# A method of IPD normalization to eliminate IP timing covert channels

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

[a] *National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), 115409, 31 Kashirskoe shosse, 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 eliminate the information leakage via IP timing covert channels by inter-packet delays normalization in the process of packets' sending. The advantage of our approach is that the influence of counteraction tool on the communication channel's capacity is negligible. The novelty of the investigation undertaken is that the covert channel is eliminated 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; 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.

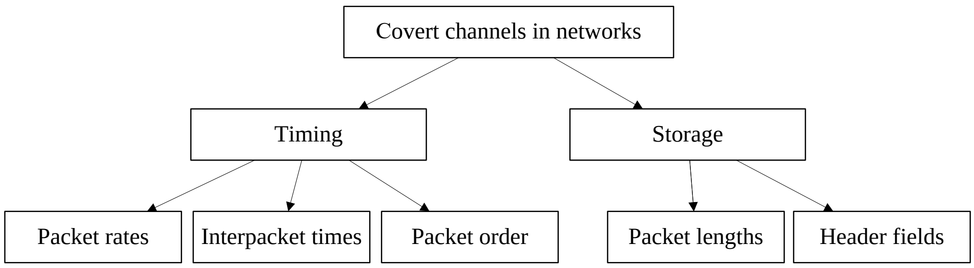


**Fig. 1.** Types of network covert channels.

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

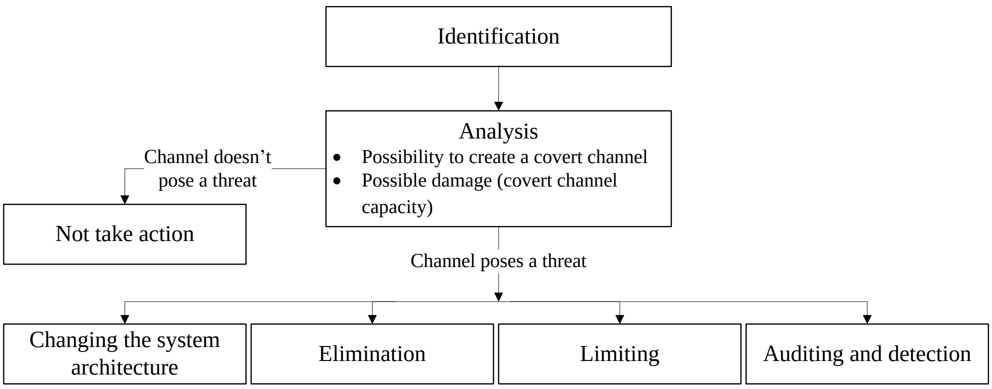


**Fig. 2.** Covert channels handling.

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 IPT normalization method parameters. In Chapter 4, we provide experimental results to demonstrate its effect on network performance. Our conclusions are presented in Chapter 5.

## 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].

Covert timing channels bandwidth can be limited by applying the following measures: packet transmission at a constant rate during a certain time interval $T$; changing of packet rate by a predetermined fixed value at time points $T \cdot k$. In this case covert communication is possible as follows. Covert channel sender can affect the data rate change by transmitting packets to the channel. For example: rate change means that "0" was encoded; no rate change means "1". The capacity of the covert channel is $1/T$ bits per unit time.

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. 3). 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 (Fig.) [16, 17]. The PUMP uses an average of rate of high's ACKs as the rate of sending ACKs to low. For every packet from high to the trusted high process an average of high's ACKs rate is updated. When the trusted low process receives a packet from low it inserts the packet into the buffer, and then sends an ACK to low after a delay. The delay is a random variable chosen from an exponential distribution with the mean equal to the current average of high's ACKs rate. The PUMP model does not eliminate the covert channel, but it does significantly decreases its capacity.

