

Long PN Code Based DSSS Watermarking

Junwei Huang, Xian Pan, Xinwen Fu and Jie Wang
University of Massachusetts Lowell
Email: {jhuang, xpan, xinwenfu, wang}@cs.uml.edu

Abstract—Cyber crimes often involve complicated scenes. In this paper, we investigate unidentified crimes committed through anonymous communication networks. We developed a long Pseudo-Noise (PN) code based Direct Sequence Spread Spectrum (DSSS) flow marking technique for invisibly tracing suspect anonymous flows. By interfering with a sender’s traffic and marginally varying its rate, an investigator can embed a secret spread spectrum signal into the sender’s traffic. Each signal bit is modulated with a small segment of a long PN code. By tracing where the embedded signal goes, the investigator can trace the sender and receiver of the suspect flow despite the use of anonymous networks. Benefits of the Long PN code include its resistance to previous discovered detection approaches. We may also use the vast number of long PN code at different phases to conduct parallel traceback without worrying about the interference between codes. Using a combination of analytical modeling and experiments on Anonymizer, we demonstrate the effectiveness of the long PN code based DSSS watermarking technique.

I. INTRODUCTION

As the pervasive and ubiquitous wireless mobile computing platforms and Internet become an integrated part of our daily life, the number of cyber crimes has also been increasing drastically. These crimes, including sexual exploitation of children, intellectual property theft, identity theft, financial fraud, espionage, and many others, often involve complicated scenes. Criminals may abuse professional anonymous communications systems including Anonymizer [1] and Tor [2]. For example, unable to trace the true criminals, German authorities have been making intermittent arrests of Tor administrators, who have become scapegoats for anonymous criminals downloading child pornography via Tor.

Yu *et al.* [3] proposed to use the Direct Sequence Spread Spectrum (DSSS) flow marking technique for tracing anonymous crimes. In their approach, an investigator (*interferer*) can embed a short PN code encoded signal into the suspect traffic by interfering with the traffic rate at one side of the anonymous communication network. Another investigator (*sniffer*) eavesdrops on the suspect’s (*receiver*) inbound traffic at the other side of the anonymous communication network. If the same PN encoded signal is recovered in the receiver’s traffic, the investigators know that the receiver received the suspect traffic, such as data containing child pornography.

In [4], Jia *et al.* proposed a scheme based on *Mean-Square AutoCorrelation (MSAC)* of a single modulated flow’s traffic rate time series, which can detect the self-similarity and then expose the existence of the DSSS based traceback. When MSAC is applied to the target traffic marked with such a signal, periodic peaks will be shown since a single short PN

code is used to modulate all the bits of the signal individually in sequence.

In this paper, we developed a new flow marking technique called long PN code based DSSS watermarking for invisible traceback. In this technique, a long PN code is shared by both investigators (*interferer* and *sniffer*). The long PN code is used to modify a signal. However, one segment of the long PN code is used to spread only one bit of the signal. Different bits of the signal will be encoded with different segments of the long PN code. Basically, we are using different codes to modify different signal bits. This defeats the MSAC based detection, which can only detect a spread signal with the same short PN code spreading all the signal bits individually in sequence. Moreover, the number of long PN codes are abundant and we can use different long PN code for parallel traceback of different suspect flows. We may even use partial long PN codes with different phases for parallel traceback.

We have conducted extensive analysis and experiments to show the effectiveness of this new technique. We are able to prove that MSAC based detection cannot detect the long PN code modulated traffic, and different codes in parallel traceback will have limited interference with each other. We developed a suite of tools and performed real-world Internet experiments over Anonymizer, a popular commercial anonymous communication network. Our data validates the theory and demonstrate that our long PN code based DSSS watermarking technique can invisibly trace anonymous traffic flow over Anonymizer. The approach is also applicable to Tor.

The rest of the paper is organized as follows: In Section II, we review related work. In Section III, we introduce the long PN code based traceback. We then analyze the benefits of the long PN code based traceback in Section IV. The real-world experimental results are presented in Section V. The paper is concluded in Section VI.

