Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **6**, November **2015**

SkyLen: A Simple Covert Timing Channel Based on Huffman Encoding

Wen Tian, Jiang-Tao Zhai*

School of Electronics and Information
Jiangsu University of Science and Technology
No.2 Mengxi Road, Zhenjiang, Jiangsu, P.R.China
csusttianwen@163.com
*Corresponding author: jiangtaozhai@gmail.com

Ming-Qian Wang

School of Information Engineering
Changzhou Institute of Mechachronic Technology
No.26 Minxin Road, Changzhou,Jiangsu,P.R.China
wmq1989219@126.com

Received June, 2015; revised September, 2015

Abstract. *Covert timing channel is a mechanism that can be exploited by an attacker to conceal secrets in timing intervals of transmitted packets. With the development of this technique, the undetectability of covert timing channel has improved greatly. However, due to the complexity of network environment, it has become increasingly difficult to build a covert timing channel with both well stealth and robustness. In this paper, after analyzing the previous work, a simple covert channel based on Huffman encoding is proposed. In our method, firstly, mimic function is used as the basis and RTT (Round-Trip Time) feedback is added as the improvement to mimic the behavior of legitimate traffic. Secondly, the symbols from Huffman encoding are divided into two parts. Finally, the entropy detection test is utilized to evaluate the undetectability of our approach and the error rate is calculated to evaluate the robustness of our covert traffic. Compared with the previous work, the experimental results show that this simple covert channel can successfully evade detection and possesses high robustness.*

Keywords: Covert channels, Mimic function, Information hiding.

1. **Introduction.** Covert channel, which is originally studied in the context of multi-level secure systems, is first introduced by Lampson[1]. However, compared with information hiding over things, covert channel has no quantitative limit of secret information and allows secret information transmitting during a long period of time. Ever since Girling started the study of covert channel in the network scenario[2], and the security threat posed by network covert channel has attracted increasing attention.

Traditionally, network covert channel can be classified into covert storage channel and covert timing one. For the former one, the sender embeds the secret message by modifying some inconspicuous or unused bits of covert traffic (e.g., packet headers)[3, 4, 5]. While for the latter one, the secret messages are modulated into the timing information of covert traffic (e.g., inter-packet delays)[6, 7]. Recently, with the development of the Deep Packet Inspection (DPI) and Deep Flow Inspection (DFI) techniques, the storage channel exposes some limit in security. However, due to the complexity of network, timing channel reveals better performance in undetectability. Taking the previous work based on Huffman encoding as an example, it is found that they are quite difficult to detect. As it is known to all, undetectability is a main goal for designing covert channels. Here undetectability means the adversary cannot detect the existence of covert channel by distinguishing between covert and legitimate traffic [8]. Some popular ways like pattern recognition methods or statistical tests are designed to detect covert storage channels [3], among which Kullback-Leibler (K-L) divergence test [9], Kolmogorov-Smirnov (K-S) test [10] and entropy test [11] are utilized frequently. Meanwhile, the characteristic of legitimate traffic must be considered when designing the covert channel. The previous works[12] did pretty well in undetectability. However, as network jitter always exists, especially for the growing wireless links, signals over wireless channel always suffer from several channel disruptions including channel noise, fading, pathloss and interference [13, 14]. Therefore, the adversary may disrupt the covert channel by maliciously adding additional noise. In fact, good robustness is the foundation for guaranteeing the success of covert communication. However, the previous work is easily affected by the network noise, failing to transmit secret message accurately. Therefore, an improved method based on the previous works is introduced to improve its robustness. There are two high lights in this paper:

1)Compared with the previous work, the modeling process is less complexity than before. The different symbolizations which are distributed from different order MBCTC are all divided into two parts. Meanwhile, the bits string can be decoded one by one instead of three or four bits.

2)A simple Huffman MBCTC is proposed, the robustness of which becomes well enough to resist the network jitter and channel noise.

The rest of this paper is organized as follows. In the next section, the related works in covert timing channels are reviewed. In section 3, the proposed scheme is introduced. In section 4, experimental results are presented and analyzed. Finally, the paper is concluded in section 5.

