International Journal of Networked and Distributed Computing, Vol. 1, No. 3 (August 2013), 144-158

On Effectiveness of Hopping-Based Spread Spectrum Techniques for Network Forensic Traceback

Wei Yu [1] , Xinwen Fu [2] , Erik Blasch [3] , Khanh Pham [4] , Dan Shen [5] , Genshe Chen [5] , and Chao Lu [1]

[1] *The Department of Computer and Information Science, Towson University, MD 21252*
E-mail: {wyu,clu}@towson.edu

[2] *The Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854*
E-mail: xinwenfu@cs.uml.edu

[3] *The Air Force Research Laboratory, Information Directorate, Rome, NY 13441*
E-mail: erik.blasch@rl.af.mil

[4] *The Air Force Research Laboratory, Space Vehicles Directorate, Kirtland AFB, NM 87117*
E-mail: pham.dai.khanh@gmail.com

[5] *The Intelligent Fusion Technology, Inc., 20271 Goldenrod Lane, Suite 2066, Germantown, MD 20876*
E-mail: {dshen,gchen}@intfusiontech.com

Received 15 April 2013

Accepted 28 June 2013

Abstract

Network-based crime has been increasing in both extent and severity and network-based forensics encapsulates an essential part of legal surveillance. A key network forensics tool is traceback that can be used to identify true sources of suspects. Both accuracy and secrecy are essential attributes of a successful forensic traceback. In this paper, we study a class of hopping-based spread spectrum techniques for forensic traceback, which fully utilize the benefits of the spread spectrum approach and preserves a greater degree of secrecy. Our investigated techniques, including Code Hopping-Direct Sequence Spread Spectrum (CH-DSSS), Frequency Hopping-Direct Sequence Spread Spectrum (FH-DSSS), and Time Hopping-Spread Spectrum (TH-DSSS), operate to randomize the effects of marking traffic in both time and frequency domains. Our theoretical analysis, simulations, and real-world experiments validate these DSSS techniques in terms of accuracy and secrecy to benefit network forensics and deter cyber crimes.

Keywords: Traceback, Secrecy, Accuracy, Hopping, DSSS, Cyber Situation Awareness

1. Introduction

In this paper, we address the issue of developing efficient forensic traceback techniques to deal with cyber crimes through anonymous communication networks. Societies in the current world are more dependent on the cyber space, in which commercial and military communication and information dissemination are realized. However, it has also led to cyber security issues. As the number of cyber crimes has been increasing with the convergent and fast growing cyber world, network forensics plays a more important role to support legal surveillance.

Our focus in this paper is addressing the criti-

cal issue of one category of anonymous cyber-attack scenes described below. Particularly, valuable network services for anonymous communication, such as Tor [1] and Anonymizer [2], can enhance privacy by supporting anonymous publishing and browsing, and protect users's privacy from malicious eavesdroppers. However, such anonymous communication systems can be subverted and used for crime, including illegal file sharing or private information distribution. In addition, cyber activities pose new challenges for law enforcement that uses digital forensics to combat the growing number of anonymous cyber crimes. For example, cyber terrorists may communicate and share information through anonymous communication networks.

To address these cyber issues, we develop network forensic traceback techniques to identify anonymous enemies in the challenging cyber-attack scene. As most anonymous nodes do not keep necessary logs for forensic investigations in the aftermath of attacks, defending against such dynamic and anonymous enemies requires real-time data collection, analysis and response. One fundamental network-based forensic technique is *traceback* [3,4]. Both accuracy and secrecy of traceback are important for successful network forensics. Accurate traceback makes surveillance possible, while traceback secrecy prevents suspects from knowing that they are under the target of surveillance. To achieve those goals, the spread spectrum based traceback technique was initiated in [4]. In this approach, spread spectrum (SS) is a transmission technique that uses a pseudo-noise (PN) code, independent of the original data signal, to "spread" the signal in the data transmission. On reception, the signal is recovered ("despread") by making use of the same PN code. Spread spectrum techniques are resistant to interference and interception. For example, Yu *et al.* [4] proposed a direct sequence spread spectrum (DSSS) based traceback technique, showing the strengths of spread spectrum approaches to conduct network traceback. In this developed traceback technique, investigators can modulate a suspect's traffic flow rate using a secret PN code. The moderately changed traffic rate does not show obvious regularity (e.g., the periodic pattern [5]). Although the DSSS-based traceback technique in [4] has the above benefits, we have determined that it is vulnerable to detection and cannot preserve traceback secrecy against attacks even when cryptographically secure PN codes are used [6].

In this paper, we investigate a class of hopping-based spread spectrum techniques for network forensic traceback, which preserves spread spectrum accuracy while providing a greater degree of secrecy. We first provide a generic framework for applying the hopping technique into spread spectrum, which is robust, accurate, and covert to secretly trace illegal cyber activities. Under this framework, we then develop three new hopping based spread spectrum traceback techniques, including *Code Hopping-DSSS* (*CH-DSSS*), *Frequency Hopping DSSS* (*FH-DSSS*), and *Time Hopping DSSS* (*TH-DSSS*). Our developed techniques are inherently less prone to interception and detection by a third party. Traceback schemes based on these hopping techniques can randomize the traffic pattern in code, frequency, or time domain to a markedly greater extent than DSSS alone.

We also investigate the secrecy and effectiveness of our proposed hopping-based spread spectrum forensic traceback techniques. For the secrecy, we first consider one recently discovered threat [6] that detects DSSS marks using *determination coefficient* of traffic rate on the marked flow. We then investigate the secrecy of our proposed techniques over the other attacks based on the traffic aggregation of multiple marked flows. We also formalize the traceback as a communication channel and derive the capacity. Through theoretical analysis, extensive simulations, and real-world experiments, we demonstrate the use of our developed new techniques, in their capacity of preserving network forensic traceback secrecy against cyber crimes along with maintaining a high traceback accuracy.

The rest of the paper is organized as follows. We give the background and related work in Section 2. In Section 3, we present a class of hopping-based spread spectrum techniques for network forensic traceback. In Section 4, we investigate the secrecy and efficiency of our investigated techniques. In Section 5 and Section 6, we use ns-2 simulation and

experiments over *Tor* to validate the effectiveness of our proposed techniques, respectively. We conclude the paper in Section 7.

2. Background and Related Work

In this section, we first give an overview of a *mix network*. We then introduce the spread spectrum based flow marking for network forensic traceback, followed by literature review. Mix networks [7] have been widely used by anonymous communication systems, including *Tor* as shown in Figure 1. In a mix network, a sender transmits data packets through a series of mixes to a receiver, in which a mix manipulates the packet delivery to prevent the traffic analysis through correlating input traffic and output traffic of mix networks.