II. RELATED WORK

Because of space limits, we only reviewed here the most related work. A good review of various anonymous communication systems can be found in [2], [5]. There has been much research done on degrading anonymous communication through mix networks. To determine whether Alice is communicating with Bob, through a mix network, similarity between Alice’s outbound traffic and Bob’s inbound traffic may be measured. For example, Zhu *et al.* in [6] proposed the scheme of using mutual information for the similarity measurement. Levine *et al.* in [7] utilized a cross correlation technique. Murdoch *et al.* in [8] also investigated the timing based threats

on Tor by using some compromised Tor nodes. Fu *et al.* [9] studied a flow marking scheme. Overlier *et al.* [10] studied a scheme using one compromised mix node to identify the “hidden server” anonymized by Tor. Yu *et al.* [3] proposed a direct sequence spread spectrum (DSSS) based traceback technique, which could be maliciously used to trace users of an anonymous communication networks.

Interval-based watermarks are proposed to trace attackers through the stepping stones. Wang *et al.* in [11] proposed a scheme that injected nondisplayable content into packets. Wang *et al.* in [12] proposed an active watermarking scheme that was robust to random timing perturbation. They analyzed the tradeoffs among the true positive rate, the maximum timing perturbation added by attackers, and the number of packets needed to successfully decode the watermark. Wang *et al.* in [13] also investigated the feasibility of a timing-based watermarking scheme in identifying the encrypted peer-to-peer VoIP calls. By slightly changing the timing of the packets, their approach can correlate encrypted network connections. Nevertheless, these timing-based schemes are not effective at tracing communication through a mix network with batching strategies that manipulate inter-packet delivery timing, as indicated in [3]. Peng *et al.* in [14] analyzed the secrecy of timing-based watermarking traceback proposed in [12], based on the distribution of traffic timing.

Kiyavash, Houmansadr and Borisov [15] proposed a multi-flow approach detecting the interval-based watermarks (which modify packet timings by selectively delaying some packets [16], [17]) and DSSS watermarks [3]. The approach requires multiple watermarked flows, which may show an unusual long silence period without packets or an unusual long period of low-rate traffic. They also proposed approaches to recover the watermarking parameters and remove watermarks in the case of interval based watermarks. They applied a probabilistic model and the Markov-modulated Poisson process (MMPP) to demonstrate the principle of their approaches. The authors briefly discussed countermeasures, which require more in-depth discussion. Note that given so many flows over the Internet, it is not always easy to recognize and find a relatively large number of flows embedded with DSSS watermarks. This multiple flow attack is infeasible against the long PN code based approach in this paper since our approach basically use different codes (partial long PN code) to trace different flows. In [18], Luo *et al.* introduced an approach detecting DSSS marks. They observed regular TCP burstiness corresponding to PN code chips. One reason causing this phenomenon is that the applied interference is large. The detection approach can also be defeated by separating chips by random intervals. Noise may also introduced during those random intervals.

III. LONG PN CODE BASED DSSS TRACEBACK

In this section, we will first formally define the problem, introduce our basic idea, and then discuss the long PN code. Finally, we introduce the flow marking process on embedding a long PN code spread signal into suspect traffic and recovering it.

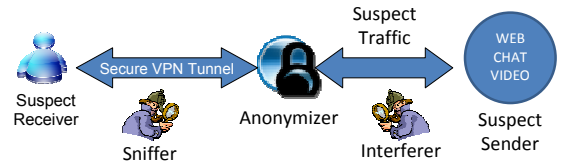


Fig. 1. Anonymizer

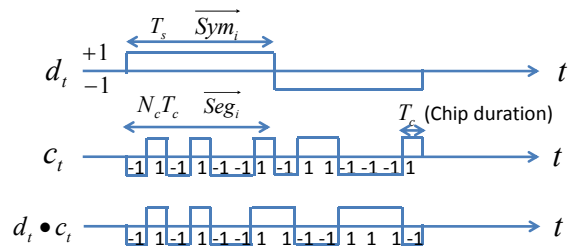


Fig. 2. Long PN Code

A. Problem Definition and Basic Idea

Figure 1 illustrates the forensic case we are studying. A suspect receiver is communicating anonymously with a suspect sender through an Anonymizer. For example, the suspect receiver could be a criminal downloading prohibited content from an illegal server, i.e., suspect sender. In this case, the suspect traffic is identified. The problem is: how can the law enforcement manipulate the suspect traffic in order to confirm it is the suspect sender who is communicating with the suspect receiver.

Our basic idea to solve this problem is that if law enforcement *interferer* embeds a signal into the suspect traffic and law enforcement *sniffer* can recover this signal from the inbound traffic into the suspect receiver, law enforcement can confirm that the suspect sender communicated with the suspect receiver. Techniques developed for this problem can be easily extended to a more general case: law enforcement can follow the traffic embedded with the signal and reconstruct the full communication path. *Anonymizer* in Figure 1 is a general concept for anonymous communication system, which can be Tor too.

