The ISC Int'l Journal of Information Security

ISeCure

January 2020, Volume 12, Number 1 (pp. 24–34)

http://www.isecure-journal.org

New High Secure Network Steganography Method Based on Packet Length

Vajiheh Sabeti [1,*], and Minoo Shoaei [2]

[1] *Assistant Professor in the Department of Engineering and Technology, Alzahra University, Tehran, Iran*

[2] *M.Sc. Student in the Department of Engineering and Technology, Alzahra University, Tehran, Iran*

ARTICLE INFO.

Article history:

Received: 16 July 2019

Revised: 27 December 2019

Accepted: 20 January 2020

Published Online: 30 January 2020

Keywords:

Covert Channel, Data Security, Network Steganography, Packet Length, Steganalysis.

Abstract

In the network steganography methods based on packet length, the length of the packets is used as a carrier for exchanging secret messages. Existing methods in this area are vulnerable to detections due to abnormal network traffic behaviors. The main goal of this paper is to propose a method which has great resistance to network traffic detections. In the first proposed method, the sender embeds a bit of data in each pair that includes two non-identical packet lengths. In the current situation, if the first packet length of the pair is larger than the second one, it shows a '1' bit, and otherwise, it shows a '0' bit. If the intended bit of the sender is in conflict with the current status, he/she will create the desired status by swapping the packet lengths. In this method, the paired packets can be selected freely, but in the second proposed method, the packets are divided into buckets, and only packets within a single bucket can be paired together. In this case, the embedding method is similar to the previous one. The results show that the second method, despite having low embedding capacity, will be more secure in real traffic compared to the other methods. Since the packet lengths of UDP protocol are more random in comparison to TCP, the proposed methods have higher embedding capacity, and they are more secure for UDP-based packets. However, these methods are only applicable to the protocols in which the packet length has not a constant value.

1 Introduction

The usage of the Internet as a global way of exchanging information and establishing remote communication is Increasing daily. This leads the subject of information security to be of interest to everyone, therefore, preventing unauthorized access to confidential data becomes a challenge.

The science of information hiding is a method suggested aiming to fulfill the need for communication security. In 1996, at first, information hiding educational workshops held at Cambridge, the concept of information hiding, and the classification of its related techniques were accepted [1]. Steganography is a branch of this science that plays an important role in intending to provide information security [2].

In general, steganographic methods consist of two embedding and extracting processes. A suitable carrier media is needed for the embedding process. Currently, text, audio, and video are the most common carriers. The term "network steganography" was invented by

* Corresponding author.

Email addresses: v.sabeti@alzahra.ac.ir, minoo.shoaei94@gmail.com

ISSN: 2008-2045

Szczypiorski for the first time [3]. Network steganography techniques use public network traffic as a carrier for secret data. They hide secret data in public communication in such a way that its impact on public sending becomes minimal. So, the hidden sending, which is commonly referred to as a covert channel, becomes effectively covered [4]. The concept of covert channels was introduced for the first time by Lampson in 1973 [3].

The scientific community has been using many terms such as steganography [5], [6], covert channels [7], [8], or information hiding [9] to describe the process of concealing information. This stems from the fact that the terms have not been introduced at the same time, and their definitions have evolved over time. However, drawing a distinction between steganography and covert channels is artificial, and instead, one term – network steganography – should be used. It is our belief that steganographic methods are used to create a covert channel, but that covert channels do not exist without steganography. Thus, the scope of network steganography encompasses all techniques of exchanging hidden data [4, 9, 10].

Each network steganography method can be characterized by four features: first, steganographic bandwidth, which describes the amount of secret data we are able to send using a particular method per time unit. Second, undetectability which is defined as an inability to detect a steganogram inside certain carriers. The most popular way to detect a steganogram is to analyze the statistical properties of the captured data and to compare them to the typical properties of the carrier caused by the steganogram insertion procedure. Third, the steganographic cost which depends on the type of the carrier, and if it becomes excessive, it leads to easy detection of the steganographic methods. Fourth, robustness which is defined as the amount of alteration that covert data, can withstand without being destroyed. This feature becomes useless when the channel is safe [4, 11, 12].

Several network steganography methods have been proposed so far [11–19]. Network steganography based on the packet length is a branch of the network steganography methods that transfers the bits of the intended message by modifying the length of the packets in the network. In other words, in this case, the length of the packets is used as a carrier for exchanging secret messages. Although steganographic methods based on the packet length have great resistance against intrusion, but they fail to deliver network normal traffic especially when the number of the hidden message bits is increased [20].

Although several network steganography methods have been presented so far with the aim of embedding in packet length [20–28], the number of steganalytical attacks designed for detecting these methods is very low [29, 30]. These attacks can distinct normal traffic from the traffic carrying hidden data (stego traffic) through extracting some features from network traffic and with the help of a classifier. The attack in [31] uses two features extracted from packets length histogram in two cases of normal and stego traffic to achieve this goal. But the attack in [32], in addition to the previous attack features, also uses two other features extracted directly from normal and stego traffic packet lengths and improves the function of the previous attack. For this reason, this attack is used for testing stage.

In this paper, two new methods have been presented for embedding in packet length. The first one chooses a pair of packets. In this pair, if the first packet length is larger than the second one, it carries the data bit '1', and if it is smaller, then it carries the data bit '0'. If the condition of the pair does not fit the intended data bit, then desired conditions will be provided only by swapping the packet lengths. The idea of the second method is the same, and it only adds a constraint to the packet pairing method. In these methods, the values of packet lengths do not change, and they are just swapped if necessary.

In the following, a number of network steganography methods based on packet length and the most successful attack for detecting these methods will be presented in section 2. The algorithms for two suggested methods will be explained thoroughly in section 3. In section 4, the functionality of two suggested methods will be compared to the previous methods for a few different and real datasets, and in addition, the best method between these two methods will be presented. Finally, the conclusion will be explained in section 5.

