Methods of IPD normalization to counteract IP timing covert channels

K. Kogos[1], A. Sokolov[1]

[1]*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), 31 Kashirskoe Sh., 115409, Moscow, Russia*

Abstract

Covert channels are used for information transmission in a manner that is not intended for communication and is difficult to detect. We propose a technique to prevent the information leakage via IP covert timing channels by inter-packet delays normalization in the process of packets sending. Recommendations for using the counteraction methods and choosing parameters were given. The advantage of our approach is that the covert channel is eliminated or limited preliminary, whereas state of the art methods focus on detecting active IP covert channels that may be insecure.

Keywords: Covert channel; IP timing channel; elimination; limitation; traffic normalization; inter-packet delays; capacity

1. Introduction

Covert channels were introduced by Lampson as channels not intended for information transfer at all [1]. TCSEC defines covert channel as any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy [2].

Covert channels were classified into storage and timing channels. Storage channels involve the direct or indirect writing of a storage location by the sender and the direct or indirect reading of it by the receiver. Timing channels include the sender signaling information by modulating the use of resources over time such that the receiver can observe it and decode the information.

Information in covert timing channel can be encoded by varying packets transfer rates (or inter-packet times) [3, 4, 5, 6] and by packet sorting [7]. Storage channels in networks can be encoded in packet lengths [8, 9] or packet header fields (TTL, TOS, ID, Checksum, etc.) [10, 11, 12, 13]. Network covert channels are described on Fig. 1.

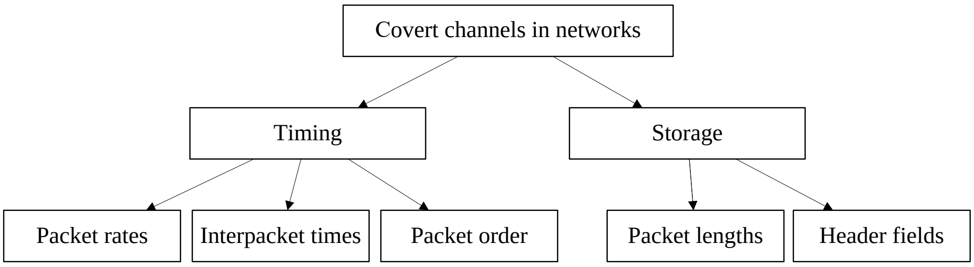

Fig. 1. Types of network covert channels.

Fig. 2 illustrates the main stages of covert channels counteraction.

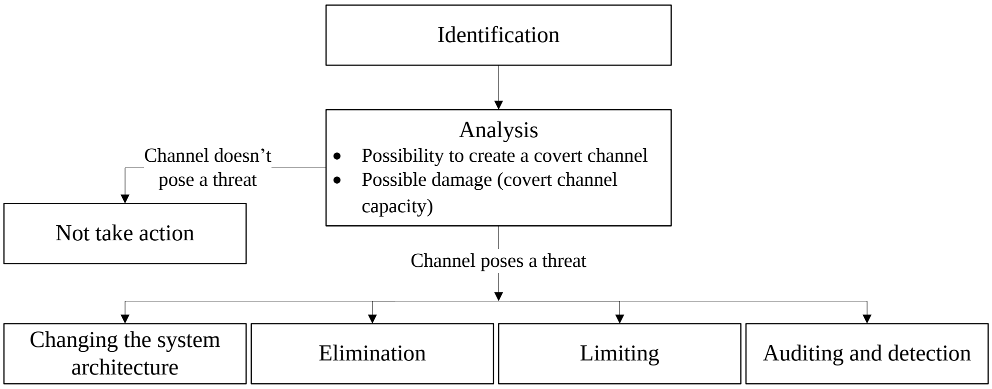
Identification
Analysis
Possibility to create a covert channel
Possible damage (covert channel capacity)
Channel doesn’t pose a threat
Not take action
Channel poses a threat
Changing the system architecture
Elimination
Limiting
Auditing and detection

Fig. 2. Covert channels handling.