B. Long PN Code

In *Direct Sequence Spread Spectrum (DSSS)*, we use *Pseudo-Noise (PN)* code to spread a signal over a bandwidth greater than the original signal bandwidth. Based on its length, there are short PN code and long PN code. In the spreading and despreading processes, the two types of PN codes are very different. In short PN code based DSSS, the same short PN code is used to spread (encode) each bit of a signal.

Figure 2 shows the long PN code based DSSS technique, in which we use different segments of the long PN code to spread different signal bits. The original signal d_t is a series of binary symbols Sym (+1 or -1). The *symbol duration* for both symbol +1 and -1 is T_s seconds, so the symbol rate is $R_s=1/T_s$. A long PN code c_t is a long sequence of chips of +1 and -1

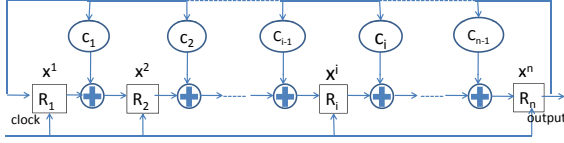


Fig. 3. MSRG

and is generated by the interferer and shared with the sniffer. Each *chip* lasts for T_c seconds, denoted as *chip duration*. The chip rate is $R_c=1/T_c$. N_c is the number of chips per symbol and is also the length of one segment from the long PN code. N_c chips construct one segment *Seg* from the long PN code. A long PN code can be very long (e.g. $2^{42} - 1$ chips). Therefore, we can use different segments of the code for spreading in order to spread different signal bits. For example, in Figure 2, we use $\{-1, 1, -1, 1, -1, -1, 1\}$ to spread signal bit 1 and $\{-1, 1, 1, -1, -1, -1, 1\}$ to spread signal bit -1.

There are mature ways to generate a long PN code by using the *Linear Feedback Register (LRF)*. There are two configurations for the LRF. One is called *Simple Shift Register Generator (SSRG)* and the other is called *Modular Shift Register Generator (MSRG)*. We use MSRG to generate a long PN code. The configuration of a MSRG is determined by the primitive polynomial coefficients [19]. In Figure 3, the primitive polynomial is

$$f(x) = 1 + c_1x + c_2x^2 + \dots + c_ix^i + \dots + c_{n-1}x^{n-1} + x^n. \quad (1)$$

where c_i is the coefficient, $i \in [1, n]$. c_i is either 0 or 1. R_i is the stage of the shift register. \oplus refers to *XOR*. Different primitive polynomials generate different long PN codes. If the degree of the primitive polynomial is n , the number of different primitive polynomials of degree n is equal to the number of different long PN codes. The total number of different PN codes produced by primitive polynomials of degree n can be calculated as follows [19],

$$\text{Number of different long PN codes} = \phi(2^n - 1)/n \quad (2)$$

where $\phi(2^n - 1)$ is the *Euler's ϕ function*.

The same primitive polynomial produces the same long PN code. We can use different long PN codes to trace different flows. A simple way to trace multiple flows is to use a long PN code at different phase shifts. A long PN code has a long period, so we may assign different phase shifts to the PN code that we can get different parts of the PN code, denoted as *partial PN codes*, to modulate different flows. These partial PN codes are shifted sequence segments from the same original long PN code. In Figure 4, we get three different PN codes by shifting the original PN code 0 bits, 5 bits and 10 bits. We use MSRG to generate a partial long PN code with a specific mask.

Therefore, we use MSRG to generate the long PN code, apply a *mask* to get the PN code shift [20] and then attain different shifted long PN code segments—partial long PN codes. To avoid the interference between different sequences,