2 A Review on the Methods based on the Packet Length

There are many network steganography methods, and different classifications have been proposed for them. One of the complete classifications is based on the way of embedding data in the carrier and is divided into two main categories: "storage" and "timing". There is also a "hybrid" category that combines two previous methods. Most existing methods belong to the first category. Storage methods hide data by modifying protocol fields, such as unused bits of a header. In general, high capacity can be achieved by using this method. Timing methods hide data in the timing of protocol messages or packets. In the network environment, this is often done through changing packet rates or changing packet delays. Timing channels are difficult to identify [4].

The scheme of length-based methods is one of the most interesting branches of designing network covert channels since it highly increases the resistance

against infiltration. In the following, a few steganographic methods will be introduced briefly and at last, the attack in [32] will be presented as the most successful steganalytical method for detecting the methods based on the packet length.

2.1 Ji Method

In [24], Ji et al. designed a protocol-independent covert channel by mimicking legitimate traffic. Let S represent a K-bit binary stream. The secret message is usually divided into several subgroups for being sent. Let W_i be the j^{th} subgroup of S with w-bits.
Alice and Bob follow five steps for transferring the secret message:
First step: Alice and Bob communicate normally. They both capture the message lengths sent by Alice and consider them as references.
Second step: Alice and Bob choose a length l from the reference by the same randomized algorithm.
Third step: in the i^{th} transmission, Alice sends Bob a message with the length of $l_{next} = l + SUM_i$. The reference is updated by adding $l_{next} \cdot SUM_i$ is calculated as in (1):

$$SUM_i = \begin{cases} [W_i]_{10} - 2^w - 1 & i\%2 = 0 \\ [W_i]_{10} - (2^w - 1 - 1) & i\%2 = 1 \end{cases} \quad (1)$$

Fourth step: Bob decrypts the i^{th} message by subtracting l of l_{next}.
Fifth step: steps 2 to 4 are repeated until the entire message is encrypted.
The shortcoming of this scheme is that the packet length distribution of the reference will gradually deviate from the normal distribution because of the continued appending operation.

2.2 NTNCC

As an improvement, Ji et al. [25] proposed another protocol-independent network covert channel based on packet length using the same model. Let S be a k-bit binary secret. The secret message S is usually divided into several subgroups with the size of w bits for each of them. W_i is the i^{th} subgroup that is being sent. The communication is being accomplished in 5 phases:
Phase 1: Alice sends N normal messages to Bob, and they both capture the message lengths as references.
Phase 2: Alice and Bob both sort the reference and divide it uniformly into 2^w buckets.
Phase 3: for the subgroup W_i, Alice converts W_i into decimal $W_i s$ and chooses a length l from the W_i bucket randomly. Then Alice sends Bob a message with the size of l.
Phase 4: Bob decrypts the i^{th} message by checking whether it belongs to the bucket length range.
Phase 5: phases 3 and 4 are being repeated until the entire message is sent.
Different from the previous one, this scheme can imitate the normal packet length histogram well. However, it cannot resist the detection method based on the second-order statistics [32].

2.3 Parity Method

Abdullaziz et al. in [27], proposed a method (with the embedding capacity of one bit in each packet) in which the secret message is expressed easily based on the parity of packet payload byte numbers. If the size of the payload is an even number (in bytes), then it encodes '0' bit of the secret message. On the other hand, if the size of the payload is an odd number (in bytes), then it represents bit '1' of the message. Thus the sender changes data sizes that do not match the desired data size, only by embedding one byte. As the authors say, although this bandwidth seems to be low compared to the other covert channel based on the packet lengths, it guarantees high undetectability, and it does not need a hidden key like a search table to realize a hidden connection. In [30, 33], two particular steganalytical methods are proposed which have been succeeded in detecting this method, though.

2.4 Empirical Distribution Function (EDF)

In [28], a method is proposed based on the empirical distribution function of packet length series taken from legitimate traffic. The empirical distribution function of the series $\{x_1, x_2, \ldots, x_n\}$ is defined as (2):

$$F(t) = \frac{1}{n} \sum_{i=1}^{n} l\{x_i \leq t\} \quad (2)$$

In which $l\{A\}$ represents the event of A and n is the number of instances in series $\{x_1, x_2, \ldots, x_n\}$.
With the assumption of embedding k bits in each packet, the steps of [28] method are as follows:

(1) Both the sender and receiver capture packet length series in normal communication.
(2) They calculate EDF for captured packet length series.
(3) The range of packet length values of normal traffic is divided into 2^k subranges in a way that the range of EDF function values is almost uniform for lengths in each range.
(4) If the decimal value of the intended k-bit data is n, then the stego packet length must be chosen in a way that exists in the n^{th} subrange. So that the receiver can extract data by receiving it and calculating corresponding EDF value.

The main purpose of this method is creating a stego traffic which its EDF value is as close to normal traffic

value as possible. The algorithm of embedding and extraction of this method is explained in details in [28].

2.5 PLD Method (Embedding in Packet Length Differences)

In [20], Sabeti et al. suggested a new method called Packet Length Differencing (PLD) network steganography. PLD method uses the idea of embedding the message in the differences between the consecutive packet pair lengths. Suppose that a block of two adjacent packets P_i and P_{i+1} with lengths l_i and l_{i+1} is chosen. The difference d in this block is equal to (3):

$$d = l_{i+1} - l_i \tag{3}$$

The amount of information to be embedded in each block depends on d value. For this purpose, the d range is divided into n subranges and a k index is assigned to each range. The capacity of each range, i.e., the number of embeddable bits, depends on its length.
Suppose that a two-packet block with the difference d is chosen and d belongs to the range k In order to embed n bits of the message bitstream in a block with a decimal value b, new d value should be calculated. Then new (l'_i, l'_{i+1}) are being calculated for the packet lengths in the stego stream. The algorithm of embedding and extraction of this method is explained in detail in [20].

