Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

A New DHT: Network Steganography Based on Distributed Coding

Peng-Fei Xue, Jing-Song Hu, Rong-Gui Hu, Han-Lin Liu, Yu Gu

Electronic Engineering Institute (EEI)
No.460 Huangshan Road, Hefei, Anhui, China
leorick092182@163.com

Received August, 2017; revised December, 2017

Abstract. *Network steganography is a new branch of information hiding. Compared with digital media steganography, it has a stronger undetectability and difficult to track. To further enhance the undetectability of network steganography, a new network steganography based on distributed coding (NS-DCM) is proposed in this paper. NS-DCM is realized by coding steganogram in the distributed system. The experimental results showed that the proposed method has an average bandwidth of 0.998 bits/packet and a better undetectability than some other network steganography.*
Keywords: Information hiding; Steganography; Network steganography; Deep hiding techniques; Distributed system.

1. **Introduction.** Network steganography is the latest branch of information hiding. The term was originally introduced by Szczypiorski [1]. All of the information hiding techniques that may be used to transmit secret information in telecommunication networks are called network steganography. Unlike traditional digital media steganography, network steganography does not use image, audio or video files as carriers, but uses data packet to transfer secret information (also called steganogram).

The goal of network steganography is to hide the communication process, in other words, to hide the existence of the data exchange process. Before network steganography appeared, people found that there were two serious shortcomings when the secret information was embedded in digital media files. First, each file can only hide a limited number of data. Second, the modified media files may be obtained by forensic experts easily. Network steganography does not have these two flaws. It can leak information for a long time. Unless all the traffic is intercepted, it will not leave anything for analysis. Therefore, network steganography is more difficult to detect than digital media steganography.

With the further research of network steganography, steganalysis technology is also developing. Steganalysis refers to the process of determining whether or not there is secret information from the observed data, analyzing the size of the data and the location of the data embedding, and finally getting the original secret information. Currently, network steganography which only use a single protocol can't resist steganalysis effectively. Petitcolas et al. [2] and Zander et al. [3] proposed many steganalysis methods. Mazurczyk et al. [4] proposed steg-tomography that using traffic visual analysis to detect network steganography. The author chose LACK [5] as the experimental subject, and successfully detects StegoSIP [6] based on LACK. They announced that this visual analysis method can also be used to detect other network steganography.

In order to improve the undetectability of network steganography, Fraczek et al. [7] proposed Deep Hiding Techniques (DHTs). These techniques describe a general approach to improve the undetectability of various network steganography. The author defines five methods that can enhance the undetectability of network steganography. The presented methods are the first attempt in the state of the art to systematically describe general solutions that can make steganographic communication more undetectable and harder to perform steganogram extraction.

To further enhance the undetectability of network steganography, a new DHT is proposed in this paper, which is called network steganography based on distributed coding method (NS-DCM), to realize network steganography by coding steganogram in the distributed system.

This paper is organized as follows. The related work is introduced in Section 2, including network steganography, steganalysis, and deep hiding techniques. In section 3, the proposed methods are presented. Experimental results and analysis are presented in section 4, followed by a conclusion in section 5.

2. **Related Works.**

2.1. **Network Steganography.**

According to classification based on steganography pattern proposed by Wendzel et al. [8], network steganography mainly include three methods, that is, storage method, timing method and hybrid method.

The storage method either hide the data in the user data (payload) or hide the data in the protocol field (non-payload). There are two non-payload modes. One is the structure change mode. It means that the structure of the PDU is changed, such as size modulation of PDUs [9, 10], sequence modulation in PDUs [11, 12, 13], adding redundancy to PDUs [14, 15] and value modulation in PDUs [16, 17, 18, 19, 20, 21]. The other is the structure retention mode, which means that the structure of the PDU is remained. But the header fields of PDUs are changed, such as reserved/unused bits in PDUs [22, 23] and random values in PDUs [12, 24].

The timing method encode the secret information by utilizing rate or throughput of network traffic [25, 26, 27, 28], inter-packet interval [29, 30, 31, 32], message sequence timing [33, 34, 35, 36], etc.

There are two well-known hybrid methods, that is, Lost Audio Packets Steganography (LACK) [37] and Retransmission Steganography (RSTEG) [38]. LACK was first proposed by Mazurczyk [37], and then got further research until now. LACK uses the timing method to select suitable RTP packet. Meanwhile the storage method is used to embed secret information into the payload of selected RTP packet. RSTEG was proposed by Mazurczyk et al. [38]. RSTEG uses the timing method to find the data packets to be retransmitted, and uses the storage method for data embedding.

2.2. **Steganalysis.**

Steganalysis is the basis to against network steganography. Many researchers have proposed a variety of steganalysis methods to detect and analyze network steganography.

Smith and Knight proposed a general detection method based on statistical inference technique [39]. The possibility of covert channel existence can be estimated by comparing and analyzing the deviation of the eigenvalue distribution between the normal flow and the detected flow.

In addition to the general detection method, there are many special means for detecting a particular method. As mentioned in Section 2.1, the storage method of network steganography hides the secret information mainly by modifying some fields in the chosen protocols. But simple modifying these fields can cause the eigenvalue distribution to

differ from the typical distribution. At present, the mainly means to detect the storage steganography are to use some machine learning algorithms. Sohn et al. demonstrated that the covert channels with a simple encoding hidden in the IP ID or the initial sequence number (ISN) of TCP (proposed by Rowland [40]) can be discovered with high accuracy by SVMs [41]. Tumoian and Anikeev used a neural network to detect TCP ISN covert channel [42]. Zander showed that C4.5 decision tree classifiers can be used to detect covert channels in the IP TTL field [43]. The results showed that the detection accuracy can be up to 95%.