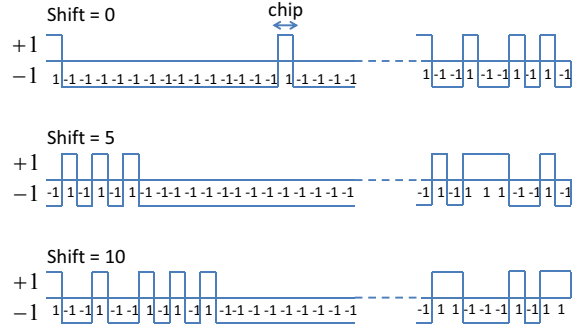


Fig. 4. Shift of Long PN Code

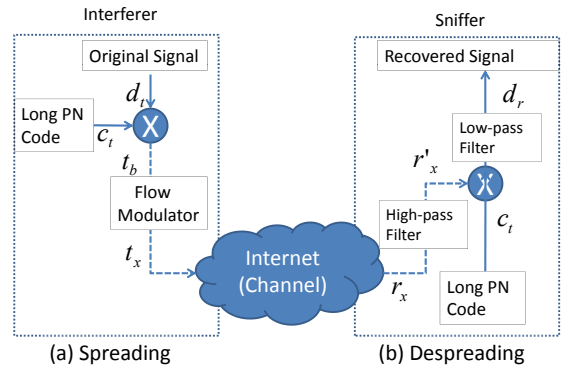


Fig. 5. Long PN Code Spreading and Despreading in DSSS

we have a minimum phase difference requirement, k , between partial long PN codes. The mask polynomial of the m th user can then be calculated as follows,

$$\text{mask} = \{\{x^{mk} \bmod f(x)\}/f(x)\}_{\text{terms of degree} < n} \quad (3)$$

where $f(x)$ is the primitive polynomial. Therefore, the number of different partial long PN codes are $\frac{2^P - 1}{k}$.

It can be seen that we can easily generate different long PN codes to modulate and trace multiple traffic flows in parallel. The low correlation property of those long PN codes makes it feasible to trace multiple flows through a mix network without interfering with each other.

C. Flow Marking

Figure 5 illustrates the framework of the flow marking. We spread a signal d_t as follows,

$$t_b = d_t \cdot c_t \quad (4)$$

where c_t is a segment of a partial long PN code and \cdot is the element-wise multiplication of two vectors. t_b is then used to modulate a target traffic flow by the interferer. We use *weak interference* against the flow when a chip is +1, so that the flow has a high rate for T_c seconds. We use *strong interference* against the flow when a chip is -1, so that the flow has a low rate for T_c seconds. We assume that the flow has an average traffic rate of D , then the high rate is $D+A$ and the low rate is $D-A$, where A is denoted as *mark amplitude*.

The target traffic flow rate should be large enough for investigators to introduce the marks by interference. Therefore, the transmitted signal t_x can be represented by

$$t_x = Ad_t c_t + D. \quad (5)$$

The modulated flow travels through the anonymous network, where there exists noise created by cross traffic and other interference. We treat all noise n as an aggregated factor. So the received signal r_x is

$$r_x = Ad_t c_t + D + n. \quad (6)$$

At the sniffer side (suspect receiver in Figure 1), in order to remove the direct current component D from the received signal, a high-pass filter is applied. So, the filtered received signal r'_x can be represented by

$$r'_x \approx Ad_t c_t + n. \quad (7)$$

We then use the same segment c_r of the shared partial long PN code to despread the filtered received signal r'_x to derive the received baseband signal r_b ,

$$r_b = Ad_t c_t \cdot c_r + n \cdot c_r. \quad (8)$$

A low-pass filter is then used to filter the high frequency noise. Thus,

$$r_b \approx Ad_t c_t \cdot c_r. \quad (9)$$

Since both *interferer* and *sniffer* have the same partial long PN code and $c_r = c_t$, $c_r \cdot c_t = 1$, we can recover the original signal.

IV. BENEFITS OF LONG PN CODE BASED DSSS TRACEBACK

In this paper, long PN code is applied in the DSSS-based technique for tracing traffic flows in an anonymous network. Benefits of using long PN code include: (i) The approach can defeat *mean-square autocorrelation (MSAC)* based detection technique proposed in [4] and makes the traceback hard to detect; (ii) it can trace multiple traffic flows in an anonymous network simultaneously. In this section, we will analyze how the long PN code based traceback achieves these two benefits.

In this section, we will first present the partial correlation of the long PN code, then analyze the invisibility of the long PN code based-DSSS watermarking and demonstrate that it can effectively trace multiple traffic flows simultaneously at the end.

A. Partial Correlation of Long PN Code

Assume a long PN code is $C = \{c_0, c_1, \dots, c_{P-1}\}$, where $c_i \in \{+1, -1\}$. The code period is P . A partial long PN code of length l from the whole long PN code is given by $C_s = \{c_s, c_{s+1}, \dots, c_{s+l-1}\}$, where $s \in [0, P-l]$ and s is the starting position to get a segment of l chips from the long PN code. We calculate the correlation on the partial PN code C_s as follows,

$$r_{C_s}(\gamma) = \sum_{i=0}^{l-\gamma-1} (c_{i+s} * c_{i+s+\gamma}) \quad (10)$$

where $l < P$ and γ is the lag.