The identification problem is to find the potential covert channels that can be realized in the analyzed system. The second step is the analysis of identified channels to assess the threat level of each covert channel. If channel poses a threat to the protected system the following measures can be applied: elimination, limiting, detection. Ideally covert channels should be identified and removed during the design phase. Covert channels in networks can be eliminated by traffic encryption and normalization (protocol headers, packet lengths, inter-packet times). If a channel cannot be eliminated its capacity should be reduced by using limiting techniques [14, 15, 16, 17, 18]. Auditing and detection methods can be used to detect the operating covert channels [4, 19, 20, 21, 22]. These methods are based on the detection of non-standard or abnormal behavior. Covert timing channels in networks can be eliminated only by normalizing inter-packet times. But this measure reduces the communication channel bandwidth. Method parameters must be correctly selected to minimize the negative impact on network performance.

The rest of the paper is organized as follows. First, we give an overview of existing methods of covert channels construction and counteraction in Chapter 2. In Chapter 3, we introduce recommendations on the choice of parameters of covert channel elimination method. In Chapter 4, we consider two ways to limit covert channels capacity. In Chapter 5, we provide experimental results to demonstrate counteraction methods effect on network performance. Our conclusions are presented in Chapter 6.

2. Related Work

2.1. Methods of covert timing channels construction

Covert information can be encoded by varying packet rates or inter-packet times. The covert sender varies packet rate between two (binary channel) or more packet rates each time interval. The receiver decodes the covert information by measuring the rate in each time interval. The sender and receiver need a mechanism for synchronization of the time intervals. Timing channel where the sender either transmits or stays silent in each time interval (on/off channel) is a special case of binary channel [3]. Authors of [5] implemented the on/off timing channel. In their scheme the covert data is divided into frames and synchronization between sender and receiver is achieved through a special start sequence at the beginning of each frame. There are variants of the timing channels that does not require synchronization between sender and receiver because the covert information is encoded directly in the inter-packet times of transmitted packets [23, 24].

Authors of [25] developed an indirect covert channel that exploits the fact that a host’s CPU temperature is proportional to the number of packets per time unit it processes and a host’s system clock skew depends on the temperature. The channel requires an intermediary that receives and sends packets to both covert sender and receiver. The covert sender either sends packets to the intermediary or stays silent. The covert receiver estimates the intermediary’s clock skew by observing a series of timestamps in packets sent by the intermediary. There are other implementations of thermal covert channels [26].

Covert timing channel can be organized through packet sorting [7]. Sender can transmit a maximum of $\log_2 n!$ bits because a set of *n* packets can be arranged in any *n*! ways. This approach requires per packet sequence numbers to determine the original packet order. The method only modifies the sequence numbers, thus keeping payload unchanged.

2.2. Methods of covert channels counteraction

Admissible covert channel capacity depends on the kind of protected information and on the amount of leaked information, which is critical. TCSEC assumes that covert channels with maximum bandwidths of less than 1 bit per second are acceptable in most application environments [2]. According to IBM guidelines, channels with bandwidths lower than 0.1 bits per second can exist. There is no special need to counteract them. Channels in range from 0.1 to 100 bits per second can exist when absolutely necessary [27].

Capacity of covert timing channels in networks can be limited by adding random delays to the packets. Fig. 3 shows the framework of using traffic control module [18]. Network covert timing channel exploitation takes place here. An innocent process request the OS kernel to send a network packet, then covert message sender can somehow interfere with this procedure (for example, delay response), after that the remote covert channel receiver eavesdrops related packets and decodes the message. No matter whether there are covert channels, the traffic controller will get in on the network packet send out procedure.

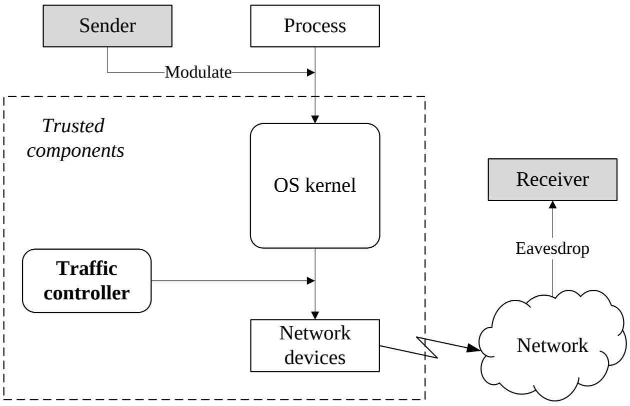

Fig. 3. Framework of using traffic controller.