Hintz proposed a method to detect covert channels encoded in the TCP timestamp option [44]. He carried out a randomness test to examine the least significant bit (LSB) of the timestamps in low-speed networks. Too much randomness reveals the presence of a covert channel. Meanwhile, in high-speed networks, the covert channel can be detected by computing the ratio of different timestamps used and the total number of timestamps.

The timing method of network steganography hides the secret information mainly by changing packet rate or inter-packet interval. Venkatraman and Newman proposed a method to audit the change of traffic over time to detect packet rate channels [45]. Cabuk et al. proposed a technique to detect inter-packet interval covert channels based on their compressibility [46]. In this approach, a series of recorded inter-packet interval are converted to strings. The strings are compressed and the compressibility of a string is used to detect the presence of a covert channel. Stillman proposed a method to detect timing channels by computing plausible covert bit strings from the inter-packet interval and scanning for these bit strings in the sender's random access memory [47].

There are several covert channels that could possibly be detected, but as Mazurczyk et al. summed up in his book, "there is no published work that describes detection approaches for these" [48]. Current techniques to counter network covert channels usually focus on specific channels instead of more general characteristics common to multiple channels. A combination of many countermeasures is required to achieve a comprehensive protection, which is problematic in practice [48].

2.3. **Deep Hiding Techniques (DHTs).** The emergence of many steganalysis methods makes network steganography need further development to enhance the undetectability. For each kind of network steganography, there is a trade-off relation between bandwidth and undetectability. User X can utilize a method naively and sends as much steganogram as is possible but it will simultaneously raise a risk of disclosure. So he/she must purposely resign from some fraction of the steganographic bandwidth to achieve undetectability.

In order to make network steganography more undetectable, Fraczek [7] proposed five DHTs methods, that is, Steganogram Scattering (SGS), Steganogram Hopping (SGH), Carrier Modifications Camouflage (CMC), Inter-Protocol Steganography (IPS) and Multi-Level Steganography (MLS). The following content is mainly about SGS. More information about the other four DHTs can be referred in Ref. [7].

The SGS method needs to divide steganogram into several pieces first. Then each piece is sent as an independent message by using different steganographic method. The SGS method includes 3 means as shown in Figure 1, that is, flows-based scattering, hosts-based scattering, flows and hosts-based scattering.

(1) Flows-based scattering

The flows-based scattering has the following four steps. First, several flows are set up between two hosts. Second, the steganogram is divided into many pieces. Third, the sender covertly transmits each piece of the steganogram through one available flow. At last, the receiver receives all the pieces and reassembles them to get the original steganogram. An example of the flows-based scattering is presented in Figure 2.

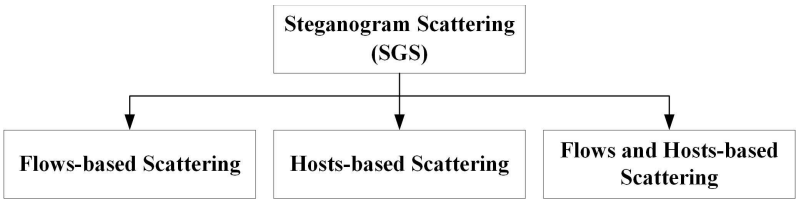

FIGURE 1. SGS classification [7]

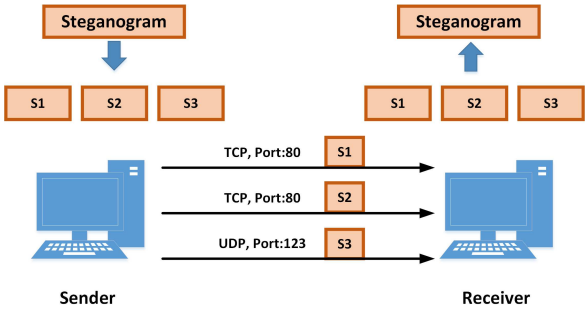

FIGURE 2. Example of the flows-based scattering

(2) Hosts-based scattering

The hosts-based scattering requires that the sender and the receiver control more than one hosts or other network devices separately. The sending hosts and the receiving hosts form a number of independent host pairs. The channel between each host pair is different from each other. An example of the hosts-based scattering is presented in Figure 3. The sender (controlling N hosts) wants to send steganogram to the receiver (controlling M hosts). The sender splits steganogram into k parts and sends them using steganographic methods that are available in different overt channels. The receiver gets k pieces of the steganogram and merges them to obtain the original secret information. In the presented example, maximum number of parts (k) into which steganogram is divided, equals the number of the different host pairs, that is $N * M$.

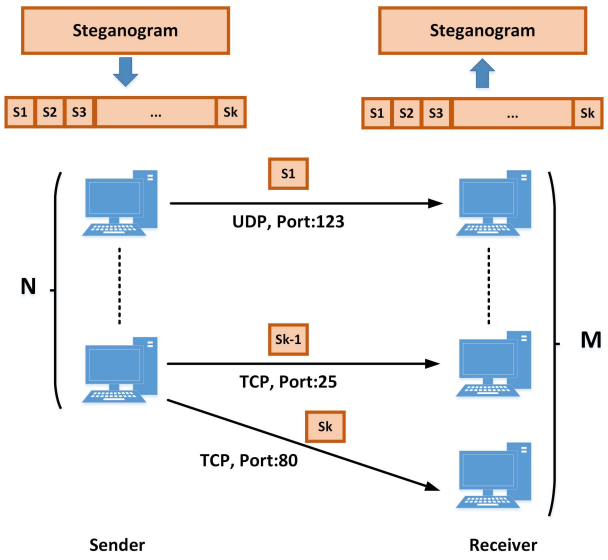

FIGURE 3. Example of the hosts-based scattering

(3) Flows and hosts-based scattering

The flows and hosts-based scattering is a hybrid of the first two method. It means that there are more than one flow in each host pair (as shown in Figure 4).

