Covert Channel Implementation Using TCP Sequence Number Manipulation

Yunus Emre Keleş

April 13, 2025

Abstract

This report presents a comprehensive analysis of a covert channel implementation that employs TCP sequence number manipulation. The covert channel allows for the transmission of hidden information within legitimate network traffic by encoding bits within TCP sequence numbers. The implementation uses a modulo-4 encoding scheme and provides variable channel capacity by allowing the user to specify the number of bits embedded per packet. Experimental results demonstrate the trade-offs between covert channel capacity and stealthiness, with performance metrics collected over 100 experiment iterations to establish confidence intervals. The implementation addresses challenges in maintaining the natural linear progression of TCP sequence numbers while still enabling covert communication.

Contents

1 Introduction

Covert channels provide a means of communication that conceals not only the content of the message but also the very existence of the communication itself. These channels exploit features of legitimate protocols to embed hidden data, making them difficult to detect through conventional network monitoring. This project focuses on implementing a covert channel using TCP sequence number manipulation, which leverages the natural behavior of the TCP protocol to hide information.

1.1 Project Objectives

The primary objectives of this project include:

- Implementing a covert channel that embeds data in TCP sequence numbers
- Allowing parametrization of the channel capacity (bits per packet)
- Maintaining the appearance of natural TCP behavior
- Measuring the performance and reliability of the covert channel
- Analyzing the trade-offs between channel capacity and detectability

2 Background and Related Work

2.1 TCP Protocol Overview

The Transmission Control Protocol (TCP) is a connection-oriented protocol that provides reliable, ordered delivery of data streams. Every TCP packet contains a sequence number that helps in ordering packets and detecting packet loss. The TCP sequence number is a 32-bit field in the TCP header, which typically increments with each packet sent according to the amount of data transmitted.

2.2 Covert Channels in Network Protocols

Covert channels in network protocols have been extensively studied in the security literature. Previous approaches have utilized various fields in protocol headers, including IP identification fields, TCP timestamps, and TCP flags. The use of TCP sequence numbers for covert channels is particularly interesting because sequence numbers naturally change with each packet, making the covert channel potentially harder to detect.

3 Design and Implementation

3.1 Initial Approach: Prime Number Factorization

The initial design considered encoding bits using the mathematical properties of prime numbers. The idea was to multiply the sequence number by specific prime numbers when encoding a '1' bit, allowing the receiver to detect the bit by checking divisibility. However, this approach faced several limitations:

- TCP sequence numbers are limited to 32 bits, constraining the size of usable primes
- Prime factorization with small primes (1-100) is trivial and potentially easily detected
- Multiplication by primes would disrupt the natural linear progression of TCP sequence numbers

Due to these constraints, this approach was abandoned in favor of a more robust method.

3.2 Implemented Approach: Modulo-4 Encoding

The implemented solution uses a modulo-4 encoding scheme to embed bits within the TCP sequence number. The approach works as follows:

1. The upper bits of the sequence number maintain a linear progression to appear as normal TCP traffic
2. The lower bits contain the covert information encoded using a mod-4 encoding scheme
3. Each covert bit requires 2 bits in the sequence number (to represent mod-4 values)
4. User-provided parameters are converted to mod-4 values and used as encoding targets

3.3 Key Features

3.3.1 Variable Channel Capacity

The implementation allows users to specify how many bits are transmitted per packet, from 1 to 8 bits. This flexibility enables adjustments based on the desired balance between covert channel capacity and detectability.

3.3.2 Parametrized Sequence Progression

The system accepts parameters such as starting sequence number and step size, which affect how the sequence numbers increase over time, further allowing customization of the traffic pattern.