2. **Related works.** Our work concentrates on the design of undetectable and robust MBCTC scheme. For the timing information of packets can be utilized to leak information, Caulk [15] et al. presented IPCTC, which is the first covert timing channel based on IP packets. The IPCTC employs a simple on/off encoding scheme. During a specific timing interval, the sender transmits a bit '1' by sending a packet and a bit'0' by not sending a packet. For transmitting message safely and effectively, Walls et al. [19] presented a detection-resistant covert timing channel relying on the idea of Jitterbug. The main idea is to insert shaping IPDs into covert traffic, so as to smooth out shape distortion generated by Jitterbug. Jitterbug presented a way, which communicates synchronously as Secure Shell, to leak secret information by knock the keyboard in different frequency. However, these channels are generally detectable due to their regularity - as it is generated by a computer program, it does not has the variations found in human-generated traffic. For this reason, mimic was proposed and an active covert channel that mimics both the shape and regularity of legitimate traffic to disguise its presence. Gianvecchio et al. [16] presented a model based covert channel, which named MBCTC in short. The distribution function for legitimate traffic is modeled by MBCTC, which has the minimum mean square error (RMSE) between mimic covert channel and legitimate traffic. The features of legitimate traffic had been analyzed and the traffic behavior observed to a model. However, not all MBCTC can evade detection successfully, Kotharil[20] et al proposed an undetectable covert timing channel based on a mechanism of Regularity Tree to mimic the irregularity tree of legitimate traffic, and he found an active CTC that is able to evade all known detection techniques while maintaining throughput of one bit per

IPD. Furthermore, Wayner[21] et al. presented a model based covert channel on the tree, which building the tree from the bottom up instead of the top down. Because of the tree consist of root, branch and leaf, thus, the encoding become convenient and easy. Besides, there are many researchers utilized CTCs based on other HTTP traffic. WI-Atawy and AI-Shaer[22] proposed a covert timing channel and utilized grouping arrangement function. However, these previous works cannot perform well in robustness.

As it is known to all, the robustness plays an important role in design covert timing channel, and lots of methods have been proposed to improve it. Liu[23] et al. utilized i.i.d traffic to mimic legitimate traffic and built a covert timing channel which means the data belong to i.i.d distribution is undetectable. And it has high performance in robustness. It utilized extended code to resist net jitter and signal distortion. However, the security of this channel cannot evade detection perfectly. Sellke et al. [18] provided two main improvements in robustness. The first is to quantify the threat posed by covert network timing channels. The other is to use timing channels to communicate at a low data rate without being detected. Kirovski[24] et al. utilized spread spectrum technology to defend synchronization attack.

This paper ultilized Huffman model based covert timing channel to transmit information. Due to Huffman tree is one kind of a regularity treeit can evade detection successfully. However, the robustness of Huffman MBCTC is low. Some improvements are proposed to improve it. A detailed description of our proposed scheme is presented in the next section.

3. **The proposed scheme.**

In this section, a mimic function is introduced firstly. It is the previous work and proposed by Jing Wang [12]. Then, the performance under the influence of network jitter has been analyzed. The results shown its error rate of transmission are a little high, which means it has low robustness in transmission. In the end, a method to improve the robustness of covert timing channel has been proposed.

3.1. **mimic function:**

Mimic functions [21], which introduced by Peter Wayner, are used to transmit hidden information as a subliminal technique. A mimic function changes input data so it assumes the statistical properties of another type of data, and consequently accomplishes the mimicry of identity. This encryption technique has been applied in various scenarios, e.g., text steganography, digital watermarking and code obfuscation.

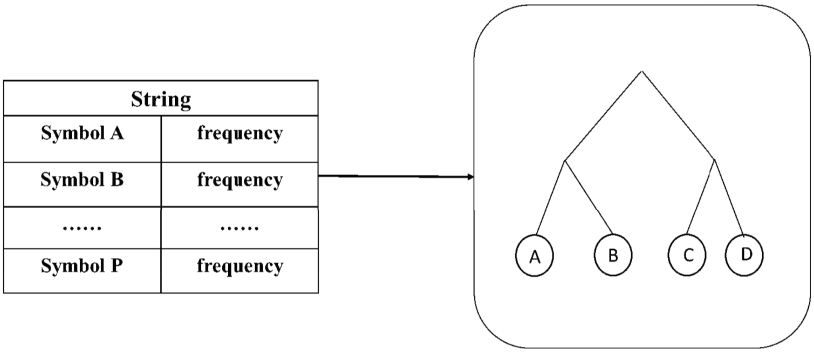
First-order Huffman mimic function: In first-order Huffman encoding, a table of occurrence frequency for each symbol in the input is required to build a variable-length code table. According to it, a binary tree of nodes, namely a Huffman tree, is generated. As a result, the symbols which occur frequently reside at the top while the rare symbols are represented as long codes and located at the deep.

The inverse of Huffman code can be used as mimic functions if the input is a random bit stream. The mimic process consists of two parts: compression phase and expansion phase. In the compression phase, the frequency table of each symbol in data set A is estimated and the corresponding Huffman tree is constructed. In the second phase, a data set B of random bits is expanded, specifically, variable length blocks are converted into fixed length blocks due to the Huffman decoding operation, which is based on the Huffman tree of A.

However, there is a problem that symbols occur in regular mimic functions output with different probabilities from the original ones. In fact, the regular model limits all symbols to have a probability which is a negative power of two, e.g., .5, .25, .125, and so on. These regular models can make codes in same length. The following technique can be used to solve this problem.