For each network connection, traffic control module maintains some information (network address, port number, connection type, previous packet's outgoing time, etc.). When an application sends out a packet, traffic controller will intercept the packet, look up the network connection information and add a random delay to the packet (fixed delays could be easy filtered by covert channel users). Delay of *n*th packet will be calculated according to the formula:

$$T_n = f(\Delta t_n, k) = Rand(k) \cdot \Delta t_n, \tag{1}$$

where Δt_n denotes the time interval between current and previous packet-sending request, *k* is a configurable parameter $(0 < k < 1)$, *Rand*(k) function generates a random number ranged from 0 to *k*. Hence, T_n will be a random value from 0 and up to $k \cdot \Delta t_n$. Experimental results shows that the covert communication achieved nearly 100% encode/decode correctness when traffic control was disabled. With the traffic control enabled, the error rate rapidly raised to about 50%.

Gateway is often used to prevent the data transmission from higher security level to lower. Gateway is located between the sender with low security level and receiver with high security level (Fig. 4). When the gateway receives a packet from low it stores it into a buffer and sends an acknowledgment (ACK) to low. Then it transmits the packet to high and waits for an ACK. If the ACK is received the gateway removes the packet from the buffer.

However, when the buffer is full the gateway must wait for high to acknowledge a received packet until another packet from low can be stored in the buffer. The time of sending an ACK to low is directly related to the time of receiving an ACK from high. High can ensure the buffer is always filled and vary the rate of its ACKs. In this manner, he can exploit the covert channel. The PUMP model reduces this covert channel capacity [16, 17].

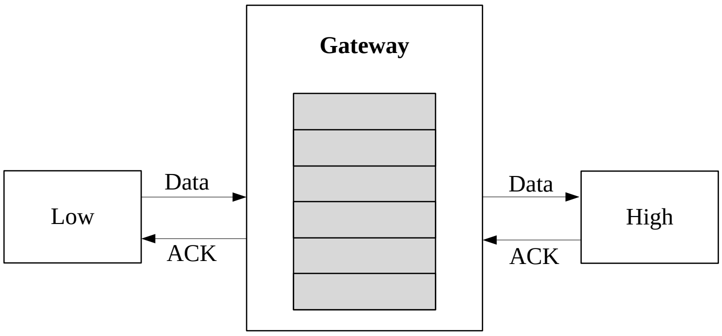

Fig. 4. Message passing from low to high using the gateway.

Network covert timing channels can be eliminated only by normalizing inter-packet times. But this measure reduces the communication channel bandwidth. Method parameters must be correctly selected to minimize the negative impact on network performance.

3. Covert timing channels elimination

Inter-packet times normalization makes it necessary to delay the transmission of packets and generate dummy packets. It reduces the network performance. So, method of covert channels elimination should be used only if the leakage of a very small amount information is unacceptable. Parameters of inter-packet times normalization method must be correctly selected to minimize the negative impact on communication channel capacity.

Input data for the calculation of the best inter-packet time value *kt* can include:

1. empirical distribution of inter-packet time over a long period of time,
2. maximum acceptable packet queue delay *lt*,
3. ε equal to the allowable part of packets which may be delayed for a time greater than *lt*.

Following conditions must be met when inter-packet times normalization to *kt* is performed:

1. communication channel bandwidth is not less than the set value,
2. percentage of packets which are delayed for a time greater than *lt* is not more than ε,
3. number of dummy packets is minimal.

One of the following values can be used instead of ε and *lt*:

1. maximum allowable average packet delay $d_{avg}t$,
2. maximum acceptable part of dummy packets.

Suppose we have a distribution of inter-packet time (Fig. 5). The minimum value of the inter-packet interval is equal to *t* and maximum equal to *mt*.

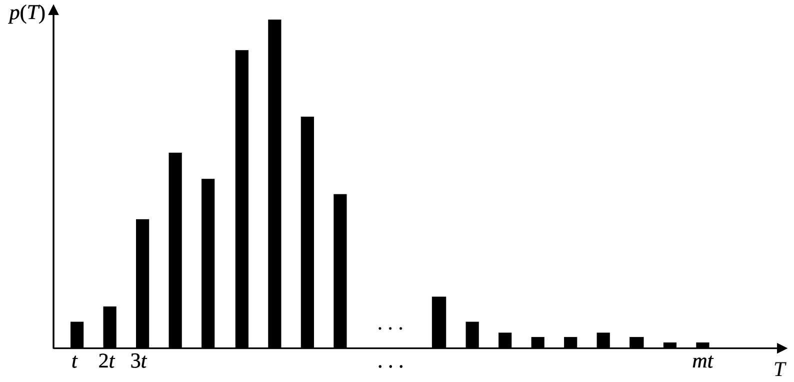

Fig. 5. Inter-packet time distribution.

