A New Network Steganographic Method Based on the Transverse Multi-Protocol Collaboration

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

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

Received December, 2016; revised February, 2017

ABSTRACT. In recent years, network steganography has become a new research direction in the field of information hiding. In this paper, a new network steganographic method called Transverse Multi-Protocol Collaboration Network Steganographic Method (TMPCNSM) is proposed. TMPCNSM is realized in the same layer in the TCP/IP protocol stack. Two kinds of collaboration modes to encode the secret information by multi-protocol permutation are proposed and experiments are conducted with three protocols (HTTP, FTP, and DNS) as the carriers. The experimental results showed that the great advantage of this method over others are higher steganography bandwidth and more undetectable characteristics due to the complexity of multi-protocol collaborative. Keywords: Information Hiding; Network Steganography; Multi-Protocol Collaborative.

1. Introduction. With the development of computer network technology, steganography carriers can no longer be digital images, audio or video files, but may be other file types or data units, such as computer network data packets, frames, etc. Network steganography is gradually attracting attentions of researchers, and has become a new branch of information hiding. The term Network Steganography was first proposed by Szczypiorski [1]. Compared with traditional digital media steganography, network steganography is more undetectable and is difficult to be eliminated in the channel.

The essence of network steganography is to use the protocols in the Open Systems Interconnection (OSI) reference model [2] as carriers to hide secret information. The goal is to hide the communication process, in other words, to hide the existence of the data exchange process. Network steganography uses overt network flow as the carrier of secret information, embeds the secret information in the communication process and ensures minimal impact on the normal communications, so that the transmission of secret information can't be detected by unauthorized third party.

Over the last decade a large number of network steganographic techniques have been developed, but most of them used the concept of redundancy. According to classification based on steganographic pattern proposed by Wendzel et al. [3], we divide the network steganographic technology into four categories: storage methods, timing methods, hybrid methods, and transform methods. Among them, the most widely used methods are storage methods and timing methods. The storage methods either hide the data in the user data (payload) or hide the data in the protocol field (non-payload). The two non-payload modes are as follows. Structure change mode: the structure of the PDU is changed, such as size modulation of PDUs [4, 5], sequence modulation in PDUs [6, 7, 8], add redundancy to PDUs [9, 10] and value modulation in PDUs [11, 12, 13, 14, 15,

16]. Structure retention mode: the structure of the PDU is remained, but the header fields of PDUs, such as reserved/unused bits in PDUs [17, 18] and random values in PDUs [7, 19] are changed. The timing methods encode the secret information by utilizing rate or throughput of network traffic [20, 21, 22, 23], inter-packet times [24, 25, 26, 27], message sequence timing [28, 29, 30, 31], etc. In general, the storage methods are easier to implement, but changing bits in the protocol field may also reduce undetectable. The timing methods are more undetectable, but they are susceptible to channel noise and have low steganography bandwidth.

At the same time, with the deepening of network steganography, the corresponding steganalysis technology is gradually developing. Single protocol has been unable to provide a strong anti-detectability. Zander et al. [32] and Petitcolas et al. [33] proposed several methods for network steganalysis. Mazurczyk proposed steg-tomography [34], a method of steganalysis using traffic visualization analysis in 2012. They chose LACK [35] as the steganographic method to detect the stegoSIP based on it [36], and pointed out that this visualization method can also be used to detect other network steganographic methods.

To further enhance the anti-detection of network steganography, a new network steganographic method is proposed in this paper, which is called Transverse Multi-Protocol Collaborative Network Steganographic Method (TMPCNSM), to realize the safety and reliable transmission of secret information.

This paper is organized as follows. The related work is introduced in section 2, including steganography based on packet reordering and inter-protocol steganography. 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. Packet Reordering Steganography. Packet reordering is a natural occurrence in the Internet. It would hardly be suspected or closely monitored. Besides, attempting to eliminate such reordering would be extremely costly and hard to achieve. Therefore, many researchers propose to use packet reordering for network steganography.

Kundur and Ahsan [30] first proposed to use different sequencing of multiple packets to implement network steganographic method and build a covert channel. For a set of nobjects, there are n! kinds of sequence to arrange all of them, i.e. n! permutations. If all the n! permutations are used to encode different secrete information, there are a maximum of $log_2(n!)$ bits that can be covertly transferred. Figure 1 shows a simple example of encoding hidden information by packet reordering. The covert sender sends a packet at the correct position to encode a zero bit, and a packet in an incorrect position to encode a one bit.