High-order Huffman mimic function: Compared with the first-order Huffman mimic function, high-order Huffman function captures more detailed statistical profile of the data. In order to maintain regularity of the data, high-order mimic Huffman functions extract the inter-symbol dependencies by estimating the frequency of each symbol that follows a specific string of length n-1. High-order mimic functions build a Huffman tree for each occurred string of length n-1, as shown in Figure 1.

As a start, the encoding of high-order mimic functions requires one possible string as a seed. Given the seed, the encoding program locates the right Huffman tree with the prefix of that selected string in the forest, and then uses the Huffman decoding operation to determine the symbol that will follow the string. The resulting symbol and its preceding string of length n-2 form a new prefix of length n-1. The encoding program will take iterations in this order and output one by one. With the increasing of the order, the results become more and more approximate to the original data.

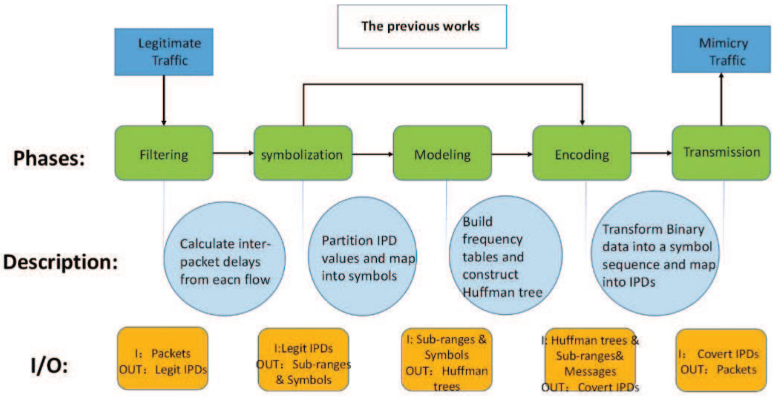


String	
Symbol A	frequency
Symbol B	frequency
……	……
Symbol P	frequency

Figure 1. Huffman tree

3.2. **The previous works scheme.** The framework includes five phases: filtering, symbolization, modeling, encoding and transmission. Figure 2 gives an overview of our framework and a concise description of each phase. Details are expanded in the following paragraphs

Filtering: In this phase, the packets from legitimate network connections are captured. The packets are then classified into individual flows according to protocol type, source and destination IP addresses and ports. Generally, different types of traffic have different statistical properties. For example, HTTP and SSH protocols are both based on TCP/IP, and the difference between their traffic behaviors also exists and has been revealed by statistical tests. Furthermore, the more specific traffic to be filtered out, the more precise



The previous works
Legitimate Traffic
Mimicry Traffic
Phases:
Filtering
symbolization
Modeling
Encoding
Transmission
Description:
Calculate inter-packet delays from eacn flow
Partition IPD values and map into symbols
Build frequency tables and construct Huffman tree
Transform Binary data into a symbol sequence and map into IPDs
I/O:
I: Packets OUT: Legit IPDs
I:Legit IPDs OUT: Sub-ranges & Symbols
I: Sub-ranges & Symbols OUT: Huffman trees
I: Huffman trees & Sub-ranges& Messages OUT: Covert IPDs
I: Covert IPDs OUT: Packets

Figure 2. The framework of previous works

statistical properties can be mimicked. For this reason, a specific application protocol is chosen as a filtering condition. After trace classification, the packet analyzer calculates timing intervals between adjacent packets from each trace.

Symbolization: The input IPDs are mapped into corresponding symbols in this phase. In our application scenario, the objective is to mimic the statistical properties of legitimate traffic and to cover up the presence of covert timing channels. If IPD values are mapped into symbols one-to-one, the symbol set will be oversized due to the high variation in HTTP traffic, resulting in overload of the encoding program. To solve this problem, all IPDs in ascending order are sorted, and partition IPD range into several sub-ranges. IPDs in the same sub-range are mapped to the same symbol.

The most important factor is to partition the sub-range. Our approach is to make symbols based on the same probability which have the same Huffman code. So the sorted IPDs which will be transmitted at once in an ascending order. This order will be very helpful to encoding in next phase.

Modeling: During the modeling phase, statistical properties of legitimate IPDs is extracted by constructing Huffman trees. This phase takes the sequence of symbolized IPDs as the modeling object and mimicry target.

The first step is building frequency tables for all occurred strings of length n-1 in order to extract the nth order statistics. The modeling function processes the input sequence in a sequential manner. When moving onto a string of length n-1, the function determines whether this pattern has already occurred. If so, the function gets the symbol that follows the string, and increments the corresponding frequency counter in the table appended to this pattern. If it is absent, then a new table is created and appended to it. These operations are repeatedly conducted until the search is completed. If the input sequence has a total of 2000 symbols, 2001-n strings of length n-1 can be collected, some of which may have the same pattern. The strings of length n-1, acting as the ”prefixes” in mimic functions, are the indexes of their frequency tables. At the end of modeling phase, each frequency table is converted into a Huffman tree. All the Huffman trees with corresponding indexes are output to the next phase.