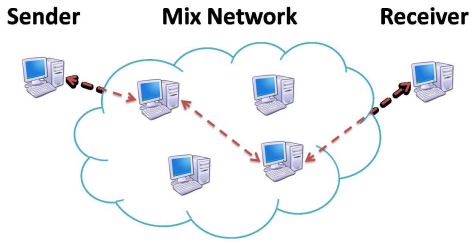

Figure 1: An Example of Mix Network

Using the spread spectrum, in our previous work [4], we developed a new traffic flow marking approach that can carry out network forensic traceback. Using Figure 1 as an example, the basic idea is illustrated as follows. To confirm the communication between a sender and a receiver who communicate with each other over the mix network, the investigator first selects a secret signal, in which each bit of a signal is spread by a secret PN code. Through interfering with the outbound traffic from the sender and manipulating the traffic rate based on the pattern defined by spread signal, the investigator then embeds secret marks into the traffic and sniffs the inbound traffic at the receiver. From the sniffed inbound traffic at the receiver, the investigator extracts the embedded marks and confirm the communication relationship.

The basic principle of spread spectrum is illustrated as shown in Figures 2 and 3 [8]. As we can see, in the spreading process, the original signal d_t at the transmitter is a series of binary symbols (e.g., $\{1\ -1\}$). We we use bits encoded as $+1$ or -1 instead of 1 or 0), although the signal can be encoded by other schemes such as QPSK (Quadrature Phase Shift Keying) [9]. With a PN code $c_t = \{1\ 1\ 1\ -1\ 1\ -1\ -1\}$, the spread signal becomes $t_b = \{1\ 1\ 1\ -1\ 1\ -1\ -1\ +1\ -1\ 1\ 1\}$, which will be in the input to the despreading process. By using the same PN code to conduct correlation, the $d_t = \{1\ -1\}$ is obtained, which is the same as d_t.

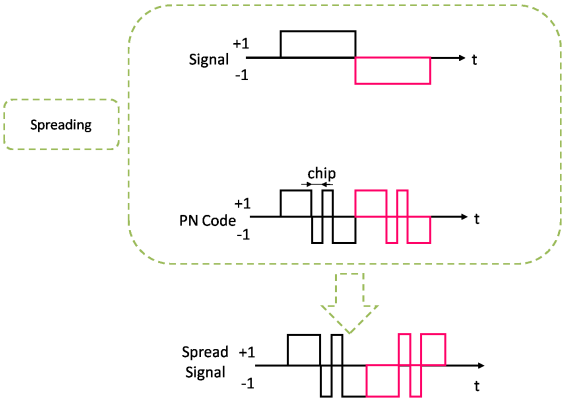

Figure 2: An Example of Spreading in DSSS

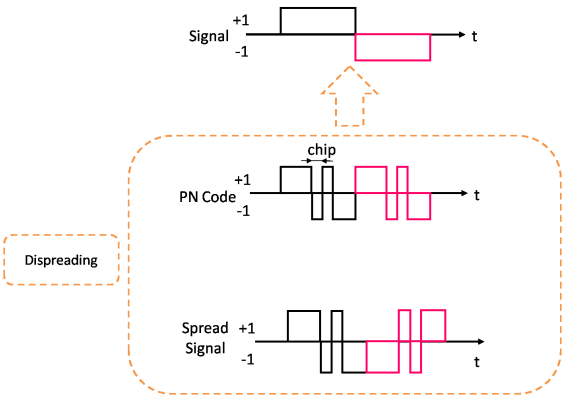

Figure 3: An Example of Despreading in DSSS

Owning to the secrecy of PN code, only those who know the PN code can correctly recover the original signal and identify the communication relationship. The PN code modulated signal also appears as innocent noise in both time and frequency

domains, so it is difficult for other to detect the presence of such a signal in the host traffic. As such, using a spread spectrum technique, anonymous communication can be traced while evading detection by suspects. Nonetheless, one discovered sophisticated traffic analysis attack proposed in [6] shows that traceback using PN code is still visible to the suspects if a *determination coefficient* is used. In Section 3, we will demonstrate a class of hopping-based spread spectrum techniques for network forensic traceback, which can effectively defend against those sophisticated attacks.

There has been a number of traffic analysis attacks against anonymous communication through mix networks in the past. The existing research showed that the private information can be leaked from encrypted and anonymized network traffic by examining the patterns of traffic rate, packet size, and timing [4,10,11,12,13,14,15]. One type of traffic analysis techniques is to record the traffic and identify the similar pattern in the traffic between sender and receiver [10,11]. As an example, Levine *et al.* [11] relies on a cross correlation technique to compute the similarity in the traffic between sender and receiver. Another type of traffic analysis techniques is to embed specific secret signal (or marks) into the target traffic [3,4,16,17,18]. For examples, Yu *et al.* [4] developed a flow marking scheme to accurately confirm the communication relationship through mix networks. Wang *et al.* [3] investigated a timing-based watermarking scheme to confirm the caller and callee parties using the encrypted peer-to-peer voice over IP traffic flow.

3. Techniques

In this section, we first present the design of a class of hopping-based spread spectrum techniques for forensic traceback, which can achieve both forensic traceback accuracy and secrecy. We then introduce several techniques, including Code Hopping-DSSS (CH-DSSS), Frequency Hopping-DSSS (FH-DSSS), and Time Hopping-DSSS.

3.1. Basic Idea

Figure 4 illustrates the general framework for the hopping-based spread spectrum for the network forensic traceback. The basic idea is illustrated as follows. The original signal x and a PN code are selected in the same way as [4]. Then, a *hopping* component directed by the *Hopping Control Code* (HCC) is used to randomize the PN code. At the *transmitter*, each bit of a signal is spread through a random sequence controlled by HCC and then the spread signal is used to modulate the traffic characteristics (e.g., traffic rate, timing, and packet size) and the signal is embedded into the target traffic. Note that using the same PN code and original signal x, the actual spread sequence X used to modulate the traffic flow will be varied owning to the randomization introduced by HCC. At the *receiver*, the spread signal is extracted from the target traffic flow by a digital filter and the same HCC and PN codes are used to despread and retrieve the original signal x. If the original signal x can be recovered at the receiver, the communication relationship is confirmed.