They also explored the information hiding method based on IP packet sorts at the network layer. Unlike the network steganographic method that modifies packet headers, the carrier of this steganographic approach based on packet ordering steganography is the sequency of packets rather than a single packet.

It should be noted that as n increases, the maximum transferable information bits will gradually increase. However, the process is not endless because of a growing complexity of steganographic algorithm with the n increasing. At the same time it will also produce a large amounts of data traffic in the network that may attract the attention of steganography detection.

Chakinala et al. further developed the concept of reordered channels [31] based on Kundur and Ahsan's research and modeled the channel formally.

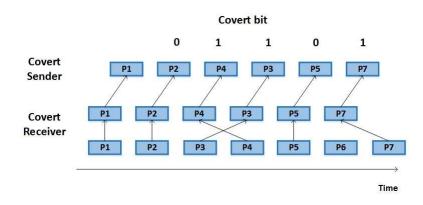


FIGURE 1. An example of (re)ordering packets to encode hidden communication [37]

In 2007, Luo et al [28] pointed out that the covert channels steganography bandwidth is low because of the dynamic network environment and a variety of network monitoring. For example, timing channels based on inter-packet delay are very sensitive to network delay jitter, while packet loss affects both timing and storage channels. On the other hand, the storage channel is not affected by the delay, but the information that is simply encoded may be detected or even modified by the network supervisor. Thus, they proposed Cloak [28], a timing channel that provides reliable covert communication. This covert channel and other time-domain covert channels are fundamentally different. First, Cloak encodes the secret information by uniquely distributing N packets in X TCP streams, which will increase the channel capacity. Second, Cloak offers 10 different encoding and decoding methods, each of which requires trade-offs in several important areas, such as channel capability and packet marking requirements. Third, Cloak-modulated packets can mimic normal TCP traffic and reduce the risk of anomaly detection. At last, even being faced with packet loss, delay jitter, packet reordering and packet replication and other issues, 100% decoding accuracy can still be guaranteed. This reliability is mainly due to using TCP data stream as a carrier, and using a fixed number of TCP packets to encode and decode the secret information which both contribute to avoiding a lot of time-domain channel inherent synchronization errors. According to the authors, Cloak is the first network covert channel that uses enumerated sets to deliver secret information.

10 different encoding methods provided by Cloak are based on the famous Twelvefold Way in the field of counting combination. The Twelvefold Way involves 12 basic counting problems that deal with the number of methods to put N balls into X boxes. Cloak channel can provide reliable covert communication with a certain anti-detection property. However, as mentioned above, steganography bandwidth and anti-detection are two properties of mutual checks and balances. An increase in steganography bandwidth will limit the anti-detection effect. Conversely, an increase in anti-detection effect will limit the steganography bandwidth. Since the Cloak channel has strong anti-detection performance, its steganography bandwidth is not high enough.

Atawy and Al-Shaer invented another kind of covert channel based on packet ordering [38], which used out-of-order packets to represent some information. This representation does not depend on packet payload, and is not sensitive to inter-packet delay jitter. The number of different queues can represent the steganography bandwidth. For traditional circuit-switched networks, once the order of transmission is determined, the order of reception is confirmed, and the channel steganography bandwidth is zero. But for the current packet-switched network, packet out-of-order is a normal phenomenon. Unlike Kundur et al., Atawy notes that for n packets, the steganography bandwidth is between $log_2(n!) \sim nlog_2 n$. For example, to transfer a 700MB file in which each packet size is 1.5KB, then there is 8.7Mbit~8.8Mbit steganography bandwidth in the transmission channel. However, the large number of reordering packets will not only greatly affect the normal communication, but also reduce the anti-detection of the channel. So Atawy chooses a subset of the *n* permutations of packets, that is, only the adjacent packets can be reordered. 0 represent that the adjacent packets in the normal order, and 1 represent that they are exchanged. So that the steganography bandwidth can be compressed to 233Kbit~240Kbit. In other words, the information entropy of this channel is 1bit/symbol, that is, each packet provides 0.5bits steganography bandwidth.