Encoding: Based upon the modeling results, the encoding phase converts an arbitrary binary stream into a sequence of symbols, which has similar statistics with the mimicry target. The encoding function has three components: randomization, mimic encoding process, and inverse mapping.

Covert messages are mostly encoded into binary sequences with ASCII scheme, and each letter is represented as a code with the length of 8 bits. Due to different occurrence frequency of letters, these binary sequences are non-random, whereas the input of mimic functions is required to be random. To solve this problem, each binary sequence is randomized by XORing with a pseudo-random bit stream. This stream is assumed to be known by the sender and receiver.

After randomization, the input of the mimic encoding process is an arbitrary binary stream. To start the encoding of the nth order mimic functions, the first string of length n?1 is chosen as a seed. The right Huffman tree of the specific string is located according to its index. Then the process performs the operation of walking the tree. From the root node, the process selects left or right branches according to the input bit, until it arrives at a leaf node, which represents a symbol. The most recently generated symbol and its preceding string of length n?2 forms a new string of length n?1. The encoding program will take iterations in this order and output one by one. In the end, a sequence of symbols is obtained.

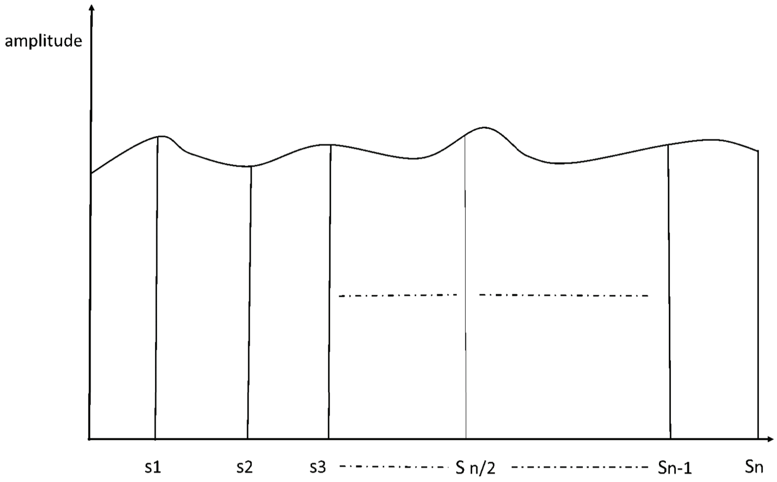


Figure 3. Symbols in Huffman coding

Inverse mapping refers to an operation of inversely mapping symbols to IPD values. First, one symbols is mapped into the corresponding sub-range. Then, an IPD value is randomly selected from the sub-range.

Transmission: In this phase, the sender of the covert timing channel modulates the timing of packets corresponding to the sequence of IPDs, and then forwards the packets to the Internet.

3.3. **The previous work and our improved scheme.** Given the advantage of high-order Huffman mimic function, the previous works of Jing Wang which transmit the input information into a string of binary bits, divide the string into several blocks and the length of each block is n. Blocks compared with the Huffman code, each block can be found out in each located symbol. However, the problem is quite obvious that the information transmitted in the internet would be affected by network jitter. Meanwhile, the original sub-range of symbols cannot figure out the right IPDs location. The error rate of the 3rd-MBCTC, the 4th- MBCTC and the 5th- MBCTC are calculated in experiments, and the results are 84%, 94% and 96% in the most. These results show that the previous works cannot meet the requirement of practical transmission. To solve this problem, the RTT (Round-Trip Time) from legitimate traffic had been calculated. In order to improve the correct rate, RTT is utilized to expand the size of symbol because of the RTT can keep unchanged in short time, and symbols are divided into two parts 0 and 1 as shown in Figure 3. (S1-$S_{n/2}$ are defined as part 0 and $S_{n/2}$-S_n are part 1)

The size, which expanded by RTT feedback, called the extra volume $\boldsymbol{l_R}$. Compared with the original symbols size $\boldsymbol{l}$ the practice size finally encoded is $\boldsymbol{l_n}$ (the sub-range of a part).With the network dithering interference, the expanded size can successfully resist it. Meanwhile, the model based CTCs proposed by this paper has high capacity of robustness.