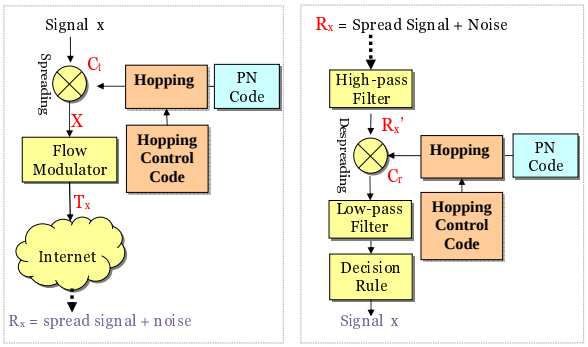

Figure 4: A Framework of Hopping-Based Spread Spectrum for Network Forensic Traceback

The secrecy of traceback refers to the difficulty of detecting the forensic traceback by anyone other than investigators. It is also an important goal for any forensic traceback technique. As shown in [4], the PN code itself, original signal length, and chip duration, PN code length, and mark amplitude, impact how well traceback can be performed. In addition to these, our hopping-based spread spectrum approach

provides the following mechanisms for preserving a better secrecy: (i) *Secrecy of HCC*. The secrecy of HCC makes the marks secret. The randomization introduced by HCC provides increased flexibility to make the marks undetectable to suspects. Because the suspects do not know HCC, it is very difficult for them to recognize the existence of marks in the traffic. (ii) The marks generated by the hopping-based spread spectrum show a white noise-like pattern in both time and frequency domains. The modulated traffic appears random for those who do not know both HCC and PN codes. Our investigated hopping-based traceback approach leverages the benefits of DSSS technique as well. As indicated in [4], by selecting a carefully chosen mark amplitude, which represents the strength of embedding signal in comparison with the strength of host traffic, can be very small in comparison with noise so that the DSSS mark is masked. The recognition process will still effectively restore the spread signal to its narrow band and recover the original signal from the noise.

3.1.1. Code Hopping-Direct Sequence Spread Spectrum (CH-DSSS)

In CH-DSSS, we use multiple codes to spread different signal bits at the transmitter. That is, the hopping module at the transmitter in Figure 4 selects from multiple PN codes to spread each signal bit. This selection sequence is directed by the HCC. To recover the signal, the investigator at the receiver must use the same selection sequence of PN codes to conduct despreading based on HCC.

As shown in Figure 5, we use the following simple example to show the property of this scheme, which makes marks randomly embedded in the traffic flow. For example, investigators use code $\vec{C_0} = \{c_0, \cdots, c_4\}$ to spread a signal bit x_0 (-1 or 1) and code $\vec{C_1} = \{c'_0, \cdots, c'_4\}$ to spread signal bit x_1. If $\vec{C_0}$ and $\vec{C_1}$ are orthogonal PN codes, i.e., $\vec{C_0} \cdot \vec{C_1} = 0$, then the *dot* product of two spread bits $\vec{C_0}x_0 \cdot \vec{C_1}x_1 = 0$. As an ideal case, we use a different orthogonal code for each signal bit. One limitation is that if a too short code is used, the number of orthogonal codes in the same length is inadequate. To overcome this, we can use codes with variable lengths. Another way to address the limited number of orthogonal codes is to collect a number of codes in a pool, and then generate a pseudorandom sequence of selections from the pool to modulate the outgoing traffic. The receiver uses the same sequence of codes from the pool to demodulate the traffic.

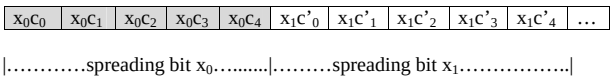

Figure 5: Code Hopping-Direct Sequence Spread Spectrum (CH-DSSS)

3.1.2. Frequency Hopping-Direct Sequence Spread Spectrum (FH-DSSS)

The frequency hopping spread spectrum (FHSS) is also a popular spread technique in the wireless communication. In FHSS, signal bits are modulated and carried by different frequency bands or channels. As shown in Figure 6, after communicating on one channel for a predefined small amount of time, the *dwell time*, the transmitter and receiver switch to another channel. As time goes, the synchronized transmitter and receiver communicate on a series of channels, which is known as the *hopping sequence*. The hopping sequence is controlled by a pseudorandom number sequence. In general, a channel from the hopping sequence comes from a set, a limited number of channels/frequencies used by the communication system. The technique of FHSS can effectively escape detection [19]. The output power of FHSS signals is spread over a large bandwidth and the spectrum has a very low power spectral density. The low spectral density may not even be recognized as valid communication but instead appears to be noise.

We leverage FHSS to carry out network traceback in the following way. At the transmitter, the hopping module will vary the chip duration of a PN code, which is used for spreading a signal bit. The chip duration is generated by HCC, in which each different chip duration corresponds to a different frequency. The spreading process is equivalent to modulating each signal bit to a different frequency channel. Consider the example illustrated in Figure 7: a signal bit x_0 can be spread by a PN code $\vec{C}$ with a

chip duration t_0, denoted as $\vec{C}_0$, and another signal bit x_1 can be spread by the same PN code $\vec{C}$ with a different chip duration t_1, denoted as $\vec{C}_1$. Hence, x_0 is analogous to transmitting at a frequency channel around $f_0 = \frac{1}{t_0}$hz and x_1 at a frequency around $f_1 = \frac{1}{t_1}$hz. If an m-sequence code is used as the PN code, $\vec{C}_0 x_0 \cdot \vec{C}_1 x_1 \approx 0$, because an m-sequence code's autocorrelation approaches zero for lags not equal to zero. To recover the signal at the receiver, the investigator uses the same PN code while varying the chip frequencies according to HCC to despread the signal as shown in Figure 4.

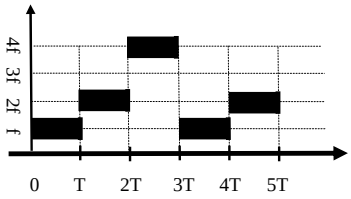

Figure 6: Principle of Frequency Hopping Spread Spectrum (FHSS)

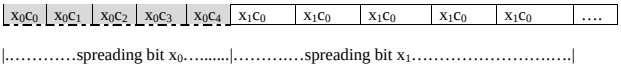

Figure 7: Frequency Hopping-Direct Sequence Spread Spectrum (FH-DSSS)

As shown in Figure 7, for signal bits x_0, $\cdots$, x_{w-1}, we use a PN code with chip frequency f_0 to modulate x_0, the same PN code with chip frequency f_1 to modulate x_1, and so on. In this way, the ordered set of frequencies f_0, $\cdots$, f_{w-1} is the hopping sequence (HCC). To recover the signal at the receiver, an investigator can use the same sequence of frequencies controlled by the same HCC for carrying out despreading as shown in Figure 4. Theoretically, the hopping sequence can be infinite. In practice, we need to consider the limitations of flow duration. A low chip frequency implies a long chip duration, and requires a long interfering session to embed the spread signal into the target traffic. A more general form of FH-DSSS will vary the duration for each and every chip of PN code, rather than just varying chip durations at signal bit boundaries. This approach was actually used for our experiments.