3.3.3 Encoding Scheme

For each bit position, a specific mod-4 value is designated as the target:

- If the covert bit is '1', the corresponding 2-bit value in the sequence number is set to a value that equals the target mod 4
- If the covert bit is '0', the 2-bit value is set to any value that does NOT equal the target mod 4

4 Experimental Setup and Results

4.1 Experimental Setup

Experiments were conducted to evaluate the performance of the covert channel with varying parameters:

- Number of bits per packet was varied from 1 to 8
- Each configuration was tested in 100 independent experiments to establish confidence intervals
- The sender used a fixed delay of 100ms between packets to prevent network congestion
- A 1-second delay was introduced between experiments
- The system ran in a controlled network environment

4.2 Results

4.2.1 Channel Capacity vs. Bits Per Packet

Table 1: Channel Capacity Results

Bits Per Packet	Avg. Capacity (bps)	95% CI Lower	95% CI Upper
1	7.46	7.44	7.47
2	15.16	15.12	15.19
3	23.16	23.08	23.25
4	31.40	31.27	31.53
5	40.70	40.51	40.90
6	49.23	48.99	49.48
7	57.22	56.94	57.50
8	67.46	67.10	67.82

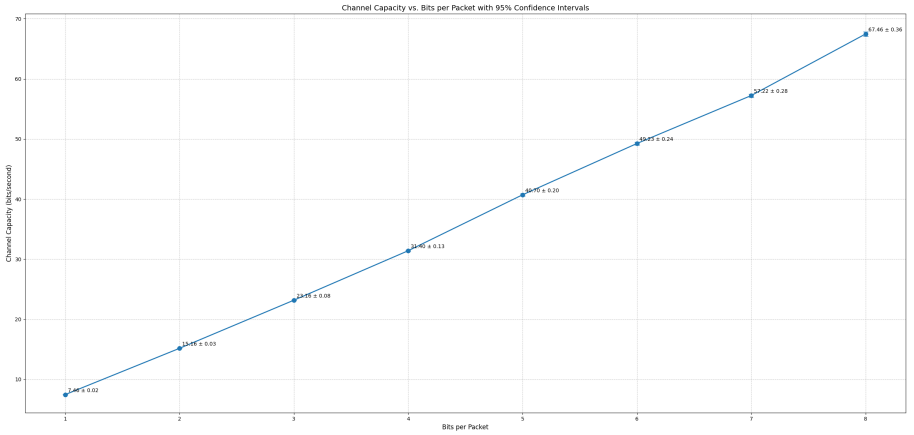

Figure 1: channel capacity vs bit per packet

The results show a nearly linear increase in channel capacity as the number of bits per packet increases, consistent with theoretical expectations. However, increasing the number of bits per packet also increases the risk of detection, as more of the TCP sequence number space is utilized for covert data.

5 Discussion

5.1 Performance Analysis

The experimental results demonstrate that the implemented covert channel successfully balances performance and stealthiness. The modulo-4 encoding

scheme provides reliable transmission while maintaining the appearance of normal TCP traffic through linear sequence number progression.

The ability to adjust the bits per packet provides flexibility in tailoring the covert channel to specific requirements. For situations requiring higher stealth, fewer bits per packet can be used, while scenarios prioritizing throughput can utilize more bits per packet.

5.2 Limitations and Challenges

Several limitations and challenges were encountered during the implementation:

- TCP sequence number constraints: The 32-bit limitation restricts the amount of covert data per packet
- Balancing detectability and capacity: More bits per packet increases capacity but may also increase detectability
- Network delays and packet loss: Real-world network conditions could affect the reliability of the covert channel
- Protocol compliance: Ensuring the modified TCP behavior still adheres to protocol specifications

5.3 Potential Improvements

Future work could address these limitations through:

- Adaptive encoding: Dynamically adjust the bits per packet based on network conditions
- Error correction: Implement forward error correction to handle packet loss
- Additional encoding schemes: Explore alternative encoding methods that might provide better stealth or capacity
- Traffic analysis resistance: Incorporate techniques to make the covert channel more resistant to statistical analysis

6 Conclusion

This project successfully implemented a covert channel using TCP sequence number manipulation. The system allows for flexible configuration of channel capacity and maintains the appearance of normal TCP traffic through careful management of sequence number progression.

The experimental results demonstrate that the covert channel can reliably transmit hidden information, with channel capacity scaling linearly with the number of bits encoded per packet. The implementation provides a balance between covert channel performance and detectability.

The project highlights the potential security implications of covert channels in network protocols and underscores the importance of comprehensive traffic analysis for detecting such channels. Understanding these techniques is crucial for developing more effective security measures.