2.2. Inter-Protocol Steganography. A typical network steganographic method is usually to modify a single network protocol. As mentioned above, the steganographic method consists of storage methods, timing methods, and a combination of both. All of them are the intra-protocol steganographic methods. Currently, there are many detection methods that can detect the intra-protocol steganography.

In order to improve the weakness of the intra-protocol steganography, some researchers proposed an Inter-Protocol Steganography (IPS) method. The IPS method was originally proposed by Jankowski et al. [39], which refers to using the relationships between two or more different protocols to build covert channels. These different protocols can be in the same layer or different layers of TCP/IP protocol stack.

Fraczek further summarized the IPS method as one of the deep hiding techniques (DHTs) [40]. DHTs include five general methods that can improve the anti-detection performance of network steganography, making the general network covert channel more subtle, being difficult to be detected, doubted or extracted. An example of the IPS approach is the PadSteg steganography system [41] proposed by Jankowski. In the LAN environment, this system uses ARP, TCP, ICMP and UDP et al. to build covert channel. These protocols are also known as the carrier protocols. That means, the secret information needs to be embedded in the padding bits of the frame, but it must be determined by the upper protocol whether the padding bit contains secret information. Fraczek et al. points out that PadSteg is the first information hiding method to realize inter-protocol steganography, and is an extension of the literature [39].

However, due to the current fuzzy standardization of network protocols, the realization of frame padding mechanism is different among NICs. Besides, some NIC drivers incorrectly pad frames with zeros. Because of memory leaks, the Ethernet frame padding may contain some kernel information. This defect is named as Etherleak [42] by Arkin. The existence of Etherleak provides an opportunity for network steganography, since it is difficult for detectors to distinguish between steganography and Etherleak. Through the experiment, the author found many network cards still contain Etherleaks.

3. The Proposed Method. As described in section 2, packet reordering steganography has strong undetectable performance, but steganography bandwidth is low. In contrast, the Inter-Protocol Steganography has high bandwidth, but some problems still exist. On the basis of these two methods, we propose a new network steganographic method based on the transverse multi-protocol collaboration (TMPCNSM), which belong to IPS method, to obtain a higher steganography bandwidth while ensure a strong anti-detection performance. TMPCNSM selects and make a comprehensive utilization of several protocols in the same layer of the TCP/IP stack to implement network steganography. From simple to complex, there are two phases. In each phase, we proposed one kind of steganographic method, embedding fragmentation into multi-protocol and encoding the secret information by multi-protocol permutation. The first method that in the phase 1 is the foundation of the second method that in the phase 2.

3.1. Embed Fragmentation into Multi-Protocol. This method divides the secret information into several parts. And then each part is embed into different protocols (shown in Figure 2).

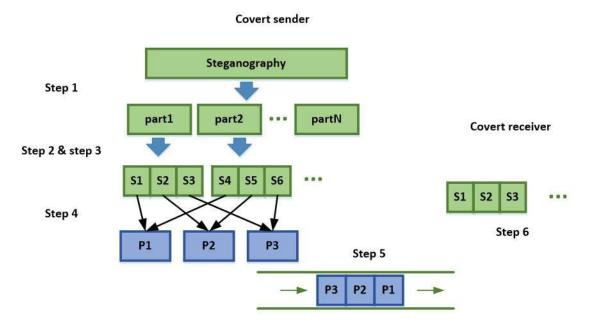


FIGURE 2. Each secret information part embed into different protocols

Suppose that the size of the secret information is S_v , generically $S_v > V$, then one must divide the secret information into multiple parts. The Embed Fragmentation into Multi-Protocol (EF2MP) algorithms are described in Algorithm 1.

As usual steganalysis only monitors a certain kind of network data flow, judges and analyzes whether the data stream contains abnormal information. While in this method, we embed the secret information into many different protocols, which will make the general steganalysis difficult. Meanwhile, a variety of protocols can better imitate nomal network traffic and camouflage covert channel, which will not produce anomalous network traffic. EF2MP method is a simplest kind of steganographic method we have proposed. It is also the foundation of the following method.

