Covert Channel Construction Method Based on HTTP Composite Protocols

Longxing Jin, 1 Zijun Liu, 1 Fuquan Huang, 1 Zizhao Lin, 1 and Ming Li 2

1 Department of System Operation, Shenzhen Power Supply Bureau Co., Ltd., Shenzhen, Guangdong 440304, China
2 Industry Information Security Center, Xingtang Communication Technology Co., Ltd., Beijing 100191, China

Correspondence should be addressed to Ming Li; lm_619090@126.com

1. Introduction

With the rapid development of information network technology, the use of the network has become ubiquitous, and various terminal devices have been connected to the network one after another, and the number of people interacting through the network has also increased exponentially [1]. While the network brings many conveniences to people’s work and life, all kinds of information security problems come one after another. In order to fight against the increasingly bad network environment, people pay more and more attention to the security of information transmission, and information encryption has become a necessary security means in information transmission. However, encryption technology only makes the data unreadable, which is easy for attackers to detect, violently destroy, or cut off the data transmission path. Moreover, with the rapid development of processor performance and cryptography, the difficulty of password cracking gradually decreases. As a communication mechanism that can violate security policy and is difficult to detect, the core of covert channel lies in insensitivity. Therefore, how to ensure the concealment of the channel has become the focus of covert channel design. While ensuring the concealment of hidden channels, hidden channels have strong reliability and certain channel capacity is the key to improve the performance of hidden channels.

This paper focuses on how to improve the concealment of covert channels and proposes a method of constructing covert channels based on different combinations of HTTP packet information. The encoding table is constructed by using different combinations of User-Agent fields in HTTP packets, and the hidden information is converted into binary sequences to correspond with the mapping table; at the same time, the multisite access mechanism is used to improve the concealment.
network transmission protocol were mainly used to embed covert information in a filling way to realize covert communication. For example, replace some redundant, reserved or random fields with hidden information [3, 4], or embed Cookies, URL, Entity body, HTML page code, and other contents in the network protocol into hidden information [5, 6]. Although most of these covert channels can reach higher channel capacity, they are weak in concealment and easy to be detected or destroyed. In order to improve the concealment of stored covert channels, researchers began to use random fields such as packet length and serial number to carry covert information. LUO [7] et al. proposed a covert channel (Cloak), which uses multiple TCP streams to combine and transmit covert information, and constructed a channel (Cloak), which uses multiple TCP streams to combine and transmit covert information. LUO [7] et al. proposed a covert channel (Cloak), which uses multiple TCP streams to combine and transmit covert information, and constructed a mathematical coding method, which conforms to the 12-fold way model. WANG [8] et al. proposed an HTTP multiplexing covert channel based on browser application, which uses multipath transmission to reduce the correlation of packets between single paths. However, in order to solve the problem of packet arrival order in different paths, this method sacrifices large channel capacity to add redundant information.

Time covert channel mainly modifies the time attribute of data packet to make it have certain regularity. For example, IPCTC [9] realizes coding by whether to send data packets within a fixed time interval. Jitterbug, a time-based covert channel proposed by SHAH [10], is a passive covert channel, which does not need to build its own data packet, and the sender will send a data packet to the server while tapping the keyboard, and automatically add a time delay to the data packet when sending, thus realizing covert information transmission. Qian Yuwen [11] and others realized the full-duplex covert channel RCTC by using HTTP protocol, which greatly improved the reliability of this kind of channel. However, the robustness of this channel decreases with the increase of the distance between the two parties and will be greatly affected by network jitter when the distance is far away, resulting in high bit error rate and channel failure.

The application of signature detection method [12] can effectively detect the hidden channel constructed by modifying the format or content of some protocol fields in the data packet. In a normal HTTP application, keywords and most field contents will not change at will, and data packets that are quite different from the conventional settings will be judged as abnormal data. In order to detect hidden channels based on behavioral features, DUSI [14] et al. proposed fingerprint detection. The method calculates the protocol fingerprint value according to the packet size and time interval in the packet sequence, trains through a large number of packet extraction features generated by normal traffic to obtain the fingerprint distribution of normal traffic packets, compares the audited HTTP protocol packet fingerprint information with the HTTP fingerprint information generated by normal traffic, and judges that the audited channel contains hidden information if there is significant difference between them. Guo Chuanchuan [14] proposed a combined detection model to detect fields with high randomness, such as ISN and timestamp, which used chaos theory and Markov model to detect fields with no specific value, such as reserved fields, timestamp, and ISN, with good detection effect.

