University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Faculty Publications from the Department of Electrical and Computer Engineering

Electrical & Computer Engineering, Department of

2014

Achieving Robustness and Capacity Gains in Covert Timing Channels

Fahimeh Rezaei
University of Nebraska-Lincoln, fahimeh.rezaei@huskers.unl.edu

Michael Hempel
University of Nebraska-Lincoln, mhempel2@unl.edu

Pradhumna Lal Shrestha
University of Nebraska-Lincoln, pshrestha@huskers.unl.edu

Hamid Sharif
University of Nebraska-Lincoln, hsharif@unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/electricalengineeringfacpub

Part of the Computer Engineering Commons, and the Electrical and Computer Engineering Commons

Rezaei, Fahimeh; Hempel, Michael; Shrestha, Pradhumna Lal; and Sharif, Hamid, "Achieving Robustness and Capacity Gains in Covert Timing Channels" (2014). *Faculty Publications from the Department of Electrical and Computer Engineering*. 307.
http://digitalcommons.unl.edu/electricalengineeringfacpub/307

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications from the Department of Electrical and Computer Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Achieving Robustness and Capacity Gains in Covert Timing Channels

Fahimeh Rezaei, Michael Hempel, Pradhumna Lal Shrestha, Hamid Sharif
Department of Computer and Electronics Engineering
University of Nebraska-Lincoln
fahimeh.rezaei@huskers.unl.edu, mhempel2@unl.edu, pshrestha@huskers.unl.edu, hsharif@unl.edu

Abstract –**In this paper, we introduce a covert timing channel (CTC) algorithm and compare it to one of the most prevailing CTC algorithms, originally proposed by Cabuk et al. CTC is a form of covert channels – methods that exploit network activities to transmit secret data over packet-based networks – by modifying packet timing. This algorithm is a seminal work, one of the most widely cited CTCs, and the foundation for many CTC research activities. In order to overcome some of the disadvantages of this algorithm we introduce a covert timing channel technique that leverages timeout thresholds. The proposed algorithm is compared to the original algorithm in terms of channel capacity, impact on overt traffic, bit error rates, and latency. Based on our simulation results the proposed algorithm outperforms the work from Cabuk et al., especially in terms of its higher covert data transmission rate with lower latency and fewer bit errors. In our work we also address the desynchronization problem found in Cabuk et al.'s algorithm in our simulation results and show that even in the case of the synchronization-corrected Cabuk et al. algorithm our proposed method provides better results in terms of capacity and latency.**

Keywords-Covert Communication; Covert Timing Channel; Hidden Information; Capacity; Latency; Network Security

I. INTRODUCTION

Taking advantage of a communication medium, its characteristics and its resources to send secret information to specific recipients is known as covert communication. While it has its origins in ancient times, it found a dramatic resurgence with the proliferation of the Internet. Several diverse methods of using exploiting this communication medium for hidden information exchange purposes are introduced and investigated in [1-7]. Based on the specific technique of how this covert communication is accomplished, it can be classified into three major categories. One of the simplest and most straightforward methods of covert communication in networks is to utilize specific header fields of the overt network packets that are not used for regular communication and substitute their information with covert data. This technique is known as *Covert Storage Channel (CSC)*, and has been studied in articles such as [1] and [2]. The authors in [1] establish a covert channel based on the Session Initiation Protocol (SIP) signaling during the signaling phase of Voice-over-IP (VoIP). The authors discuss different parts of the SIP signaling message that can be used to embed covert data and determine the amount of data that can be embedded within the generated covert channel. In [2] the authors discuss the possibility of embedding hidden information in TCP/IP packets and verify the vulnerability of these types of covert storage channels against wardens.

Another existing method to transfer secret data is called *Covert Timing Channel (CTC)*, which manipulates the timing of overt network packets to achieve a desired pattern [3-5]. This pattern is what is used to convey the covert information. In [3], the authors introduce a CTC that transfers a covert "1" by sending a packet during a given time interval and a covert "0" by not sending a network packet. Another method, introduced in [4], is known as the Jitter Bug covert channel, where during a network terminal session the transmitted keystroke timing is manipulated by applying delays to the corresponding packets. In another approach [5], the authors designed a covert timing channel that encodes N covert symbols to the inter transmission time of L TCP/IP overt packets.