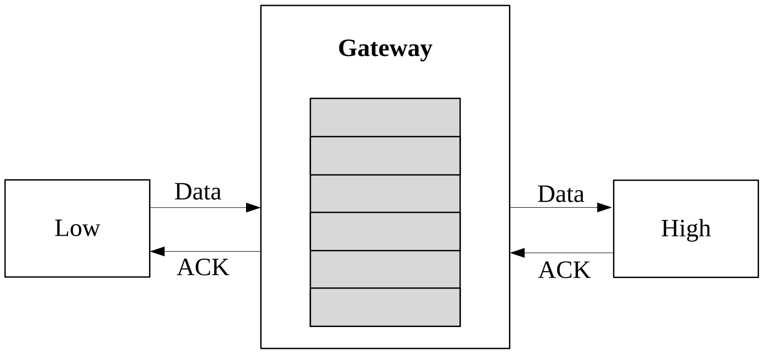


**Fig. 3.** Message passing from low to high using the gateway.

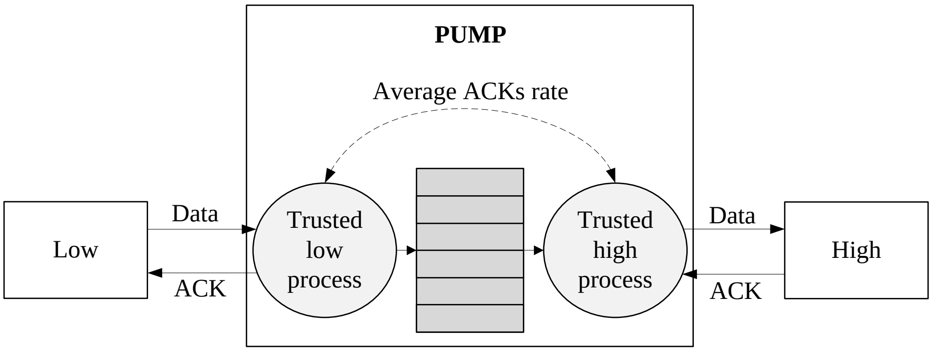


**Fig. 4.** Message passing from low to high using the PUMP.

Capacity of covert timing channels in networks can be limited by adding random delays to the packets. Fig. 5 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.

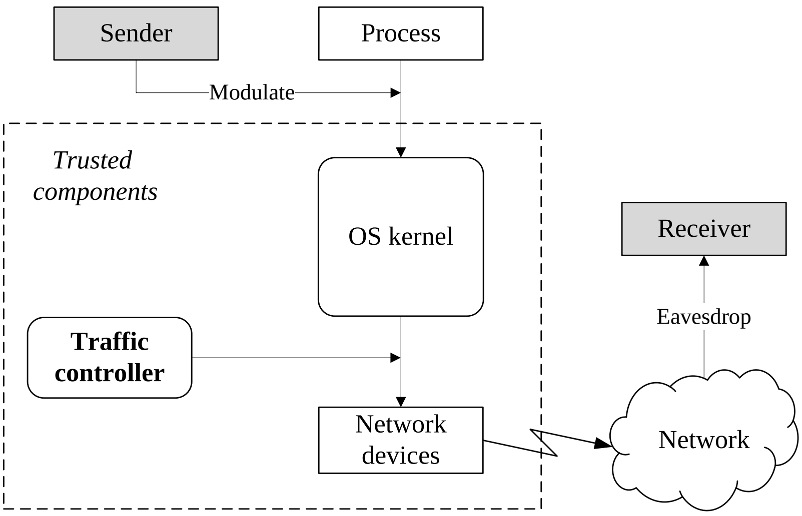


**Fig. 5.** Framework of using traffic controller.

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:

$$T_n = f(\Delta t_n, k) = Rand(k) \cdot \Delta t_n , \qquad (1)$$

where $\Delta t_n$ 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%.

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. Inter-packet times normalization

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. $\varepsilon$ 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 $\varepsilon$,
3. number of dummy packets is minimal.

One of the following values can be used instead of $\varepsilon$ 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. 6). The minimum value of the inter-packet interval is equal to *t* and maximum equal to *mt*.

