electronics

MDPI

Article

An Enhanced Steganography-Based Botnet Communication Method in BitTorrent

Gyeonggeun Park [1], Youngho Cho [1,*] and Gang Qu [2]

[1] Department of Cyber Security and Computer Engineering, Graduate School of Defense Management, Korea National Defense University, Nonsan 33021, Republic of Korea; unicopgg@gmail.com
[2] Department of Electrical and Computer Engineering and Institute for System Research, University of Maryland, College Park, MD 20742, USA; gangqu@umd.edu
* Correspondence: younghocho@korea.kr

Abstract

In a botnet attack, significant damage can occur when an attacker gains control over a large number of compromised network devices. Botnets have evolved from traditional centralized architectures to decentralized Peer-to-Peer (P2P) and hybrid forms. Recently, a steganography-based botnet (Stego-botnet) has emerged, which conceals command and control (C&C) messages within cover media such as images or video files shared over social networking sites (SNS). This type of Stego-botnet can evade conventional detection systems, as identifying hidden messages embedded in media transmitted via SNS platforms is inherently challenging. However, the inherent file size limitations of SNS platforms restrict the achievable payload capacity of such Stego-botnets. Moreover, the centralized characteristics of conventional botnet architectures expose attackers to a higher risk of identification. To overcome these challenges, researchers have explored network steganography techniques leveraging P2P networks such as BitTorrent, Google Suggest, and Skype. Among these, a hidden communication method utilizing Bitfield messages in BitTorrent has been proposed, demonstrating improved concealment compared to prior studies. Nevertheless, existing approaches still fail to achieve sufficient payload capacity relative to traditional digital steganography techniques. In this study, we extend P2P-based network steganography methods—particularly within the BitTorrent protocol—to address these limitations. We propose a novel botnet C&C communication model that employs network steganography over BitTorrent and validate its feasibility through experimental implementation. Furthermore, our results show that the proposed Stego-botnet achieves a higher payload capacity and outperforms existing Stego-botnet models in terms of both efficiency and concealment performance.

Keywords: BitTorrent; network steganography; botnet; C&C communication

check for updates

Academic Editors: Jiacun Wang, Bin Hu and Xiwang Guo

Received: 9 September 2025
Revised: 10 October 2025
Accepted: 16 October 2025
Published: 17 October 2025

Citation: Park, G.; Cho, Y.; Qu, G. An Enhanced Steganography-Based Botnet Communication Method in BitTorrent. *Electronics* **2025**, *14*, 4081. https://doi.org/10.3390/electronics14204081

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Steganography is a common technique used in cyberattacks, and it is difficult to detect its presence. Unlike cryptography, which relies on encryption and can be easily identified, steganography conceals messages within ordinary media files, making detection highly challenging [1]. A botnet is a network of devices infected with malware and remotely controlled by a single entity, known as a *bot-herder* or *botmaster*, through command and control (C&C) messages [2]. Malicious actors can utilize botnets to take control of legitimate web services or to steal sensitive personal and financial information. Well-known examples of cyberattacks involving botnets include ransomware, distributed denial-of-service

(DDoS) attacks, spyware, and illegal information harvesting [3]. Botnet architectures have evolved from traditional centralized models that employ the HTTP protocol [4,5], to Peer-to-Peer (P2P) botnets that adopt decentralized communication structures [6]. In addition, hybrid botnets have been introduced, combining the advantages of both centralized and P2P models.

Recently, steganography-based botnets, referred to as *Stego-botnets* [3,7,8], have emerged. These botnets conceal C&C messages using steganographic techniques to evade conventional detection systems [9–12]. They typically appear to exchange normal files via social networking service (SNS) platforms [13], allowing them to bypass existing security mechanisms by embedding malicious instructions within common types of cover media such as images, audio, and video files. Among these, video files are particularly attractive due to their larger payload capacity compared to other media formats [14–18]. However, Stego-botnets cannot fully utilize their payload capacity due to the transmission limits imposed by SNS platforms. For instance, KakaoTalk allows a maximum file transfer size of 300 MB, Instagram permits up to 40 MB per photo, and Threads limits uploads to 6 MB per photo.

A P2P network has no limit on the size of transportable files and accounts for a considerable portion of total network traffic [19]. Suppose that a botnet employing steganography is constructed. In this case, an attacker can covertly communicate by using regular network traffic as a cover medium. Such steganographic botnets utilize protocol data units (PDUs) and the timing relationships between PDUs within normal traffic to embed secret information [20].

Network steganography techniques using P2P networks, including multimedia and VoIP services such as Skype and WeChat, have been extensively studied [21–23]. Wang [24] proposed a covert communication channel employing *Bitfield* messages in BitTorrent, demonstrating resistance to machine learning-based steganalysis. However, this approach failed to achieve a satisfactory payload compared to traditional digital steganography, securing only about 1 KB of hidden data per 255 KB of cover medium. Unlike SNS platforms, P2P networks impose no transmission capacity limit, allowing for the construction of digital Stego-botnets that can secure large payloads. Nevertheless, because an attacker shares only a single file, the communication availability is limited to one peer, increasing the risk of tracing the attack's origin or having the file hash registered as malicious [25].

To address these challenges, we propose a new communication model that applies a network steganography technique using *Piece* messages in the BitTorrent protocol. This model enables a botnet C&C communication mechanism that embeds large payloads within regular traffic while maintaining a high degree of stealth and resistance to detection.

Our contributions in this paper can be summarized as follows:

- We propose a novel communication method that enables continuous C&C message transmission over BitTorrent. By exploiting a vulnerability in the BitTorrent Piece message exchange, secret messages are covertly embedded and transferred during ordinary file exchanges between benign peers. As a result, the covert traffic generated by our model preserves normal BitTorrent behavior and statistical properties, making it difficult to distinguish from legitimate traffic.
- We implement the proposed method and evaluate its payload capacity and threat potential through experimental comparison with existing approaches. Our results show that the proposed method achieves at least a twofold increase in payload capacity compared to prior work, enabling more efficient delivery of C&C messages and substantially increasing the covert-channel threat surface. We propose a novel communication method that allows continuous C&C message transmission using BitTorrent.

The rest of our paper is organized as follows. In Section 2, we review the BitTorrent protocol and introduce existing studies related to Stego-botnets. In Section 3, we propose the Stego-botnet C&C communication model in BitTorrent. In Section 4, we implement our model in BitTorrent. In Section 5, we discuss and conduct a comparative analysis against previous studies. Finally, we conclude in Section 6.

2. Background and Related Works

2.1. Overview of BitTorrent

BitTorrent is a distributed network used by more than two billion people worldwide. Unlike the traditional client-server model, which shares an entire file through a centralized server, each peer in BitTorrent can hold part or all of a file and share fragments with other peers simultaneously. This decentralized structure enables the transfer of large files much faster than in a conventional client-server system [26].

2.2. BitTorrent Protocol Components and File Sharing Procedure

The components of BitTorrent are illustrated in Figure 1. BitTorrent divides the file to be shared into multiple pieces. *Seeders* are peers who have already downloaded the entire file; therefore, they possess all the pieces. *Leechers* are peers who lack some of the necessary pieces. Both seeders and leechers are collectively referred to as peers. In addition, *Swarms* exist for each shared file and contain the file's hash, along with a list of peers participating in its distribution. *Trackers* are servers that manage peer information related to file sharing and swarm coordination.

