysis technique.

Fenlin Liu was born in in Jiangsu Province, China, 1964. He received his B.S. from Zhengzhou Institute of Science and Technology in 1986, M.S. from Harbin Institute of Technology in 1992, and Ph.D. from the Eastnorth University in 1998. Now, he is a professor of Zhengzhou Institute of Science and Technology. His current research interests include Networking and Information Security.