2.6 Steganalysis of the Methods for Embedding in Packet Length

Recently, Sur et al. [32] presented a detection scheme for network covert channel based on the packet length. In this method, in order to distinguish the normal traffic from stego traffic, four features are used: standard deviation (δ) and center of mass (COM) are extracted from the histogram of packet length sequence (X) as first order features. Calculation of these two standards is done using (4) and (5):

$$COM = \frac{\sum_{i=0}^{m} iXdft}{\sum_{i=0}^{m} Xdft} \tag{4}$$

$$\delta = \sum_{n} |2X(e) - X(e-1) - X(e+1)| \tag{5}$$

In (4), $Xdft$ is DFT conversion of X and m is the length of this vector. In (5), n is number of columns (bins) in X and e is the position of a maximum or a minimum.
The adjacency histogram $H_l^d(P)$ is defined as the number of times that two packets with the distance l and the absolute difference d appear in packet length series P. This parameter is used for calculating the third and the fourth standards and its normalized form $R_l^d(P)$ is defined as (6):

$$R_l^d(P) = \frac{H_l^d}{Len(P)} \tag{6}$$

Here, $Len(P)$ is the number of packets.
The normalized adjacency histogram of normal $(R_l^d(N))$ and stego $(R_l^d(S))$ packet sequences with $l = 1, 2$ and $d = 0$(i.e. R_1^0, R_2^0) are taken as second-order statistical features.
The network legitimate and stego traffic features mentioned above are used for training a supervised learning-based classifier. Sur's experimental results show that the proposed classifier could detect existing covert channels based on packet length with high accuracy.

3 Our Proposed Methods

The main problem of existing network steganography methods based on packet length is their inability to create carrier data traffic (stego traffic) with features similar to a normal one. This weak point appears more when the number of intended message bits is increased. There exists a successful steganalytical method that distinguishes normal traffic from the traffic carrying hidden data by extracting some features of network traffic and with the help of a classifier [32]. In the following, two methods are proposed which existing attacks are less likely to detect them in real situations. The second proposed method is the improved version of the first one that works much better. Both methods are explained in the following:

3.1 Network Steganography Method based on Packet Length Using Relocations (swap-PLN)

Although most image steganography methods are also used for embedding in packet length, but the image field methods have limitations due to the nature of the images. There are dependencies between image pixels so that changing them excessively or moving them would reduce image quality metrics such as $PSNR$ and MSE. As a result, it is not possible to use some image embedding ideas. One of these ideas is to swap two pixels of the image in such a way that the resulting composition can represent a particular bit of data. But due to the lack of these restrictions or the much fewer constraints in a network steganography method based on packet length, this idea will be useful. Suppose P is a normal network traffic that the sender knows about. There is no need to inform the receiver of normal traffic. The sender embeds data in P using the intended algorithm and by creating stego traffic S, he/she prepares packets with this traffic lengths and sends them to the receiver. P includes L_P packets and D includes L_D one and zero data

bits. The steps of the embedding algorithm are shown as follows:

(1) $S \leftarrow P$
(2) According to (7), using the key agreed between the sender and the receiver, the initial kernel of the rand function is initialized:

$$rng(key) \tag{7}$$

(3) Using (8), a random permutation of numbers 1 to L_P is generated, which has 2^*L_D members:

$$seq \leftarrow randperm(L_P, 2 * L_D) \tag{8}$$

(4) The following steps are repeated in order to embed each data bit $d_k(\forall k1 \leq k \leq L_D)$:
   (a) Using (9) and (10), two packet lengths are specified for embedding:

$$l_1 \leftarrow P[seq[2k-1]] \tag{9}$$

$$l_2 \leftarrow P[seq[2k]] \tag{10}$$

   (b) If $(l_1 < l_2, d_k = 1)$ or $(l_1 > l_2, d_k = 0)$ then l_1 and l_2 in stego traffic S, should be swapped using (11) and (12):

$$S[seq[2k-1]] = l_2 \tag{11}$$

$$S[seq[2k]] = l_1 \tag{12}$$

The idea of this method is very simple. Figure 1, shows an example of how to perform this algorithm. A pair of packets is chosen. If the first packet length is larger than the second one, then this pair carries the data bit '1', and if it is smaller, then it carries the data bit '0'. If the condition of the pair does not fit the intended data bit, then desired conditions will be provided only by swapping the packet lengths. In this method, the pairs with the same packet length values are ignored. With the knowledge of the key, the receiver can easily extract the data embedded in S traffic with the following steps:

(1) Using key, he/she initializes the initial kernel of rand function (same as (7)).
(2) Using randperm, he/she creates the sequence of lengths in which the data is embedded (same as (8)).
(3) The following steps are repeated in order to extract each data bit $d_k(\forall k1 \leq k \leq L_D)$:
   (a) Based on (13) and (14), two packet lengths in a pair are determined:

$$l'_1 \leftarrow S[seq[2k-1]] \tag{13}$$

$$l'_2 \leftarrow S[seq[2k]] \tag{14}$$

   (b) Using (15), d_k is calculated:

$$d_k = \begin{cases} 1 & l'_1 > l'_2 \\ 0 & l'_1 < l'_2 \end{cases} \tag{15}$$

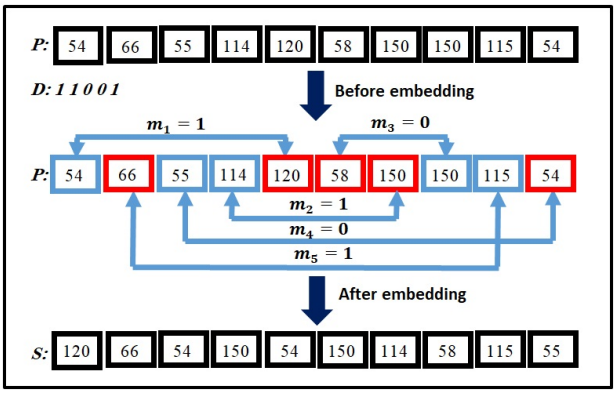
Due to the lack of change packet length values, the packet length histogram in stego traffic is same as the normal one. So, it is completely predictable that the attacks using histogram changes for detection, would be completely unsuccessful in detecting suggested method $swap - PLN$.