In addition to these two major categories, papers such as [6] and [7] introduce covert communication methods that are a combination of storage and timing techniques. There, a given network packet is filled with covert data and sent intentionally late by the covert sender. An overt receiver that is not aware of the covert channel algorithm will ignore this packet because it appears to arrive late. However, the covert receiver gets the late packet and extracts the covert data.

Among all of these CTC algorithms, the one proposed by Cabuk et al. in [3] is the one that has been used in many research efforts and investigations of covert communications. This algorithm is well accepted by the research community and is used as the basis for numerous other concepts in covert communication such as detection and modulation. For example in [8], the authors introduce Cabuk et al.'s CTC algorithm as a covert communication technique that can be employed in Building Automation Systems and provide a solution to prevent covert communication in BAS. The authors in [9] developed an attack that uses a covert channel mechanism such as Cabuk et al.'s CTC, to inject a watermark signature into the network flow via virtual machines in cloud environments. A network forensics collection system called Horizon Extender is presented in [10] to avoid information leakage in HTTP traffic such as the one introduced by Cabuk et al. In [11] Cabuk et al.'s CTC is one of the case studies for the general mathematical model that is proposed to predict the capacity of CTCs in networks.

Although Cabuk et al.'s algorithm is one of the most influential methods of covert communication, it has some

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

flaws such as low covert data rate, high latency, and high bit error rate due to desynchronization events.

In this paper, we propose a CTC algorithm that solves the issues of low capacity and high latency for Cabuk et al. and verifies it by simulation. We will show that we could successfully reduce the bit error rate during desynchronization events. However, to prevent these desynchronization errors we also suggest a solution that eliminates this problem of Cabuk et al.'s CTC algorithm and we will show that the Bit Error Rate (BER) is significantly decreased.

This paper is organized as follows: In section II these different CTC methods are described in detail. This is followed by section III in which we provide a comparison and analysis of these approaches. In section IV, we explain our simulation parameters and implementation approach. Simulation results are presented and discussed in section V followed by the summary and conclusions in section VI.

II. Background on utilizied CTC Algorithms

One of the most influential methods of CTC was presented by Cabuk et al. in [3]. In this method, covert bits are sent over the communication channel by considering a constant time interval, which is known by both the covert sender and the covert receiver. On the network, an overt transmitter communicates with an overt receiver. This packet exchange is intercepted and manipulated by the covert transmitter on the network path the packets take. Farther along this path the covert receiver is located. It observes the packets before they reach the overt receiver. The covert transmitter manipulates the timing of packets intercepted from the overt transmitter in order to conform to the specified time interval. If during that time it allows a packet to be delivered towards the receiver, this encodes a covert "1", whereas if no packet is allowed it is encoding a covert "0". Therefore, from the covert receiver's point of view, the decoding is based on whether a packet is observed during the known time interval. This method is illustrated in Figure 1.

Aiming to improve upon the original Cabuk et al., we introduce our proposed CTC method in which a covert "1" is sent after a specific time delay that is known by the covert sender and receiver. However, for sending a covert "0", network packets are transferred normally. On the covert receiver side, the given interval is monitored to check if any network packet arrives after this interval or not. If any packet was received within this interval, it will be interpreted as a covert "0", after which the covert receiver resets the observation interval. If a packet was received after the given timeout it represents a covert "1".

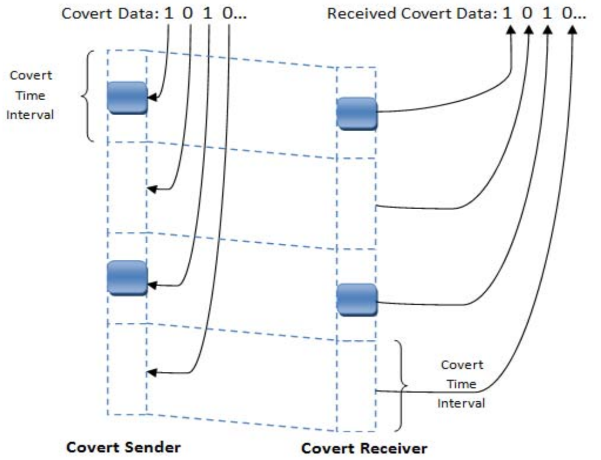

Figure 1. Cabuk et al.'s CTC algorithm (Modified picture from [3])

Our proposed CTC technique shares the idea of utilizing timeouts with LACK, proposed in [6] and [7]. In LACK the contents of the late packet are modified and used as covert information carrier. In our proposed method, however, which we call Delayed Packet One Indicator (DPOI) all the covert information is derived from the timing exclusively, which significantly reduces detectability of our algorithm. Our proposed CTC algorithm is shown in Figure 2.