3.2. Multi-Protocol Permutation Encoding the Secret Information. As we described in Section 2, the packet reordering steganography and the inter-packet steganography mostly encode secret information according to the different permutations of different protocol packets. The network flow are generally binary string, such as 000100001111100. If we only use one protocol to represent the binary string, then we can denote 1 by transmitting two packets consecutively in a time unit, and denote 0 by transmitting just one packets in a time unit. This encode method is very simple, but we must use 1.5 packets to represent 1bit on average, which will produce a large network load and make the covert channel easy to detect. Further, we use two kinds of protocol to represent 0 and 1, that means P_1 represent 0 and P_2 represent 1. Thus, an information bit requires only one packet rather than 1.5 packets to represent. But this will also induce a larger network load and a loss of data packets, which means a loss of information and will have negative effect on the transmission of confidential information.

Algorithm 1 Embed Fragmentation into Multi-Protocol (EF2MP)

Input:

k protocols $P_i(i = 1, 2, 3...k)$; the hidden capacity of the *i*th protocol v_i ; the size of the secret information S_v ;

Output:

A number of packets containing secret information fragmentation;

- 1: Dividing the secret information into several parts and ensuring the size of each part equal to the capacities of k protocols, one finds the amount of parts $N = \left\lceil \frac{S_v}{V} \right\rceil$. Here V is defined by $V = \sum_{i=1}^k v_i$;
- 2: Continue dividing each part into k fragmentations to make each fragmentation size meet capacity v_i of each protocol;
- 3: Calculate the location ID, S_{id} , of each fragmentation in the original secrete information for step 6. Assume the fragmentation in part n(n = 1, 2, N) will be insert into *i*th protocol, then $S_{id} = i + (n - 1) \times k$;
- 4: As V is an aliquant part of S_v , the remnant information, which size is $S_v modV$, should be divided and inserted into each protocol according to step 3 completely;
- 5: Covert sender should send every packet that contain secrete information fragmentation to covert receiver consequently;
- 6: The covert receiver should reassemble secrete information according to S_{id} after successfully reception.

More generally, if we have select k kinds of protocols in one TCP/IP layer. The *i*th protocol is represented by $P_i(i = 1, 2, 3...k)$. The k protocols can have a variety of permutations to represent the different coding. If each protocol in the k protocols is not reused, then the total number of permutations is A(k, k), that is k!. Each of the permutations can represent up to $\lceil \log_2 k! \rceil$ bits. So we need $\frac{k}{\lceil \log_2 k! \rceil}$ packets to represent one bit. We assume that the value of k change from 2 to 100, then the number of packets to represent one bit is shown in Figure 3. It can be seen from the Figure 3 that as

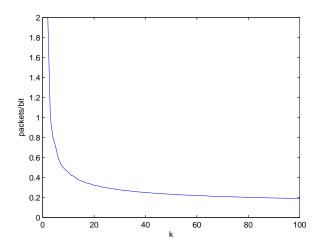


FIGURE 3. Indicates the trend of the number of packets required for an information bit

the number of selected protocol types increases, the number of packets required for each information bit is gradually decreased. However, the available protocols is limited. If we

use same protocol excessively, there will be significant changes in network traffic. To solve such problem, we propose the following two modes in this phase.

Mode A: Multi-protocol permutation encoding existence of fragmentation(MPPEEOF) Before establishing the covert channel, the covert sender and receiver share a same relationship table. Assume that the covert sender has select 3 protocols as carriers, then the relationship between the multi-protocol permutation and the binary string is shown as Table 1. Different from k! permutations of k protocols mentioned above, here each permutation should represent k bits binary string, so the total number of the string is 2^k . Along with the increase of k, 2^k will far larger than k!. That means there will be an amount of binary string that cannot be represented by one permutation. So each protocol should be reused in each permutation. Then the number of permutations will be up to k^k . When k > 2, there is $k^k > 2^k$. It ensures that each binary string can be uniquely represented by one permutation of packets.

Multi-protocol permutation	Binary string
$P_1P_2P_3$	001
$P_1P_3P_2$	010
$P_2P_1P_3$	100
$P_2P_3P_1$	011
$P_3P_1P_2$	101
$P_{3}P_{2}P_{1}$	110
$P_1P_1P_1$	111
$P_3P_3P_3$	000
$P_2P_2P_2$	

TABLE 1. The relationship between the multi-protocol permutation and the binary string

Figure 4 shows the basic idea of mode A.

The Multi-protocol permutation encoding existence of fragmentation (MPPEEOF) algorithms are described in Algorithm 2.