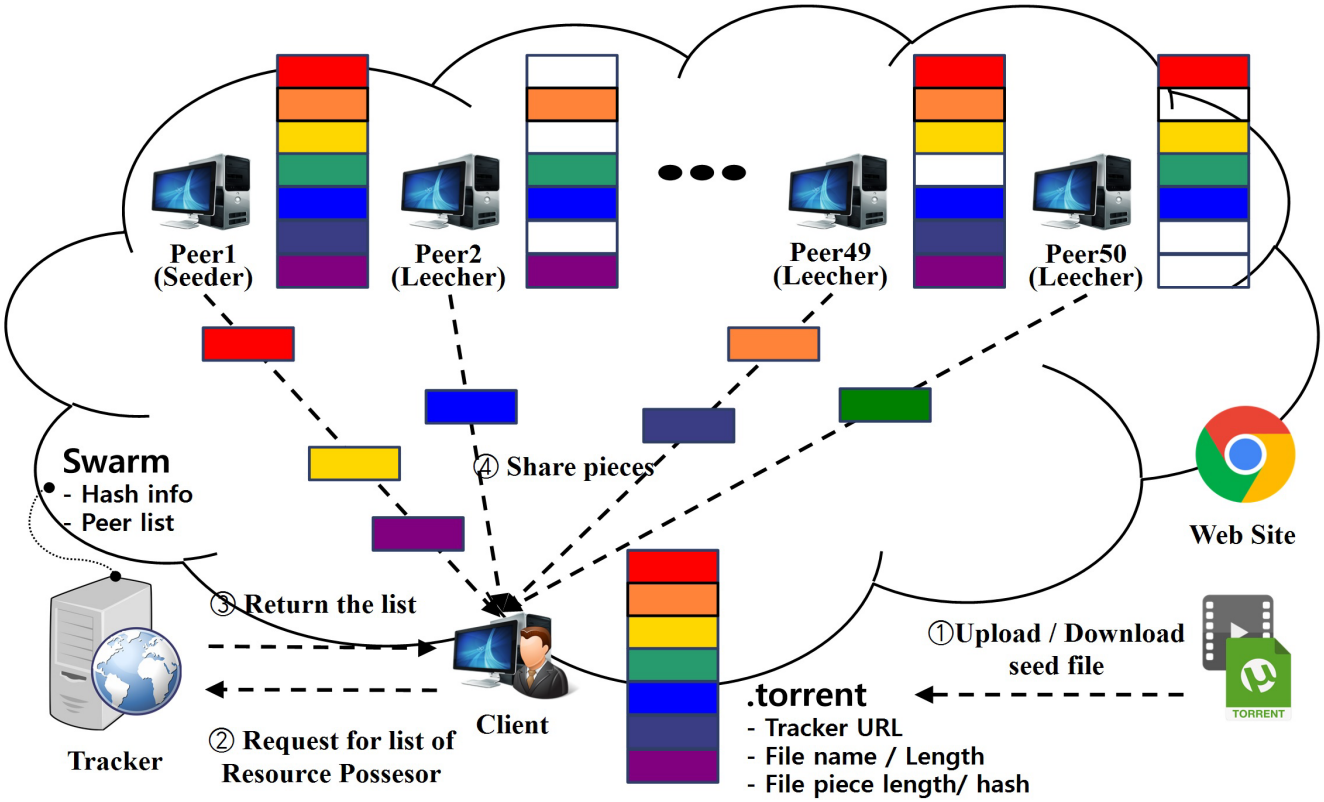


Figure 1. BitTorrent protocol.

BitTorrent requires a **.torrent file** (also called a seed file), which contains metadata about the file to be shared. Running the seed file in a BitTorrent client enables the file to be shared among peers. As shown in Figure 1.

(1) A peer creates a seed file and posts it on a website or an SNS platform;
(2) A client downloads the seed file and runs it, sending its own information and the hash value of the desired file to the tracker;
(3) The tracker returns a list of peers to the client;
(4) The client downloads file fragments from those peers [27].

2.3. BitTorrent Message

BitTorrent uses message packets to share file pieces. As shown in Figure 2,

(1) A peer uploads a seed file to be shared, and clients download the seed file via a website or an SNS platform;
(2) The client runs the seed file and requests a peer list from the tracker;
(3) The tracker returns the peer list to the client;
(4) The client performs a three-way handshake with the peer, followed by the exchange of Bitfield, Have, Negotiation, and Piece messages for fragment transmission.

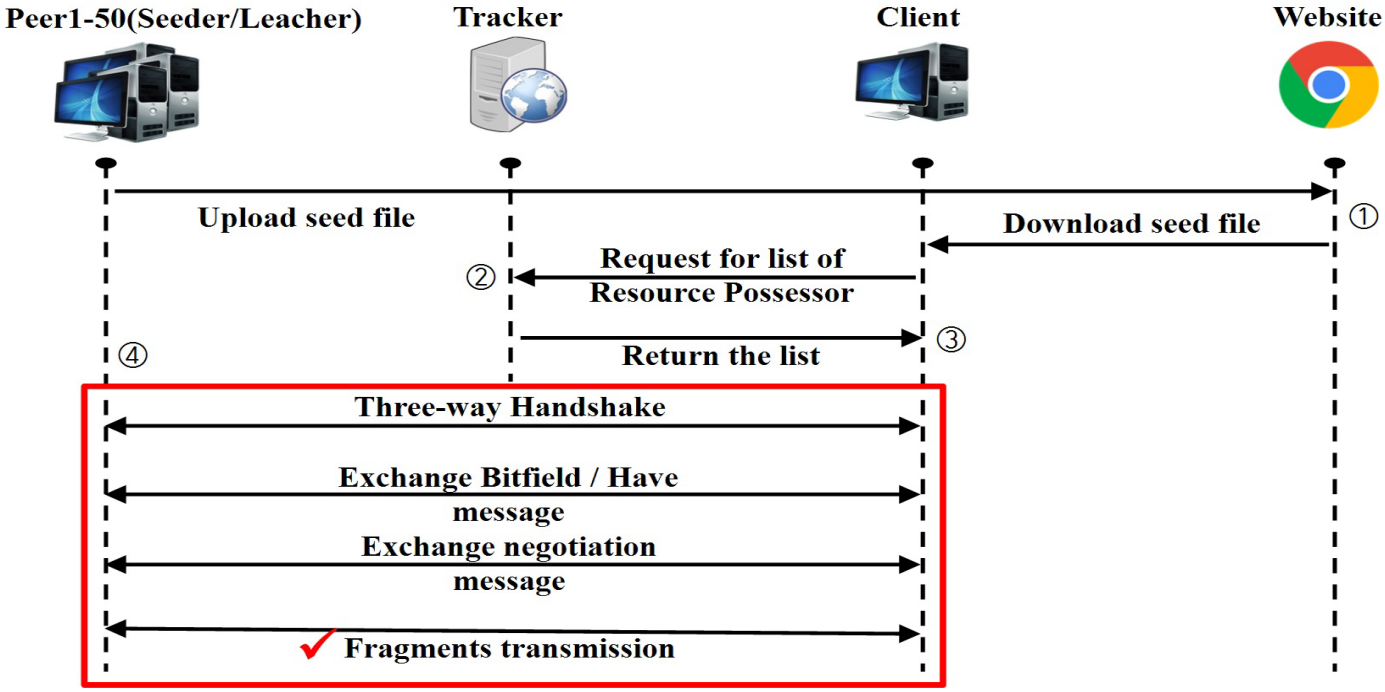


Figure 2. Communication sequence diagram of BitTorrent file sharing.

Table 1 shows the nine types of BitTorrent message packets. Among them, there are two message types with payloads: the Bitfield message and the Piece message.

Table 1. BitTorrent message type.

Message Type	ID	Payload
CHOKE	00	NO
UNCHOKE	01	NO
INTERESTED	02	NO
NOT INTERESTED	03	NO
HAVE	04	NO
BITFIELD	05	YES
REQUEST	06	NO
PIECE	07	YES
CANCEL	08	NO

In BitTorrent, files are divided into multiple pieces, each typically 256 KB in size. The Bitfield message indicates the indexes of the file pieces a peer already possesses. The format of the Bitfield message is illustrated in Figure 3. The bitfield, consisting of X bytes, represents the possession status of specific pieces. The piece corresponding to the highest bit of the first byte is checked first. A bit value of "1" indicates that the corresponding piece is possessed, whereas a bit value of "0" means that the piece is not owned by the peer. Because the number of pieces depends on the total file size, the length of the bitfield is variable. The Bitfield message is transmitted only after the three-way handshake is completed. Since the BitTorrent protocol does not provide an authentication mechanism for Bitfield messages, their modification may not be detected abnormally [28].