The operation of both Cabuk et al.'s CTC and the proposed DPOI algorithm is based on the assumption that the time interval between network packets does not exceed the covert time interval.

With both of these algorithms introduced we can now compare them in terms of aspects such as bit error rate, capacity, latency and more.

III. Analysis Methodology

In this section we compare Cabuk et al.'s CTC algorithm against our proposed DPOI CTC algorithm in terms of their impact on the overt traffic, capacity for covert data, Bit Error Rate (BER) of covert data and implementation constraints for covert sender and receiver.

A. Impact on the overt traffic

As shown in Figures 1 and 2, the principles by which the two CTC algorithms encode covert information has observably different impact on overt traffic. In Cabuk et al.'s algorithm, for sending a covert bit zero no packet is allowed during a given time interval. Therefore, there is a silent period between overt network packets that are used to transmit the covert data. This gap results in a decrease in the bandwidth of the overt network and a corresponding increase in detectability of this covert activity on the network. However, for the DPOI algorithm, covert "0" bits are sent normally and do not have any impact on overt network traffic. For sending covert "1" bits, both scenarios introduce the same latency to the overt network traffic. Thus, DPOI has less impact on overt traffic and lower detectability.

According to this discussion, we define the following parameters to explain the relationship between the overt network and covert data in both algorithms:

- T_r: A given duration of time, which is used as a reference to determine the number of overt network packets considered.
- P_r: Number of overt network packets within T_r.
- N_{ct}: Number of covert bits (total number of zeros and ones).
- O_{ct}: Number of covert "1" bits within a covert bit string.

- T_{ct}: Covert time interval that is known by the covert receiver and the covert sender.
- T_p: Time duration to transfer one overt packet from a sender to a receiver. T_p includes channel delay, operational delay and other existing delays that affect the overt packet arrival time.

According to these definitions, in Cabuk et al.'s algorithm the bandwidth of the overt traffic is bounded to the number of covert "1" bits. This arises from the fact that the overt network packets are only transmitted if the current covert bit is one, otherwise no packet is sent over the network. Therefore the number of overt network packets within a given time T_r equals to:

$$P_r = \frac{O_{ct}\times T_r}{O_{ct}\times(T_{ct}+T_{\mathrm{p}})} \quad (1)$$

In the DPOI CTC, transferring covert bits zero does not require a silent time interval, which considerably improves the overt traffic bandwidth. However, the delay after sending each covert "1" bit results in the reduction of overt traffic bandwidth within a given time T_r so that:

$$P_r = \frac{N_{ct}\times T_r}{(T_p\times N_{ct})+(O_{ct}\times T_{ct})} \quad (2)$$

According to this discussion, sending a given number of covert "1" bits contributes to the same P_r in both CTC algorithms. In addition, the maximum P_r for Cabuk et al.'s algorithm is reached when a covert bit string of all ones is sent. As the number of covert "1" bits increases in the covert data, the bandwidth of the overt network is improved proportionally.

In our DPOI algorithm, the maximum overt traffic bandwidth is achieved by sending a string of zeros covertly. As the number of ones increases in the covert data, P_r decreases based on the covert time interval T_{ct}.

In general, the proposed DPOI CTC algorithm is more suitable to be employed in networks where the latency of overt traffic and detectability are important factors. Since Cabuk et al.'s CTC algorithm diminishes the bandwidth of the overt network traffic, this CTC algorithm can best be employed in case of low network traffic channels.

