CENG519 Term Project Phase 2 Report

Emre Burak Bakdemir
2580272

April 13, 2025

Abstract

This report presents the design, implementation, and performance evaluation of a covert channel that encodes hidden data in the ordering of TCP SYN options via permutations. The covert data is first XOR encrypted with a shared key, and a 16-bit header encoding the message length (in bytes) is prepended to the encrypted bitstream. Two configurations are evaluated: one using 4-bit encoding (using 4 TCP options) and one using 5-bit encoding (using 5 TCP options). The channel capacity is computed as the total number of transmitted bits (header plus message bits) divided by the elapsed transmission time. The experiments, conducted over a range of message lengths, demonstrate that the 5-bit mode consistently provides higher throughput than the 4-bit mode, and that the results are highly consistent as indicated by narrow 95% confidence intervals.

1 Introduction

Covert channels enable communication by embedding information into regular network traffic. In this project, a covert channel is implemented by modulating the order of TCP SYN options. The covert data (after XOR encryption with a shared key) is prepended with a 16-bit header, which indicates the length of the encrypted message in bytes. This header allows the receiver to accurately determine when the full message has been received and to discard any extra padding. The system is parameterized so that the number of bits encoded per packet can be selected (4 or 5 bits), which in turn determines the size of the permutation space (4-bit: $2^4 = 16$ values; 5-bit: $2^5 = 32$ values).

2 Implementation

The covert channel is implemented in Python using the Scapy library. The sender maps fixed-length bit chunks (derived from the XOR-encrypted message and a 16-bit header) to unique TCP option permutations which are:

- **For 4 bit chunks:** Maximum Segment Size (MSS), Window Scale (WScale), No Operation (NOP), Selective Acknowledgment Permitted (SAckOK)
- **For 5 bit chunks:** Maximum Segment Size (MSS), Window Scale (WScale), No Operation (NOP), Selective Acknowledgment Permitted (SAckOK) and Timestamp (Timestamp)

Sender then transmits TCP SYN packets accordingly. The receiver captures these packets, reconstructs the transmitted bitstream, reads the header to determine the expected number of message bits, and then decodes and decrypts the covert data.

3 Experimental Setup

Two Docker containers are used:

- **Sender (Container `sec`):** Runs the covert sender script.
- **Receiver (Container `insec`):** Runs the covert receiver script, which utilizes an early-exit mechanism based on the 16-bit header.

For each configuration (combination of bit encoding mode and message length), multiple trials are conducted and the capacity is computed as:

$$\text{Capacity (bits/s)} = \frac{\text{16 bits for header} + 8 \times \text{(message length in bytes)}}{\text{Elapsed Time (s)}}$$

The experimental results are aggregated over several trials to compute averages and 95% confidence intervals.

4 Results

The following tables summarize the covert channel performance metrics for both 4-bit and 5-bit modes.

4.1 Results for 4-bit Mode

Msg Length (bytes)	Avg. Elapsed (s)	Avg. Capacity (bits/s)	95% CI Low	95% CI High
4	1.3145	36.57	35.02	38.12
8	1.6442	48.85	45.56	52.13
12	2.0655	54.28	52.44	56.12
16	2.4770	58.16	56.73	59.60
20	2.8582	61.65	59.39	63.91
24	3.1922	65.22	63.02	67.43
28	3.4701	69.23	66.97	71.49
32	3.7342	72.94	70.14	75.74
64	6.8582	77.02	75.40	78.63
96	9.4306	83.23	80.29	86.16
128	12.5163	83.17	80.50	85.83

4.2 Results for 5-bit Mode

Msg Length (bytes)	Avg. Elapsed (s)	Avg. Capacity (bits/s)	95% CI Low	95% CI High
4	1.1939	40.22	39.29	41.15
8	1.4529	55.12	53.27	56.97
12	1.7902	62.61	60.73	64.50
16	2.0711	69.75	65.58	73.92
20	2.5278	69.97	64.68	75.27
24	2.8196	73.93	70.28	77.58
28	2.9707	80.87	78.10	83.64
32	3.2003	85.05	82.65	87.45
64	5.4063	97.74	94.75	100.73
96	8.0386	97.59	94.92	100.27
128	10.2613	101.46	97.92	105.00