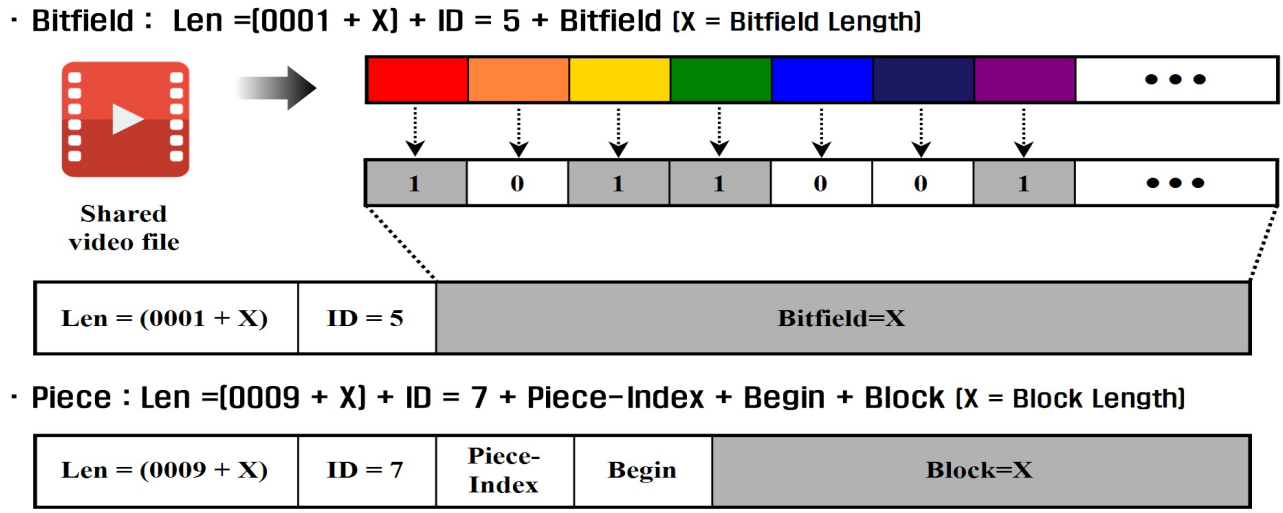


Figure 3. Bitfield/Piece message format.

Once divided into pieces, the file is further subdivided into 16 KB blocks before transmission, which are sent as Piece messages. The format of the Piece message is shown in Figure 4. The piece index field specifies the index number of the piece, and the block of X bytes contains the metadata about the file. Unlike the Bitfield message, the Piece message performs hash verification at the destination; if the hash value does not match, the corrupted data are discarded, and the pieces are re-downloaded from other peers or leechers [24].

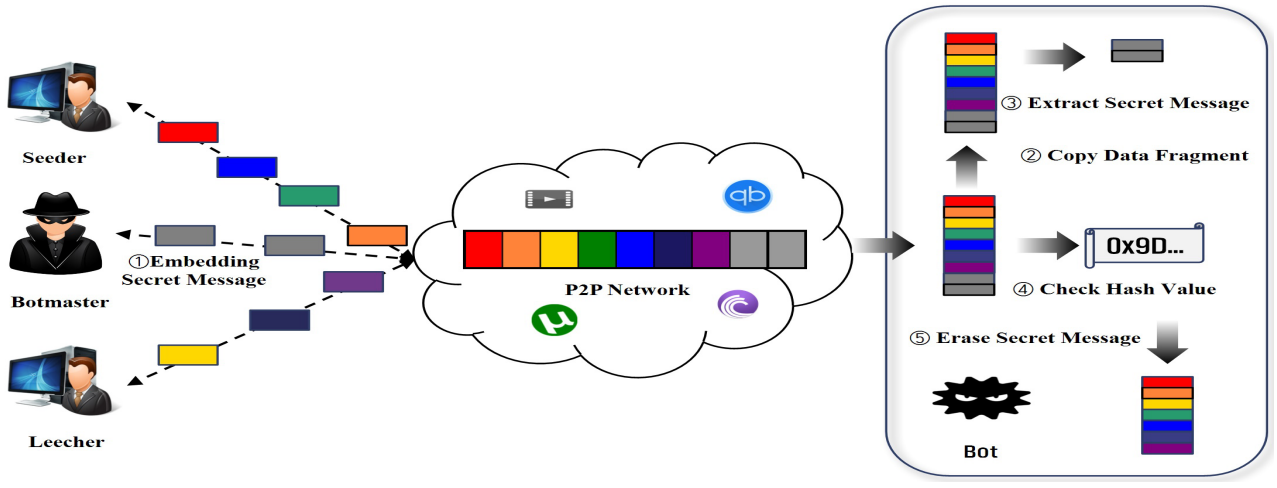


Figure 4. Proposed model.

2.4. Existing Studies and Their Limitations

Nagaragja et al. [7] were the first to study the feasibility of building a botnet based on steganography and proposed the concept of a stego-botnet. Hiney et al. [13] investigated the possibility of constructing a stego-botnet on the Facebook platform, focusing on the compression of image files during communication to determine the conditions under which hidden messages remain intact. Jeon et al. [3] introduced an image-based stego-botnet implemented through the KakaoTalk messenger, while Kwak et al. [8] proposed a video stego-botnet operating on the Telegram platform. However, due to the inherent limitations of SNS platforms—particularly the restricted size of transferable files—it remains difficult to conceal and transmit large payloads to bots.

Several studies have also explored hidden communication using network steganography techniques. Wu et al. [29] surveyed VoIP-based steganography, noting that Voice over IP (VoIP), as a service of IP networks, minimizes communication costs while meeting diverse user needs such as voice and fax transmission. These advantages have made VoIP a dominant form of streaming media communication. In this context, the network layer manages packet activity within the network, the transport layer provides end-to-end communication between hosts, and the application layer handles application-specific operations—making it possible to embed secret information. However, previous studies have failed to achieve sufficient transmission capacity or have shown weaknesses in detection evasion [23,29,30].

BitTorrent, as a representative torrent client of peer-to-peer (P2P) networks, is also well-suited for covert communication, and thus various studies have been conducted in this area. Zishuai et al. [31] proposed a steganographic method that modifies the metadata of seed files, which proved effective at the time but is no longer applicable due to protocol updates. Kopiczko et al. [32] achieved a bandwidth of 270 b/s by presenting a network steganography method using the UTP protocol employed in BitTorrent P2P exchanges. However, the payload capacity was small, and the detection rate remained high. Wang et al. [24] proposed a covert communication channel exploiting the lack of authentication in Bitfield messages. Nevertheless, since the Bitfield is transmitted only once at the beginning of a peer session, the method could achieve only 1 KB of payload per 256 KB of cover medium, which is relatively low compared to digital steganography.

Steganography-based detection techniques have also been actively studied [33–36]. While traditional simple malware can often be detected through reverse engineering, steganography-based malware is significantly more challenging to identify using such methods. Consequently, detection techniques employing AI, machine learning, and deep learning have emerged. Denysiuk et al. [35] demonstrated improved detection performance over existing steganalysis networks such as SRNet, Xu-Net, and YeNet by analyzing frequency spectrum variations in images using discrete cosine transform (DCT) analysis. Revyakina et al. [36] proposed a machine learning-based malware detection technique of network steganography.

However, as detection technologies evolve, concealment techniques are also advancing through the use of AI and machine learning, making steganography-based detection increasingly challenging.