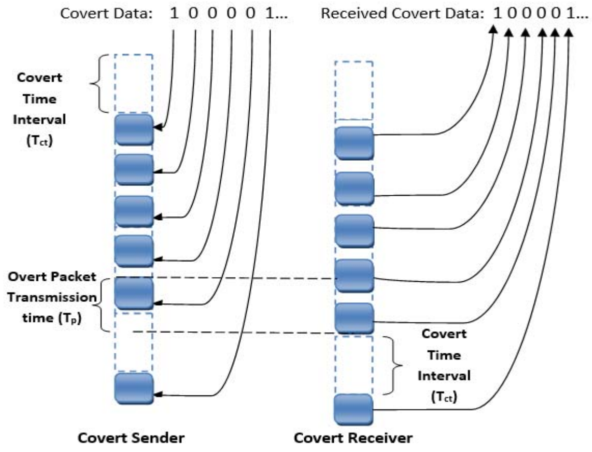
Covert Data: 1 0 0 0 0 1...
Received Covert Data: 1 0 0 0 0 1...
Covert Time Interval (T_{ct})
Overt Packet Transmission time (T_P)
Covert Time Interval (T_{ct})
Covert Sender
Covert Receiver

Figure 2. Proposed CTC algorithm: DPOI

B. *Covert Channel Capacity*

The capacity of a covert channel, C_{ct}, is defined as the number of covert bits that can be transferred within a specific time T_r from the covert sender to the covert receiver. Evidently, this capacity depends on the channel latency, noise and other network factors. However, for the maximum covert channel capacity analysis we can consider an ideal channel that does not have the mentioned limitations. Therefore, for the Cabuk algorithm, the capacity of the covert channel depends on the covert time interval. This arises from the fact that each covert bit, regardless of being one or zero, is transferred within T_{ct}. Consequently, the capacity of the covert channel for this algorithm, also reported in [11], is defined as:

$$C_{ct} = \frac{1}{T_{ct}} \quad (3)$$

For the proposed DPOI CTC algorithm the covert channel capacity depends on the covert bits being "1" or "0". In case of a covert "0" bit, the capacity of the covert channel equals the number of overt packets that are transferred. However, in case of covert "1" bits the covert channel capacity is calculated using equation (3). Hence, the covert channel capacity depends on the probability of a covert bit being one or zero. If we assume that the probability of a covert bit being one is P_1, then the capacity of a covert timing channel for the second scenario is:

$$C_{ct} = (1-P_1)\times\frac{1}{T_p} + P_1\times\frac{1}{T_{ct}} \quad (4)$$

From this discussion, we can conclude that the capacity of the DPOI CTC algorithm is larger than the capacity of Cabuk et al. Furthermore, DPOI's capacity is bounded to the number of ones in the covert data. As the number of covert "1" bits increases the capacity is reduced and approaches equation (3). Generally we can observe that for the CTC algorithms discussed in this paper the covert time interval plays a significant role in the capacity of the covert data. If the covert interval is not long enough, the network delay jitter will result in decoding errors in the covert bits, which we will discuss in the following section in detail.

C. *Covert Channel Bit Error Rate*

One key target of CTC algorithms, or in fact any networking approach, is to be able to transmit data with as few errors as possible. The Bit Error Rate (BER) for both CTC algorithms depends on various aspects of the network and implementation of the covert channel algorithms. The main reasons for getting covert bit errors for the covert timing channel algorithms are:

1) *Network Delay*

Although a short covert time interval is desirable for higher capacity, if it is too short network delay can affect the decoding result of the covert receiver. Let's assume that the covert time interval is 30 ms and the sender transmits a network packet at the middle of this time interval (covert "1"). The covert receiver will check the arrival of the network packet within this 30 ms interval. If the network delay is such that the receiver observes the packet after its covert time interval ends, then it will wrongly decode a covert “0” as the next bit instead of the cover “1” that was intended. Therefore, the covert time interval should be chosen in such a way as to diminish and ideally eliminate the network delay effects on the covert BER.

For Cabuk et al.’s CTC algorithm, network delay contributes to both zero-to-one and one-to-zero errors. However, in the proposed DPOI CTC algorithm, network delay will only cause zero-to-one errors.