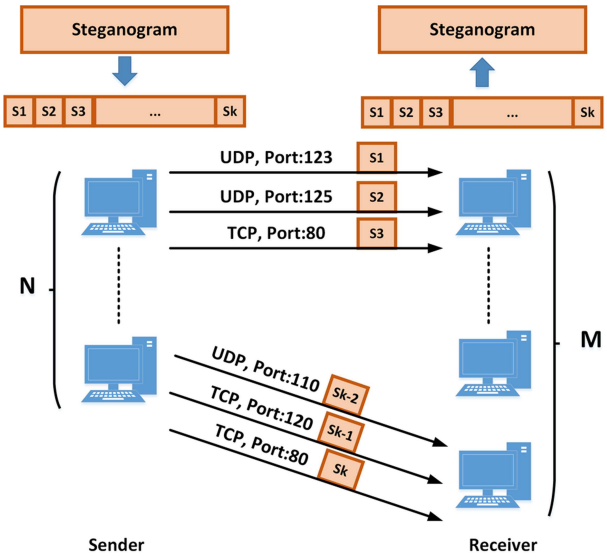

FIGURE 4. Example of the flows and hosts-based scattering

This hybrid method has more available flows than the first two methods. But it does not means that each flow must be used for steganography. The author says that some pieces of the steganogram can be redundant and can be sent using different flows. It will increase resilience and the chance of the successful steganogram reception by this way, even if some pieces are lost or removed.

3. **The Proposed Method.** As described in section 2, the simple network steganography based on a single protocol is unable to resist a large number of steganalysis means. The emergence of DHTs has broadened the field of network steganography and enhance its performance. On the basis of the SGS method proposed by Fraczek [7], a new DHT of network steganography based on distributed coding method (NS-DCM) is proposed in this paper. NS-DCM implements steganography by utilizing the different channels in the distributed system to encode and covertly transmit steganogram. There are two phases in NS-DCM, that is, embedding phase and extracting phase.

3.1. **Embedding Phase.** To implement this method, a distributed system should be constructed firstly. An example of the distributed system is shown in Figure 5. There are N sending hosts and M receiving hosts. The total number of host pairs is equal to $N*M$. Each host pair can be represented by $(i,j)(i=1,2,3...N; j=1,2,3...M)$. In each host pair, there is only one channel that can be used to transmit steganogram covertly. Each channel is different with each other, so the number of independent channels is $N*M$.

Before the secret information is transmitted, in order to make it easy for steganography, the steganogram need to be represented in the form of a binary string. If the steganogram is a text, it can be transformed to a binary string directly. If the steganogram is an image, it should be reduced from 2-dimension to 1-dimension first, and then be transformed to a binary string. There are a lot of dimensionality reduction method. The Zigzag Scan is used in this paper to perform reducing dimensions. If there is a matrix of $k*k$ as shown in Figure 6. The Zigzag method scans and gets cells in the matrix from the top left corner

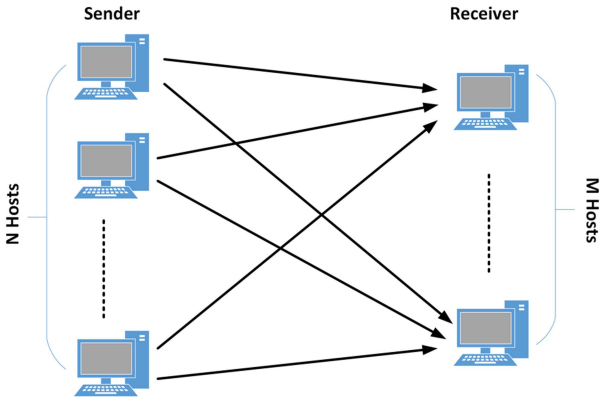

FIGURE 5. Distributed system of N sending hosts and M receiving hosts

to the bottom right corner one by one followed as the Z sharp. Because of its simple algorithm and low time complexity, it is widely used in steganography. The Zigzag Scan is also a scrambling method, which can make the steganogram be scrambled and improve the security.

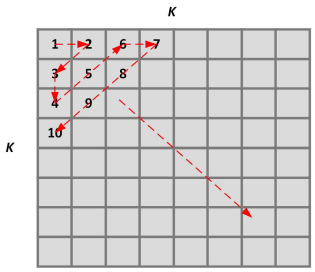

FIGURE 6. Example of Zigzag Scan

The sender constructs a matrix of $N*M$(called sending matrix). Then the steganogram represented by a binary string is filled in the sending matrix row by row (as shown in Figure 7). After the sending matrix is filled up, the value of cell (i, j) represent the message which the ith sending host should send to the jth receiving host.