3. Stego-Botnet C&C Communication Model in BitTorrent

3.1. Problem Description and Concept of Proposed Method

Similar to SNS-based stego-botnets, a digital steganography technique that embeds command and control (C&C) messages within image or video files is required. In this study, we propose to implement such a method within the BitTorrent environment, which

supports a large number of peers for high-speed data transfer and, unlike SNS platforms, imposes no limitation on single transmission capacity.

However, attackers face the risk of identity exposure because the attacker must distribute the initial cover medium containing the embedded data before peers are created, and the file generated at the destination inherently contains information from the original cover medium.

To address these issues, we propose a novel communication and control (C&C) model based on network steganography, designed to mitigate the risk of identity leakage. The proposed model is illustrated in Figure 4, and operates as follows:

(1) The botmaster embeds the C&C message into a Piece message using a network steganography technique and transmits it to bots disguised as a normal Piece message.
(2) Before the Piece message undergoes the hash verification process, a copy of it is created in a random folder.
(3) The bot extracts the embedded C&C message from the copied Piece message.
(4) The original Piece message then proceeds through the standard BitTorrent hash verification procedure.
(5) After verification, the embedded Piece messages are deleted, and only the normal pieces are collected to reconstruct the file.

This approach enables covert C&C communication while maintaining compliance with the BitTorrent protocol, thus reducing the likelihood of detection and identity exposure.

3.2. The Procedures of the Proposed Model

The procedure of the proposed model consists of the following 11 steps, as illustrated in Figure 5.

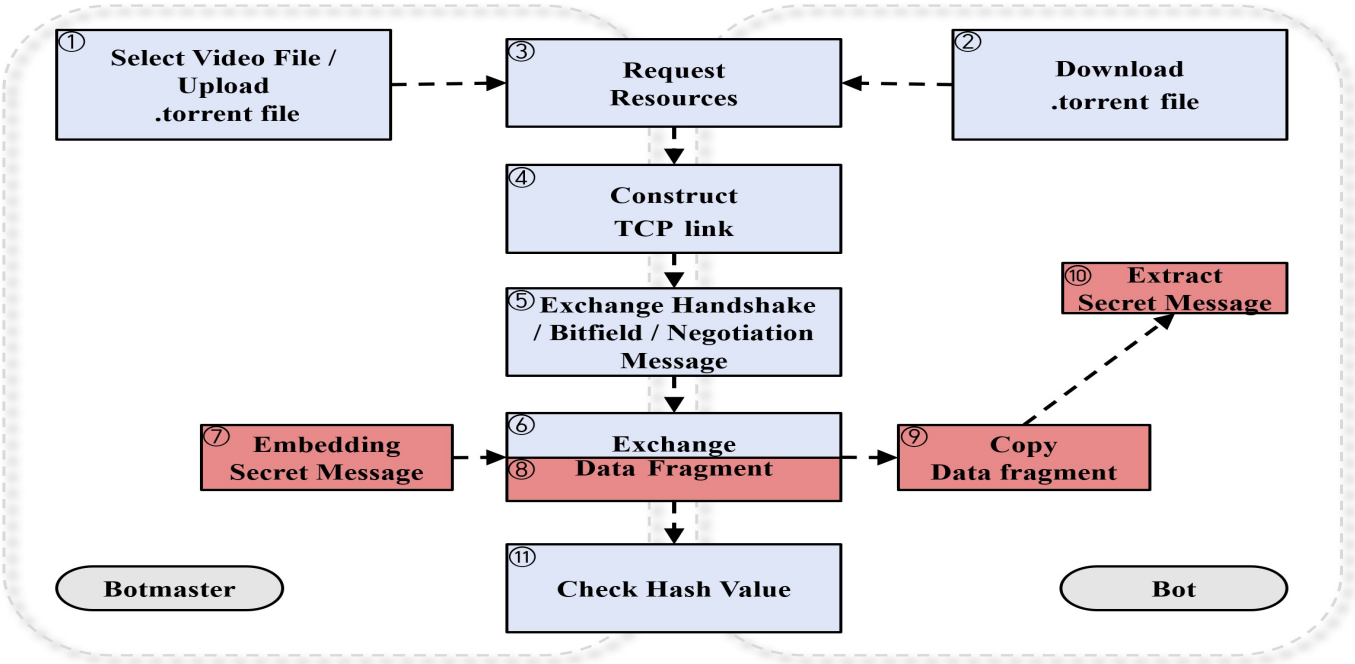


Figure 5. Procedure of proposed model.

(1) The botmaster selects a video file of an appropriate size, depending on the size of the C&C message to be embedded.
(2) The botmaster delivers the corresponding seed file to the bot through a website of an SNS platform.

(3) The bot executes the seed file and attempts to connect to peers sharing the same file.
(4) The bot establishes normal TCP connections with the botmaster and other peers possessing the file.
(5) Through the exchange of the three-way handshake, Bitfield, and Negotiation messages, information such as session details, fragment ownership, and fragment transfer requests is obtained by the bot.
(6) The bot and peers then exchange Piece messages, and the bot collects file pieces from all peers that own the file.
(7) After establishing a normal session with the bot, the botmaster embeds the C&C message into a Piece message.
(8) The embedded Piece message is transmitted to the bot (as shown in Figure 4, Step ①).
(9) Upon receiving the Piece message, the bot copies and stores it in a random folder before the hash verification process begins (as shown in Figure 4, Step ②).
(10) The bot extracts the embedded C&C message from the copied Piece message (as shown in Figure 4, Step ③).
(11) The original Piece message is then processed through the standard hash verification procedure (as shown in Figure 4, Step ④). After verification, any modified pieces are deleted, and only verified fragments are collected to reconstruct a normal file (as shown in Figure 4, Step ⑤).

This process allows the botmaster to transmit hidden C&C commands within normal BitTorrent traffic while maintaining protocol compliance, thereby reducing the likelihood of detection.

4. Experiments

4.1. Experimental Purpose and Methods

The purpose of this experiment is to validate that the proposed C&C communication model can be effectively implemented within the BitTorrent environment and to evaluate its efficiency through a comparative study with existing techniques [24]. More specifically, the experiment aims to verify whether the communication model successfully copies the fragments sent by the attacker to a designated folder before the hash verification process.

Since the embedded Piece messages are deleted and only the normal pieces are collected to complete the file, the proposed model can be considered valid in the BitTorrent client if the file downloaded at the destination matches the file transmitted from the source.

The experimental setup consisted of the following hardware and software configurations:

- Laptop: Intel Core i7 (10 cores), 16 GB RAM, Windows 10.
- Desktop: Intel Core i7 (10 cores), 32 GB RAM.
- Virtualization: VirtualBox 6.1.26.
- Client software: BitTorrent Classic 7.10.5.

This experiment assumes that bots are already infected, and we demonstrate whether covert communication is possible without being detected by the botmaster. Thus, we assume that infected bots can exchange seed files freely.

Four peers participated in the covert communication experiment. Peer1, Peer2, and Peer3 were controlled by the sender to cooperatively transmit the hidden message, while Peer4 acted as the receiver. The size of the secret message was 36 bytes, and the shared video file used as the cover medium was 105 MB in size. The bot software copied the packets transmitted during the session, and these were compared with the packets captured at the destination using Wireshark 3.4.9, HashMyFiles 1.95 was utilized to compare the hash values between the original and the downloaded video files, ensuring data integrity.

All experiments were conducted in an isolated and controlled environment to prevent external interference.

4.2. Implementation and Operation

As shown in Figure 6, all message packets arriving before the hash verification process are stored in a randomly designated folder to confirm file creation. The stored file is observed to be identical to the normal Piece message. Furthermore, as shown in Figure 7, the regular Piece message packets are verified to be reconstructed into a standard video file after the hash verification process, as defined by the BitTorrent protocol, regardless of whether packet copies exist. These observations confirm that the proposed model remains unaffected by modifications to Piece messages and that normal communication is maintained without any additional abnormalities.