2) Network Jitter

The error introduced by network delay jitter is a random phenomenon that cannot be predicted by the receiver. If the jitter causes an arbitrary delay that is not expected in the normal traffic network, then the covert receiver does not observe the network packet within the expected covert time interval and produces covert BER. For example, if the common delay in the network follows normal distribution and the covert time interval is adjusted between the covert sender and the covert receiver, then arbitrary jitter can change the timing of overt traffic, which results in covert data BER. Similar to network delay, jitter can result in zero-to-one and one-to-zero errors in Cabuk et al.’s algorithm but only zero-to-one errors in our DPOI CTC algorithm.

3) Desynchronization between the Covert Sender and the Covert Receiver

Another aspect factor contributing to Covert BER is the desynchronization of the covert sender and receiver. If the sender does not compensate for network delay, then the covert receiver will receive the network packets outside of the expected covert arrival time interval bounds. Consequently, the covert receiver will produce decoding errors. In essence, the covert communication partners become desynchronized if this delay remains uncompensated for.

In Cabuk et al.’s algorithm, sporadic desynchronization causes an extra covert time interval to be inserted into the received covert data. Hence, all following bits are shifted and thus potentially differ from the expected bits until synchronization is restored. In our proposed algorithm, if the covert sender and receiver are not synchronized, then a one bit zero-to-one or one-to-zero error type can occur. This arises from the fact that an empty covert time interval from a covert receiver’s perspective is not an indicator for covert data. Instead, a late packet indicates a covert “1”. The extra covert time interval, which results from the desynchronization problem, can be ignored in the covert receiver if the desynchronization issue occurs while sending covert bits “1”. If the desynchronization happens when the cover bits “0” are sent, one of the normal packets that indicates covert “0” will arrive late and the covert receiver will interpret that as a covert “1”. Therefore, a zero-to-one bit error appears.

4) Network Packet Loss

Although the packet loss in modern Internet communication links is very low and is assumed to be zero in [3] and other covert channel publications [4-7], packet loss can nevertheless cause errors in covert data. For the two CTC algorithms discussed in this paper packet loss can result in various error types. In Cabuk et al.’s CTC algorithm, packet loss will contribute to one-to-zero errors. However, in the proposed algorithm, it can cause both zero-to-one and one-to-zero errors depending on whether the lost packet was intended to be sent as normal or delayed traffic.

IV. IMPLEMENTATION AND SIMULATION PARAMETERS

We implemented both covert timing channel algorithms to verify the previous discussions and observe their differences. These algorithms are implemented using C++,

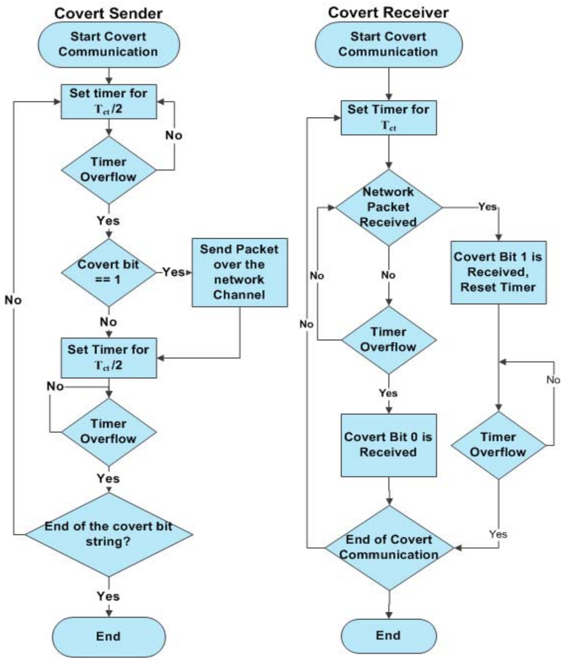

Figure 3. CTC Flowchart of Cabuk et al.’s Algorithm.