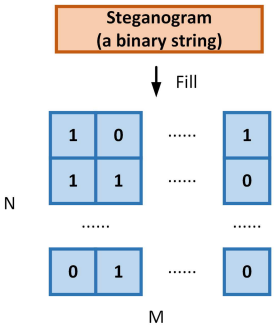

FIGURE 7. Filling the sending matrix with the steganogram represented by a binary string

The binary string only has two kinds of number, that is, 0 and 1. So the value of each cell in the sending matrix which has been filled up by the sender is either 0 or 1. Two commonly used protocols P_1 and P_2 in network communication are chosen to represent 0

and 1 respectively. The reason for choosing commonly used protocols is to avoid causing the traffic type exception in the network communication environment, and to reduce the possibility of being detected. The following step is the most important step of encoding steganogram. In the sending matrix as shown in Figure 7, if the value of cell (i, j) is 0, then the ith sending host should send P_1 to the jth receiving host. And if the value of cell (i, j) is 1, then the ith sending host will send P_2 to the jth receiving host.

In general, a sending host can send multiple messages to more than one receiving host at the same time. More than one send host may also send multiple messages to one receiving host simultaneously. To improve the accuracy of information reception, the latter transmission mode is used, that is, sending messages according to the sending matrix shown in Figure 7 row by row. An example of sending protocols is presented in Figure 8.

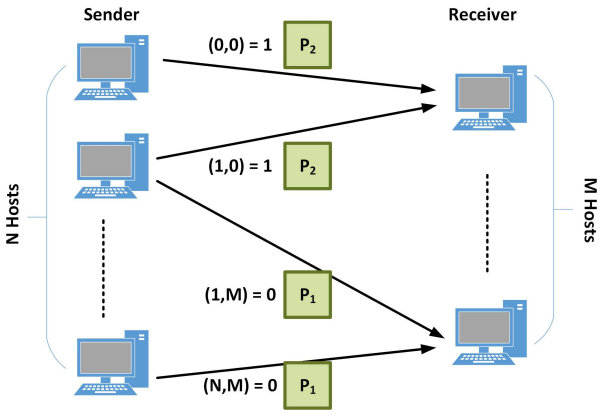

FIGURE 8. Example of sending protocols

Since the chosen protocols are commonly used in network communication, they may be confused with the same protocols that are not used as steganography in the network environment. In order to avoid this confusion, the sender and the receiver should agree on a special flag set in the header field of the protocol so that the receiver can distinguish the protocols used for steganography from all the protocols exist in the same network environment.

3.2. **Extracting Phase.** The extracting phase is the inverse of the embedding phase. The main process of the extracting phase is shown in Figure 9.

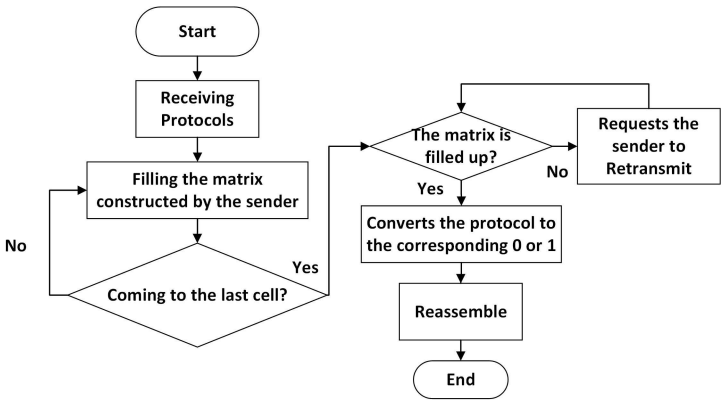

FIGURE 9. Example of sending protocols

At the beginning of the covert communication, the receiver should establish a matrix (called receiving matrix), which is consistent with the sending matrix. Since the sender sends protocols according to the matrix row by row, so each receiver host will receive a message at the same time. The message (protocol) that the jth receiving host received from the ith sending host should be filled in the (i, j) position of the receiving matrix.

When the filling process comes to the last cell (bottom right corner of the receiving matrix), the receiver will check whether the whole receiving matrix is filled up. If it is true, the covert communication is complete. If the matrix is not filled up, the receiver should locate the position of the empty cell. And then the receiver will request the sender to retransmit the protocol that not received.

Each cell in the receiving matrix is either P_1 or P_2. By reading the receiving matrix and converting each cell in it from protocol P_1 or P_2 to 0 or 1 respectively row by row, the receiver can get a binary string. If the steganogram is a text, the receiver can recover the original steganogram from the binary string directly. If the steganogram is an image, an inverse Zigzag Scan need to be performed first on the binary string to get a 2-dimension date. Then the original steganogram can be recovered.

To sum up, the Network Steganography Based on Distributed Coding Method can be described in Algorithm 1.

Algorithm 1 Network Steganography Based on Distributed Coding Method (NS-DCM)

Input:

steganogram S;
2 protocols P_1 and P_2;
N sending hosts and M receiving hosts.

Output:

extracted steganogram S_e.

1: The sender construct an empty sending matrix A_s, which size is $N * M$. The receiver establish a receiving matrix A_r, which is the same as A_s;
2: If the steganogram is a text, then it is converted to a binary string representation directly. If the steganogram is an image, a Zigzag Scan should be done first before converting;
3: Fill A_s with the binary string row by row. The value of cell (i, j) represent the actual bit value which the ith sending host should send to the jth receiving host;
4: Let P_1 represent binary '0', and P_2 represent binary '1'. Then each cell of A_s filled with the steganogram can be represented by the protocol P_1 or P_2;
5: The sender sends protocols to the receiver according to A_s row by row;
6: The receiver fills the receiving matrix A_r row by row with the receiving protocols. The protocol that the jth receiving host gets from the ith sending host should be filled into the (i, j) position in A_r;
7: When the last cell is filled in, the receiver will check whether the whole A_r is filled up. If it is true, the covert communication is complete and then turn to step 8. If it is false, the receiver should locate the position (e, f) of the empty cell, and then request the eth sending host to retransmit the protocol to the fth receiving host;
8: After the whole A_r is filled up, the receiver converts each cell from protocol P_1 or P_2 to 0 or 1 respectively;
9: The receiver can get a binary string by reading A_r row by row. If the steganogram is a text, the receiver can extract the steganogram S_e from the binary string directly. If the steganogram is an image, an inverse Zigzag scan need to be performed first on the binary string to get a 2-dimension date. Then the steganogram S_e can be recovered.