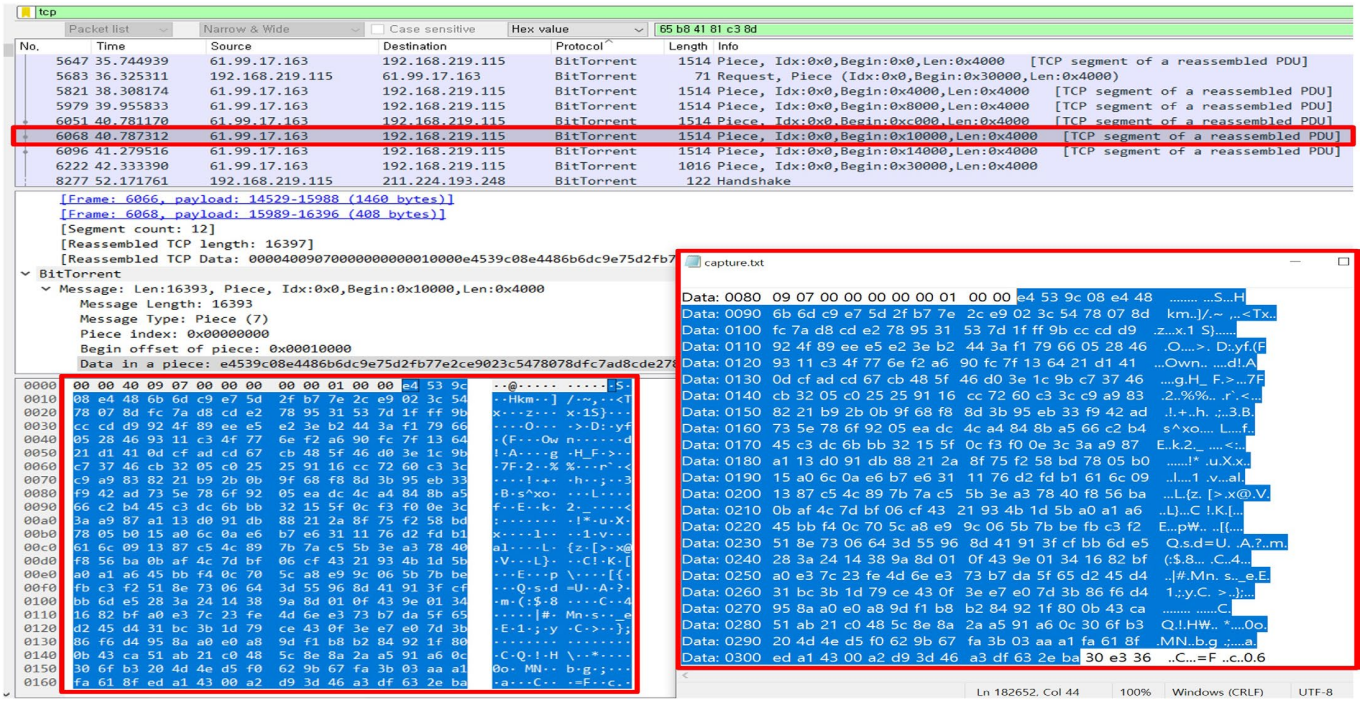


Figure 6. Copying packets to the specified folder.

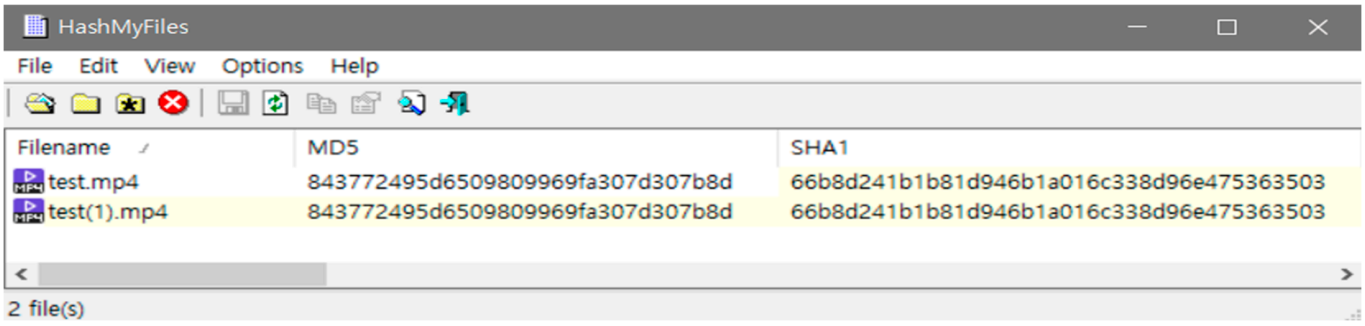


Figure 7. Downloading the video files.

Based on the experimental results, it can be concluded that the proposed model is successfully implementable within the BitTorrent environment and operates correctly and stably under normal protocol conditions.

4.3. Experiment Result

In this experiment, we conduct a comparative analysis to demonstrate that the proposed model is more efficient in securing payload capacity than existing approaches [24]. Specifically, the comparison focuses on video files of average size, calculated by averaging the sizes of the 100 most shared video files.

According to data collected from a popular torrent seed file sharing site, the sizes of the top 100 shared videos ranged from 258 MB to 35.7 GB, with an average size of 1.98 GB. Therefore, video files of 2 GB were used in our experiments to represent the average case. This average value was also used to estimate how many video files would be required to embed a given C&C message when 2 GB video files serve as the cover medium. The maximum value indicates the largest cover medium capacity, and the minimum value indicates the smallest. If the cover medium capacity is large, the required number of cover media will decrease. Conversely, if the cover medium itself is small, the required number will increase.

The analysis results are presented in Table 2 and Figure 8. Using the Bitfield message, a payload of approximately 8 MB can be secured per 2 GB video file. Consequently, two video files are required to transmit a 10 MB C&C message, and 128 video files are needed to transmit 1 GB of C&C data.

Table 2. Number of cover media needed to send secret messages of various sizes.

Size of Secret Message	Bitfield			Piece		
	Avg.	**Max.**	**Min.**	**Avg.**	**Max.**	**Min.**
10 MB	2	1	10	1	1	1
20 MB	3	1	20	1	1	1
30 MB	4	1	30	1	1	1
40 MB	5	1	40	1	1	1
50 MB	7	1	50	1	1	1
100 MB	13	1	100	1	1	1
200 MB	25	2	200	1	1	2
500 MB	63	4	500	1	1	2
1 GB	128	8	1024	1	1	4