Mode B: Multi-protocol permutation encoding the location of fragmentation (MPPELOF)

Similar to mode A, the covert sender and receiver share a same relationship table before establishing the covert channel. But in this mode, the permutation of protocols is mapping to the location ID of fragmentation in the original information rather than one binary string. The relationship table is shown as Table 2. There are k^k permutations of k protocols, mapping the location ID from 1 to k^k .

Figure 5 shows the basic idea of mode B. The covert sender divides the secret information into several fragmentations and record location ID of each fragmentation. Then arrange all the fragmentations out of order. At last, insert each fragmentation into a packet and send it to the covert receiver according to Table 2.

The Multi-protocol permutation encoding the location of fragmentation(MPPELOF) algorithms are described in Algorithm 3.

Neither of two modes above uses the permutations of protocols to encode the secret information directly. Instead the permutations of protocols are used to control the dividing and reassembling of secrete information. This method can provide a higher bandwidth than others, such as that of Kundur et al. At the same time, the method will be more undetectable.

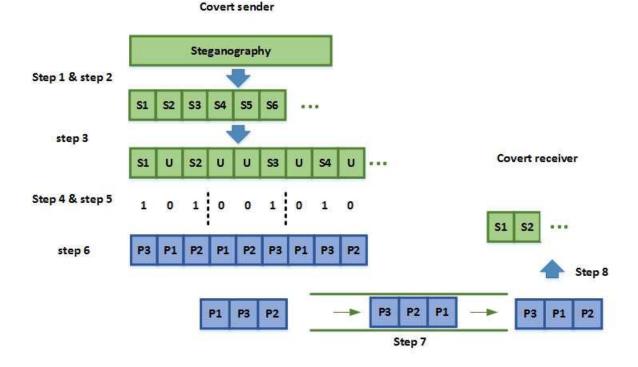


FIGURE 4. Encoding existence of fragmentation by permutation

Algorithm 2 Multi-protocol permutation encoding existence of fragmentation (MPPEEOF)

Input:

k protocols $P_i(i = 1, 2, 3...k);$

the hidden capacity of the *i*th protocol v_i ;

the size of the secret information S_v ;

Output:

A number of packets containing secret information fragmentation;

- 1: Assume each protocols hidden capacity \bar{v} is defined by $\bar{v} = min(v_1, v_2, v_3, ..., v_k)$;
- 2: Divide the secret information into n fragmentations with equal length of \bar{v} , i.e. $n = \left\lceil \frac{S_v}{\bar{v}} \right\rceil$. Mark each fragmentation as $S_i (i = 1, 2, 3, ..., n)$;
- 3: Insert *m* fragmentations *U*, i.e. *U*, *U* which do not carry secret information, between each two adjacent secret information fragmentations and get a new sequence like $S_1U, US_2U, U, U, U, S_3$. Here *m* is a random value between 0 and *k*;
- 4: In the sequence mark S_i (i = 1, 2, 3, ..., n) as 1 and each U as 0. Then one gets a binary string;
- 5: Divide the binary string into several parts. Each part contains k bits;
- 6: According to the mapping relationship in Table 1, find the protocol permutation of each part obtained in step 5 and then insert each fragmentation into corresponding packet;
- 7: The covert sender should send every packet to the covert receiver consequently;
- 8: The covert receiver should convert the received packet sequence back to binary string according to step 4 and extract the secret information fragmentations to reassemble.

Multi-protocol permutation	Location ID
$P_1P_2P_3$	1
$P_1P_3P_2$	2
$P_2P_1P_3$	3
$P_2P_3P_1$	4
$P_3P_1P_2$	5
$P_3P_2P_1$	6
$P_1P_1P_1$	7
$P_3P_3P_3$	8
$P_2P_2P_2$	9

Covert sender

TABLE 2. The relationship between the multi-protocol permutation and the location ID of secret information fragmentation