In this algorithm, the most important aspect of correctly and successfully transmitting secret information is to ensure that both the sending and the receiving matrices are complete, correct and consistent. The seventh step shows that the algorithm has the capability of error detection and error correction. The error detection mechanism could accurately locate the unsuccessfully receiving message. The error correction mechanism means that the receiver can request the sender for retransmission. It can be seen that the theoretical steganographic bandwidth of it is 1bit/packet. Actually, the steganographic bandwidth needs to be verified by the following experiment.

4. **Experimental Results and Analysis.** TCP and UDP are two commonly used protocols in the network communication. In this paper, these two protocols are used as steganographic carriers to avoid causing the abnormal network traffic. The proposed method has been achieved mainly by using C++, Matlab tools and Scapy, which will be introduced in the following part. First, a feasibility verification is conducted. Second, the bandwidth of the proposed method is measured, and the undetectability is analyzed.

4.1. **Experiment Environment.** The experiment environment is as follows.

A. Network environment: 100Mb/s switched Ethernet, LAN, five sending hosts and six receiving hosts (IP address is shown in Table 1);

Table 1. IP address configuration

IP addresses of the 5 sending hosts	IP addresses of the 6 receiving hosts
192.168.1.11	192.168.1.21
192.168.1.12	192.168.1.22
192.168.1.13	192.168.1.23
192.168.1.14	192.168.1.24
192.168.1.15	192.168.1.25
	192.168.1.26

B. Host configuration: Windows 7 64-bit system, Intel Core i5-2300 CPU, 8G memory, 100Mbps NIC;

C. Tools: Microsoft Visual Studio 2015, Matlab R2013b, WinPcap development kit, Python 2.6.3, Scapy2.3.1;

D. The steganogram: 255*255 grayscale image, the name is 'lena.bmp' (as shown in Figure 10), bit depth of which is 8.

Figure 10. The steganogram

4.2. **Feasibility Verification.** Feasibility verification of the proposed method is carried out first. The steganogram is a grayscale image (as shown in Figure 10), the resolution of which is 255 dpi, and the bit depth of which is 8.

Before all the start, there are some preparations to be done. First, the steganogram needs to be converted into a 2-dimension matrix representation by using Matlab tools (as shown in Figure 11). The size of the matrix is 255*255. The value of each cell is a decimal number that represents the pixel value of the corresponding pixel in the image. The bit depth of the image is 8, which means that each pixel value occupies one byte.

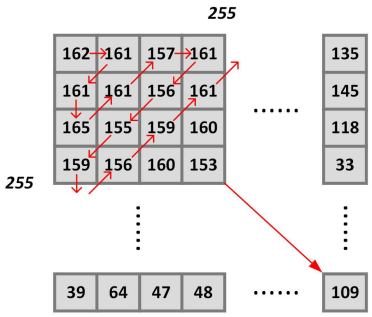

FIGURE 11. The matrix representation of lena.bmp

Second, the dimension should be reduced to get 1-dimension representation by using Zigzag method shown in Figure 11. The value of each cell in the 1-dimension array obtained after the dimensionality reduction is still in decimal form (as shown in Figure 12). Third, the 1-dimension array of decimal representation needs to be convert to a binary representation (as presented in Figure 12). Up to now, the preparations is completed.

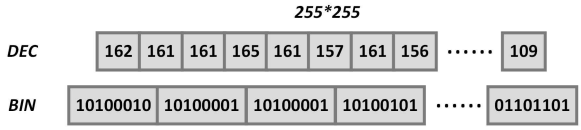

FIGURE 12. Dimensionality reduction

When $N = 5$ and $M = 6$, the sender constructs a sending matrix of 5 rows and 6 columns. And the receiver constructs a receiving matrix the same as the sending matrix. The binary string presented in Figure 12 is filled in the sending matrix row by row. The total length of the binary string is 520200 bits, which is much larger than the size of the sending matrix. So the sending matrix will be used many times. Figure 13 shows the result of the first 30 bits of the binary string are filled in the sending matrix.

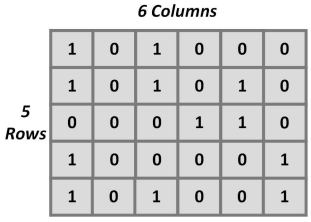

FIGURE 13. The result of the first 30 bits of the binary string are filled in A_s

As described in Algorithm 1, the 5 sending hosts send the secret information bit in the matrix (as shown in Figure 13) to the 6 receiving hosts row by row. After the receiver gets the secret information bits, he/she fills the receiving matrix. Then the receiver gradually recover the original steganogram from the receiving matrix as Algorithm 1 describes.