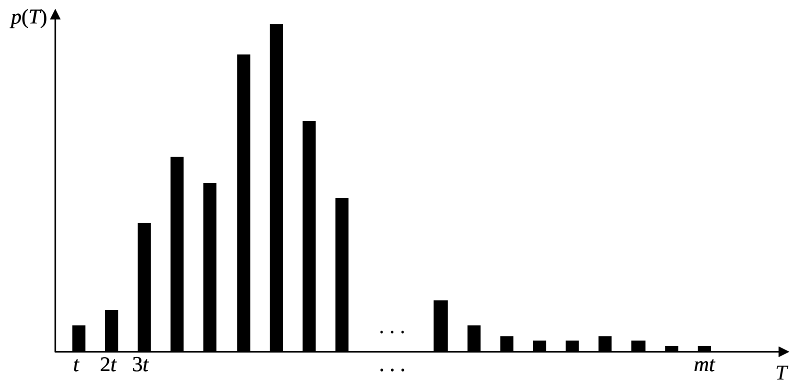


**Fig. 6.** Inter-packet time distribution.

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 $a_1 t, a_2 t, ..., a_n t$ intervals come to the device, delay of the (*i*+1)th packet is equal to

$$d_i t = d_{i-1} t + (N_{d_i} + 1)kt - a_i t, \quad (2)$$

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

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

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 $\xi$ obeying the distribution law in Table 1.

**Table 1.** Distribution law of $\xi$

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

The generating function of the random variable $\xi$ is:

$$\varphi_\xi(z) = p_1z^t + p_2z^{2t} + \ldots + p_mz^{mt}. \tag{4}$$

The random variable $n\xi$ takes the values of the sum of *n* random inter-packet times ($\sum_{i=1}^{n} a_it$). The generating function of the $n\xi$ is:

$$\varphi_{n\xi}(z) = \left(\varphi_\xi(z)\right)^n = \left(p_1z^t + p_2z^{2t} + \ldots + p_mz^{mt}\right)^n. \tag{5}$$

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

$$dt = (n + N_d)kt - \sum_{i=1}^{n} a_it , \tag{6}$$

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:

$$kt \le E(\xi) , \tag{7}$$

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

The probability that the (*n* + 1)th incoming packet will be delayed for longer than *lt* is:

$$P\left((n + N_d)kt - \sum_{i=1}^{n} a_it > lt\right) = P\left(\sum_{i=1}^{n} a_it < (n + N_d)kt - lt\right) = \sum_{i=nt}^{(n+N_d)kt-lt-t} \varphi_{n\xi}\left[z^i\right], \tag{8}$$

where $\varphi_{n\xi}\left[z^i\right]$ is the coefficient in front of $z^i$ in $\varphi_{n\xi}$ which is:

$$\varphi_{n\xi}\left[z^i\right] = \sum_{\substack{b_1,\ldots,b_m: \\ b_1+b_2+\ldots+b_m=n, \\ b_1t+b_2\,2t+\ldots+b_m mt=i}} \left(\frac{n!}{b_1!b_2!\ldots b_m!} p_1^{b_1} p_2^{b_2} \ldots p_m^{b_m}\right). \tag{9}$$

One should choose a value of *kt* for witch this probability is not greater than $\varepsilon$. Furthermore, the value of probability should be as close to $\varepsilon$ 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. 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.