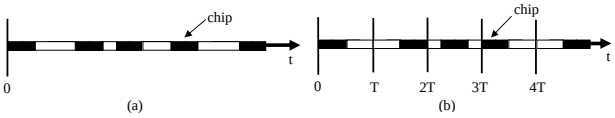

Figure 8: Principle of Time Hopping Spread Spectrum (THSS)

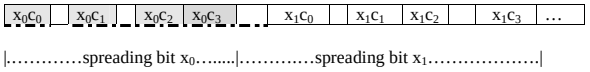

Figure 9: Time Hopping-Direct Sequence Spread Spectrum (TH-DSSS)

3.1.3. Time Hopping-Direct Sequence Spread Spectrum (TH-DSSS)

The time hopping spread spectrum (THSS) can also be used to reduce the probability of interception and recognition [19]. Figure 8 illustrates the principle of THSS. For THSS, there is non-zero intervals between chips within a PN code. The duration of those intervals are varied according to the pseudo random control code (i.e., HCC). We implement THSS for network traceback in the following way. At the transmitter in Figure 4, the hopping module will vary a PN code's inter-chip intervals, which is under the control of HCC. This spreading process spreads signal bits through time. Hence, PN codes used to spread each signal bit are actually different. As shown in Figure 9, a chip is transmitted in a random position relative to a period T. The actual position of the chip within each interval T is actually determined by HCC. A variant of THSS is illustrated in Figure 9. In this approach, a chip is transmitted in a random position relative to a period T. The actual position of the chip within each interval T is determined by HCC. To recover the signal at the receiver, an investigator can use the same sequence of PN codes with a corresponding variation of inter-chip intervals controlled by the same HCC

for despreading as shown in Figure 4. As we can see, the essential differences between FH-DSSS and TH-DSSS is that the TH-DSSS spreads signal bits through time rather than frequency.

4. Analysis

In this section, we investigate the secrecy and accuracy of our developed techniques. For traceback secrecy, we first consider one recently discovered the attack based on the self-similarity traffic analysis [6]. We then investigate the secrecy of our techniques against the attack based on multiple-flow traffic analysis [18]. Finally, for the efficiency of forensic traceback, we formalize the forensic traceback as a communication channel and derive the capacity of the channel.

4.1. Traceback Secrecy

4.1.1. Secrecy against Self-Similarity Traffic Analysis Attack

In [6], the deficiency of the DSSS modulated mechanism was investigated. In particular, this work demonstrated that although it is hard to recognize the existence of PN code modulated traffic from the pattern in time and frequency domains, other features may disclose the existence of marks. The feature studied in [6] is to use the *determination coefficient*, which will be explained in the following. This feature is to measure the similarity of a PN code modulated traffic segment and a time-shifted version of the same segment. If the same PN code is used to spread each bit of a signal on a traffic flow as used in [4], there is a way to synchronize a time-shifted segment of the traffic with the original segment. Hence, the PN code will reinforce rather than cancel and an observable phenomenon will arise.

As shown in [6], the *correlation coefficient r* measures the strength of similarity between two random variables and is defined in

$$r = \frac{\sum_{x,y}(x-\bar{x})(y-\bar{y})}{\sqrt{\sum_{x}(x-\bar{x})^2}\sqrt{\sum_{y}(y-\bar{y})^2}} = \frac{\sum_{x,y}(x-\bar{x})(y-\bar{y})}{\sigma_x\sigma_y}. \quad (1)$$

Denote $\vec{x} = \{x_0, \cdots, x_{w-1}\}$ as the signal, a series of bits, where the number of bits w is *window* size. Hence, a window contains w complete bits. Denote a PN code as $\vec{C} = \{\ c_0, \cdots, c_{l-1}\}$, where l is the code length. $r^2(\tau)$ is the *determination coefficient* of spread signal $\vec{X}$ and a time-shifted $\vec{X}$ with lag τ. From the results of Theorem 1 in [6], $r^2(\tau)$ demonstrates a *periodicity* with τ ($0 \leqslant \tau < wl$) as

$$E\{r^2(\tau)\} \approx \begin{cases} A^4, & \tau = 0, \\ \frac{A^4}{w-k}, & \tau = kl, 0 < k < w, \\ 0, & \tau \neq kl, 0 < k < w. \end{cases} \quad (2)$$

The result explains the reason why the PN code modulated traffic can be detected as illustrated in Figure 10 [6]. The code in the time-shifted traffic can synchronize with the one in the original traffic, leading to *periodic peaks* in the *determination coefficient*, which implies the *self-similarity* of embedded DSSS marks at regular intervals. Hence, the adversary can infer the code length l based on the periodicity of $r^2(\tau)$. These distinguishing properties provide features of DSSS marks and enable the detection by suspects.

According to [6], PN code modulated traffic: DSSS-based traceback fails to preserve secrecy owning to a single PN code is used to spread each bit of the signal. This leads to self-similarity behavior with a period corresponding to the PN code length in the modulated traffic. Because of the randomization introduced by HCC as shown in Section 3, we now show that our proposed hopping-based spread spectrum techniques can effectively render such self-similarity analysis ineffective.

Recall that in CH-DSSS, to recover the signal, the investigators will use the same selection sequence of PN codes determined by the HCC for despreading. CH-DSSS preserves traceback secrecy as illustrated in Figure 11. The investigators use code $\vec{C_0} = \{c_0, \cdots, c_4\}$ to spread signal bit x_0 (-1 or 1) and code $\vec{C_1} = \{c'_0, \cdots, c'_4\}$ to spread signal bit x_1. If $\vec{C_0}$ and $\vec{C_1}$ are orthogonal PN codes, i.e., $\vec{C_0} \cdot \vec{C_1} = 0$, then the *dot* product of two spread bits $\vec{C_0}x_0 \cdot \vec{C_1}x_1 = 0$. Hence, no self-similarity will be detected in the spread signal, the *determination coefficient* will not show a periodic pattern, and the se-

x_0c_0	x_0c_1	x_0c_2	x_0c_3	x_0c_4	x_1c_0	x_1c_1	x_1c_2	x_1c_3	x_1c_4					
					x_0c_0	x_0c_1	x_0c_2	x_0c_3	x_0c_4	x_1c_0	x_1c_1	x_1c_2	x_1c_3	x_1c_4

Figure 10: Self-similarity of PN code Modulated Traffic

x_0c_0	x_0c_1	x_0c_2	x_0c_3	x_0c_4	x_1c_0'	x_1c_1'	x_1c_2'	x_1c_3'	x_1c_4'					
					x_0c_0	x_0c_1	x_0c_2	x_0c_3	x_0c_4	x_1c_0'	x_1c_1'	x_1c_2'	x_1c_3'	x_1c_4'

Figure 11: Secrecy of CH-DSSS

crecy of CH-DSSS traceback can be preserved.