To sum up, the existing hidden channel construction methods are still difficult to balance concealment, reliability, and channel capacity. Memory covert channel has poor concealment. Using the hidden channel of multiplex transmission requires a large resource overhead and reduces the channel capacity. Time covert channel is greatly affected by network jitter in long-distance transmission or poor network environment, and its reliability is low.

3. Hidden Channel Construction Method Based on HTTP Protocol Combination

3.1. Algorithm Framework. The core idea of the construction method of covert channel based on HTTP protocol combination (HUCC) is to study a covert channel, which can avoid conventional feature detection, is not affected by network jitter, and is suitable for various application scenarios, and the basic principle framework is shown in Figure 1.

The whole construction method includes three main functional modules: information coding module, anti-detection processing module, and information decoding module. Before information coding, first build a coding mapping table to correspond the distribution of different HTTP requests on browser applications to binary sequences. Second, the hidden information is compressed and encrypted to obtain the binary sequence $u$, the binary sequence is converted into the encoded information $V$ by using the encoding mapping table, and the corresponding browser combination is selected according to $V$ to prepare to send the HTTP request. Then, the concealment is enhanced by antidetection processing, including packet length simulation, time interval simulation, and multiwebsite access mechanism. Finally, the information decoding module reverses the encoded information $V'$ through the encoding mapping table to obtain the binary sequence $u'$, decrypts, and decompresses the hidden information. Because the bottom layer of HUCC method relies on TCP to ensure the transmission reliability and ensure that the order of data packets will not change during transmission, there will be no disorder between groups of data packets, that is, $V' = v$. Only $u' = u$ can ensure error-free transmission.

3.2. Information Coding. The main idea of the information coding module is to design a coding method, which is not easy to be perceived by the detection system and has a certain channel capacity. Web browsers are the most widely used HTTP applications, and each Web server gateway will inevitably receive a large number of HTTP packets generated by different browsers. Therefore, this paper chooses to simulate the scene where users visit web pages using browsers to build hidden channels. Firstly, N browser-like HTTP applications are deployed at the sender, and the hidden information is embedded by distributing $M$ HTTP requests to N different browser applications in different combinations. In order to avoid alerting some intelligent application-level gateways based on machine learning, all
HTTP requests constructed by clients come from general browsers. The User-Agent keyword represents the type and version of each browser, operating system and version, browser kernel, and other information, so this field of each browser must be different. Therefore, the User-Agent field can be used to explicitly distinguish the packets generated by each browser. In addition, the data packet generated by browser application will not change the order and format of keywords and parameters in the protocol header, so it can resist the application of fingerprint detection method.

The sender divides the compressed and encrypted binary sequence \( u \) into several groups, and each group corresponds to a binary bit segment \( U_i \). Firstly, we set a set of binary fragment \( U_i \) corresponding to \( m \) HTTP requests, and select \( n \) different browser applications; secondly, \( m \) HTTP requests are distributed on \( n \) different browser applications, and different combinations are established, and each combination \( V_i \) corresponds to a binary fragment \( U_i \).

From the perspective of combinatorial mathematics, this problem can be simplified to the problem that there are several combinations of \( m \) balls in \( n \) boxes. Since there is no obvious difference between each HTTP request, and the browser application can be explicitly distinguished according to the User-Agent field, this problem is transformed into the problem that there are several combinations of \( m \) identical balls in \( n \) different boxes.

A total of \( \binom{N}{N-M+1} \) combinations can be obtained by the plug-in method, that is, after one-time coding. There are 11 possible coding methods in the program, so \( \log_2 \binom{N}{N-M+1} \) bit data can be transmitted every time. Assuming that four different HTTP applications are selected in one sending process, and every five data packets are set to correspond to a group of data, C83 can be constructed in total, that is, 56 combinations, and 5 bit of data can be transmitted each time.