In definition, t_R is the feedback from RTT, and t_s is the difference between the largest IPD and the smallest IPD in a part. The number of IPDs which a part involved is l. The size function defined in Eq. (1) (2):

$$l_R = \frac{t_R}{t_s} \times l \tag{1}$$

$$l_n = l_R + l \tag{2}$$

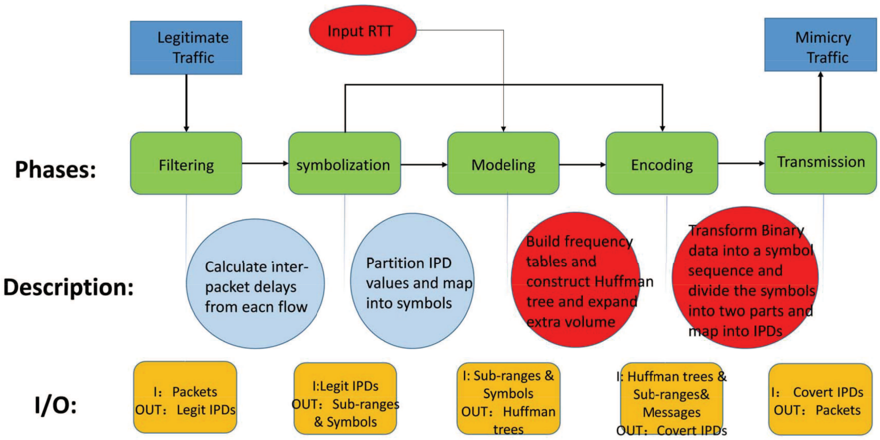

Legitimate Traffic
Input RTT
Mimicry Traffic
Phases:
Filtering
symbolization
Modeling
Encoding
Transmission
Description:
Calculate inter-packet delays from eacn flow
Partition IPD values and map into symbols
Build frequency tables and construct Huffman tree and expand extra volume
Transform Binary data into a symbol sequence and divide the symbols into two parts and map into IPDs
I/O:
I: Packets OUT: Legit IPDs
I:Legit IPDs OUT: Sub-ranges & Symbols
I: Sub-ranges & Symbols OUT: Huffman trees
I: Huffman trees & Sub-ranges& Messages OUT: Covert IPDs
I: Covert IPDs OUT: Packets

Figure 4. The framework of our improved method

The improved framework includes five phases similar to the previous work: filtering, symbolization, modeling, encoding and transmission. However, some improvement in modeling and encoding are proposed. Figure 4 gives an overview of our framework and a concise description of each phase. The red in this figure is the improvement of our scheme. Details are expanded in the following paragraphs.

Modeling: The first step is building frequency tables for all occurred strings of length n?1 in order to extract the nth order statistics. Then, RRT feedback added before building the frequency tables to determine the extra volume that size should be expanded.

Encoding: The encoding program will take iterations in one order and output one by one. Then, to make the number of symbols become less, when a sequence of symbols is obtained, they are divided into two parts equally according to priority, the former part is 0 and the later one is 1.

Inverse mapping refers to an operation of inversely mapping parts to IPD values. First, one part is mapped into the corresponding sub-range. Then, an IPD value is randomly selected from the sub-range.

3.4. **Designs Details. Prior agreements**The sender and receiver should share the same mimic target and have the same modeling results. One choice is to send modeling results to the receiver which requires a large amount of communication traffic before channel built up. The other choice is to collect the same legitimate traffic as the mimic target by the two parties. This method can solve decoding successfully even if some packets lost. Our partitioning method is based on the cumulative distribution function of IPDs, and thus the slight changes during the transmission almost have no effect on the overall distribution. Moreover, slight changes have little influence on the generated Huffman trees.

Decreasing error rate: In the symbolization phase, the distribution of IPDs in each part is treated as the uniform distribution approximately. Accordingly, an IPD value is uniformly selected from the corresponding sub-range in the inverse mapping operation. When the selected IPD value is close to the cut-of point between the adjacent sub-ranges, a transmission error will probably arise due to network jitters. One solution is adding error correcting bits to covert data. In addition, the areas with a certain distance apart from cut-off points can be set with the feedback from RTT, and the symbols are divided into two parts which means there is only one cut-off point. However, this solution theoretically has a little influence on the mimicry effectiveness.

Table 1. The mean of test score and comparative score

Test	legitimate	3rd-MBCTC		4th-MBCTC		5th-MBCTC	
	Mean	Mean	Diff.	Mean	Diff.	Mean	Diff.
EN	1.8760	1.8830	0.0043	1.8785	0.0025	1.8768	0.0008

4. Experiments and analysis.

A series of experiments are conducted to validate the effectiveness of our mimicry approach. The emphasis of our experiments is on determining if the covert timing channel exploited from the mimicry framework can evade the detection tests and resist the network jitters.

4.1. Experiment Setup.

The detection tests are based on the statistical analysis, therefore a large volume of network data is necessary in order to perform these tests. The publicly available data sets which from the communication of YY audio-a network service based on P2P, which satisfies the above requirement. Thus, it is adopted as the normal carrier in the implementation of this paper. The normal traffic of YY audio is captured from an intermediate router during the communication of the two hosts within our campus networks. The data sets contain the mixed traces of diverse network protocols.

4.2. Stealthness and robustness.

In the experiments, the most commonly used detection method-the entropy detection method is utilized to examine the shape and regularity respectively. In this section, the detection method is detailed and the results are also shown.