In FH-DSSS, the HCC randomizes the duration for transmitting each original signal. Consider the example illustrated in Figure 12: a signal bit x_0 can be spread by a PN code $\vec{C}$ with chip duration t_0, denoted as $\vec{C}_0$, and another signal bit x_1 can be spread by the same PN code $\vec{C}$ with a different chip duration t_1, denoted as $\vec{C}_1$. Hence, x_0 is analogous to transmitting at a frequency channel around $f_0 = \frac{1}{t_0}$hz and x_1 at a frequency around $f_1 = \frac{1}{t_1}$hz. If an m-sequence code is used as the PN code, $\vec{C}_0 x_0 \cdot \vec{C}_1 x_1 \approx 0$, because an m-sequence code's autocorrelation approaches zero for lags not equal to zero. In the context determined by the HCC, no self-synchronization will occur when the lag is not zero, since signals which otherwise might synchronize, now occur at different frequencies. Attempting to detect FH-DSSS marks using the determination coefficient will fail as well.

In TH-DSSS, the HCC randomizes the inter-chip intervals. Figure 13 shows the marks on a flow and its time-shift version for the self-similarity analysis used by the attack. From this example, we can see that the marks corresponding to the first bit of original signal and marks corresponding to the second bit of original signal can not be synchronized due to the different inter-chip intervals. Hence, no self-similarity will be detected in the spread signal, the *determination coefficient* will not show any periodic pattern, and the secrecy of TH-DSSS approach will be preserved.

4.1.2. Secrecy over Multi-flow Attack

In [18], Kiyavash *et al.* introduced a multi-flow attack that is capable of detecting interval-based marks [3,20]. This approach can be also used to detect DSSS marks [4]. The key idea of this approach is to let the adversary to learn the information of marks by observing a number of marked flows. Regardless of whether marking schemes are implemented, it is possible that the adversary can correlate and synchronize the marks on different flows. As a result, the adversary can synchronize marks of different flows and average them, exposing a highly visible traffic rate drop within some intervals. Nonetheless, given so many flows over the Internet, it is not always easy to find a relatively large number of flows embedded with DSSS marks. Our hopping-based spread spectrum techniques can maximally randomize the marks on different flows and can significantly reduce the probability for the adversary to synchronize the marks on different flows and identify the highly visible traffic rate drop on aggregated traffic rate in some intervals.

Recall that in TH-DSSS, we introduce the HCC component to adapt the PN code as directed by a HCC. The output of HCC will be used to randomize each bit of a spread signal. For a multi-flow attack, we know that the adversary intends to average the rate of multiple flows. Same as [4], in TH-DSSS, when the spread signal bit is -1, the strong interference traffic will be launched to reduce the host flow rate.

Denote n as the total number of flows embedded with marks, $x = \{x_0, \cdots, x_{w-1}\}$ as the original signal, a series of bits, where w is the length of original signal, $C = \{c_0, \cdots, c_{l-1}\}$ as the PN code, where l is the code length. We assume that each flow lasts for mT_c, where $m \geqslant lw$ and T_c is the chip duration. Theorem 1 shows the average number of flows, in which marks modulated by -1 bit are synchronized by the

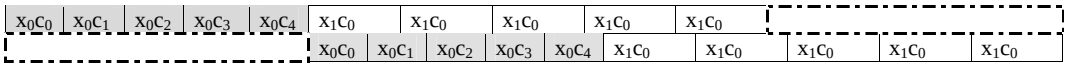

Figure 12: Secrecy of FH-DSSS

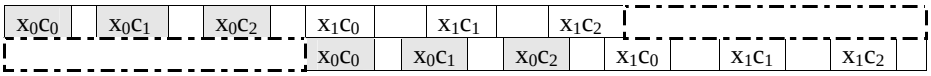

Figure 13: Secrecy of TH-DSSS

multi-flow attack. The detailed proof this theorem can be found in Appendix A.

Theorem 1. *Within n flows with embedded marks, the average number of flows embedded with marks modulated by the synchronized* -1 *bit is*

$$s = \sum_{k=n}^{1} [kC_n^{n-k} p^j (1-p)^{n-k}], \tag{3}$$

where $p = \frac{wl}{2m}$ *and n is the total number of flows.*

We now illustrate the results with practical examples. We choose the chip duration $T_s = 0.3$ second, original signal length $w = 7$, and PN code length $l = 7$. We also assume that the flow lasts for $T = 200T_s = 60$ seconds. When the total number of flows is $n = 30$, and the average number of flows embedded with marks modulated by a synchronized -1 bit is 6.5. Using a simple case with a 7 bits PN code as an example, from the analysis in [4], to achieve 95% detection rate, $A = 0.25\sigma$, where σ is the background noise introduced by the interference from other network flows. For a multi-flow attack, to achieve a desired low false positive rate (i.e., 2%), we assume its detection threshold $T_r = 3\sigma$. For the detection of TH-DSSS, the detection signal strength becomes $6.5 * A = 6.5 * 0.25 * \sigma = 1.65\sigma < T_r = 3\sigma$. Hence, the multi-flow attack becomes ineffective to detect marks masked by TH-DSSS. Owning to the same reason, FH-DSSS and CH-DSSS make the multi-flow attack ineffective as well.

4.2. Traceback Effectiveness

In the following, we analyze the capacity of hopping-based forensic traceback techniques. As we showed in Section 4.1, the hopping-based spread spectrum techniques can preserve the forensic traceback secrecy and render the attack based on the self-similarity analysis and other attack ineffective. However, for real-world forensic practice, the effective bit rate of a forensic traceback technique is also critical. Recall that the hopping-based spread spectrum techniques leverage the host traffic from the suspect sender to the receiver as a covert channel to transmit marks. Using the concept of channel capacity in communication theory, we can derive the efficiency of our approach. Channel capacity defined by Shannon provides a theoretical bound for measuring the information transmission capability over a noisy channel [21]. This theory provides the theoretical bound for communication system research and the design of coding schemes to increase the resistance of digital communication to channel noise. The purpose of channel coding is to minimize the overall effect of channel noise on the system. Theorem 2 provides a closed formula for capacity of hopping-based spread spectrum techniques. The detail proof this theorem can be found in Appendix B.