Let the inter-packet times are normalized to *kt*. The device processes packets for an infinitely small time and its queue is empty at the moment. When two packets with *at* interval arrive to the device, there will be the following. The second packet will be delayed for $kt - at$, if $at \le kt$. The second packet will be delayed for $\left(\left\lceil \frac{at}{kt} \right\rceil - 1\right)kt + (kt - at) = \left\lceil \frac{at}{kt} \right\rceil kt - at$ and $\left\lceil \frac{at}{kt} \right\rceil - 1$ dummy packets will be generated and sent by the device, if $at > kt$.

So, when *n*+1 packets with $a_1t, a_2t, ..., a_nt$ intervals come to the device, delay of the (*i*+1)th packet is equal to

$$d_it = d_{i-1}t + (N_{d_i} + 1)kt - a_it\,, \tag{2}$$

where $d_0t = 0$ and N_{d_i} is a number of dummy packets sent after receiving the *i*th packet (during the *a*a_it):

$$N_{\phi_i} = \left\lceil \frac{a_it - d_{i-1}t}{kt} \right\rceil - 1\,. \tag{3}$$

The smaller the inter-packet time *kt*, the less packet queue delay and the greater the number of dummy packets.

Let the inter-packet time is a discrete random variable ξ obeying the distribution law in Table 1.

Table 1. Distribution law of ξ.

ξ	t	$2t$	…	$(m-1)t$	mt
$P(\xi = it)$	p_1	p_2	…	p_{m-1}	p_m

Queue delay of (*n*+1)th packet is given by:

$$dt = (n + N_d)kt - \sum_{i=1}^{n} a_it\,, \tag{4}$$

where N_d is a number of dummy packets sent during the $\sum_{i=1}^{n} a_it$ after receiving the first packet.

The inter-packet time after normalization should not exceed the average value in the initial distribution to avoid the constant increase in queue length. That is, the following inequality must be satisfied:

$$kt \le E(\xi)\,, \tag{5}$$

where $E(\xi)$ is the expected value of a variable ξ.

One should choose a value of *kt* for witch this probability is not greater than ε. Furthermore, the value of probability should be as close to ε as possible to minimize the amount of dummy packets. In choosing the value of *kt* based on the maximum acceptable part of dummy packets ($\frac{N_d}{n+N_d}$) one should select the minimum suitable *kt* value to minimize packet delays.

We consider two use cases of communication channel:

1. file transfer only,
2. real-time data transfer (e.g. VoIP, Skype).

The maximum packet queue delay is not too important, if the channel is not being used for real-time data transmission. Allowable average packet delay or acceptable percentage of dummy packets should be used as input data in this case. If you use a channel for real-time data transfer, it is essential to ensure good communication quality. Therefore, the inter-packet time *kt* should be calculated based on the maximum acceptable packet delay. For example, packet jitter should not exceed 30 milliseconds to provide an acceptable quality of a Skype call [28, 29].

4. Covert timing channels limitation

If a non-zero covert channel capacity is allowed one can use partial inter-packet times normalization. Such methods have less negative impact on the communication channel.

4.1. Normalization by several inter-packet times

Let two values of inter-packet times after traffic normalization be allowed: k_1t and k_2t. Inter-packet time equal to k_1t will be observed at the output if the queue is not empty in k_1t after sending the last packet. If the queue is empty at this moment the inter packet time equal to k_2t will appear at the output. It will be a dummy packet if the queue also is empty in k_2t after sending the last packet.

Violator can affect the inter-packet times by passing packets to the channel and use covert channel. Let the random variable X take the values 0 or 1 in accordance with the inter-packet times (k_1t or k_2t) at the output. p is the probability of observing packets with an interval equal to k_1t. Then entropy of X is equal to:

$$H(X) = -p\log_2 p - (1-p)\log_2(1-p)\,. \tag{6}$$

The average time between two consecutive outgoing packets is $pk_1t + (1-p)k_2t$. So, capacity of covert timing channel that can be built is:

$$C = \frac{-p\log_2 p - (1-p)\log_2(1-p)}{pk_1t + (1-p)k_2t}\,, \tag{7}$$

where $(1-p)^{k_1t} = p^{k_2t}$ holds.