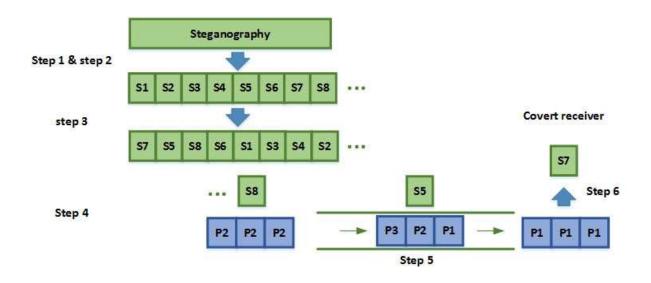


FIGURE 5. Encode location ID by permutation

4. Experimental Results and Analysis. At present, the network steganographic methods mainly focus on the reserved fields, padding fields or optional fields of network layer and transport layer protocols, such as IP, ICMP, TCP, and UDP, et al. These methods have disadvantages of low stegnographic bandwidth, complex process, being easy to detect, and being difficulty to traverse NAT and firewall. There are a lot of protocols in the application layer, and existing firewalls and routers generally do not check the application protocols, which will make it easy to implement network steganography. The mainly application layer protocols include HTTP, FTP, SMTP, P2P, DNS and several other classes.

In this paper, we choose HTTP, FTP, and DNS as carriers to hide information. HTTP traffic is second only to P2P protocol. It is one of the most widely used application protocols on the Internet. At the same time it has a complex header for information hiding. The FTP has a larger set of commands. There is a special type among them. In some specific conditions, such as receiving commands of NOOP, ALLO, and ABOR, the server just receives but do nothing. The NOOP command prevent the FTP control connection from being disconnected for a long period of time. The server does not process

Algorithm 3 Multi-protocol permutation encoding location of fragmentation (MP-PELOF)

Input:

k protocols $P_i(i = 1, 2, 3...k);$

the hidden capacity of the *i*th protocol v_i ;

the size of the secret information S_v ;

Output:

A number of packets containing secret information fragmentation;

- 1: The hidden capacity of each permutations V is defined by $V = (v_1 + v_2 + v_3 + ... + v_k);$
- 2: Divide the secret information into *n* fragmentations with equal length of *V*, i.e. $n = \left\lceil \frac{S_v}{V} \right\rceil$. Mark each fragmentation as $S_i (i = 1, 2, 3, ..., n)$;
- 3: Reorder all the fragmentations to make them out of order. Find the permutation of each fragmentation in Table 2 according to its location ID;
- 4: Insert each fragmentation into corresponding packet;
- 5: The covert sender should send every packet to the covert receiver consequently;
- 6: The covert receiver should convert the received packet sequence back to location ID according to Table 2 and extract the secret information fragmentations to reassemble.

any received NOOP commands. The ALLO command is used to request the server to allocate space and receive files. For some servers which do not allow uploading files, the ALLO command has the same effect as the NOOP command. ABOR command terminates the command which the server has not completed. If the previous command has been completed, the server would do nothing. In certain specific case, all the three commands tell the server not to carry out any treatment [43, 44]. The DNS protocol is also one of the usual protocols in the Internet. We can construct a DNS request message carrying the secret information to imitate the normal DNS packet. The advantage of using DNS for steganography is that there is no effective detection for DNS protocol, and most firewalls do not filter DNS protocol.

4.1. Experiment 1:Embed Fragmentation into Multi-Protocol. The experiment environment is as follows.

A. Network environment: 100Mb/s switched Ethernet, LAN, two hosts (One is the sender, which IP address is 192.168.1.17. The other is the receiver, which IP address is 192.168.1.18);

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

C. Tools: Microsoft Visual C++ 6.0, WinPcap development kit, Python 2.6.3, Scapy2.3.1.

According to the characteristics of HTTP, FTP and DNS protocols and what was described in the section 3.1, the total capacity of three protocols is given by $V = v_{HTTP} + v_{FTP} + v_{DNS}$. Among them, $v_{FTP} = 1bit/packet$, $v_{DNS} = 4bits/packet$. For the HTTP protocol, some researchers claim that it can be hidden up to 60bytes in one packet, that is, 480bits. But hiding so much secret information in a packet will increase the risk of detection, so in this experiment the upper limit of secret information that a HTTP packet can hide is set to be $v_{HTTP} = 60bits/packet$. Then V = 65bits.