3.2 Network Steganography Method based on Packet Length Using Restricted Relocations (Rswap-PLN)

The idea of this method is the same as the previous one, and the only difference is adding a constraint to the packet pairing method. In $swap - PLN$ method, there is no constraint on choosing two members of the packets length sequence for embedding a data bit. These packets could be chosen freely. But it is better to reduce changes in the order of the packets, given that in an attack, it is possible to extract parameters based on the order of the packets.
These changes are done in $swap - PLN$ method with no constraints, but another method is suggested for restricting it called $Rswap - PLN$.
Assume that P is the normal network traffic with the length of L_P. First, the sender divides the sequence P into consecutive buckets in which there are $Bucket\_size$ number of packets. Then, the pairing procedure in each bucket should be done in a pseudo-random way. In consequence, the elements of each packet pair are in the same bucket. After choosing a random packet pair from each bucket, an algorithm similar to the same as previous is used in order to embed all data bits (formulas (9) to (12)). Having the parameter $Bucket\_size$, the receiver performs the division in a way similar to the sender and by specifying the packet pair used for embedding data, he/she extracts data same as the previous method (formulas (13) to (15)).

4 Implementation Results

Network steganography methods based on packet length make use of data embedding in the lengths of network packets in order to exchange data. Most of these methods recreate the packet lengths in the stego traffic carrying data, using the packet lengths in normal traffic and a certain algorithm. The function of these methods, like all steganographic methods in different media, could be assessed based on three metrics: embedding capacity, undetectability, and security against attacks. All methods are implemented with the aim of comparing proposed methods with each other and with other previous methods. The results of this comparison may be different for several traffic flows. In other words, the efficiency of a steganographic method could depend on the normal traffic characteristics used.
Some methods have measured their efficiency in a



$L_P = 10 \quad L_D = 5$
Seq=[1,5,4,7,6,9,3,10,9,2]

	Embedding				Extraction		
k	l_1	l_2	d_k	Swap	l'_1	l'_2	d_k
1	54	120	1	Yes	120	54	1
2	114	150	1	Yes	150	114	1
3	150	58	0	Yes	58	150	0
4	55	54	0	Yes	54	55	0
5	115	66	1	No	115	66	1

Figure 1. An example of the embedding algorithm (the first packet in each pair is shown with blue and the second one is shown with red)

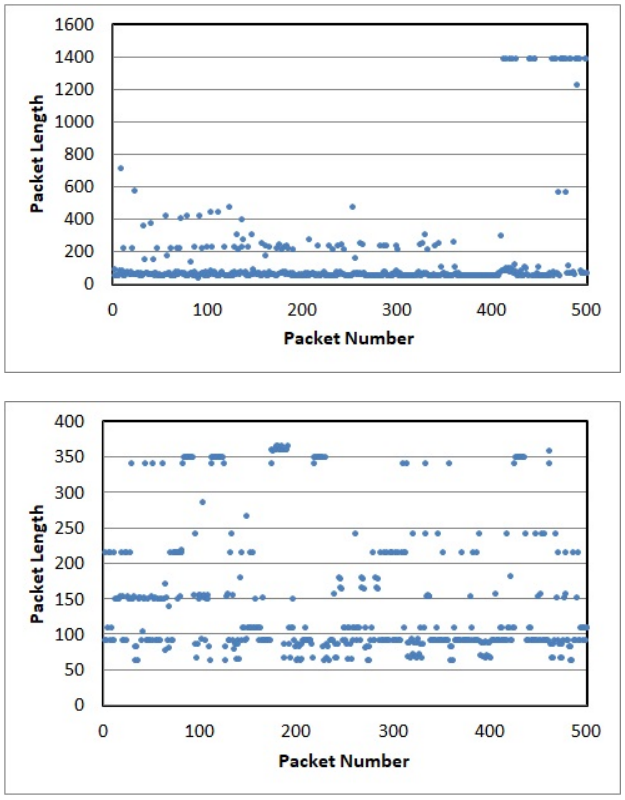


Figure 2. The length distribution for 500 consecutive packets in (a) Net_TCP and (b) Net_UDP

state that considers normal traffic as a sequence of random numbers, but this sequence of numbers can not be real normal traffic of a network. In order to make the test conditions real and due to the lack of standard datasets, the results presented below are calculated for a normal and actual traffic flow, which is sampled from Alzahra University faculty of engineering using Wireshark. By using this software, 20000 consecutive packet lengths are captured. This sampling is done for four different cases and under the conditions below:

(1) TCP-based packets of Skype protocol (Skype_TCP dataset).

(2) UDP-based packets of Skype protocol (Skype_UDP dataset).

(3) TCP-based packets of the whole network (Net_TCP dataset).

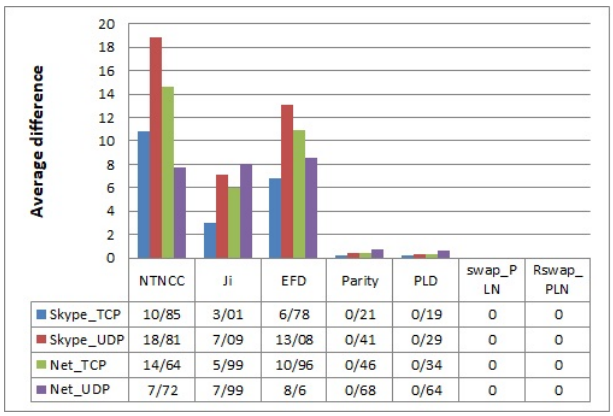
(4) UDP-based packets of the whole network (Net_UDP dataset).