It is possible to use more than two values of inter-packet delays.

4.2. Normalization by several packet transfer rates

Let there are two inter-packet delays: k_1t and k_2t ($k_1t < k_2t$) which correspond to two fixed packet transfer rates. During each interval T inter-packet times are fixed and equal to k_1t or k_2t (packets are transmitted at a constant rate). Rate can change or remain the same at the time points $T \cdot i$ (i = 1, 2,…). Selected transfer rate depends on the number of packets received at the last part of the T and the number of packets in the queue. Lower rate (which correspond to k_2t) will be set at the time $T \cdot j$ if

$$\frac{T'}{N_{T'} + N_q} > k_2t\,, \tag{8}$$

where $N_{T'}$ is the number of packets received during the time interval ($T \cdot j - T'\,, T \cdot j$); N_q is the amount of packets in the queue at the moment. Otherwise, a high data transfer rate will be established.

When using this method, covert channel capacity is:

$$C = \frac{\log_2 r}{T}, \tag{9}$$

where r is the number of transfer rates. In this case $r = 2$.

Parameter T should be chosen depending on the predetermined allowable capacity of covert channel C_a.

$$T = \frac{\log_2 r}{C_a}. \tag{10}$$

5. Experimental results

This chapter provides experimental results to demonstrate the effect of inter-packet times normalization on network performance. Two use cases of network are reviewed: file transfer only and real-time data transfer. For each of these cases we have two empirical distribution of inter-packet time (under high and low network load). The best values of inter-packet time was calculated for several input data sets.

5.1. Covert channels elimination during file transfer

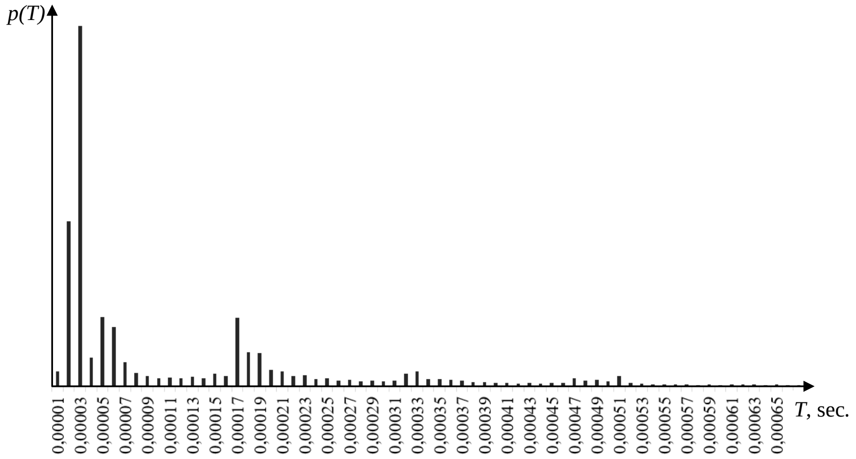

Fig. 6. Inter-packet time distribution under high network load ($E(T)$ = 0.00017 sec.).

Table 2. Results of calculation of *kt* based on the acceptable part of dummy packets (high network load).

$\frac{N_d}{n + N_d}$	*kt*, sec.	$d_{avg}t$, sec.
0.1	0.00016	0.00611
0.3	0.00012	0.00053
0.5	0.00009	0.00019

Table 3. Results of calculation of *kt* based on the acceptable average packet delay (high network load).

$d_{avg}t$, sec.	*kt*, sec.	$\frac{N_d}{n + N_d}$
0.5	0.00017	0.002
1.0	0.00017	0.002

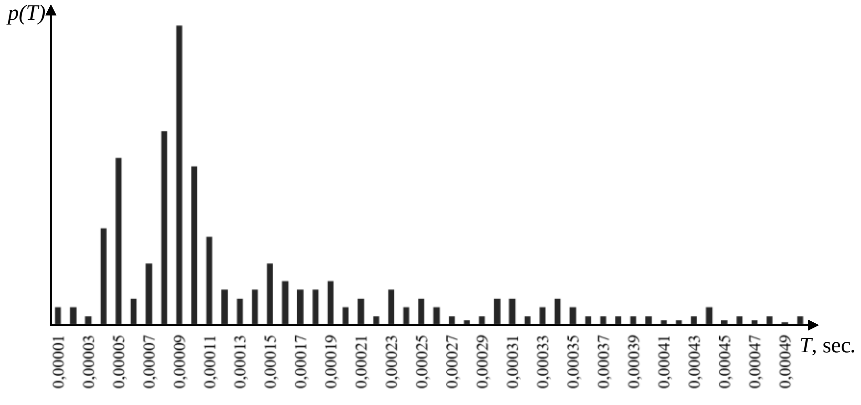

Fig. 7. Inter-packet time distribution under low network load ($E(T)$ = 2.33749 sec.).