### *4.1. File transfer*

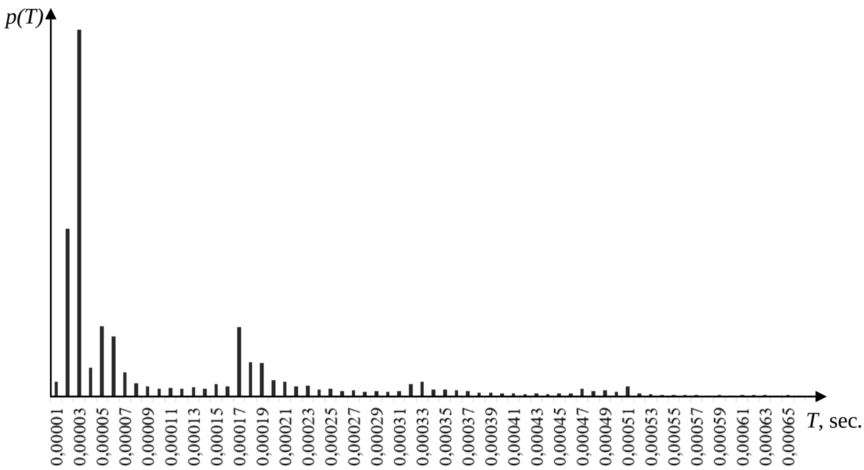


**Fig. 7.** Inter-packet time distribution under high network load ($E(T)$ = 0.00017 sec.).

**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 |

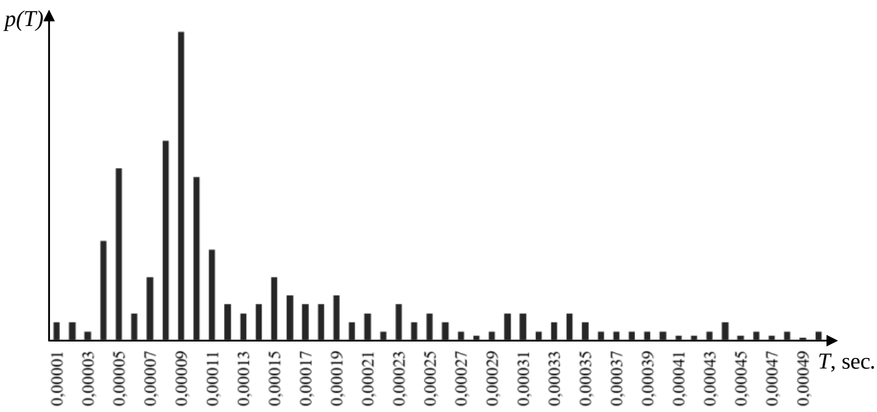


**Fig. 8.** Inter-packet time distribution under low network load ($E(T)$ = 2.33749 sec.)

**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 |

### *4.2. Real-time data transfer*

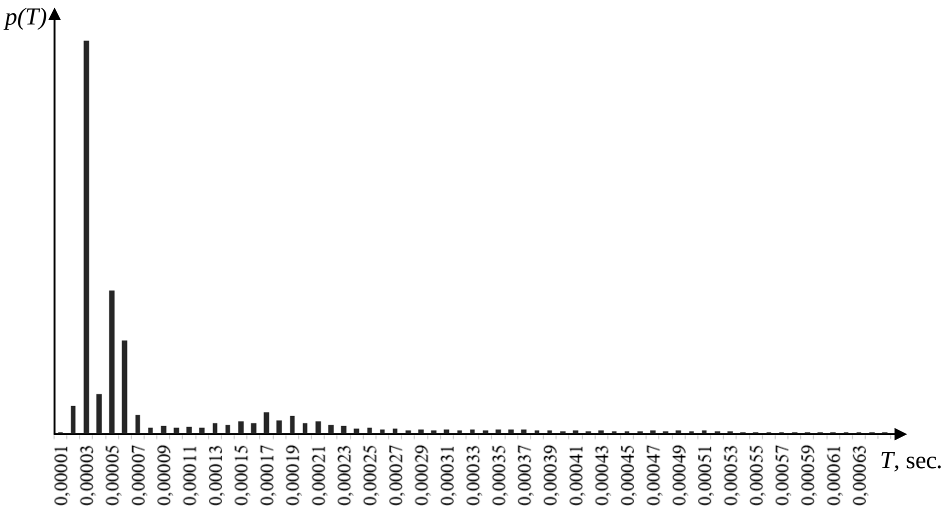


**Fig. 9.** Inter-packet time distribution under high network load ($E(T)$ = 0.00025 sec.)

**Table 6.** Results of calculation of *kt* based on the maximum acceptable packet delay (high network load; $\varepsilon$ = 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 |