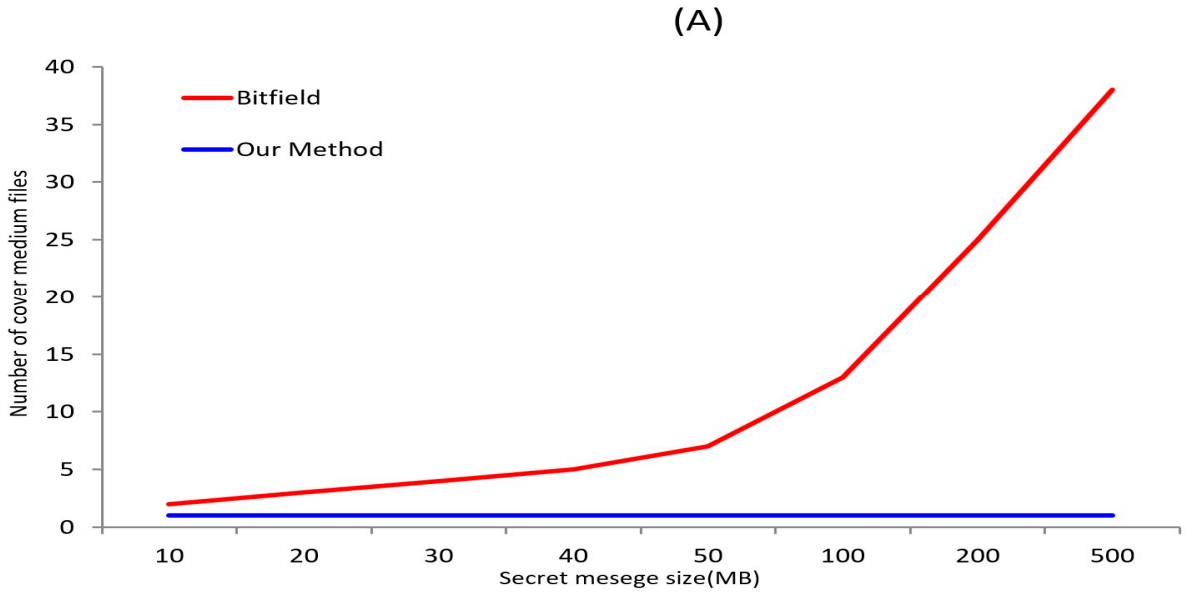


Figure 8. *Cont.*

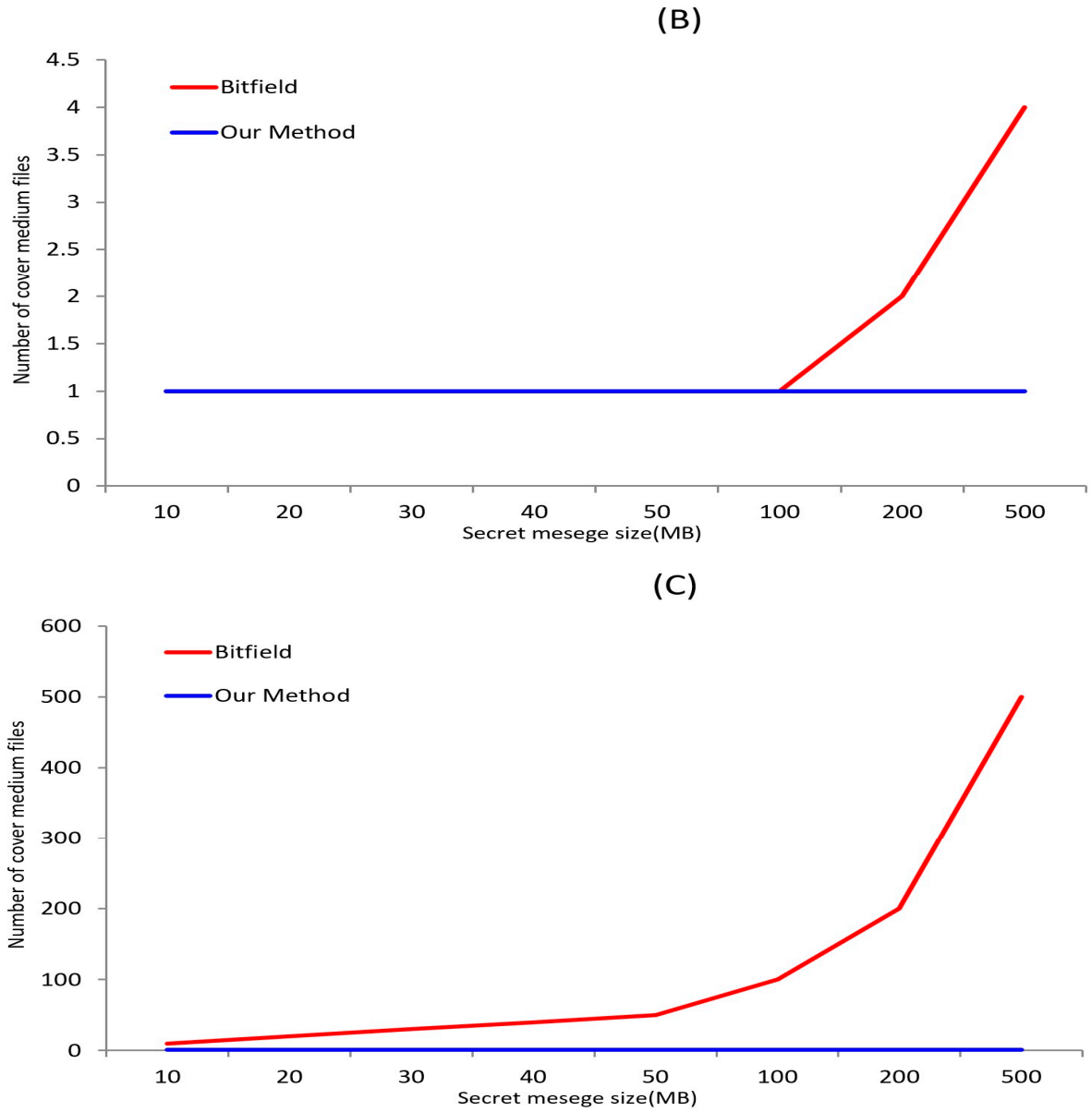


Figure 8. Number of cover medium files measure for Bitfield and Piece message used in experiments—(**A**) Avg., (**B**) Max., and (**C**) Min.

In contrast, when steganography is performed using the Piece message, the entire video file size can be allocated as the payload. This allows for a significantly larger payload capacity compared to the existing technique. This result is noteworthy, as it effectively reduces the cover medium overhead and transmission cost from the attacker's perspective.

As shown in Figure 9, the normal traffic generated during the download of a standard video file and the traffic produced by the proposed model were captured using Wireshark and compared. Since no significant differences were observed between the two types of traffic, the proposed model is confirmed to operate stealthily, indistinguishable from normal BitTorrent communication.

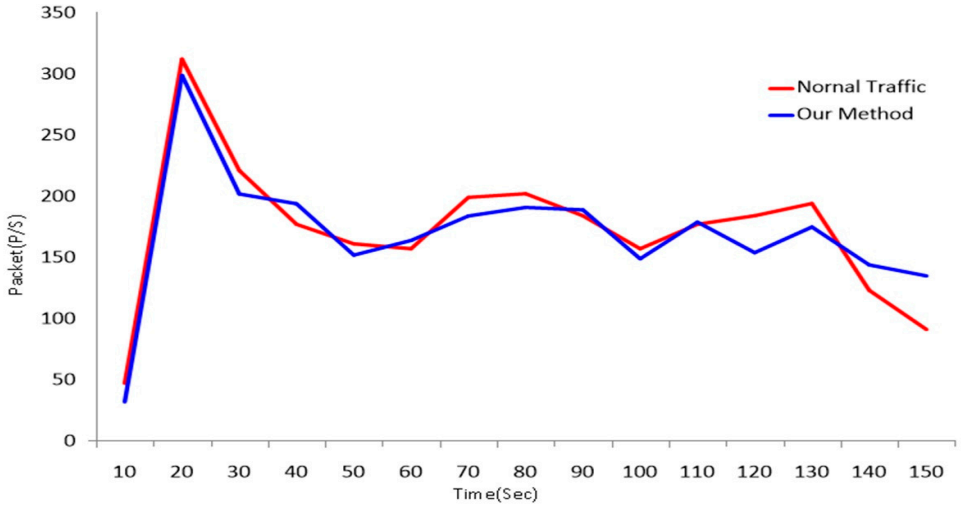


Figure 9. Wireshark-based comparison between normal traffic and traffic from our method.

5. Discussion

This section discusses three aspects of our findings—(1) consistency and integrity, (2) robustness, and (3) stealthiness—and explains the limitations of the proposed model.