The packet length distributions of TCP and UDP are commonly different. The reason is that the TCP and UDP protocols have different fragmentation mechanisms and different applications [28].

The length distribution of 500 packets of Net_TCP dataset and Net_UDP dataset is shown in Figure 2 with purpose of observing this difference. Comparing these charts show that UDP protocol packet lengths are varied, and the sequence of numbers is more random, so most methods only use this protocol for tests. But in this paper, the packets of both protocols are used for testing to compare the performance of the proposed methods for both cases.
According to the high repetition of some packet lengths in real datasets, neither $NTNCC$ method nor EDF are usable in embedding cases with lots of bits, and they can not embed successfully using the mentioned algorithms. Thus in the accomplished tests, the w parameter is considered equal to 1 in $NTNCC$, EDF, and J_i methods. The test results and the comparison of previous methods with proposed methods for three intended metrics are presented below:

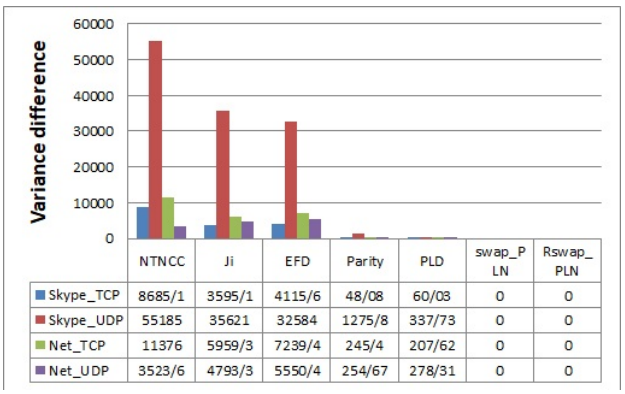
4.1 Distinction Undetectability

If the steganography method can generate a stego traffic that more similar to normal network traffic, then it is less likely to be detected by attacks. Since the attacks analyze the network traffic statistically, and if there is a significant difference between the statistical features of the present and normal traffic, they identify the present traffic as stego one. Therefore,



	NTNCC	Ji	EFD	Parity	PLD	swap_PLN	Rswap_PLN
Skype_TCP	10/85	3/01	6/78	0/21	0/19	0	0
Skype_UDP	18/81	7/09	13/08	0/41	0/29	0	0
Net_TCP	14/64	5/99	10/96	0/46	0/34	0	0
Net_UDP	7/72	7/99	8/6	0/68	0/64	0	0

Figure 3. The mean difference between normal traffic in several datasets and stego traffic generated by various embedding methods



	NTNCC	Ji	EFD	Parity	PLD	swap_PLN	Rswap_PLN
Skype_TCP	8685/1	3595/1	4115/6	48/08	60/03	0	0
Skype_UDP	55185	35621	32584	1275/8	337/73	0	0
Net_TCP	11376	5959/3	7239/4	245/4	207/62	0	0
Net_UDP	3523/6	4793/3	5550/4	254/67	278/31	0	0

Figure 4. The variance difference between normal traffic in several datasets and stego traffic generated by various embedding methods

one criterion for evaluation and comparison of network steganography methods is the rate of alteration in network traffic parameters. One of these parameters is network traffic histogram (the frequency of the packet length in traffic). In order to perform this comparison, the difference between histograms of the normal and stego traffic (derived from the proposed method algorithm) is extracted, and the mean and standard deviation are calculated for this difference. The smaller the mean and standard deviation values, the more similarity of stego traffic to a normal one. These two parameters can be used for showing distinction undetectability.

The mean and variance difference between normal and stego traffic are presented in Figures 3 and 4 respectively for several embedding methods, different datasets, and the embedding level of 0.1bpp. Examining these results indicates that the proposed methods are very successful in maintaining the mean and variance of normal traffic for creating stego traffic. This difference is absolutely zero in *Swap_PLN* and *Rswap_PLN*. Otherwise stated, the stego traffic mean and variance of these two methods are equal to the mean and variance of the real and initial normal traffic.

4.2 Embedding Capacity

Each steganographic method has the ability to embed a maximum number of specified bits in a cover media. In the network steganography methods based on packet length, the unit of a number of bits per each packet (bpp) can be used for expressing this ability, and the metric is shown with $C_{\max}$ This metric can have two meanings in different methods. In some methods, $C_{\max}$ number of bits is really embedded in each packet length. But in adaptive methods, the number of bits embedded in each packet length could differ depending on the circumstances. Therefore, according to the embedding algorithm, the $C_{\max}$ number is specified for some methods, and for some other ones, this number depends on the normal traffic used. In these methods, $C_{\max}$ could be calculated according to the number of the packets in the traffic (*Traffic_size* parameter) and the maximum number of the bits that could be embedded in the whole traffic (*Max_bit* parameter) as expressed in (16):

$$C_{\max} = \frac{Max\_bit}{Traffic\_size} \tag{16}$$

The $C_{\max}$ numbers for previous and the proposed methods are presented in Table 1. In $NTNCC$, EDF, and J_i methods, the $C_{\max}$ value depends on the w parameter. The algorithm of these methods is in such a way that, if there are many duplicate data in the dataset, then there is no way to embed data for large w values (and not even for many duplicates in $w = 1$) Since real datasets are used, and there is a large number of repetitions in them, here in these methods, a maximum of one bit per packet length is embedded in order to reduce the embedding problems. According to the algorithm, the parity method is allowed to embed one bit in each packet length. But the maximum embedding capacities of PLD, $Swap\_PLN$, and $Rswap\_PLN$ methods depend on the normal traffic used. They do not have constant and certain values. In an effort to calculate these values for different datasets using normal traffic with 1000 packets, the maximum number of embeddable bits is specified and the $C_{\max}$ value is calculated using the formula (16). This process is repeated 500 times and the average of these repetitions for these three methods are shown in Table 1.