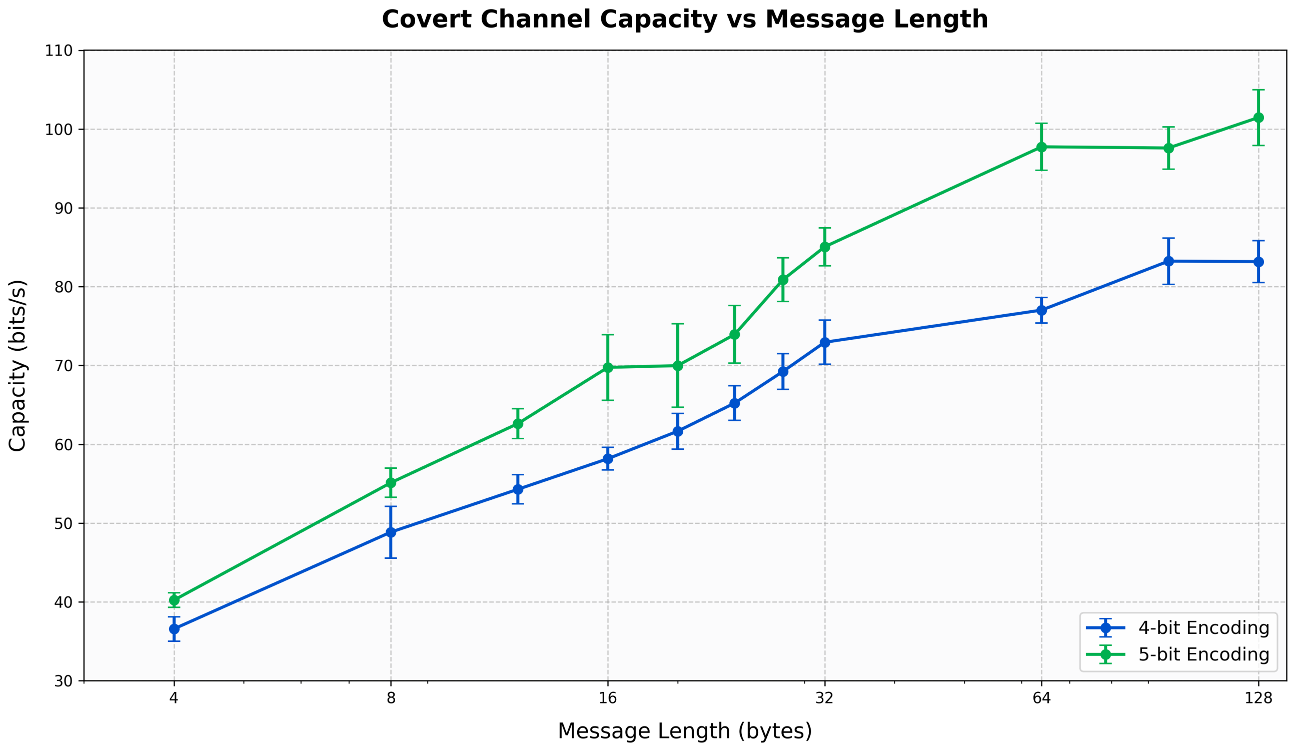

Figure 1: Covert Channel Capacity(bits/s) vs Message Length(bytes) for 4-bit and 5-bit Encodings

5 Discussion

The experimental campaign reveals several key observations:

- **Effect of Message Length:** In both modes, the average elapsed transmission time increases approximately linearly with the message length, which is expected since the number of packets increases. However, the increase in the covert channel capacity shows diminishing returns at higher message lengths, particularly in 4-bit mode—where capacity saturates around 83 bits/s for larger messages.

- **Comparison between 4-bit and 5-bit Modes:** The 5-bit mode consistently yields a higher capacity across all message lengths. For shorter messages (4–12 bytes), the capacity in 5-bit mode is noticeably higher (e.g., 40.22 vs. 36.57 bits/s for 4 bytes, and 55.12 vs. 48.85 bits/s for 8 bytes). For longer messages, the 5-bit mode still provides an advantage (e.g., 101.46 bits/s vs. 83.17 bits/s for a 128-byte message).

- **Consistency and Reliability:** The 95% confidence intervals for each configuration are narrow, indicating consistent performance across trials. This suggests that the covert channel is robust and its throughput is reliably reproducible under the tested conditions.

- **Overhead Impact:** The fixed 16-bit header has a larger impact on the overall capacity for shorter messages, but as message length increases the header overhead becomes negligible.

6 Conclusion

This report presented a covert channel that encodes data by permuting TCP SYN options, with the covert bits being encrypted via a simple XOR operation and accompanied by a fixed-length header for precise decoding. The experimentation campaign, which evaluated both 4-bit and 5-bit encoding modes over a range of message lengths, demonstrated that the 5-bit mode consistently achieves higher throughput. Moreover, the channel performance is stable as evidenced by the narrow 95% confidence intervals across multiple trials.