The transmission rate of the secret information is strongly related to the sending rate of the packet. With a determined steganographic capacity, the transmission rate of the secret information will increase with the packet sending rate. It is not generalized to use the transmission rate for the steganography bandwidth description. Therefore, the average steganography capacity of each packet (bit/packet) is used to represent steganography bandwidth. In the ideal network environment (network congestion-free, no packet loss), the steganography bandwidth is 3.79bits/packet after several experiments. Figure 6 shows the comparison of the bandwidth between EF2MP and other methods. IT can be seen from this figure that resulted bandwidth of our method is only half of using only the HTTP. However, using the HTTP for steganography only will generate a large amount of HTTP traffic within a short time, which will make it easy to be detected. EF2MP uses a variety of protocols as carriers to reduce the flow of each protocol and lower the risk of being detected. The bandwidth of EF2MP is higher than the methods which Kundur and Atawy proposed. Our bandwidth is close to that of Anands method. However, the multi-protocols we selected as carriers make our method more undetectable than his.

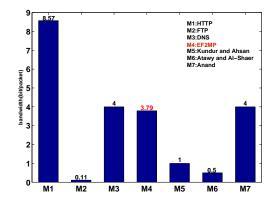


FIGURE 6. Compare EF2MP with other methods

4.2. Experiment 2:Multi-Protocol Permutation Encoding the Secret Information. On the basis of experiment 1, we use the permutation of multi-protocols to encode the secret information, and carried out two encoding modes. The experimental environment is the same as experiment one.

Mode A: MPPEEOF

In this steganography, the average bandwidth of our method is 1.94 bits/packet as proved by repeated experiments. The comparison between the bandwidth of our method and others is shown in figure 7. It shows that the bandwidth of MPPEEOF is lower than that of EF2MP. The reason of bandwidth decreasing is that we insert a number of packets which do not carry secret information between each two adjacent packets. This increases the total amount of sending packets but reduces bandwidth. According to the trade-off relation between the bandwidth and the anti-detection, the anti-detection performance of our method will get promotion.

Mode B: MPPELOF

In this mode our average bandwidth is higher, which is 3.82 bits/packet. Figure 8 shows a comparison of the method MPPELOF with MPPEEOF in mode A and several other methods.

As shown in Figure 8, the bandwidth of the method MPPELOF is higher than the method MPPEEOF, and even slightly higher than the method EF2MP. But MPPELOF uses the permutation of multi-protocol to encode the position of the secret information fragment. So it is well disguised and performs better in anti-detection than the method EF2MP.

4.3. Analysis of undetectable. First of all, the three protocols selected in this paper belong to the application layer. Currently, most firewalls and gateways do not filter and

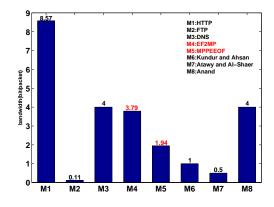


FIGURE 7. Compare Mode A(MPPEEOF) with other methods

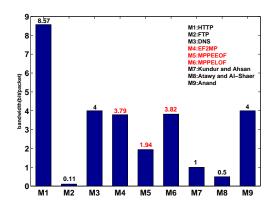


FIGURE 8. Compare Mode B(MPPEEOF) with Mode A and other methods

detect application layer protocols. Therefore, covert channels built using application layer protocols can pass through firewalls without being detected.

Secondly, there is no tool for steganography detection in real-life networks. Every member of a hidden group can exchange almost unlimited number of steganograms and remain undiscovered. Although there are no effective tools, some researchers still propose three steganalysis methods as follows.

1) Header field channels: This method mainly detects some special field in the header. In this paper, the secret information is not embedded in the protocol header field, so this detection method is invalid.

2) Timestamp channels: This method mainly detects the randomness of the LSB of the TCP timestamp. Strong randomness indicates the existence of covert channel. All the methods we proposed do not modify the TCP timestamp, so they can resist such detection method.

3) Packet rate and timing channels: This method mainly detects the interval between two packets. A regular delay implies the presence of covert channels. There is no deliberate setting of the time intervals between two packets, so that there is no regularity in our methods.

In summary, the proposed steganographic method can resist most of the detection methods, and has a certain anti-detection property. 5. Conclusions. In this paper, we propose a new network steganographic method, which uses multi-protocol collaboration in the same layer of the TCP/IP stack. Based on this method, there are two kinds of implementations. One is MPPEEOF and the other is MPPELOF. We select three protocols (HTTP, FTP and DNS), all in the application layer, as carriers. Through analyses of experimental results, it is found that our method can achieve a higher steganography bandwidth than others. At the same time, owe to the collaboration of multi-protocols, our method is more undectectable. Our method is belong to the IPS method, and extend the DHTs. In the future, we will develop more multi-protocol collaboration steganographic methods in the network layer and the transport layer. Also, there are some noisy channels in the real network communication environment. We are going to use vector quantisation (VQ) method [45] and multiple description vector quantisation (MDVQ) [46] to improve the robustness of our methods.