Table 4. Results of calculation of *kt* based on the acceptable part of dummy packets (low network load).

$\frac{N_d}{n+N_d}$	*kt*, sec.	$d_{avg}t$, sec.
0.1	2.09752	47.1353
0.3	1.64243	12.9241
0.5	1.16614	5.45441

Table 5. Results of calculation of *kt* based on the acceptable average packet delay (low network load).

$d_{avg}t$, sec.	*kt*, sec.	$\frac{N_d}{n+N_d}$
0.5	0.25739	0.89
1.0	0.41155	0.82
5.0	1.11260	0.52

5.2. Covert channels elimination during real-time data transfer

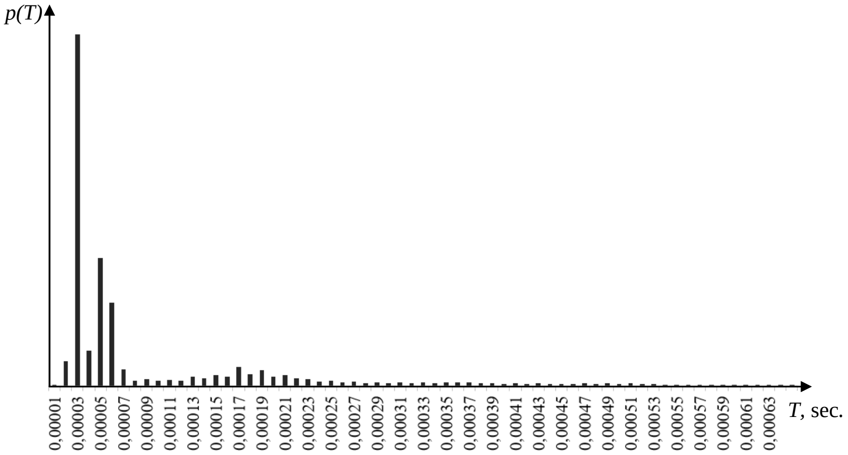

Fig. 8. Inter-packet time distribution under high network load ($E(T)$ = 0.00025 sec.).

Table 6. Results of calculation of *kt* based on the maximum acceptable packet delay (high network load; ε = 0.001).

lt, sec.	*kt*, sec.	$d_{avg}t$, sec.	$\frac{N_d}{n+N_d}$
0.005	0.00012	0.00063	0.52
0.010	0.00014	0.00117	0.44
0.020	0.00016	0.00216	0.36

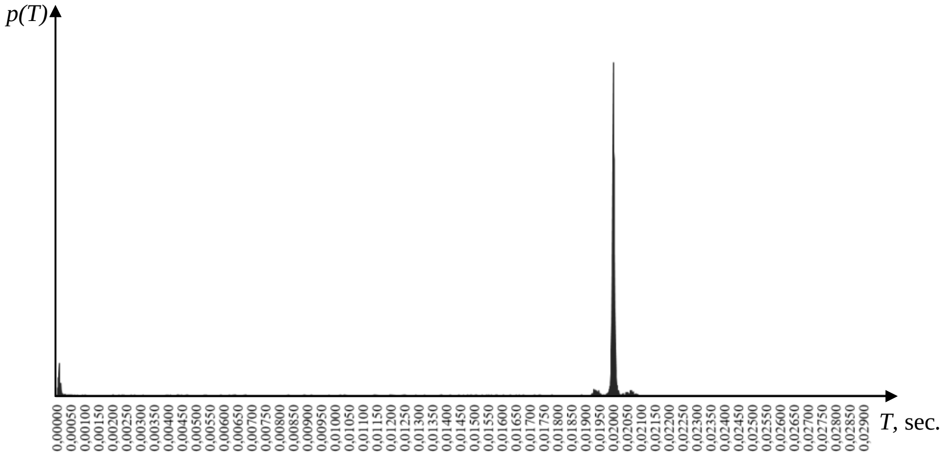

Fig. 9. Inter-packet time distribution under low network load ($E(T)$ = 0.02472 sec.).