Theorem 2. *The channel capacity for different hopping-based spread spectrum techniques for network forensic traceback can be estimated by,*

$$C_t = \frac{\log_2(1 + \frac{A^2}{\delta^2})}{2E\{\mathscr{T}\}}, \tag{4}$$

where A^2 *is signal power density and* δ^2 *the noise variance.* $E\{\mathscr{T}\}$ *is the mean time required for transmitting one signal bit and can be derived as follows*

for different spread spectrum techniques

$$E\{\mathscr{T}\} = \begin{cases} lT_c, & DSSS, \\ E\{l\}T_c & CH\text{-}DSSS, \\ lE\{\mathscr{T}_c\}, & FH\text{-}DSSS, \\ l(T_c + E\{I\}), & TH\text{-}DSSS, \end{cases} \quad (5)$$

where l is the code length for spreading one signal bit using DSSS without hopping, $E\{l\}$ is the mean of the PN code length in CH-DSSS. T_c is a constant chip duration for DSSS, CH-DSSS and TH-DSSS, $E\{\mathscr{T}_c\}$ is the mean chip duration for FH-DSSS, and $E\{I\}$ is the mean inter-chip interval for TH-DSSS.

Assume that the signal to noise ratio SNR, A^2/δ^2, is the same for the four spread spectrum traceback techniques. There are a few observations from Theorem 2. Here the signal to noise ratio SNR, A^2/δ^2, is the same for the DSSS and TH-DSSS.

- A general intuition is that the mean transmission time for one bit will be longer for the TH-DSSS based approach. The FH-DSSS based approach will also be longer since the $E\{\mathscr{T}_c\}$ will inevitably be longer than the minimal chip duration $\mathscr{T}_c$ used in DSSS and CH-DSSS to achieve the same accuracy. The channel capacities of DSSS and CH-DSSS are generally greater than the channel capacities of FH-DSSS and TH-DSSS. In the later two approaches, some channel capacity is sacrificed for increased secrecy. Hence, there is a tradeoff between accuracy and secrecy in the three hopping-based spread spectrum traceback techniques. Please refer to Sections 5 and 6 for evaluation results. In general, all hopping-based spread spectrum techniques sacrifice the channel capacity for secrecy.
- The channel capacity increases as the chip amplitude increases. This allows us to transmit embedded signals at a high rate to trace the suspect traffic. However, a large amplitude will make the embedded signal protrude out of the host traffic, which is modulated to carry the embedded signal. Hence, there is a tradeoff between the capacity and invisibility.
- It looks as if we reduce the chip duration, thus $E\{\mathscr{T}\}$, the channel capacity will increase. This is not always true in reality. As we know, the chip duration is controlled by the interference. When interference is applied to the target traffic, it takes time for the target traffic to respond. When the interference starts, there is a transition time during which the traffic rate reduces. When the interference is turned off, there is a transition time during which the traffic rate increases. Especially, Transmission Control Protocol (TCP) uses loop control mechanism and the rate varies with the path's response to the interference. Hence, the average chip amplitude A is influenced by the time taken for the interference is applied, i.e, the chip duration. Moreover, because an average traffic rate is used, the noise variance changes as the sampling interval is adjusted. We expect a complicated relationship between C_t, A and $E\{\mathscr{T}\}$, depending on the interfering process and traffic flow path status.

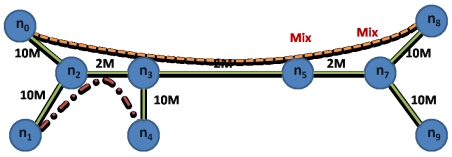

Figure 14: Topology in ns-2

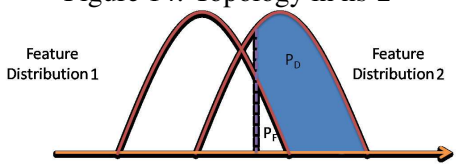

Figure 15: Calculation of P_D and P_F

5. Performance Evaluation over ns-2 Simulation

We have studied the secrecy and efficiency of our investigated hopping-based spread spectrum techniques for network forensic traceback. In this section, we use ns-2 simulator to validate the effectiveness of the investigated techniques in terms of accuracy and secrecy. Results of empirical evaluation on Tor will be presented in Section 6. Notice that we have conducted a large number of simulations and only limited results are shown in this section due to space limitation. In the following, we first present

the simulation methodology and then demonstrate the simulation results.

5.1. Simulation Methodology

Figure 14 shows the simulation topology, where n_5 and n_7 are Tor-like mixes. In our experiment, we do not consider batching or reordering mechanisms for the reason similar to [4]. The file Transfer Protocol (FTP) flow from node n_0 to node n_8 is used as a target traffic flow through the simulations, together with cross flows as noise traffic. In our simulations, the *interferer* uses UDP (User Datagram Protocol) CBR (Constant Bit Rate) traffic to modulate the target file FTP flow. The CBR traffic is an on-off traffic source and transmitted from n_1 to n_4. The CBR flow shares the link between n_2 and n_3 with the target file FTP flow. Owning to TCP flow control, when the CBR traffic rate increases, the FTP traffic rate decreases. In our simulation, the CBR traffic used for interference is turned off when a chip within a signal modulated by the PN code is $+1$ and it is turned on when the chip is -1. In this way, we mark the interested file FTP flow by manipulating its rate through the interference of the CBR traffic.

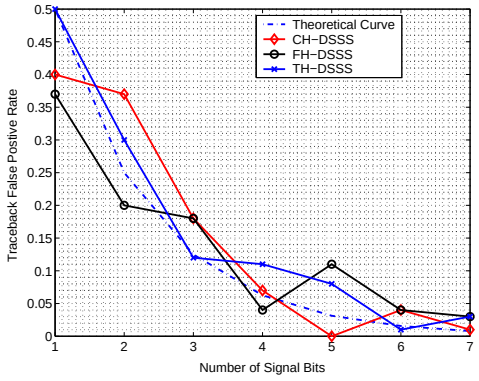

Figure 16: False Positive Rate

To measure the accuracy of traceback, we use the traceback successful rate and traceback false positive rate. The traceback successful rate refers to the probability that marks are correctly recognized. The traceback false positive rate refers to the probability that marks are mistakenly recognized in the scenario, in which spread signal is not embedded into the target traffic flow. For measuring the secrecy of traceback, we use a detection rate P_D and a false positive rate P_F as our evaluation metrics, which are illustrated in Figure 15. To displaying the relationship between P_D and P_F, we use the *Receiver Operating Characteristic* (*ROC*) curve, which is a plot of P_D versus P_F. In reality, when an adversary tries to detect traffic containing DSSS marks, he wants a high detection rate and a low false positive rate. For the secrecy of forensic traceback, the ideal scenario is that the false positive rate is as high as the detection rate.