The comparison of the original steganogram and the extracted steganogram is shown in Figure 14. From an intuitive point of view, the original steganogram and the extracted steganogram are almost exactly the same. From an objective point of view, two indicators are used to measure whether there is a difference between the extracted steganogram and the original steganogram, that is, Bit Error Rate (BER) and Normalized Coefficient (NC). Generally speaking, the smaller the BER is, the smaller the difference will be. The larger the NC, the smaller the difference will be. After calculation, the BER is 0 and NC is 1, which indicating that the extracted steganogram S_e is exactly the same as the original steganogram S.

FIGURE 14. The comparison of the original steganogram and extracted steganogram

The success of one experiment can't effectively demonstrate the feasibility of the proposed method. Under the established experimental conditions, the number of sending hosts and receiving hosts are changed to build several different distributed systems. Assuming that N is in the range of 3 to 5, and M is in the range of 4 to 6. Then there are 9 pairs of different (N, M), which means there will be 9 different distributed systems. A total of 100 experiments are conducted among these 9 different systems, and the results is recorded as shown in Table 2.

TABLE 2. The results of 100 experiments

(N, M)	The matrix size	Times	Average BER(%)
$(3, 4)$	12	10	0
$(3, 5)$	15	10	0
$(3, 6)$	18	10	0
$(4, 4)$	16	10	0
$(4, 5)$	20	10	0
$(4, 6)$	24	10	0
$(5, 4)$	20	10	0
$(5, 5)$	25	10	0
$(5, 6)$	30	20	0

Among all the 100 experiments, the steganogram can be covertly transmitted from the sender to the receiver successfully, and the average BER is 0%. It is proved that the proposed method is feasible.

4.3. **Bandwidth and Undetectability Analyses.** In general, the steganographic bandwidth can be represented by average steganography capacity of each packet (bit/packet). It can be calculated by the formula(1).

$$bandwidth = \frac{size\ of\ the\ steganogram(bits)}{total\ number\ of\ used\ packets} \tag{1}$$

In the ideal network environment (no packet loss and no congestion), the theoretical steganographic bandwidth of the proposed method is 1bit/packet. But in fact, the matrix constructed by the receiver may not be filled up one-time. Some information may be lost during the transmission, so the sender is requested to retransmit. The actually steganographic bandwidth measured in the experiment is about 0.998 bits/packet. Figure 15 shows the comparison of the bandwidth between NS-DCM and other methods. Kundur and Ahsan [33] are the first person who proposed a steganographic method based on packet reordering to encode secret information. Atawy and Shaer [49] invented another kind of covert channel based on packet sorting. He used an out-of-order packet to represent a message. Such representation does not depend on the packet payload. As for the SGS method proposed by Fraczek, he did not give a detailed experimental process, and there is no some experimental results. However, according to his description, SGS method mainly embeds secret information into some fields of the chosen protocols, so the steganographic bandwidth is greater than 1bit/packet.

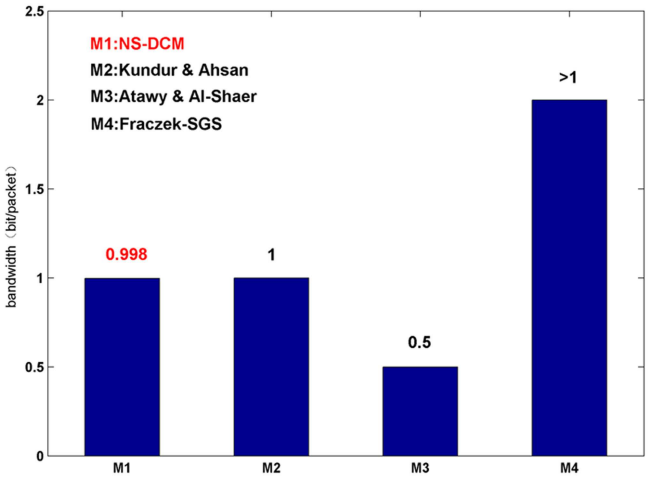

FIGURE 15. The comparison of the bandwidth between NS-DCM and other methods

As can be seen from Figure 15, the steganographic bandwidth of NS-DCM is almost the same as that of Kundur's method. It is higher than the bandwidth of Atawy's method, but lower than the bandwidth of SGS method theoretically. As known to all, steganographic bandwidth and undetectability is two important factors to measure the performance of network steganography. There is a trade-off relationship between these two factors. Increasing the steganographic bandwidth will result in an undetectability drop. So in network steganography, the higher bandwidth does not mean that the method is better. According to the experimental results, NS-DCM method has better undetectability than SGS method.

In addition to analyzing the undetectability based on the steganographic bandwidth, it can also be analyzed according to some detection means. As described in section 2.2, there are mainly 3 steganalysis methods to detect 3 different covert channels as follows.

1) Detecting header field channels: This steganalysis method mainly detects some special fields in the header of a protocol. The NS-DCM method does not embed the secret information into the protocol header field, so this steganalysis method is invalid.

2) Detecting timestamp channels: This steganalysis method mainly detects the randomness of the LSB of the TCP timestamp. Strong randomness indicates the existence of covert channel. The NS-DCM method does not modify the TCP timestamp, so it can resist such steganalysis method.

3) Detecting packet rate and timing channels: This steganalysis method mainly detects the interval between two packets. A regular interval implies the presence of covert channels. The NS-DCM method does not use the time intervals between two packets to represent secret information, so that there is no regularity in the proposed method.

In summary, the NS-DCM method can resist most of the steganalysis methods, and has a stronger undetectability than some other network steganography.