The mean value of the partial correlation for the PN code is presented in Lemma 1. The detailed proof of Lemma 1 is available in Appendix A.

Lemma 1: $E\{r_{C_s}(\gamma)\}$ shows the mean value of the partial correlation, and γ means shift.

$$E\{r_{C_s}(\gamma)\} = \begin{cases} l, \gamma = 0 \\ -\frac{l-\gamma}{P}, \gamma \neq 0 \end{cases} \quad (11)$$

B. Invisibility of Long PN Code Based DSSS Traceback

The long PN code based DSSS watermarking technique makes it difficult to detect the existence of traceback by the suspect (receiver) being traced. The traffic flow modulated by a long PN code shows white noise-like pattern in both frequency and time domains so that investigators can not detect those watermarks in the frequency and time domains. We will also demonstrate that the *mean-square autocorrelation (MSAC)* method fails to detect the watermarks too. The MSAC method is based on the fact that the same short PN code is repeatedly used to spread each signal bit. In our new technique, each bit is spread by successive different segments from a long PN code. Basically, different signal bits are spread by different codes.

Recall that our objective is to prove the invisibility of the long PN code based DSSS watermarking technique that can defeat the MSAC detection method. Denote $\vec{x} = x_0, \dots, x_{N-1}$ as the signal, where N is the number of signal bits. x_i is either A or $-A$, where A is the watermark amplitude. Denote $\vec{C} = \{\vec{C}_0, \vec{C}_1, \dots\}$ as a long PN code, where \vec{C}_i is a segment from the long PN code. We take a segment to spread one signal bit. Assume the length of each PN segment is l , that is, we use l chips to spread one signal bit. c_j represents one chip and c_j is either 1 or -1 . We assume that bits x_i and x_j ($i \neq j$) are independent.

The modulated signal \vec{X} can be written as follows,

$$\vec{X} = (x_0 \vec{C}_0, x_1 \vec{C}_1, \dots, x_{N-1} \vec{C}_{N-1}) \quad (12)$$

$$= (x_0 c_0, \dots, x_0 c_{l-1}, x_1 c_l, \dots, x_1 c_{2l-1}, \dots, x_{N-1} c_{(N-1)l}, \dots, x_{N-1} c_{Nl-1}) \quad (13)$$

Since x_i s are independent and identically distributed, $P(x_i c_j = A) = 1/2$ and $P(x_i c_j = -A) = 1/2$, thus $E(x_i c_j) = 0$ and the standard deviation $\sigma = A$. The following formula can be used to estimate the autocorrelation of a time series represented by \vec{X} ,

$$r(\gamma) = 1/(N - \gamma) \sum_{i=0}^{N-1-\gamma} (a_{i_j} * a_{i_j+\gamma}), \quad (14)$$

where γ is the lag, $a_{i_j} = x_i c_{il+j}$ is the i^{th} item of \vec{X} , and $i \in [0, N-1], j \in [0, l-1]$.

The MSAC method reveals the existence of the short PN code based DSSS watermarks by calculating $E(r^2(\gamma))$. $r^2(\gamma)$

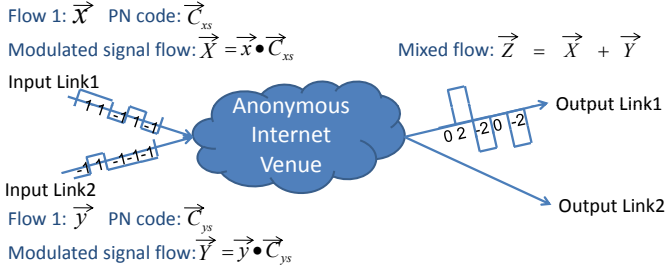


Fig. 6. Tracing Multiple Flows

is the square autocorrelation of spread signal \vec{X} and a time-shifted \vec{X} with lag γ . By calculating $E(r^2(\gamma))$, periodic peaks with a period of l will show up.

Theorem 1 shows there are no periodic peaks in our long PN code based watermarking technique under this MSAC detection method. The long PN code based DSSS watermarking technique is invisible for suspect *sender* and *receiver*. The detailed proof of Theorem 1 is in Appendix B.

Theorem 1: The mean value of $E(r^2(\gamma))$ is

$$E(r^2(\gamma)) \approx \begin{cases} A^4, \gamma = 0 \\ 0, \gamma \neq 0 \end{cases} \quad (15)$$

According to Theorem 1, it is secret to use the long PN code based DSSS watermarking technique to trace traffic flows. There is only one peak shown in the MSAC detection method at the point $\gamma = 0$. Unlike using the short PN code based DSSS watermarking technique in [3], which reveals the self-similarity of embedded DSSS watermarks occurring at regular intervals, no periodic peaks show up for the long PN code based traceback.