Analyzing the results of Table 1 leads to two results: 1. More embedding capacity of UDP packets in comparison to TCP packets: due to the randomness of most packet lengths in UDP traffic, $Swap\_PLN$ and $Rswap\_PLN$ could embed more secret data in UDP traffic. 2. $Swap\_PLN$ and $Rswap\_PLN$ methods have much less embedding capacity compared

Table 1. The average C_{max} value for different embedding methods in four datasets

	Skype_TCP	Skype_UDP	Net_TCP	Net_UDP
NTNCC	1	1	1	1
Ji	1	1	1	1
EFD	1	1	1	1
Parity	1	1	1	1
PLD	0.60	0.64	0.67	1.06
Swap-PLN	0.14	0.15	0.18	0.44
Rswap-PLN	0.11	0.12	0.13	0.33

Table 2. The maximum C_{max} value for two suggested methods in four datasets

	Skype_TCP	Skype_UDP	Net_TCP	Net_UDP
Swap-PLN	0.27	0.42	0.42	0.49
Rswap-PLN	0.19	0.34	0.33	0.40

to the existing methods. But does this low embedding capacity make these methods unusable in the real world? To answer this question, two points must be taken care of: the first one is that the mentioned numbers are the average $C_{\max}$ values for different traffic types in datasets. To put it simply, there are normal traffic flows in these datasets that if they are used for embedding data, then the $C_{\max}$ the value would be greater than the values in Table 1 for the suggested methods *Swap_PLN* and *Rswap_PLN*. The maximum $C_{\max}$ value for a normal traffic flow in four datasets for these two methods are shown in Table 2. The less duplicate numbers in the normal traffic use, the greater the embedding capacities of two *Swap_PLN* and *Rswap_PLN* suggested methods, but the maximum value is 0.5 bpp.

The second point is the security of a method with high embedding capacity against attacks. To clarify, if the methods with high embedding capacity use their capacity thoroughly, are they not detected by attacks? Therefore, the more important factor in steganographic methods is the amount of data an embedding method can embed without being detected by attacks.

4.3 Security Against Attacks

As mentioned before, there are two steganalytical attacks with the aim of detecting network steganography methods based on packet length. As the second attack [32] is more complete in comparison to the first one [31], only the second one is used here. The detection accuracy of this attack at the level of 0.1 bpp for four datasets is shown in Table 3. The meaning of attack accuracy is the normalized area below the ROC chart for attack output relative to the original diameter. As the ROC chart farther away from the original diameter, and as the area increases, the attack becomes more successful in detecting the

Table 3. The accuracy of the attack in [32] in detecting different methods in four datasets at the level 0.1 bpp

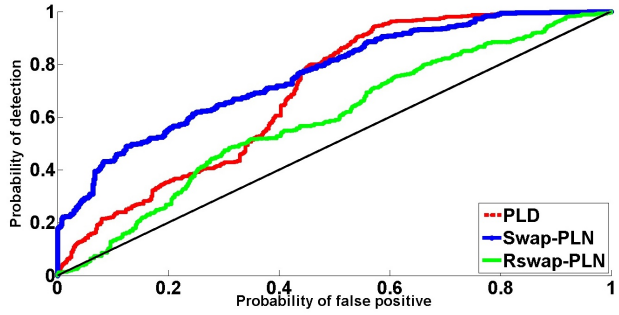
	Skype_TCP	Skype_UDP	Net_TCP	Net_UDP
NTNCC	0.8668	0.5079	0.4446	0.4891
Ji	0.9979	0.9885	0.9504	0.5905
EFD	0.9963	0.6952	0.4964	0.5962
Parity	1	0.9858	0.9822	0.4475
PLD	0.9607	0.8740	0.8800	0.3760
Swap-PLN	0.8027	0.6162	0.4785	0.5121
Rswap-PLN	0.5958	0.4562	0.4199	0.1599

method, and the resistance of the method against the attack decreases. A study on Table 3 shows that in all datasets, *Rswap_PLN* has the least detection accuracy and the most resistance against the attack. The results show that most of the previous methods are easily discovered with this low embedding rate and despite their high embedding capacity (according to Table 1), they are not practically usable. Among previous methods, only *NTNCC* method is close to the *Rswap_PLN* method in some cases. In this method, both the sender and the receiver need to be aware of normal traffic, and this need is restrictive in practical and real use. Furthermore, according to the results of Figures 3 and 4, the *NTNCC* method is not successful at maintaining the mean and the variance values of the stego traffic.

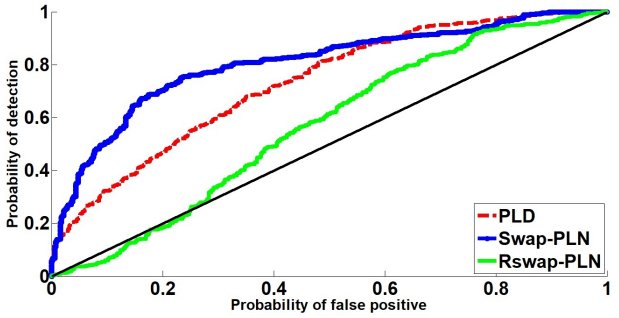
Another noteworthy point in Table 3 is the good performance of the *Rswap_PLN* method in the Net_UDP dataset. Due to the possibility of embedding more data in this dataset, the comparison of various methods is also done in this dataset for embeddings at the level of 0.2 bpp and 0.3 bpp and the results are shown in Table 4. These results also indicate the impressive superiority of *Rswap_PLN* method compared to the all other ones. For a better understanding, the ROC charts of three *PLD*, *Swap_PLN*, and *Rswap_PLN* methods for Net_UDP dataset at 0.1 bpp and 0.2 bpp embedding levels are displayed in sections (a) and (b) of Figure 5, respectively. A review of all represented results indicates that most methods, despite their large embedding capacity, are easily detected in low embedding percentages. Thus the large embedding capacity in these methods is not considered as a significant advantage. On the contrary, the *Rswap_PLN* method, despite having low embedding capacity, performs the embedding process in a way that is not detectable for the existing attack. This embedding method does not have high discovery accuracy for all datasets but for Net_UDP dataset, it has higher embedding capacity and much less detectability. The final point is the effect of the *Bucket_size* parameter value on the performance of the *Rswap_PLN* method. This parameter shows the number of packets in each bucket. Pairing operations are performed per each