Table 7. Results of calculation of *kt* based on the maximum acceptable packet delay (low network load; ε = 0.001)

lt, sec.	*kt*, sec.	$d_{avg}t$, sec.	$\frac{N_d}{n+N_d}$
0.005	0.00192	0.00108	0.92
0.010	0.00367	0.00209	0.85
0.020	0.00720	0.00421	0.71

5.3. Covert channels limitation

The following dependencies were identified using a covert channel limit method that allows two packet transfer rates (Fig. 10, 11).

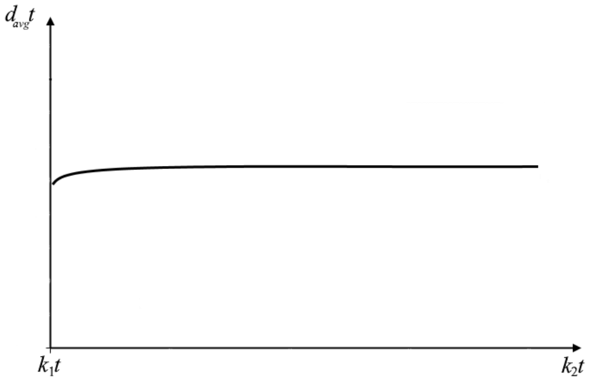

Fig. 10. The dependence of average packet delay on k_2t for a fixed k_1t (normalization by 2 packet transfer rates).

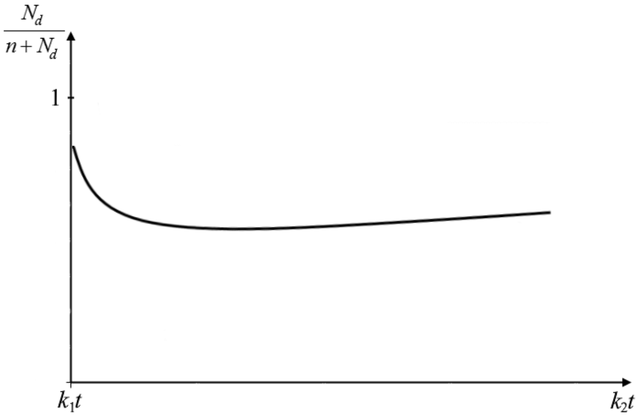

Fig. 11. The dependence of the part of dummy packets on k_2t for a fixed k_1t (normalization by 2 packet transfer rates).

5.4. Comparison of counteraction methods

Three techniques of inter-packet delays normalization were applied under the same conditions. The requirement for the average packet delay was specified. The parameters of each method have been calculated to ensure a minimum effect on the communication channel performance. The values of the covert channel capacity and part of dummy packets are shown in the Table 8.

Table 8. Comparison of covert channels counteraction methods.

Average packet delay, sec.	Normalization method	Part of dummy packets	Covert channel capacity, bit/sec.
0.1	One inter-packet time	0.911	0
	Two inter-packet times	0.340	664
	Two packet transfer rates	0.602	1
0.5	One inter-packet time	0.898	0
	Two inter-packet times	0.100	13
	Two packet transfer rates	0.546	1

6. Conclusions

Inter-packet times normalization makes it necessary to delay the transmission of packets and generate dummy packets. Parameters of inter-packet times normalization method must be correctly selected to minimize the negative impact on communication channel capacity. Channel performance requirements may be different. They depend on how you use the channel. The results also show that the packet delays and the number of dummy packets are strongly depend on the communication channel load.

Complete normalization of inter-packet delays is only way to eliminate covert timing channels. However, this measure greatly reduces the communication channel capacity and should be used only if the leakage of a very small amount information is unacceptable. In other case, one can use methods of partial inter-packet times normalization which limit covert channel capacity.

Acknowledgements

This work was supported by Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Professional Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

References

[1] Lampson BW. A note on the confinement problem. Communications of the ACM, 1973: 613–615.

[2] Department of defense standard. Department of defense trusted computer system evaluation criteria, 1985; 116 p.

[3] Padlipsky MA, Snow DW, Karger PA. Limitations of end-to-end encryption in secure computer networks: technical report. Bedford: MITRE Corporation, 1978; 11 p.