In order to simulate the real situation as much as possible, this paper makes statistics on the market share of browsers. The results show that the share of Google browser has reached about 50%, and the proportion of other commonly used browsers is roughly the same. Therefore, in this paper, only the first 16 combinations are selected to construct the coding table in sequence, and B1 is set as Google browser, so that the probability of its appearance reaches 44/80, that is, 55%, which is similar to the real scene. Compared with the traditional method that each browser corresponds to one encoding method, although this method loses a certain amount of channel capacity, it greatly improves the concealment, which is actually a trade-off between performances. If a covert channel wants to have high concealment, it needs to sacrifice a certain amount of channel capacity, so it is acceptable to improve concealment at a lower performance cost. Let B1, B2, B3, and B4 be four different browser applications in which B1 is Google browser and the other three are Firefox, IE, and Safari browsers. The corresponding relationship between combination mode \( V_i \) and binary fragment \( U_i \) is shown in Table 1.

3.3. Antidetection Treatment. Concealment and reliability are the two core performance indexes of covert channel. The HUCC is based on HTTP protocol, and the bottom layer establishes reliable connection through TCP protocol, depending on the sequence number field and retransmission mechanism in TCP protocol, it can ensure that HTTP data packets in the connection path are sent in sequence, and ensure that each group of data packets arrives at the receiving end in turn; thus, avoiding the disorder of sequence among groups, the receiving end can analyze the corresponding hidden information according to every \( m \) data packets. Therefore, HUCC can realize error-free transmission without extra enhancement of reliability.

Detection resistance schemes mainly focus on protocol fingerprint detection algorithms based on statistical features. If a large number of packets carrying hidden information are generated in a short period of time, the HTTP traffic received by the server increases sharply, which may easily lead to the identification of some traffic audit detection methods. Therefore, the simulated browser sending HTTP requests needs to mimic the access interval of a real user. According to the statistics and analysis of actual data in the literature [15], the time interval of packets of most network traffic

![Figure 1: Basic principle framework of algorithm.](image-url)
roughly follows the Poisson distribution. Assuming that requests in HTTP flows also follow the Poisson distribution, Poisson generator is used to generate discrete time series, as shown in the following formula:

\[ T_n = \text{PoissonDist}(\lambda, n). \]

The length of HTTP packets generated by simulation is mostly fixed, and embedding covert information by packet length is a common covert channel. Detection system based on packet length is also common. Although HUCC method does not carry covert information by packet length, in order to avoid being identified by detection algorithm based on packet length, HUCC method needs to simulate the length of data stream packet generated by normal session to realize the encoding process. By counting the packet length of ordinary traffic, add some information to the HTTP request packet to increase the packet length, so that the packets generated by HUCC method are similar to ordinary traffic and simulate normal request behavior.

In order to avoid detection by intelligent application-level gateways, the behavior pattern of simulated browser access must be as similar as possible to the behavior of ordinary users accessing websites. Because the sender is to simulate the behavior of the browser to visit the website to generate HTTP requests, the simulated browser sends HTTP requests only to the fixed access target to build packets, which will cause certain exceptions and may cause security applications to warn [16]. In order to eliminate this anomaly, this paper uses a multisite access mechanism. The basic idea is to preselect X web pages commonly used by ordinary users as potential access objects of hidden information and mark them as \( W_0, W_1, \ldots, W_{X-1} \). After using a Web page to complete a set of covert information transmission, the access page \( W_{\text{index}} \) will be selected again, and the index calculation method is shown in the following formula:

\[ \text{index} = H(\text{Seg}) \mod X. \]

Including \( \text{Seg} \), is the hidden information to be sent, \( H(\cdot) \) is the hash function, which is used to convert the string into a number sequence, index is the sequence number of web pages accessed, and \( X \) is the number of potential pages accessed. The receiver and sender can add or modify this group of web pages after a period of time to further increase the randomness of access behavior. Through the multisite access mechanism, the visited web page can be updated continuously, and the next web page visited has great randomness, which is similar to the access behavior of ordinary users.

### 3.4. Information Decoding

The decoding module is relatively simple, and the data packet can be parsed through the mapping table.

1. Listening HTTP request: the receiver listens for all HTTP at the gateway. Request and respond normally if legal HTTP request is received; if the request of the IP address of the sender is received, the decoding module is started, and the User-Agent field of each data packet in the target HTTP stream is extracted and transmitted to the next module.