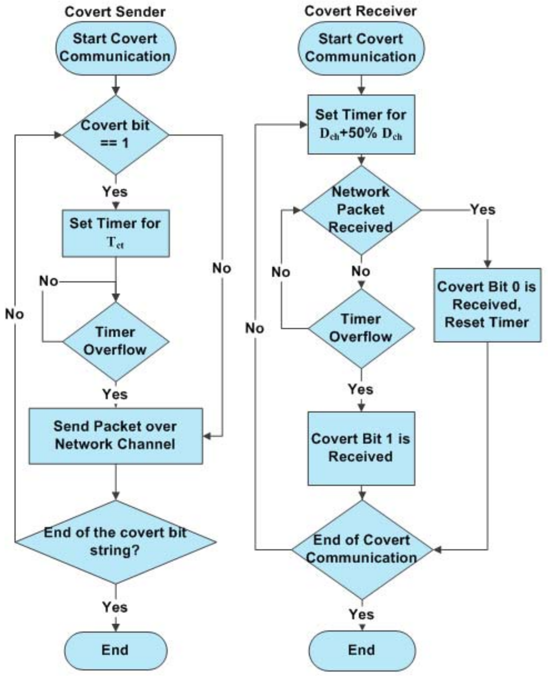
Covert Sender
Start Covert Communication
Covert bit == 1
Yes
Set Timer for T_{ct}
No
Timer Overflow
Yes
No
Send Packet over Network Channel
No
End of the covert bit string?
Yes
End
Covert Receiver
Start Covert Communication
Set Timer for D_{ch}+50% D_{ch}
Network Packet Received
Yes
Covert Bit 0 is Received, Reset Timer
No
No
Timer Overflow
Yes
No
Covert Bit 1 is Received
End of Covert Communication
Yes
End

Figure 4. CTC Flowchart of the proposed DPOI algorithm

using a multithreaded simulation framework where the covert sender, covert receiver and network channel run on three separate threads. One of the significant advantages of this approach is the simultaneous interaction between covert sender, covert receiver and network channel. Figures 3 and 4 show the overall flowcharts of these two algorithms. The most important factor that we want to observe from our simulation results for these two CTC algorithms is the capacity, latency, and their vulnerability to channel delay. The simulation parameters and assumptions are collected in Table 1. In order to represent a wide variety of environments we have defined different scenarios of covert time interval of the sender versus the channel delay. The channel latency is considered as a uniform random distribution in the range of [45ms, 55ms].

Table 1. Simulation Parameters and Assumptions

Parameter	Setting and value
Channel Conditions	Delay channel with uniform distribution over [45ms,50ms]
Covert Time Interval: T_{ct} (ms)	250, 200, 175, 150, 100
Size of Bit Strings	80 bits from 10 ASCII characters
Clock Cycles and Bit Error Results	Averaged over 3 bit stings
Simulation iterations	All tests were performed 10 times over three 80-bit strings and the mean value is presented.

In the next two sections we demonstrate the impact of these two covert timing channels on the overt traffic and covert data.

V. Comparisons and Evaluations

A. Capacity and Latency

In Figure 5, the percentage of covert transfer delay reduction we achieved with our DPOI algorithm compared to Cabuk et al.'s algorithm is shown. This result is obtained from transmitting the test's 80 covert bits to the covert receiver based on different covert time intervals T_{ct}. As we can see in this figure, our proposed DPOI algorithm improves the covert message transfer time by more than 40% on average for all the time intervals. We can observe that our proposed DPOI algorithm significantly outperforms Cabuk et al.'s and can transfer bit strings almost twice as fast. As discussed before, the duration that it takes to transfer 80 covert bits for the proposed DPOI algorithms depends on the number of covert “1” bits.

Figure 6 shows the average capacity for all the values of covert time intervals and is based on the mentioned assumptions for the channel delay. The simulations are conducted 10 times for 3 strings of 80 covert bits and the average capacity is calculated. From this figure we can observe that the covert data capacity is considerably improved in the proposed DPOI algorithm and our proposed algorithm can transfer approximately twice the amount of covert data compared to Cabuk et al.'s algorithm. For example, in the case of 175ms covert time interval our proposed DPOI algorithm transfers 9.99 bits/sec on average. However, we can achieve only 5.69 bits/sec using Cabuk et al.'s algorithm for the same simulation parameters. We should mention the fact that these results may change slightly based on operating system, operational speed and other factors.