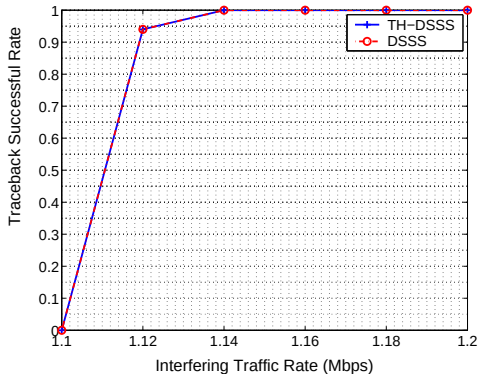

Figure 17: ROC Showing Accuracy of Time Hopping-based Spread Spectrum Techniques for Forensic Traceback

5.2. Results

Figure 16 shows the forensic traceback false positive rate when we try to recognize a signal from traffic where no marks exist (sender and receiver are not communicating). In our simulation, we vary signal length from 1 to 7, and for each fixed signal length, we measure the false positive rates for codes of different lengths (from 2 to 7). The false positive rate for each signal length is computed as the average of the "detection rate" for the different code lengths tested with that signal. From Figure 16, we can see that the false positive rate decreases exponentially with the increasing signal length. The theoretical curve (from the result in [4]) matches the empirical curve very well.

Figure 17 illustrates the forensic traceback accuracy (recognizing marks by investigators) of the

three new forensic traceback techniques: CH-DSSS, FH-DSSS and TH-DSSS. For the experiments on FH-DSSS, the hop set is $\{\frac{1}{0.5}hz, \frac{1}{0.6}hz, \cdots, \frac{1}{1.0}hz, \}$. For TH-DSSS, we use the scheme in Figure 8(b) and the pulse refers to one spread bit. The interval between two pulses is between $[1s, 5s]$. It can be observed that with the three new techniques, when the interfering traffic rate is high enough, the detection rate is 100%, while the false positive rate is also low in our simulation, consistent with results from [4]. However, there exists difference of detection rate by the three approaches.

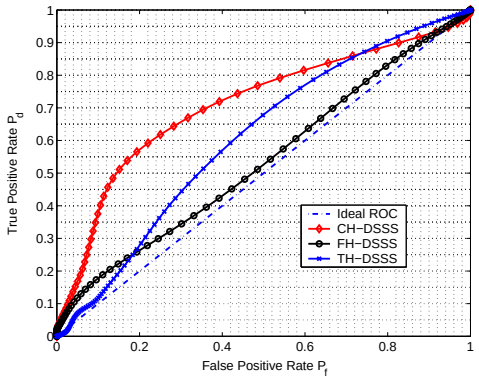

Figure 18: Hopping-based Spread Spectrum Techniques Preserving Forensic Traceback Secrecy

Figure 18 illustrates the results of CH/FH/TH-DSSS for preserving forensic traceback secrecy against the attack based on the self-similarity analysis. We use two m-sequence codes of length 7 to randomly modulate the signal, the windows size is 3 and the number of segments in each window is 2. We can see that the empirical ROC curve approaches the ideal ROC curve for the secrecy: as the detection rate increases, the false positive rate appears approximately linearly. For example, given a small false positive rate (i.e., 5%), the detection rate is also very small (i.e., 25%). This renders the detection results untrustworthy and the secrecy of forensic traceback is preserved. We also observe that the simulations for the CH-DSSS, FH-DSSS and TH-DSSS based forensic traceback have similar observations. From Figure 17 and Figure 18, we can see that there are tradeoffs on accuracy and secrecy among the three hopping-based spread spectrum forensic traceback techniques.

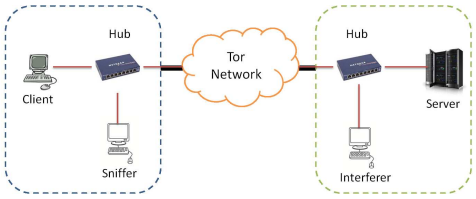

Figure 19: Experiment Setup

6. Experiments over Tor

To validate the effectiveness of our proposed schemes, we also conducted real-world experiments over *Tor* [1], a popular anonymous communication system. The experimental setup is shown in Figure 19, which represents a typical use of *Tor* for carrying out anonymous file transfer or web browsing activities. All computers are configured with Fedora Core 3. We download a file from a web server on a campus to an off-campus computer, as a client. The downloading software is the command line utility *wget*, which has an appropriate proxy configuration for using Tor. To carry out the network forensic traceback, we set up two more computers. One computer is used as an *interferer* that sends an appropriate volume of traffic to the server and the other computer is used as a *sniffer* to collect the traffic destined for the client computer. The *interferer* and server are connected through hubs, as were the *sniffer* and the client computer.

Figure 20 as shown in [6] illustrates detecting DSSS mark when hopping marks are not used. The upper figure illustrates the traffic rate varying with time. The lower figure illustrates the periodicity that shows the determination coefficient of traffic segments from the upper data set. The parameters of this DSSS mark detection experiment include the chip duration, $t_c = 3$ seconds and the code length, $l = 7$. Hence, the bit duration is 21 seconds as used in [4]. From Figure 20, we observe that the *determination coefficient* shows a periodicity at positions of multiple 21 seconds, showing that DSSS marks can be detected blindly.

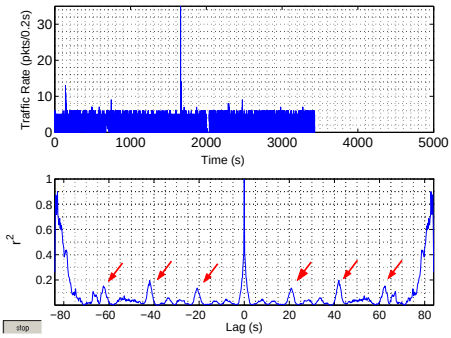

Figure 20: Detecting DSSS Marks Using Single-Flow Attack over Tor

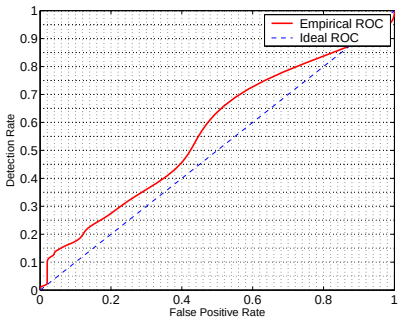

Figure 21: Secrecy of TH-DSSS on Self-Similarity Based Attack over Tor

To validate the secrecy of CH-DSSS, we use the same modulation approach, randomly modulating the time to insert marks on traffic flows, as for simulations in Section 5. Figure 21 shows the ROC curve. We can see that the empirical ROC curve approaches the ideal ROC curve for secrecy. As the detection rate increases, the false positive rate also linearly increases. For example, given a small false positive rate (i.e., 5%), the detection rate is also very small (i.e., 15%). This renders the adversary's decision rule void and the secrecy of the traceback can be preserved. The experiments of FH-DSSS and TH-DSSS over Tor have similar observations.