Entropy detection method: In information theory, the entropy (EN)of a discrete random variable X is a measure of amount of uncertainty associated with the value of X. If the X is the set of all messages $(X_1, \ldots, X_n)$ that X could be, and p(x) is the probability of some x belong to X. Then the entropy, EN, of X is defined in Eq. (3) :

$$EN(x) = -\sum_{x \in R} p(x) \log p(x) \tag{3}$$

The EN need data from legitimate traffic and mimic traffic. Thus, the 3rd-MBCTC, the 4th- MBCTC and the 5th- MBCTC mimic traffic's EN are calculated and shown in Table 1. According to Table 1, the 5th- MBCTC's EN is the most closely similar to the legitimate traffic, furthermore the 3rd-MBCTC and the 4th- MBCTC's EN is close to the legitimate traffic too. Meanwhile, the EN of legitimate traffic is 1.8760, while the EN of 3rd-MBCTC is 1.8803 and the difference between them is 0.0043. Besides, the EN of 4th-MBCTC is 1.8785 and the difference between it and legitimate traffic is 0.0025. The EN of 5th-order Huffman is 1.8768 and the difference is 0.0008. The conclusion shows that the higher the order is, the similarity the mimic traffic will be.

To be more convincing, Kolmogorov-Smirnov test also ran to against MBCTC. The Kolmogorov-Smirnov test quantifies the maximum distance between two empirical distribution functions, so it can be used to examine the shape of traffic. If the test score is low, it implies that the sample is close to the legitimate behavior. For the regularity test, network traffic is separated to several windows with the same size, and the standard deviation is computed for each window. In general, the regularity score of legitimate traffic is high. That is because the legitimate traffic changes over time. This test can be used to examine the regularity of traffic. Our results showed that the k score of these three MBCTC are 0.0087, 0.0056 and 0.0012. That means our MBCTC can evade detection successfully.

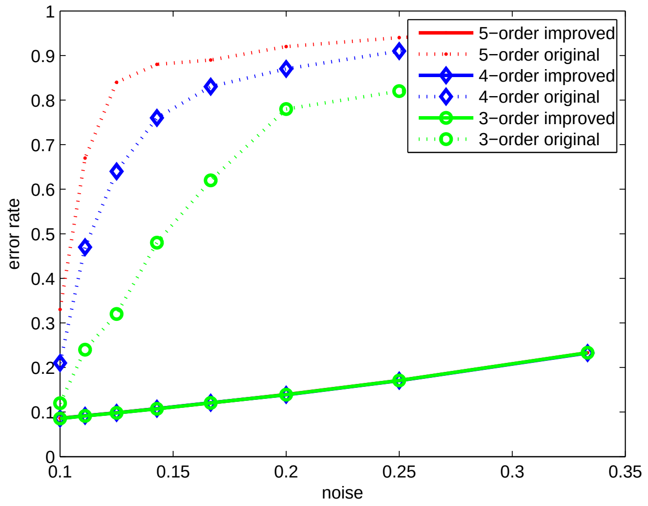


Figure 5. The comparison between improved and original

Robustness test: To test the robustness of these MBCTCs, the error rates in network jitters are calculated. To calculate the error rate, the inverse XORing of 0-1 bit string is decoded which come from the input information IPDs transmitted by sender, and it compared with the original 0-1 bit string. To mimic network jitters, the average of IPDs which shared by sender and receiver are made as the basic, one-tenth, one-ninth, one-eighth until one-third of average are used to interfere the legitimate traffic and the mimic traffic. The test result shown in Figure 5, Figure 6 and Figure 7.

In this figure, MBCTC by distributing symbols is improved into two parts 0 and 1 as it had set before. Obviously, the original MBCTC have high error rate that it almost cannot be used to do practical transmission. Not only the 3rd-MBCTC has high error rate, but also the 4th- MBCTC and the 5th- MBCTC have higher error rate neither. Thus, the more blocks the symbols divided into, the higher error rate the IPD mapped will be. Meanwhile, the blocks are distributed into two parts to decrease the error rate. The experiment's results verify our suspicion successfully. Due to symbols, which distributed in different order mimic function, are divided into 2 parts. So, no matter the 3rd-MBCTC, the 4th-MBCTC or the 5th- MBCTC all have the similar curve about error rate. Furthermore, due to MBCTC have been improved, the error rate decrease obviously in the Figure 5, which decrease almost up to 0.2 (20%) in transmission. However, the error rate of 20% can't meet the actual requirements that the error rate is still too high. Therefore, RTT feedback added to further decrease the error rate, which can control the extra volume to incase the IPD mapped into wrong part. The experimental results are shown in Figure 6.