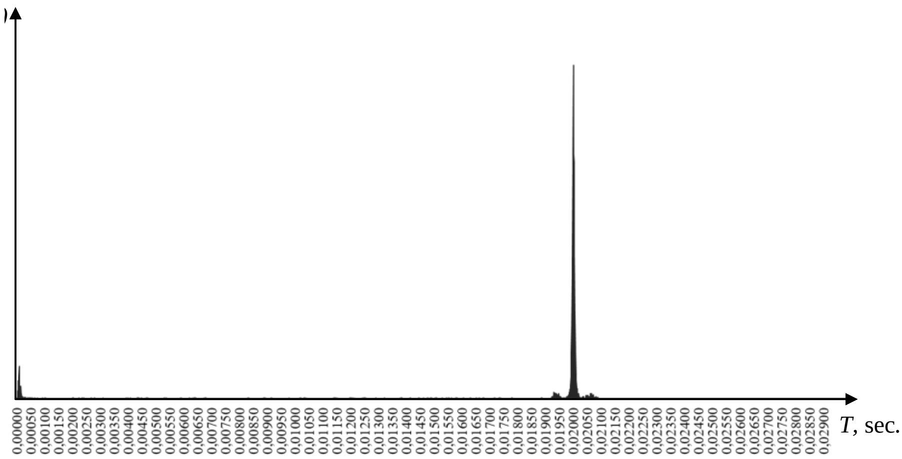


**Fig. 10.** Inter-packet time distribution under low network load ($E(T)$ = 0.02472 sec.)

**Table 7.** Results of calculation of *kt* based on the maximum acceptable packet delay (low network load; $\varepsilon$ = 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. Conclutions

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.


## Acknowledgements

This work was supported by the Competitiveness Program of MEPhI.


## References


[1] Lampson, B. W. A note on the confinement problem / B. W. Lampson // Communications of the ACM. — 1973. — P. 613–615.

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

[3] Padlipsky, M. A. Limitations of end-to-end encryption in secure computer networks : technical report / M. A. Padlipsky, D. W. Snow, P. A. Karger. — Bedford: MITRE Corporation, 1978. — 11 p.

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

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

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

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

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

[9] Zhang, L. Network Packet Length Covert Channel Based on Empirical Distribution Function / L. Zhang, G. Liu, Y. Dai // Journal of networks. — 2014. — Vol. 9. no 6. — P. 1440–1446.

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

[11] Lucena, N. B. Covert Channels in IPv6 / N. B. Lucena, G. Lewandowski, S. J. Chapin // Proc. Privacy Enhancing Technologies. — 2005. — p. 147–166.

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

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

[14] Hu, W. M. Reducing timing channels with fuzzy time / W.M Hu // Journal of computer security. — 1992. — Vol. 1. no 3—4. — P. 362–372.

[15] Trostle, J. T. Modelling a fuzzy time system / J. T. Trostle // Journal of Computer Security. — 1993. — Vol. 2, no. 4. — P. 291–310.

[16] Kang, M. H. A pump for rapid, reliable, secure communication / Kang M. H., Moskowitz I. S. // Proc. First ACM Conference on computer and communications security. — 1993. — P. 119–129.

[17] Kang, M. H. A network version of the pump / M. H. Kang, D. C. Lee, I. S. Moskowitz // Proc. 1995 IEEE Computer society symposium on research in security and privacy. — 1995. — P. 144–154.

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

[19] Mahajan, A. N. Detecting Covert Channels in TCP/IP Header with the Use of Naive Bayes Classifier / A. N. Mahajan, I. R. Shaikh // International Journal of Computer Science and Mobile Computing. — 2015. — Vol. 4, no. 6. — P. 1008–1012.

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

[21] Venkataramani, G. Detecting Hardware Covert Timing Channels / G. Venkataramani, J. Chen, M. Doroslovacki // Journal IEEE Micro. — 2016. — Vol. 36, no. 5. — P. 17–27.

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

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

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

[25] Murdoch, S. J. Hot or not: revealing hidden services by their clock skew / S. J. Murdoch // Proc. 13th ACM conference on computer and communications security. — 2006. — P. 27–36.

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

[27] IBM Knowledge Center [Electronic resource]. — Access mode: http://www-01.ibm.com/support/knowledgecenter (05.01.2017)

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

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