Table 4. The accuracy of attack [32] in detecting different methods for Net_UDP dataset traffic in more embeddings

	0.2 bpp	0.3 bpp
NTNCC	0.6173	0.7201
Ji	0.8245	0.9488
EFD	0.7400	0.8756
Parity	0.6256	0.8312
PLD	0.4419	0.6026
Swap-PLN	0.6229	0.7028
Rswap-PLN	0.1558	0.1671



(a)



(b)

Figure 5. The ROC chart of the attack [32] for embeddings at the levels of (a) 0.1 bpp and (b) 0.2 bpp for Net_UDP dataset traffic

Table 5. The accuracy of attack [32] in detecting Rswap_PLN method for Net_UDP dataset traffic in different Bucket_sizes

	0.1 bpp	0.2 bpp
2	0.1599	0.1558
4	0.1849	0.1884
10	0.1948	0.2251

bucket. In Table 5, the attack accuracy is shown in this method for Net_UDP dataset traffic at two levels of 0.1 bpp and 0.2 bpp with three different values for *Bucket_size* including 2, 4, and 10. The results show that the *Rswap_PLN* method, in each level, shows its best performance in the case where the value of the *Bucket_size* parameter is equal to 2. So, all previous results for this method are obtained in this case.

5 Conclusion

Embedding data in packet lengths is known as a network steganography method. Different methods of this area manipulate the packet lengths of traffic in such a way that the receiver can extract the intended data based on the lengths received. In this paper, two different methods are proposed in this area that the second one, called *Rwap_PLN*, has better performance. In this method, the sender divides existing packets in a normal traffic into buckets. Afterward, he/she chooses two packets of a bucket, and if they do not have the same length, he/she can embed a data bit in them. If the first packet length is larger than the second one, the pair shows a '1' bit, and otherwise, it shows a '0' bit. If the intended bit of the sender does not match the current status, he/she will create the desired status by swapping the packet lengths. The results show that the histogram of packet lengths in Stego traffic generated by the $Rswap-PLN$ method is fully compatible with the histogram of packet lengths in normal traffic. Previous methods are vulnerable to existing attacks despite having high embedding capacities. But our proposed method has far greater security against attacks for different datasets. The results show that the use of the $Rswap-PLN$ method for UDP-based packets with a detection accuracy of less than 20% is the best option for network steganography based on packet lengths.

References

[1] J. Lubacz, W. Mazurczyk, and K. Szczypiorski. Principles and overview of network steganography. 2015.

[2] B.G. Banik and S.K. Bandyopadhyay. Review on steganography in digital media. *International Journal of Science and Research (IJSR)*, 4:1–10, 2015.

[3] A.P. Dhamade and K.J. Panchal. Packet data based network steganography. *International Journal of Advance Engineering and Research Development*, 2(5):1520–1526, 2015.

[4] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski. Information hiding in communication networks: Fundamentals, mechanisms, applications and countermeasures. In *IEEE Press Series on Information & Communication Networks Security*, 2016.

[5] F. Petitcolas, R. Anderson, and M. Kuhn. Information hiding: a survey. *IEEE. Special Issue on Protection of Multimedia Content*, 87(7):1062 – 1078, July 1999.

[6] S. Zander, G. Armitage, and P. Branch. A survey of covert channels and countermeasures in computer network protocols. *IEEE Commun*

Surv Tutor, 9(3):44–57, 2007.

[7] B. Lampson. A note on the confinement problem. 16(10):613–615, 1973.

[8] DoD Orange Book. National computer security center, us DoD,. In *Trusted Computer System Evaluation Criteria, , Tech. Rep. DOD 5200.28-STD*, 1985.

[9] W. Mazurczyk. VoIP steganography and its detection – A survey. 2012.

[10] J. P. Black. Techniques of network steganography and covert channels. In *PhD diss., Sciences*, 2013.

[11] W. Fraczek, W. Mazurczyk, and K. Szczypiorski. Multi-level steganography: Improving hidden communication. In *Networks*, 2011.

[12] S. Wendzel, M. Wojciech, and Z. Sebastian. Unified description for network information hiding methods. *J. UCS 22.11*, pages 1456–1486, 2016.

[13] A. Stančić, I. Grgurevic, and V. Vyroubal. Usage of the steganography within highway information and communication network. In *4th International Virtual Research Conference In Technical Disciplines (RCITD)*, 2016.

[14] M.M. Pontón Loaiza. Steganography using rtp packets. In *University of Abertay Dundee, Dundee*, 2014.

[15] A. Swinnen, R. Strackx, P. Philippaerts, and F. Piessens. Protoleaks: A reliable and protocol-independent network covert channel. In *International Conference on Information Systems Security*, pages 119–133, 2012.

[16] W. Mazurczyk and J. Lubacz. LACK: a VoIP steganographic method. *Telecommunication Systems: Modelling, Analysis, Design and Management*, 45(2–3):153–163, 2010.

[17] W. Mazurczyk, J. Lubacz, and K. Szczypiorski. On steganography in lost audio packets. In *International Journal of Security and Communication Networks*, 2012.

[18] W. Mazurczyk, M. Smolarczyk, and K. Szczypiorski. On information hiding in retransmissions. *Telecommunication Systems*, 52(2):1113–1121, 2013.

[19] B. Jankowski, W. Mazurczyk, and K. Szczypiorski. PadSteg: Introducing inter-protocol steganography. *Telecommunication Systems*, 52(2):1101–1111, 2013.