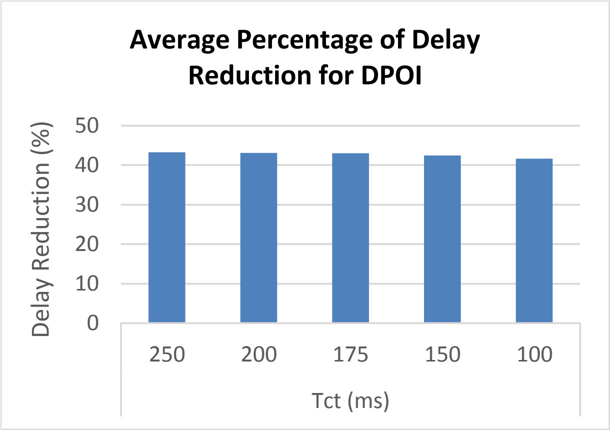

Figure 5. Average Percentage of Covert Transfer Delay Reduction of DPOI compared to Cabuk et al.'s algorithm

B. Reliability

In this section we discuss the results for BER obtained from our simulation. As discussed before, one of the main causes of errors in Cabuk et al.'s CTC algorithm is the desynchronization problem. If the observation covert time intervals at the transmitter and receiver are not aligned, packets may not arrive until after the receiver's current covert time interval ends, resulting in a covert “0” being decoded even though originally transmitted was a covert “1”. Subsequent bits are then affected as well. In our simulations we assumed correlated clocks between sender and receiver and thus could achieve interval synchronization by using a fixed delay to the start of the receiver's covert timing interval. We observed that by having a delay of $0.3 * T_{ct}$ for sending the first covert bit, we can resolve the desynchronization problem. However, this approach does not apply to real-world implementations and different synchronization approaches need to be developed. The results shown in Figure 7 include scenarios for the original Cabuk CTC algorithm (no

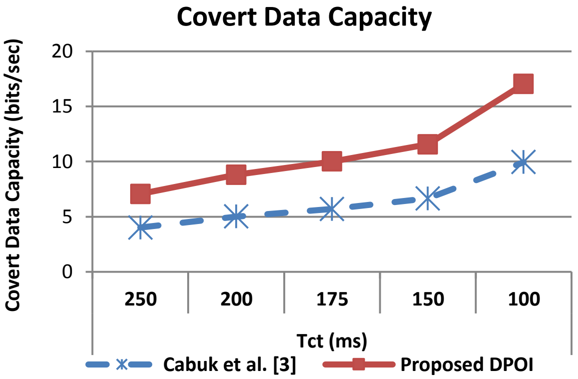

Figure 6. Covert Data Capacity of CTC Algorithms: Cabuk [3] vs. DPOI

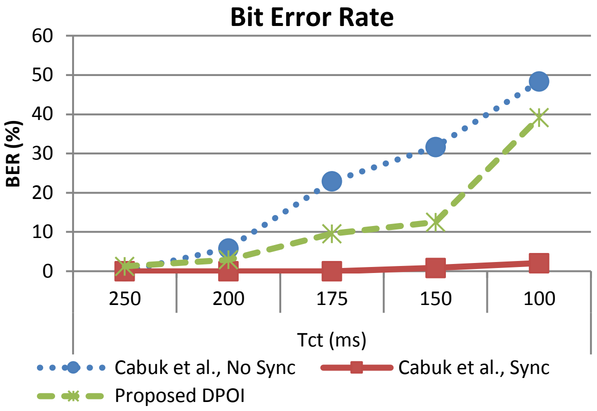

Figure 7. Bit Error Rate for the Covert Timing Algorithms

sync), a modification that resolves the desynchronization problem (sync), as well as our proposed DPOI algorithm.

In [3] the authors have tested their CTC algorithm using a test bed implementation. However, the details and different parameters of their test bed are not stated in their paper. The desynchronization problem is addressed by string to string correction after receiving the covert bits and the character accuracy is calculated for different covert time intervals (20, 30, 40, 50, 60, and 80 ms) in a normal traffic channel with an average round trip time of 31 ms.

From Figure 7 we can also observe that in cases where the covert sender interval time is more than 4 times the delay of the channel ($T_{ct}: 200\ or\ 250\ vs. D_{ch}: [45 - 55]$), only few bit errors are observed and the covert data accuracy remains above 90%. However, as T_{ct} decreases the number of bit errors increases, for the reasons explained in section III part C, especially the desynchronization problem. When we corrected the Cabuk's desynchronization issue in our simulations, then more accurate results can be obtained for Cabuk's algorithm. While our proposed DPOI algorithm shows a higher BER than the Cabuk's corrected version, it also achieved almost twice the capacity of Cabuk et al.'s algorithm and unlike the modified version of Cabuk can be used in real environments as well.