7. Conclusion

In this paper, we investigated a class of hopping-based spread-spectrum techniques for network forensic traceback that fully utilize the benefits of the spread spectrum approach while preserving a greater degree of secrecy. Our developed techniques are accurate, robust, and difficult to detect, and can be extended to other applications. Because our approaches can spread signal bits through code, time and frequency domains to a markedly greater extent and make it not fall prey to the statistical traffic analysis attack. We investigated the secrecy of those techniques against two known traffic analysis threats and showed the effectiveness of those techniques. Our theoretical analysis, extensive simulations, and real-world experiments validate that our investigated hopping-based spread spectrum techniques for network forensic traceback will support network surveillance and deter cyber crime accurately and secretly. We believe that this paper lays a solid foundation for further studies of forensic traceback schemes.

1. R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
2. Anonymizer, Inc., "Anonymizer," `http://www.anonymizer.com/`, 2007.
3. X. Wang, S. Chen, and S. Jajodia, "Tracking anonymous peer-to-peer voip calls on the internet," in *Proceedings of the 12th ACM Conference on Computer Communications Security (CCS)*, November 2005.
4. W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, May 2007.
5. X. Fu, Y. Zhu, B. Graham, R. Bettati, and W. Zhao, "On flow marking attacks in wireless anonymous communication networks," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, April 2005.
6. W. J. Jia, F. P. Tso, Z. Ling, X. Fu, D. Xuan, and W. Yu, "Blind detection of spread spectrum flow watermarks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, April 2009.
7. Wikipedia, "Mix network," `http://en.wikipedia.org/wiki/Mix_network`, 2012.
8. ir.J.Meel, "Spread spectrum (ss) - introduction," `http://www.sss-mag.com/pdf/Ss_jme_denayer_intro_print.pdf`, 1999.
9. T. F. Wong, "Spread spectrum and code division multiple access," `http://wireless.ece.ufl.edu/~twong/notes1.html`, August 2000.

10. Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Proceedings of Workshop on Privacy Enhancing Technologies (PET)*, May 2004.
11. B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix-based systems," in *Proceedings of Financial Cryptography (FC)*, February 2004.
12. D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *Proceedings of 10th USENIX Security Symposium*, August 2001.
13. Q. X. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. L. Qiu, "Statistical identification of encrypted web browsing traffic," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, May 2002.
14. M. Liberatore and B. N. Levine, "Inferring the Source of Encrypted HTTP Connections," in *Proceedings of the ACM conference on Computer and Communication Security (CCS)*, October 2006.
15. C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob?," in *Proceedings of the 16th Annual USENIX Security Symposium (Security)*, August 2007.
16. X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of inter-packet delays," in *Proceedings of the 2003 ACM Conference on Computer and Communications Security (CCS)*, November 2003.
17. S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of tor," in *Proceedings of the IEEE Security and Privacy Symposium (S&P)*, May 2006.
18. N. Kiyavash, A. Houmansadr, and N. Borisov, "Multi-flow attacks against network flow watermarking schemes," in *Proceedings of the 17th USENIX Security Symposium*, July 2008.
19. G. Smillie, *Analogue and Digital Communication Techniques*, Butterworth-Heinemann, 1999.
20. P. Peng, P. Ning, and D. S. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in *Proceedings of the IEEE Security and Privacy Symposium (S&P)*, May 2006.
21. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.
22. S. Verdu, "On channel capacity per unit cost," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 1019–1030, November 1990.

Appendix A: Proof of Theorem 1

In this appendix, we prove Theorem 1.

Proof.

Because each flow lasts for $m \cdot T_c$ ($m \geqslant w_l$), we consider each flow with m slots (each slot lasting for T_c). According to TH-DSSS technique, each slot can be embedded randomly with marks. Given a slot and total number of marks wl, the probability of having marks in that slot is $\frac{wl}{T}$ and the probability of not embedding marks in that slot is $1-\frac{w_l}{T}$. Because the number of $+1$ and -1 will be almost equal in a PN code, the probability of embedding marks modulated by -1 bit is $p=\frac{wl}{m}$.

Given a time slot for all n flows, the probability of n flows embedded with a mark modulated by a synchronized -1 bit is p^n, the probability of $n-1$ flows embedded with a mark modulated by a synchronized -1 bit is $C_n^1 p^{n-1}(1-p)$, the probability of $n-2$ flows embedded with a mark modulated by a synchronized -1 bit is $C_n^2 p^{n-2}(1-p)^2$, the probability of $n-m$ flows embedded with a mark modulated by a synchronized -1 bit is $C_n^m p^{n-m}(1-p)^m$. Hence, given n flows, the average number of flows embedded with a mark modulated by -1 bit become $s=\sum_{k=n}^{1}[kC_n^{n-k}p^k(1-p)^{n-k}]$.

□

Appendix B: Proof of Theorem 2

In this appendix, we prove Theorem 2.

Proof.

We assume that the channel model in our network forensic traceback is a discrete and memoryless channel (DMC), the signal amplitude, A in Equation (4), plays an important role for preserving secrecy. Obviously, if it is too large in comparison with the noise, the secrecy property cannot be preserved. The signal amplitude square A^2 refers to the signal power density and we denote the noise variance as δ^2, which is determined by the transmission media and the end-to-end path quality. The capacity C of a Gaussian channel [21] is derived in Equation (6),

$$C=\max I(X,Y)=\frac{1}{2}\log_2(1+\frac{A^2}{\delta^2}), \qquad (6)$$

where $I(X,Y)$ is mutual information of modulated signals X and received modulated signals Y.

Denote t_i as the amount of time to transmit one input bit x_i across the DMC. The random variable $\mathscr{T}$ is the time to send such a bit and the mean of $\mathscr{T}$ is represented by $E(\mathscr{T})$. The mutual information in units of *bits per second* $I_t(X,Y)$ considering the transmission time cost for a DMC is

$$I_t(X,Y) = \frac{I(X,Y)}{E\{\mathscr{T}\}}. \tag{7}$$

The capacity C_t in unites of bits per second for a DMC [22] is given by Equation (8),

$$C_t = \max \frac{I(X,Y)}{E\{\mathscr{T}\}}. \tag{8}$$

Substitute Equation (6) into Equation (8), we have

$$C_t = \frac{\log_2(1+\frac{A^2}{\delta^2})}{2E\{\mathscr{T}\}}. \tag{9}$$

The derivation of $E\{\mathscr{T}\}$ is intuitive based on the description of the DSSS in [4] and TH-DSSS in Section 3. Then, Theorem 2 is proved. □