[20] V. Sabeti and M. Shoaei. Network steganography based on PVD idea. In *8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018.

[21] M.A. Padlipsky, D.W. Snow, and P.A. Karger. Limitations of end-to-end encryption in secure computer networks. In *Tech. Rep. ESD-TR-78-158, Mitre Corporation*, 1978.

[22] C.G. Girling. Covert channels in LAN's. *IEEE Trans. Software Engineering*, 13(2):292–296, 1987.

[23] Q. Yao and P. Zhang. Coverting channel based on packet length. *Computer Engineering*, 34(3), 2008.

[24] J. Liping, J. Wenhao, and D. Benyang. A novel covert channel based on length of messages. In *International Conference on e-Business and Information System Security*, 2009.

[25] J. Liping, H. Liang, Y. Song, and X. Niu. A normal-traffic network covert channel. *In Computational Intelligence and Security*, pages 499–503, 2009.

[26] A.S. Nair, A. Kumar, A. Sur, and S. Nandi. Length based network steganography using udp protocol. In *In Communication Software and Networks (ICCSN), IEEE 3rd Intl. Conf.*, pages 726–730, 2011.

[27] O.I. Abdullaziz, V.T. Goh, and H.C. Ling. Network packet payload parity based steganography. In *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, 2013.

[28] L. Zhang, G. Liu, and Y. Dai. Network packet length covert channel based on empirical distribution function. *Journal of Networks*, 9(6), 2014.

[29] M.A. Elsadig and Y.A. Fadlalla. Survey on covert storage channel in computer network protocols: Detection and mitigation techniques. *International Journal of Advances in Computer Networks and Its Security*, 6(3):11–17, 2016.

[30] M.A. Elsadig and Y.A. Fadlalla. Packet length covert channel: A detection scheme. *1st International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–7, 2018.

[31] R. Goudar and A. Patil. Packet length based steganography detection in transport layer. *International Journal of Scientific and Research Publications*, 2(12), 2012.

[32] A. Sur, A.S. Nair, and A. Kumar. Steganalysis of network packet length based data hiding. *Circuits, Systems, and Signal Processing*, pages 1–18, 2012.

[33] M.A. Elsadig and Y.A. Fadlalla. A balanced approach to eliminate packet length-based covert channels. *4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–7, 2017.

Vajiheh Sabeti is an Assistant Professor of Engineering and Technology department at Alzahra University. She received her B.S. degree in Software Engineering in 2004 and her M.Sc. degree in Computer Architec-

ture in 2007 and her Ph.D. degree in Computer Engineering in 2012 from the Electrical and Computer Engineering Department of Isfahan University of Technology (IUT), Isfahan, Iran. Her research interests are soft computing, image processing, and information hiding (steganography, watermarking).

Minoo Shoaei is an M.Sc. graduated. She got her straight M.Sc. in Information Technologies in 2019 and her B.Sc. in Software Engineering in 2016 from Faculty of Engineering of Alzahra University, Tehran, Iran. She has practical experiences in Software Engineering, Network Programming, Database programming, Web Designs, and Microsoft Certified Solutions Expert (MCSE). Her interests are in the fields of Web Designs (HTML & CSS) and Network Security.

The ISC Int'l Journal of Information Security
ISeCure

January 2020, Volume 12, Number 1 (p. 70)
http://www.isecure-journal.org

Persian Abstract

روش پنهان‌نگاری شبکه مبتنی بر طول بسته با امنیت بالا

وجیهه ثابتی[1] و مینو شعاعی[2]

[1]استادیار، دانشکده فنی و مهندسی، دانشگاه الزهرا، تهران، ایران

[2]دانشجوی کارشناسی ارشد، دانشکده فنی و مهندسی، دانشگاه الزهرا، تهران، ایران

در روش‌های پنهان‌نگاری شبکه مبتنی بر طول بسته، طول بسته‌ها به عنوان حاملی برای انتقال پیام محرمانه استفاده می‌شود. روش‌های موجود در این حوزه به دلیل رفتار غیرعادی ترافیک شبکه، در برابر کشف آسیب‌پذیر هستند. هدف اصلی در این مقاله، پیشنهاد روشی است که امنیت بالایی در برابر حملات ترافیک شبکه داشته باشد. در روش پیشنهادی اول، فرستنده در هر زوج شامل دو طول بسته‌ی غیریکسان، یک بیت داده را جاسازی می‌کند. در وضعیت موجود، اگر طول بسته اول زوج بزرگتر از طول بسته دوم آن باشد، بیت یک و در غیراین صورت بیت صفر را نشان می‌دهد. اگر بیت موردنظر فرستنده با وضعیت موجود مغایرت داشته باشد، فرستنده با جابه‌جا کردن بسته‌ها در ترافیک وضعیت موردنظر خود را ایجاد می‌کند. در این روش، بسته‌های زوج شده می‌توانند به صورت آزاد انتخاب شوند، اما در روش پیشنهادی دوم، بسته‌ها به باکت‌هایی تقسیم‌بندی شده و فقط بسته‌های داخل یک باکت می‌توانند با یکدیگر زوج شوند. در این حالت، روش جاسازی مشابه روش قبل است. نتایج تست نشان می‌دهد که روش پیشنهادی دوم با وجود ظرفیت جاسازی کم، در ترافیک واقعی امنیت بسیار بالاتری نسبت به روش‌های قبلی دارد. با توجه به تصادفی‌تر بودن طول بسته‌های پروتکل UDP نسبت به TCP، روش‌های پیشنهادی برای بسته‌های مبتنی بر پروتکل UDP ظرفیت جاسازی بالاتر و امنیت بیشتری دارند. روش‌های پیشنهادی فقط برای پروتکل‌هایی قابل استفاده است که طول بسته‌ها مقدار ثابتی نیست.

واژه‌های کلیدی: کانال نهان، امنیت داده، نهان‌نگاری شبکه، طول بسته، نهان‌کاوی.