5.1. Consistency and Integrity

In the BitTorrent network, consistency refers to maintaining the stability and reliability of the system by ensuring data integrity, data synchronization, and data reliability across all peers. Mechanisms such as hash verification and data synchronization are essential for ensuring consistency. In our proposed model, the Piece message is embedded after synchronization and extracted before hash verification, enabling the exchange of C&C messages within the BitTorrent protocol without compromising consistency.

Integrity is another crucial property of the BitTorrent protocol, as data is shared among multiple peers. To ensure integrity, the destination node verifies the hash value of each piece upon receipt. In our model, the Piece message is embedded in the payload, copied before hash verification at the destination, and deleted through the standard BitTorrent protocol procedure. As a result, only verified and valid pieces are collected to reconstruct the final file, ensuring overall data integrity.

To prevent such potential exploitation, BitTorrent could be enhanced with blockchain, distributed ledger technology (DLT), or consensus algorithms to record transactions immutably and prevent tampering [37,38]. However, integrating these mechanisms could introduce additional issues, such as increased latency and reduced transfer efficiency.

5.2. Robustness

Robustness is a key objective in steganographic systems. It denotes the ability to successfully extract hidden data without corruption despite potential interference factors such as noise, packet loss, or network congestion. Typical methods to enhance robustness include employing advanced algorithms and encryption techniques [39–42].

However, achieving robustness often entails a trade-off with payload capacity. Previous studies [24] have demonstrated that using BitTorrent message payloads provides superior performance in terms of packet loss resilience compared with other approaches. Similarly, this study focuses on increasing payload capacity while maintaining robustness by utilizing BitTorrent message payloads.

Nevertheless, due to the absence of a dedicated verification procedure at the destination, we were unable to fully evaluate the robustness of the proposed technique

against various forms of network attacks. Future research will focus on developing enhanced algorithms that improve the robustness of the proposed model while preserving its payload efficiency.

5.3. Stealthiness

Stealthiness refers to the indistinguishability of covert communication from normal network traffic. To enhance stealthiness, secret message embedding and transmission must avoid generating abnormal or statistically detectable traffic patterns. As shown in Figure 8, the normal traffic generated during the download of a standard video file and the traffic produced by the proposed model were captured and analyzed using Wireshark. No significant differences were observed between the two traffic patterns, confirming that the proposed model operates stealthily within normal BitTorrent communication.

However, there are some challenges that need to be proven through more extensive experiments. While this experiment was conducted in a controlled environment with four peers, a broader set of P2P environments should be simulated to derive results. Also, more sophisticated detection approaches—such as deep packet inspection (DPI) or machine learning-based traffic analysis—could potentially identify anomalies by learning patterns in fragment deletion or reassembly behavior rather than direct traffic features. Addressing this limitation will be a focus of our future research, aiming to enhance resistance to advanced detection mechanisms.

6. Conclusions and Future Work

In this paper, we propose a botnet command and control (C&C) communication model that employs network steganography within the BitTorrent protocol. Compared to previous studies, our model aims to enhance both robustness and confidentiality while overcoming the payload capacity limitations of existing stego-botnet approaches. BitTorrent transfers large files and generates significant network traffic. Therefore, if an attacker utilizes BitTorrent network packets as the cover medium, it becomes suitable for constructing a stego-botnet that is not constrained by payload size. We demonstrate that our proposed model can be effectively implemented within BitTorrent networks and compare its payload capacity with that of existing network-steganography-based C&C models. Furthermore, we show that by embedding data within the payloads of piece messages, covert communication can occur seamlessly within normal traffic, ensuring stealthiness while maintaining protocol consistency and integrity. However, we also identify the need for new algorithms—such as those incorporating blockchain and encryption—to further strengthen robustness. Covert communication should not only emphasize payload capacity but also ensure channel stability and stealthiness.

Future research will focus on several directions to further enhance the effectiveness and resilience of the proposed BitTorrent-based stego-botnet communication model.

First, it is essential to explore advanced detection mechanisms capable of identifying covert communication in peer-to-peer (P2P) environments. Approaches that combine data loss prevention (DLP) systems with machine learning (ML) models may provide effective detection capabilities by analyzing statistical anomalies, behavioral patterns, and traffic metadata without relying on payload inspection.

Second, the robustness and confidentiality of the steganographic channel can be further enhanced by integrating cryptographic and blockchain-based algorithms. For instance, lightweight encryption schemes can ensure message integrity and confidentiality without significantly affecting transmission speed, while blockchain mechanisms can be used to decentralize C&C infrastructure and enhance resilience against takedown attempts and single points of failure.

Finally, research into defensive and mitigation strategies is also crucial. Developing ML-based defensive systems that can dynamically detect and block stego-botnet traffic in real time could provide practical protection against such threats. These defensive models could leverage supervised and unsupervised learning techniques to identify subtle deviations in normal BitTorrent traffic, offering adaptive and scalable countermeasures.

Through these research directions, future work aims not only to strengthen the stealth and stability of covert communication but also to contribute to the broader understanding of secure and resilient network steganography systems.

Author Contributions: Conceptualization, G.P. and Y.C.; methodology, Y.C.; software, G.P.; validation, G.P. and Y.C.; formal analysis, G.P. and Y.C.; investigation, G.P. and Y.C.; resources, G.P.; writing—original draft preparation, G.P. and Y.C.; writing—review and editing, G.P., Y.C. and G.Q.; visualization, G.P.; supervision, Y.C.; project administration, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The software code and data used to support the findings of this study are available from the author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Badar, L.T.; Carminati, B.; Ferrari, E. A Comprehensive Survey on Stegomalware Detection in Digital Media, Research Challenges and Future Directions. *Signal Process.* **2025**, *231*, 109888. [CrossRef]
2. Vormayr, G.; Zseby, T.; Fabini, J. Botnet Communication Patterns. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2768–2796. [CrossRef]
3. Jeon, J.; Cho, Y. Construction and Performance Analysis of Image Steganography-Based Botnet in Kakaotalk Openchat. *Computers* **2019**, *8*, 61. [CrossRef]
4. Acarali, D.; Rajarajan, M.; Komninos, N.; Herwono, I. Survey of Approaches and Features for the Identification of HTTP-Based Botnet Traffic. *J. Netw. Comput. Appl.* **2016**, *76*, 1–15. [CrossRef]
5. Eslahi, M.; Rohmad, M.S.; Nilsaz, H.; Naseri, M.V.; Tahir, N.M.; Hashim, H. Periodicity Classification of HTTP Traffic to Detect HTTP Botnets. In Proceedings of the 2015 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Langkawi, Malaysia, 12–14 April 2015; pp. 119–123.
6. Khattak, S.; Ramay, N.R.; Khan, K.R.; Syed, A.A.; Khayam, S.A. A Taxonomy of Botnet Behavior, Detection, and Defense. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 898–924. [CrossRef]
7. Nagaraja, S.; Houmansadr, A.; Piyawongwisal, P.; Singh, V.; Agarwal, P.; Borisov, N. Stegobot: A Covert Social Network Botnet. In Proceedings of the Information Hiding, 13th International Conference, IH 2011, Prague, Czech Republic, 18–20 May 2011; Filler, T., Pevný, T., Craver, S., Ker, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 299–313.
8. Kwak, M.; Cho, Y. A Novel Video Steganography-Based Botnet Communication Model in Telegram Sns Messenger. *Symmetry* **2021**, *13*, 84. [CrossRef]
9. Zhuang, D.; Chang, J.M. PeerHunter: Detecting Peer-to-Peer Botnets through Community Behavior Analysis. In Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing, Taipei, China, 7–10 August 2017; pp. 493–500.
10. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The Rise of Social Bots. *Commun ACM* **2016**, *59*, 96–104. [CrossRef]
11. Xue, Y.; Zhou, J.; Zeng, H.; Zhong, P.; Wen, J. An Adaptive Steganographic Scheme for H.264/AVC Video with Distortion Optimization. *Signal Process. Image Commun.* **2019**, *76*, 22–30. [CrossRef]
12. Raj, U.A.S.; Maheswaran, C.P. Secure File Sharing System Using Image Steganography and Cryptography Techniques. In Proceedings of the 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 26–28 April 2023; pp. 1113–1116.
13. Hiney, J.; Dakve, T.; Szczypiorski, K.; Gaj, K. Using Facebook for Image Steganography. In Proceedings of the 2015 10th International Conference on Availability, Reliability and Security, Toulouse, France, 24–27 August 2015; pp. 442–447.
14. Sun, Y.; Lu, Y.; Chen, J.; Zhang, W.; Yan, X. Meaningful Secret Image Sharing Scheme with High Visual Quality Based on Natural Steganography. *Mathematics* **2020**, *8*, 1452. [CrossRef]