2. Decoding mapping, code conversion is carried out once every \( m \) data packets, and browser combination is converted into binary bit fragments through code table \( U_i' \).

3. Decryption: every two bits of \( U_i \) are spliced and then converted into ASCII code, and then, the ASCII code is converted into corresponding characters. After the characters are spliced, the hidden information is successfully resolved.

### 4. Experiment and Analysis

#### 4.1. Experimental Data Source

In order to verify the performance of covert channel construction method based on HTTP protocol combination, this paper carries out comparative experiments from three aspects of covert channel concealment, reliability, and channel capacity. In order to ensure the consistency of network environment between legitimate channel and covert channel, as well as the reliability and comprehensiveness of legitimate data, 1000 HTTP communications from different periods were selected, and all HTTP packets in the whole communication process were captured. At the same time, abnormal data were eliminated, and 5000 packets were finally selected as legitimate data sets. It is used for the construction of concealment experimental detection model and the comparison of time series experimental detection. 50000 data packets from different time periods can reduce the interference caused by noise samples and improve the reliability of detection method. At the same time, 1000 HTTP data streams are constructed through HUCC in the internal network from which 20000 HTTP request packets carrying hidden information are extracted.

#### 4.2. Experimental Environment

The experimental environment includes one server, one switch, and two hosts. 1 host is installed with Windows 7 operating system as the
4.3. Covert Experiments. HUCC did not modify the message content or format, and will not be detection method based on the application signature recognition; therefore, the main sequential detection test through agreement (FBD) fingerprint method detecting data rooms regularity of interval (IPD) is in accordance with the law of legal channel, and at the same time, testing the distribution of the IPD is similar shape and legal channel.

The concealment experiment process and parameter concealment experiment process are as follows:

(1) The $\lambda$ value (IPD) of HUCC is dynamically adjusted from 5 to 100, and multiple groups of HTTP requests are generated at different time intervals, and FBD method is used to check the detection rate of HUCC under different $\lambda$ values.

(2) Select the appropriate $\lambda$ value, compare whether the IPD distribution shapes of ordinary traffic and HUCC covert data flow are similar, and draw their respective HTTP request distribution maps.

The experimental results of dynamic detection for $\lambda$ value are shown in Figure 3. The abscissa is the expectation of Poisson generator, and the ordinate is FBD’s response to $\lambda$ value.

It can be seen from Figure 3 that when $\lambda$ is in the range of 40–70, FBD detection method is basically ineffective. However, the rate of data packets sent by the channel is $1/\lambda$ (pieces/s), and the decrease of $\lambda$ value means that the data packets sent per unit time increase; that is, the channel capacity increases, so the $\lambda$ value is set to 40.

The experimental results of request distribution comparison between ordinary traffic and HUCC traffic are shown in Figure 4, the ordinate is the number of packets sent per second, and the abscissa is the timeline.

It can be seen from Figure 4 that the request distributions generated by the two communication modes are basically the same, and there is no obvious difference. Therefore, HUCC with appropriate Poisson distribution to adjust the time interval can well simulate the request behavior of normal network communication.

By comparison, it can be seen that the method in this paper has strong concealment and reliability, and the application of signature detection technology, protocol fingerprint technology, and combination model detection method is ineffective for it, and it can accurately transmit information in the harsh network environment with a packet loss rate of 15%. Although the channel capacity is quite different from that of the hidden channel that modifies the header field of the protocol in the literature [12], the method in this paper can avoid many conventional detection

![Figure 2: Network topology of the experimental environment.](image)
methods and can still safely transmit information in the gateway where the detection system is deployed. In this method, if the concealment is not considered, the User-Agent category can be raised to a higher value, which can also have high channel capacity. However, considering the universality of the channel, this method chooses to sacrifice part of the channel capacity to improve concealment, which is actually a trade-off between channel capacity and concealment. With the popularization of traffic normalization technology and covert channel detection technology, the covert channel effect of modifying the conventional fields of the protocol is reduced, which leads to a significant decrease in the concealment of such covert channels in literature [12], and even leads to the exposure of transmission personnel. However, the literature [8], which also has good concealment, does not perform well in terms of reliability and channel capacity, and the construction method used in the literature [8] needs to establish multiple TCP connection paths, resulting in more resource consumption.