It can be seen from this figure that, when the RTT feedback is added, the highest error rate of the high -order MBCTC had decreased to only 0.005 (0.5%). Meanwhile, this MBCTC can meet the requirement of practice and it can be used in common traffic. However, High-order MBCTC is more complicated than first-order one as we know. Thus, whether only high-order MBCTC could have this robustness to transmit information? Therefore, the first-order MBCTC, which add feedback of RTT, is utilized as the mimic traffic to transmit the information. Then, the error rate of it can be calculated and compared with High-order MBCTC. The results are shown in Figure 7.

The comparison results of the error rate in Fig.7 show that the first-order MBCTC has higher error rate than high-order MBCTC (the MBCTC in this figure all improved

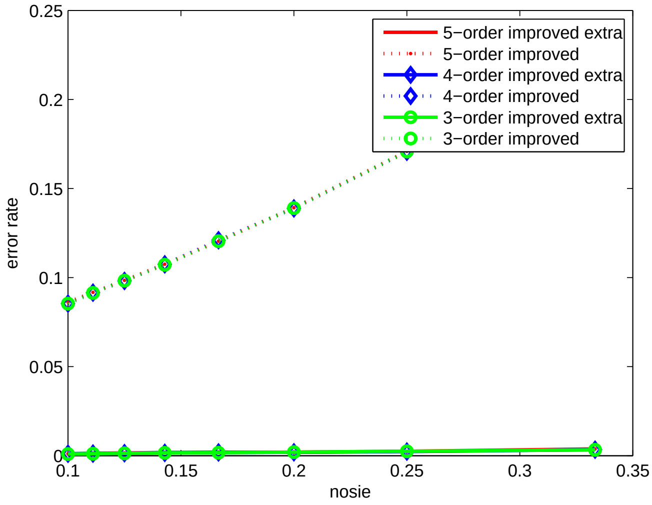


Figure 6. The comparison between MBCTC improved add extra volume and improved without extra volume

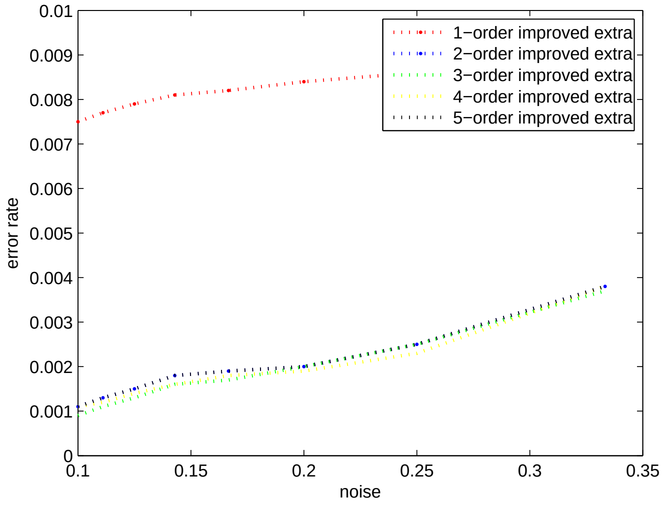


Figure 7. 1st-MBCTC-5th-MBCTC improved add extra size

to raise them robustness). Obviously, with the noise increases, the error rate of first-order MBCTC is closing to 1%. However, high-order MBCTC has lower error rate and the difference between it and first-order one is so large that shown in the figure clearly. Meanwhile, although the first-order is less complicate, but it cannot replace the high-order MBCTC.

.

5. **Conclusions.** In this paper, in consideration of undetectability and robustness, a simple covert timing channel based on the previous work has been proposed. A mimicry framework is implemented to automatically generate covert traffic. The framework includes five phases: filtering, symbolization, modeling, encoding, and transmission. The traffic generation process is as follows. Firstly, IPD values are filtered form legitimate traffic. Secondly, they are mapped into corresponding symbols and these symbols are

divided into two parts. Thirdly, the statistical properties are extracted through building Huffman trees. Next, based upon the modeling results, the encoding phase converts a binary stream into a correct part, which adds RTT feedback, and then inversely maps the part into IPD values. Finally, network packets with the mimicry IPDs are forwarded to the Internet.

In order to validate the effective ness of the mimicry approach, the detection tests added network jitters against our channel (MBCTC improved) is performed and the original channel (MBCTC original) which provided by Jing Wang is performed in the same situation too. The results show that both of them had high capacity about undetectable, but the improved MBTCT had higher robustness than the original one. Moreover, high-order MBCTC had higher robustness than first-order MBCTC.

Acknowledgment. The authors are grateful to the anonymous reviewers for their insights and fruitful comments during the reviewing process, which greatly contributed to improving the quality of the original manuscript.
This work is supported by the NSF of China (Grant No.: 61170250, 61472188) and NSF of Jiangsu province (Grant No.: BK20150472) and the project of college advantage subject construction of Jiangsu province.