Our results show that the proposed DPOI CTC algorithm outperforms Cabuk et al., especially when considering capacity and covert bit accuracy for scenarios of $T_{ct} \geq 200$. Our algorithm transfers strings of 80 covert bits within approximately 9094.33 ms for $T_{ct} = 200$, while providing more than 97% covert data accuracy. Although Cabuk et al.'s CTC shows promising results when the synchronization problem is addressed, the amount of time it takes to transmit covert data is significantly longer. Cabuk's algorithm contributes to transmission delay for all the transferred packets over the network link, which increases detectability and is not desirable in network applications. Furthermore, while we addressed the desynchronization problem of Cabuk et al. in our simulation, this approach is not feasible for actual implementations and a more suitable approach is needed. Hence, Cabuk's algorithm is more vulnerable to desynchronization between sender and receiver. By contrast, our proposed DPOI algorithm does not share this limitation.

VI. Summary and Conclusions

In this paper we introduced a new CTC algorithm called DPOI that transmits covert "0" bits by transmitting overt network packets without delay while covert "1" bits are delivered by delaying the overt packet for a specific time. We have shown that the proposed algorithm performs better than Cabuk et al. [3]. Cabuk's work is among the most cited CTC algorithms used in the literature and in ongoing research efforts. We have analyzed, compared and evaluated the two CTC algorithms to extract insights on covert data rate, latency, types and causes of bit error, and the Bit Error Rate itself. Based on the simulation results and our analysis for these two algorithms, our proposed algorithm can transmit more covert data within a specific amount of time compared to Cabuk et al. Also, one of the significant problems with Cabuk's algorithm is the desynchronization of the covert sender and receiver that causes erroneous bit insertions in the decoded data and subsequent bit error sequences. Even though we provided a modification to Cabuk et al. that addresses this desynchronization problem, our proposed DPOI CTC algorithm performs better in terms of covert data rate and latency.

VII. Acknowledgement

This study is being conducted at the University of Nebraska-Lincoln by the research faculty and students at the Advanced Telecommunications Engineering Laboratory (www.TEL.unl.edu), and is in part supported by TEL's seed funding for security engineering research projects.

References

[1] W. Mazurczyk, K. Szvzypiorski, "Covert channels in SIP for VoIP Signalling", Journal of Communications in Computer and Information Science, 2008

[2] S. J. Murduch, S. Lewis, "Embedding Covert Channels into TCP/IP ", ACM International Conference on Information Hiding, 2005

[3] S. Cabuk, C. E. Brodley, C. Shields, "IP Covert Timing Channels, Design and Detection", ACM Conference on Computer and Communications Security, 2004

[4] G. Shah, A. Molina, M. Blaze, "Keyboar and Covert Channels" USENIX Security Symposium, 2006

[5] S. H. Sellke, C. C. Wang, S. Bagchi, "TCP/IP Timing Channels: Theory to Implementation", IEEE International conference on Computer Communications, 2009

[6] W. Mazurczyk, J. Lubacz, "LACK- a VoIP Steganographic Method", Journal of Telecommunication Systems, 2008

[7] W. Mazurczyk, "Lost Audio Packets Steganography: The First Practical Evaluation", Journal of Security and Communication Networks, 2012

[8] S. Wendzel, B. Kahler, T. Rist, "Covert Channels and their Prevention in Building Automation Protocols – A Property Exemplified Using BACnet", IEEE International Conference on Green Computing and Communications, 2012

[9] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, K. Butler, "Detecting Co-Residency with Active Traffic Analysis Techniques", ACM Workshop pn Cloud Computing Security, 2012

[10] D. Gugelmann, D. Schatzmann, V. Lenders, "Horizon Extender: Long-term Preservation of Data Leakage Evidence in Web Traffic", ACM Symposium on Information, Computer and Communications Security, 2013

[11] P. Shrestha, M. Hempel, M. Alahmad, H. Sharif, "Modeling Packet Rate Covert Timing Channels", IEEE International Conference on Innovations in Information Technology, 2013.