[4] Brodley C, Cabuk S, Shields C. IP covert timing channels: design and detection. Proc. CCS 2004: 178–187.

[5] Cabuk S. Network covert channels: design, analysis, detection, and elimination : for the degree of doctor of philosophy. Indiana: Perdue University, 2006; 111 p.

[6] Hovhannisyan H, Lu K, Wang J. A novel high-speed IP-timing covert channel: Design and evaluation. Proc. 2015 IEEE International Conference 2015: 7198-7203.

[7] Ahsan K, Kundur D. Practical data hiding in TCP/IP. Proc. ACM Wksp. Multimedia Security, 2002; 8 p.

[8] Yao Q, Zhang P. Coverting channel based on packet length. Computer Engineering 2008; 34.

[9] Zhang L, Liu G, Dai Y. Network Packet Length Covert Channel Based on Empirical Distribution Function. Journal of networks 2014; 9(6): 1440–1446.

[10] Kundur D, Ahsan K. Practical Internet Steganography: Data Hiding in IP. Proc. Texas Wksp. Security of Information Systems, 2003.

[11] Lucena NB, Lewandowski G, Chapin SJ. Covert Channels in IPv6. Proc. Privacy Enhancing Technologies 2005: 147–166.

[12] Alsaffar H, Johnson D. Covert channel using the IP timestamp option of an IPv4 packet. Proc. The International Conference on Electrical and Bio-medical Engineering 2015: 48–51.

[13] Mavani M, Ragha L. Covert channel in IPv6 destination option extension header. Proc. 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications, 2014.

[14] Hu WM. Reducing timing channels with fuzzy time. Journal of Computer Security 1992; 1(3-4): 362–372.

[15] Trostle J. T. Modelling a fuzzy time system. Journal of Computer Security 1993; 2(4): 291–310.

[16] Kang MH, Moskowitz IS. A pump for rapid, reliable, secure communication. Proc. First ACM Conference on computer and communications security 1993: 119–129.

[17] Kang MH, Lee DC, Moskowitz IS. A network version of the pump. Proc. 1995 IEEE Computer society symposium on research in security and privacy 1995; 144–154.

[18] Wang Y, Chen P, Ge Y., Mao B, Xie L. Traffic controller: a practical approach to block network covert timing channel. Proc. International Conference on Availability, Reliability and Security 2009: 349–354.

[19] Mahajan AN, Shaikh IR. Detecting Covert Channels in TCP/IP Header with the Use of Naive Bayes Classifier. International Journal of Computer Science and Mobile Computing 2015; 4(6): 1008–1012.

[20] Rezaei F, Hempel M, Shrestha PL, Rakshit SM, Sharif H. Detecting covert timing channels using non-parametric statistical approaches. Proc. IEEE International Wireless Communications and Mobile Computing Conference 2015; 102–107.

[21] Venkataramani G, Chen J, Doroslovacki M. Detecting Hardware Covert Timing Channels. Journal IEEE Micro 2016; 36(5): 17–27.

[22] Rezaei F. A Novel Approach towards Real-Time Covert Timing Channel Detection : for the degree of doctor of philosophy. Linclon: The University of Nebraska, 2015; 136 p.

[23] Berk V, Giani A, Cybenko G. Detection of covert channel encoding in network packet delays : technical report. New Hampshire: Thayer school of engineering of Dartmouth college, 2005; 11 p.

[24] Sellke SH, Wang C-C, Bagchi S. TCP/IP Timing Channels: Theory to Implementation. Indiana: Purdue University, 2009; 9 p.

[25] Murdoch SJ. Hot or not: revealing hidden services by their clock skew. Proc. 13th ACM conference on computer and communications security 2006; 27–36.

[26] Masti RJ, Rai D, Ranganathan A, Muller C, Thiele L, Capkun S. Thermal Covert Channels on Multi-core Platforms. Proc. 24th USENIX Security Symposium 2015; 865–880.

[27] IBM Knowledge Center. URL: http://www-01.ibm.com/support/knowledgecenter (05.01.2017).

[28] Inside Skype for Business. URL: http://blog.insidelync.com/2012/06/a-primer-on-lync-audio-quality-metrics/ (05.01.2017).

[29] Alreja A. Understanding quality of experience alerting. Redmond: Microsoft Corporation, 2011; 15 pp.