4.4. Reliability Experiment. The reliability experiment process and parameter reliability experiment process are as follows:

(1) Build network environments with 5%, 10%, and 15% packet loss rates to simulate bad network conditions.
(2) Use HUCC to transmit 1 KB txt file and Word file with the same text content 5 times each in different packet loss rates, and count the average bit error rate at the receiving end.

The reliability test results are shown in Table 2.

It can be seen from Table 2 that HUCC can accurately transmit data under 15% network jitter, and has extremely high reliability.

5. Conclusion

In view of the limitation of existing hidden channel construction methods, it is difficult to keep a certain balance among concealment, reliability, and channel capacity, this paper proposes a hidden channel construction method HUCC. Firstly, a coding mapping table is established by combining different browsers. Then, according to the mapping table, the hidden information is converted into HTTP request, and the user behavior is simulated by multisite access mechanism. At the same time, the browser information of normal flow, data packet time interval, and data packet length are simulated to ensure concealment, and TCP reliable transmission is used to ensure robustness. Finally, the channel capacity and concealment are weighed by adjusting parameters. In order to test the performance of HUCC, experiments are carried out from three aspects: concealment, reliability, and channel capacity.

Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interests.

Acknowledgments

The authors acknowledge the Research and Demonstration Application of Key Technology of Superconducting Transmission in Urban Distribution Network of China Southern Power Grid, 090000KK52170171.
References

[1] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243–250, 1994.

[2] Y. Wang, J. Wu, and H. Zeng, "covert channel research," *Journal of Software*, vol. 21, no. 9, pp. 2262–2288, 2010.

[3] Y. Wang, J. Wu, and C. Haitao, "Research on hidden channels," *Journal of Software*, vol. 21, no. 9, pp. 2262–2288, 2010.

[4] C. G. Girling, "Covert channels in LAN's," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 292–296, 1987.

[5] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O. Gorman, "DocumentMarking and identification using both line and Word shifting," in *Proceedings of the IEEE. 4th Annual Joint Conference of the IEEE Computer and Communication Societies*, pp. 853–860, IEEE, Boston, MA, USA, April 1995.

[6] E. Brown, B. Yuan, D. Johnson, and P. Lutz, "Covert channels in the HTTP network protocol: channel characterization and detecting man-in-the-middle attacks," *Journal of Information Warfare*, vol. 9, no. 3, pp. 26–38, 2010.

[7] X. Luo, E. W. W. Chan, P. Zhou, and R. K. C. Chang, "Robust network covert communications based on TCP and enumerative combinatorics," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 890–902, 2012.

[8] F. Wang, L. Huang, H. Miao, and M. Tian, "A novel distributed covert channel in HTTP," *Security and Communication Networks*, vol. 7, no. 6, pp. 1031–1041, 2014.

[9] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert timing channels: design and detection," in *Proceedings of the ACM. 11th ACM Conference on Computer and Communications Security*, pp. 178–187, ACM, Washington DC, USA, October 2004.

[10] G. Shah, A. Molina, and M. Blaze, "Keyboards and Covert-Channels[EB/OL]," 2019, https://www.mendeley.com/catalogue/89442cbe-faaa3d0a-a936-2ab989a8bb1.

[11] Y. Qian, B. Zhao, J. Kong, and Z. Wang, "Robust covert timing channel based on web," *Journal of Computer Research and Development*, vol. 48, no. 3, pp. 423–431, 2011.

[12] Y. Qian, B. Zhao, and J. Kong, "Research on a reliable network covert time channel based on Web," *Computer Research and Development*, vol. 48, no. 3, pp. 423–431, 2011.

[13] Z. Kwecka, *Application Layer Covert Channel Analysis and Detection*, Napier University, Edinburgh, 2006.

[14] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, “Tunnel Hunter: detecting application-layer tunnels with statistical fingerprinting,” *Computer Networks*, vol. 53, no. 1, pp. 81–97, 2009.

[15] C. Guo, *Research and Implementation on the System of Storage Network Covert Channels Detection*, University of Electronic Science and Technology of China, Chengdu, 2016.

[16] J. Hao, S. Luo, and L. Pan, “Computer-aided intelligent design using deep multi-objective cooperative optimization algorithm,” *Future Generation Computer Systems*, vol. 124, pp. 49–53, 2021.