REFERENCES

- 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.
- H. Zimmermann, OSI reference model-the ISO model of architecture for open systems interconnection, *IEEE Transactions on Communications*, vol.28, no.4, pp.425–432, 1980.
- [3] 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.
- [4] C. G. Girling, Covert channels in LAN's, *IEEE Transactions on Software*, vol.13, no.2, pp.292–296, 1987.
- [5] S. J. Murdoch, S. Lewis, Embedding covert channels into TCP/IP, In Proceedings of the Information Hiding Conference 2005, Springer, pp.247–261, 2005.
- [6] A. Dyatlov, S. Castro, Exploitation of data streams authorized by a network access control system for arbitrary data transfers: tunneling and covert channels over the http protocol, *Technical Report*, Gray-World.net, 2005.
- [7] 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, 2012.
- [8] 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.
- [9] 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.
- [10] Z. Trabelsi, I. Jawhar, Messaging over ipv6 destination options, http://grayworld.net/papers/messip6.txt, 2003
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] S. Wendzel, Protocol channels as a new design alternative of covert channels, http://arxiv.org/abs/0809.1949, 2008.
- [17] M. Wolf, Covert channels in LAN protocols, In Proceedings of the Workshop on Local Area Network Security (LANSEC), pp.91–101, 1989.

- [18] M. Mehic, J. Slachta, and M. Voznak, Whispering through DDoS attack, Perspectives in Science, Springer, no.7, pp.95–100, 2016.
- [19] N. Lucena, J. Pease, P. Yadollahpour, and S. J. Chapin, Syntax and semanticspreserving applicationlayer protocol steganography, In Proceedings of the 6th Information Hiding Workshop, Springer, pp.164–179, 2004.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] StegoSIP, https://github.com/epinna/Stegosip..
- [37] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, K. Szczypiorski, Information Hiding in Communication Networks:Fundamentals, Mechanisms, Applications, and Countermeasures, *IEEE Press*, Wiley, 2016.
- [38] 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.
- [39] B. Jankowski, W. Mazurczyk, and K. Szczypiorski, Information Hiding Using Improper Frame Padding, In Proc. of 14th International Telecommunications Network Strategy and Planning Symposium (Networks 2010), IEEE, pp.27–30, Warsaw, Poland, 2010.

- [40] W. Fraczek, W. Mazurczyk, and K. Szczypiorski, How Hidden Can Be Even More Hidden?, In 3rd International Conference on Multimedia Information Networking and Security (MINES), IEEE, pp.581–585, 2011.
- [41] B. Jankowski, W. Mazurczyk, and K. Szczypiorski, PadSteg:Introducing inter-protocol steganography, *Telecommunication Systems: Modeling, Analysis, Design and Management*, ACM, Vol.52, No.2, pp.1101–1111, 2013.
- [42] O. Arkin, and J. Anderson, Ethernet frame padding information leakage, *Atstake report*, http://packetstorm.codar.com.br/advisories/atstake/atstake_etherleak_report.pdf, 2003.
- [43] X. G. Zou, Covert channel based on FTP protocol sequences of commands, Journal of Harbin institute of technology university, Vol.39, No.3, pp.424–426, 2007.
- [44] J. T. Zhai, Information hiding technology research of network communication, Nanjing university of science and technology, 2008.
- [45] J. S. Pan, M. T. Sung, H. C. Huang, B. Y. Liao, Robust VQ-based Digital Watermarking for Memoryless Binary Symmetric Channel, *IEICE Transactions on Fundamentals of Electronics, Commu*nication and Computer Sciences, vol.E-87-A, no.7, pp.1839-1841, 2004.
- [46] J. S. Pan, Y. C. Hsin, H. C. Huang, K. C. Huang, Robust Image Watermarking Based on Multiple Description Vector Quantization, *Electronics Letters*, Vol.40, No.22, pp. 1409-1410, 2004.