15. Alhaddad, M.J.; Alkinani, M.H.; Atoum, M.S.; Alarood, A.A. Evolutionary Detection Accuracy of Secret Data in Audio Steganography for Securing 5G-Enabled Internet of Things. *Symmetry* **2020**, *12*, 2071. [CrossRef]
16. Niu, K.; Li, J.; Yang, X.; Zhang, S.; Wang, B. Hybrid Adaptive Video Steganography Scheme Under Game Model. *IEEE Access* **2019**, *7*, 61523–61533. [CrossRef]
17. Yuk, S.; Cho, Y. A Time-Based Dynamic Operation Model for Webpage Steganography Methods. *Electronics* **2020**, *9*, 2113. [CrossRef]
18. Liang, X.; Kar, P. Robust and Secure File Transmission through Video Streaming Using Steganography and Blockchain. *Int. J. Inf. Comput. Secur.* **2024**, *24*, 258–283. [CrossRef]
19. Vasava, D.; Doshi, N. Study and Analysis of Network Steganography Methods. In Proceedings of the ICT with Intelligent Applications, ICTIS 2022, Ahmedabad, India, 22 April 2022; Choudrie, J., Mahalle, P., Perumal, T., Joshi, A., Eds.; Springer Nature Singapore: Singapore, 2023; pp. 71–79.
20. Wen, W.; Huang, H.; Qi, S.; Zhang, Y.; Fang, Y. Joint Coverless Steganography and Image Transformation for Covert Communication of Secret Messages. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 2951–2962. [CrossRef]
21. Lei, J.; Li, D.; Pan, Z.; Sun, Z.; Kwong, S.; Hou, C. Fast Intra Prediction Based on Content Property Analysis for Low Complexity HEVC-Based Screen Content Coding. *IEEE Trans. Broadcast.* **2017**, *63*, 48–58. [CrossRef]
22. Mazurczyk, W.; Karaś, M.; Szczypiorski, K. SkyDe: A Skype-Based Steganographic Method. *arXiv* **2013**, arXiv:1301.3632. [CrossRef]
23. Soundararajan, D.; Ramakrishnan, S. Coverless Data Hiding in VoIP Based on DNA Steganography with Authentication. *Int. Arab J. Inf. Technol.* **2023**, *20*, 190–198. [CrossRef]
24. Wang, M.; Gu, W.; Ma, C. A Multimode Network Steganography for Covert Wireless Communication Based on BitTorrent. *Secur. Commun. Netw.* **2020**, *2020*, 8848315. [CrossRef]
25. Yang, Z.; Wang, B. A Feature Extraction Method for P2P Botnet Detection Using Graphic Symmetry Concept. *Symmetry* **2019**, *11*, 326. [CrossRef]
26. Incentives Build Robustness in BitTorrent. Available online: https://www.bittorrent.org/ (accessed on 9 September 2025).
27. Xia, R.L.; Muppala, J.K. A Survey of BitTorrent Performance. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 140–158. [CrossRef]
28. Srinivasan, A.; Aldharrab, H. XTRA—EXtended Bit-Torrent PRotocol for Authenticated Covert Peer Communication. *Peer Peer Netw. Appl.* **2019**, *12*, 143–157. [CrossRef]
29. Wu, Z.; Guo, J.; Zhang, C.; Li, C. Steganography and Steganalysis in Voice over Ip: A Review. *Sensors* **2021**, *21*, 1032. [CrossRef]
30. Zhang, C.; Jiang, S.; Chen, Z.; Qian, J. MSSN: Multi-Stream Steganalysis Network for Detection of QIM-Based Steganography in VoIP Streams. *IEEE Trans. Dependable Secur. Comput.* **2025**. [CrossRef]
31. Li, Z.; Sun, X.; Wang, B.; Wang, X. A Steganography Scheme in P2P Network. In Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 15–17 August 2008; pp. 20–24.
32. Kopiczko, P.; Mazurczyk, W.; Szczypiorski, K. StegTorrent: A Steganographic Method for the P2P File Sharing Service. In Proceedings of the 2013 IEEE Security and Privacy Workshops, San Francisco, CA, USA, 23–24 May 2013; pp. 151–157.
33. Wei, S.; Wang, Z.; Zhang, X. Universal Image Vaccine Against Steganography. *Symmetry* **2025**, *17*, 66. [CrossRef]
34. Moghadasi, H.A.; Dehghani, H. Detection of Speech Steganography for VoIP Stream Based on Deep Learning Approach in G.729 Codec. *Res. Sq.* **2025**. [CrossRef]
35. Denysiuk, D.; Savenko, O.; Kvassay, M. Method for Detecting Malicious Commands Transmitted via Images Using Steganography. In Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2025), Khmelnytskyi, Ukraine, 4 April 2025.
36. Revyakina, E.; Gazizov, A. Development of Methods and Tools for Implementing and Detecting Network Steganography. *Pak. J. Life Soc. Sci. PJLSS* **2025**, *23*, 238–248. [CrossRef]
37. Ashwin, K.S.; Prakash, B.; Kulkarni, B.S.; Shivaprasad, B.J. Blockchain-Based Authentication for Secure Document Sharing with AES Encryption and Steganography. In Proceedings of the 2025 International Conference on Computing Technologies (ICOCT), Bengaluru, India, 13–14 June 2025; pp. 1–7.
38. Yuan, X.; Sun, J.; Chen, Z.; Zhang, C.; Li, M.; Zhang, Z.; Zhu, L. IAAE-Stega: Generic Blockchain-Based Steganography Framework via Invertible Adversarial Autoencoder. *IEEE Trans. Netw. Sci. Eng.* **2025**, 1–16. [CrossRef]
39. Rajba, P.; Keller, J.; Mazurczyk, W. Entropy-Aware Secret Data Embedding for Network Storage Channels. In Proceedings of the Availability, Reliability and Security, ARES 2025 International Workshops, Ghent, Belgium, 11–14 August 2025; Coppens, B., Volckaert, B., Naessens, V., De Sutter, B., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2025; pp. 327–345.
40. Gao, S.; Zhang, Z.; Li, Q.; Ding, S.; Iu, H.H.C.; Cao, Y.; Xu, X.; Wang, C.; Mou, J. Encrypt a Story: A Video Segment Encryption Method Based on the Discrete Sinusoidal Memristive Rulkov Neuron. *IEEE Trans. Dependable Secur. Comput.* **2025**. [CrossRef]

41. Liu, Y.; Wang, C.; Lu, M.; Yang, J.; Gui, J.; Zhang, S. From Simple to Complex Scenes: Learning Robust Feature Representations for Accurate Human Parsing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5449–5462. [CrossRef] [PubMed]
42. Wang, C.; Zhang, Q.; Wang, X.; Zhou, L.; Li, Q.; Xia, Z.; Ma, B.; Shi, Y.Q. Light-Field Image Multiple Reversible Robust Watermarking Against Geometric Attacks. *IEEE Trans. Dependable Secur. Comput.* **2025**. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.