5. **Conclusions.** Deep Hiding Techniques (DHTs) are general techniques that can be used to improve the undetectability of network steganography and make steganogram extraction harder to perform. On the basis of DHTs, a new network steganography based on distributed coding (NS-DCM) is proposed in this paper. Through analyses of experimental results, it is found that the proposed method can achieve a bandwidth of 0.998 bits/packet. Meanwhile, it has a strong undetectability to resist 3 steganalysis methods. Currently, network steganography based on a single protocol or some other simple methods have been developed completely. In the future, the research of network steganography will be focused on multi-protocol fusion and multi-carrier association based on DHTs to further enhance the performance of steganography. In particular, the focus will be on the combination of image steganography and network steganography, because the image steganography is an important aspect of digital media steganography. Weng et al. proposed a novel reversible data hiding (RDH) scheme in 2008 [50], which is based on invariability of the sum of pixel pairs and pairwise difference adjustment (PDA) to significantly reduce the capacity consumed by overhead information. In 2016, Weng et al. further proposed a RDH method based on pixel value ordering (PVO) and prediction-error expansion [51]. The PDA method and PVO method can be used to extend our works to get a high performance.

REFERENCES

[1] K. Szczypiorski, Steganography in TCP/IP networks, *Proceedings of State of the Art and a Proposal of a New System-HICCUPS*, Institute of Telecommunications'seminar, Warsaw University of Technology, Poland, 2003.

[2] F. A. Petitcolas, R. J. Anderson, and M. Kuhn, Information hiding-a survey, *Proceedings of the IEEE*, IEEE, vol. 87, no. 7, pp.1062–1078, 1999.

[3] S. Zander, G. Armitage, and P. Branch, A survey of covert channels and countermeasures in computer network protocols, *Communications Surveys & Tutorials*, IEEE, vol. 9, no. 3, pp.42–57, 2006.

[4] W. Mazurczyk, K. Szczypiorski, and B. Jankowski, Towards steganography detection through network traffic visualisation, *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on. IEEE*, IEEE, pp.947–954, 2012.

[5] W. Mazurczyk, and J. Lubacz, LACK: a VoIP steganographic method, *Telecommunication Systems: Modelling, Analysis, Design and Management*, Springer, vol. 45, no. 2-3, pp.153–163, 2010.

[6] Epinna, StegoSIP, https://github.com/epinna/Stegosip. 2016.

[7] W. Fraczek, W. Mazurczyk, and K. Szczypiorski, How Hidden Can Be Even More Hidden?, *3rd International Conference on Multimedia Information Networking and Security (MINES)*, IEEE, pp.581–585, 2011.
[8] S. Wendzel, S. Zander, B. Fechner and C. Herdin, A pattern-based survey and categorization of network covert channel techniques, *ACM Computing Surveys*, vol. 47, no. 3, pp.50–51, 2015.
[9] C. G. Girling, Covert channels in LAN's, *IEEE Transactions on Software*, vol. 13, no. 2, pp.292–296, 1987.
[10] S. J. Murdoch, S. Lewis, Embedding covert channels into TCP/IP, *In Proceedings of the Information Hiding Conference 2005*, Springer, pp.247–261, 2005.
[11] A. Dyatlov, S. Castro, Exploitation of data streams authorized by a network access control system for arbitrary data transfers: tunneling and covert channels overthe http protocol, *Technical Report*, Gray-World.net, 2005.
[12] R. Rios, J. Onieva, and J. Lopez, HIDE-DHCP:covert communications through network configuration messages, *In Proceedings of the IFIP TC 11 27th International Information Security Conference*, Springer, pp. 162-173, 2012.
[13] X. Zou, Q. Li, SH. Sun, and X. Niu, The research on information hiding based on command sequence of FTP protocol, *International, Proceedings Of The Th Systems, Conference On Knowledge-Based Intelligent*, Springer, pp.1079–1085, 2005.
[14] Z. Trabelsi, I. Jawhar, Covert file transfer protocol based on the IP record route option, *Journal of Information Assurance and Security*, vol. 5, no. 1, pp.64–73, 2010.
[15] T. Graf, Messaging over ipv6 destination options, *http://grayworld.net/papers/messip6.txt*, 2003
[16] S. Wendzel, B. Kahler, and T. Rist, Covert channels and their prevention in building automation protocols: a prototype exemplified using BACnet, *In Proceedings of the 2nd Workshop on Security of Systems and Software Resiliency*, IEEE, pp.731–736, 2012.
[17] N. Lucena, G. Lewandowski, and S. Chapin, Covert channels in IPv6, *In Proceedings of the 5th International Workshop on Privacy Enhancing Technologies*, Springer, pp.147–166, 2006.
[18] S. Zander, G. Armitage, and P. Branch, Covert channels in the IP time to live field, *In Australian Telecommunication Networks and Applications Conference (ATNAC)*, pp.298–302, 2006.
[19] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, Covert messaging through TCP timestamps, *In Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, Springer, pp.194–208, 2003.
[20] T. Handel, and M. Sandford, Hiding data in the OSI network model, *In Proceedings of the First International Workshop on Information Hiding*, Springer, pp.194–208, 2003.
[21] S. Wendzel, Protocol channels as a new design alternative of covert channels, http://arxiv.org/abs/0809.1949, 2008.
[22] M. Wolf, Covert channels in LAN protocols, *In Proceedings of the Workshop on Local Area Network Security (LANSEC)*, pp.89–101, 1989.
[23] M. Mehic, J. Slachta, and M. Voznak, Whispering through DDoS attack, *Perspectives in Science*, Springer, no. 7, pp.95–100, 2016.
[24] N. Lucena, J. Pease, P. Yadollahpour, and S. J. Chapin, Syntax and semanticspreserving application-layer protocol steganography, *In Proceedings of the 6th Information Hiding Workshop*, Springer, pp.164–179, 2004.
[25] L. Yao, X. Zi, L. Pan, and J. Li, A study of on/off timing channel based on packet delay distribution, *Computers & Security*, vol. 28, no. 8, pp.785–794, 2009.
[26] S. Cabuk, C. E. Brodley, and C. Shields, IP covert timing channels: design and detection, *In Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, Springer, pp.178–187, 2004.
[27] M. A. Padlipsky, D. W. Snow, and P. A. Karger, Limitations of end-to-end encryption in secure computer networks, http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=A059221&Location=U2&doc=GetTRDoc.pdf, 1978.
[28] W. Li, and G. He, Towards a protocol for autonomic covert communication, *In Proceedings of the 8th International Conference on Autonomic and Trusted Computing*, Springer, pp.106–117, 2011.
[29] S. H. Sellke, C. Wang, S. Bagchi, and N. B. Shroff, Covert tcp/ip timing channels: theory to implementation, *In Proceedings of the 28th Conference on Computer Communications (INFOCOM)*, IEEE, pp.2204–2212, 2009.
[30] S. Gianvecchio, and H. Wang, Detecting covert timing channels: an entropy-based approach, *In Proceedings of the 14th ACM Conference on Computer and Communication Security (CCS)*, ACM, pp.307–316, 2007.