REFERENCES

[1] B. W. Lampson A note on the confinement problem, *J. Communications of the ACM,* vol.16, pp. 613-615, 1973.

[2] C. G. Girling, Covert Channels in LAN's *J. IEEE Trans. on Software Engineering,* , vol. 2, pp. 292-296,1987.

[3] S. J. Murdoch, S. Lewis, Embedding covert channels into TCP/IP, *Proc. on Conference of Information Hiding*, Springer Berlin Heidelberg, pp. 247-261, 2005.

[4] L. Zhang, G. Liu, Y. Dai, Network Packet Length Covert Channel Based on Empirical Distribution Function, *J. Journal of Networks*, vol.9, pp. 1440-1446, 2014.

[5] W. Mazurczyk, VoIP steganography and its Detection-A survey, *J. ACM Computing Surveys (CSUR)*, vol. 46, pp. 20, 2013.

[6] G. Liu, J. Zhai, Y. Dai, Network covert timing channel with distribution matching, *[J]. Telecommunication Systems*, vol. 49, pp. 199-205, 2012.

[7] A. Houmansadr, N. Borisov, CoCo: coding-based covert timing channels for network flows, *Proc. on conference of Information Hiding*, Springer Berlin Heidelberg, pp. 314-328, 2011.

[8] Liu Y, Ghosal D, Armknecht F, *et al.* Robust and undetectable steganographic timing channels for iid traffic, *Conference of Information Hiding*, Springer Berlin Heidelberg, pp. 193-207, 2010.

[9] R. Archibald, D. Ghosal, A comparative analysis of detection metrics for covert timing channels, *J. Computers and Security,* vol. 45, pp. 284-292, 2014.

[10] A. Bates, B. Mood, J. Pletcher, *et al.* On detecting co-resident cloud instances using network flow watermarking techniques, *J. International Journal of Information Security*, vol. 13, pp. 171-189, 2014

[11] A. Chen, W. B. Moore, H. Xiao, *et al.* Detecting covert timing channels with time-deterministic replay, *Proc. on Conference of USENIX Symposium on Operating System Design and Implementation (OSDI),* 2014.

[12] J. Wang, L. Guan, L. Liu L, *et al.* Implementing a Covert Timing Channel Based on Mimic Function, *Information Security Practice and Experience. Springer International Publishing,* pp. 247-261, 2014.

[13] M. Shirvanimoghaddam, Y. Li, B. Vucetic, Near-Capacity Adaptive Analog Fountain Codes for Wireless Channels *J. IEEE Communications Letters*, vol. 17, pp. 2241-2244, 2013.

[14] M. Shirvanimoghaddam, Y. Li, B. Vucetic, Adaptive analog fountain for wireless channels, *Proc. on Conference of Wireless Communications and Networking Conference (WCNC), 2013 IEEE,* pp. 2783-2788, 2013.

[15] S. Cabuk, C. E. Brodley, C.Shields, IP covert timing channels: design and detection, *Proceedings of the 11th ACM conference on Computer and communications security., ACM,* pp. 178-187, 2004.

[16] S. Gianvecchio, H. Wang, D. Wijesekera, *et al.* Model-based covert timing channels: Automated modeling and evasion,*Proc. on Conference of Recent Advances in Intrusion Detection,* Springer Berlin Heidelberg, pp. 211-230, 2008.

[17] G. Shah, A. Molina, M. Blaze, Keyboards and Covert Channels *Proc. on Conference of USENIX Security,* 2006

[18] S. H. Sellke, C. C. Wang, S. Bagchi, *et al.* TCP/IP timing channels: Theory to implementation, *Proc. on Conference of INFOCOM 2009, IEEE.,*pp. 2204-2212, 2009.

[19] R. J. Walls, K. Kothari, W. Wright, Liquid: A detection-resistant covert timing channel based on IPD shaping, *J. Computer Networks,* vol. 55, pp. 1217-1228, 2011.

[20] K. Kothari, W. Wright, Mimic: An active covert channel that evades regularity-based detection, *J. Computer Networks*, vol. 57, pp. 647-657, 2013.

[21] P. Wayner, Mimic functions,*J. Cryptologia,* vol. 16, pp. 193-214, 1992.

[22] E. Al-Shaer, W. Marrero W, A. El-Atawy, *et al.* Network configuration in a box: Towards end-to-end verification of network reachability and security, *Proc. on ICNP 2009. 17th IEEE International Conference on Network Protocols. IEEE,* pp. 123-132, 2009.

[23] Y. Liu, D. Ghosal, F. Armknecht, *et al.* Robust and undetectable steganographic timing channels for iid traffic, *Proc. on Conference of Information Hiding*, Springer Berlin Heidelberg, pp. 193-207, 2010.

[24] D. Kirovski, H. S. Malvar, Spread-spectrum watermarking of audio signals, *J. IEEE Trans. on Signal Processing*, vol. 51, pp. 1020-1033, 2003.