[31] V. Berk, A. Giani, and G. Cybenko, Detection of covert channel encoding in network packet delays, *Technical Report TR2005-536*, Department of Computer Science, Dartmouth College, http://www.ists.dartmouth.edu/library/149.pdf, 2005.
[32] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, Model-based covert timing channels: automated modeling and evasion, *International Workshop on Recent Advances in Intrusion Detection*, Springer, pp.211–230, 2008.
[33] K. Ahsan, and D. Kundur, Practical data hiding in TCP/IP, *In Proceedings of the ACM Workshop on Multimedia Security*, ACM, vol. 2, no. 7, 2002.
[34] D. Kundur, and K. Ahsan, Practical Internet steganography: data hiding in IP, *In Proceedings of the Texas Workshop on Security of Information Systems*, vol. 2, 2003.
[35] R. C. Chakinala, A. Kumarasubramanian, R. Manokaran, G. Noubir, C. P. Rangan, and R. Sundaram, Steganographic communication in ordered channels, *In Proceedings of the 8th International Workshop on Information Hiding*, Springer, pp.42–57, 2006.
[36] X. Luo, E. W. W. Chan, and R. K. C. Chang, Cloak: a ten-fold Way for reliable covert communications, *In Proceedings of European Symposium on Research in Computer Security (ESORICS)*, Springer, pp.283–298, 2007.
[37] W. Mazurczyk, K. Szczypiorski, Steganography of VoIP streams, *In proceedings of the 3rd International Symposium on Information Security (IS'08)*, Springer, pp. 1001–1018, 2008.
[38] W. Mazurczyk, M. Smolarczyk, and K. Szczypiorski, Retransmission steganography and its detection, *Soft Computing*, vol. 15, no. 3, pp.505–515, 2011.
[39] R. W. Smith, G. S. Knight, Predictable design of network-based covert communication systems, *In Proceedings of the IEEE Symposium on Security and Privacy*, IEEE, pp. 311–321, 2008.
[40] C. H. Rowland, Covert channels in the TCP/IP protocol suite, *First Monday*, vol. 2, no. 5, 1997.
[41] T. Sohn, J. Seo, and J. Moon, A study on the covert channel detection of TCP/IP header using support vector machine, *In Proceedings of 5th International Conference on Information and Communications Security*, Springer, pp. 313–324, 2003.
[42] E. Tumoian, M. Anikeev, Network based detection of passive covert channels in TCP/IP, *In Proceedings of 1st IEEE LCN Workshop on Network Security*, IEEE, pp. 802–809, 2005.
[43] S. Zander, Performance of selected noisy covert channels and their countermeasures in IP networks , *Swinburne University of Technology*, 2010.
[44] A. Hintz, Covert channels in TCP and IP headers, http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-hintz-covert.ppt.
[45] B. R. Venkatraman, R. E. Newman-Wolfe, Capacity estimation and auditability of network covert channels, *In Proceedings of IEEE Symposium on Security and Privacy*, IEEE, pp. 186–198, 1995.
[46] S. Cabuk, C. E. Brodley, and C. Shields, IP covert channel detection, *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 4, pp.22:1-22:29, 2009.
[47] R. M. Stillman, Detecting IP covert timing channels by correlating packet timing with memory content, *In Proceedings of IEEE SoutheastCon*, IEEE, pp. 204–209, 2008.
[48] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski, Information Hiding in Communication Networks:Fundamentals, Mechanisms, Applications, and Countermeasures, *IEEE Press, Wiley*, 2016.
[49] A. El-Atawy, and E. Al-Shaer, Building covert channels over the packet reordering phenomenon, *In Proceedings of the 28th Annual IEEE Conference on Computer Communications (INFOCOM)*, IEEE, pp.2186–2194, 2009.
[50] S. Weng, Y. Zhao, J.S. Pan and R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, *IEEE Signal Processing Letters*, vol. 15, pp.721-724, 2008.
[51] S. Weng, J.S. Pan and L. Li, Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding, *Information Sciences*, vol. 369, pp.144-159, 2016.