C. Tracing Multiple Flows

By using long PN code based DSSS watermarking technique, we can trace multiple flows in parallel to achieve efficient and flexible traceback. Different flows will be modulated by different partial long PN codes and may reach the same destination. However, because the partial long PN codes, i.e., differently shifted long PN codes from the same original PN code, have low correlation and will not interfere with each other too much, we can still recover signals embedded in those flows modulated by these long PN codes. Therefore, we can easily trace multiple flows in parallel.

We now formally analyze the effectiveness of the long PN code based DSSS watermarking technique on tracing multiple flows through an anonymous network. We use Figure 6 as an example. Flow 1 is \vec{x} , and flow 2 is \vec{y} . We use the first partial long PN code, \vec{C}_{xs} , to modulate flow 1 on input link 1 and a second partial long PN code, \vec{C}_{ys} , to modulate flow 2 on input link 2. The modulated flow 1 and flow 2 transmit through an anonymous Internet venue such as Anonymizer. Without loss of generality, we will demonstrate that the sniffer can recover the original signal flow 1 from the mixed output flow.

$\vec{C} = \{c_0, c_1, \dots, c_{P-1}\}$ is the long PN code.

$$\vec{C}_{xs} = (\vec{C}_{xs_0}, \vec{C}_{xs_1}, \dots, \vec{C}_{xs_{n-1}}) \quad (16)$$

$$= (c_{xs}, c_{xs+1}, \dots, c_{P-1}, c_0, \dots, c_{xs-1}) \quad (17)$$

where xs is the phase shift and the starting point of the long PN code. \vec{C}_{xs_i} is the i^{th} code segment of \vec{C}_{xs} , and the length of each segment is l . That is, each signal bit is modulated by l chips. We use \vec{C}_{xs} to modulate the flow 1.

$$\vec{C}_{ys} = (\vec{C}_{ys_0}, \vec{C}_{ys_1}, \dots, \vec{C}_{ys_{n-1}}) \quad (18)$$

$$= (c_{ys}, c_{ys+1}, \dots, c_{P-1}, c_0, \dots, c_{ys-1}), \quad (19)$$

where ys is the phase shift and the starting point of the same long PN code. \vec{C}_{ys_i} has the same meaning as \vec{C}_{xs_i} . We use \vec{C}_{ys} to modulate the flow 2. Signals to be embedded in flow 1 and flow 2 are $\vec{x} = (x_0, x_1, \dots, x_{n-1})$, and $\vec{y} = (y_0, y_1, \dots, y_{n-1})$. Therefore, the modulated signal flows \vec{X} and \vec{Y} can be written as follows:

$$\vec{X} = \vec{x}\vec{C}_{xs} = (x_0\vec{C}_{xs_0}, x_1\vec{C}_{xs_1}, \dots, x_{n-1}\vec{C}_{xs_{n-1}}) \quad (20)$$

$$\vec{Y} = \vec{y}\vec{C}_{ys} = (y_0\vec{C}_{ys_0}, y_1\vec{C}_{ys_1}, \dots, y_{n-1}\vec{C}_{ys_{n-1}}) \quad (21)$$

\vec{Z} is the mixed flow. From (20) and (21), we have

$$\vec{Z} = \vec{X} + \vec{Y} \quad (22)$$

$$\begin{aligned} &= (x_0\vec{C}_{xs_0} + y_0\vec{C}_{ys_0}, \\ & \quad x_1\vec{C}_{xs_1} + y_1\vec{C}_{ys_1}, \\ & \quad \dots, \\ & \quad x_{n-1}\vec{C}_{xs_{n-1}} + y_{n-1}\vec{C}_{ys_{n-1}}) \end{aligned} \quad (23)$$

Assume *sniffer* can get the mixed flow \vec{Z} . To recover flow x , we use \vec{C}_{xs} to despread the \vec{Z} . We have

$$\begin{aligned} \vec{Z}\vec{C}_{xs} &= (x_0\vec{C}_{xs_0}\vec{C}_{xs_0} + y_0\vec{C}_{ys_0}\vec{C}_{xs_0}, \\ & \quad x_1\vec{C}_{xs_1}\vec{C}_{xs_1} + y_1\vec{C}_{ys_1}\vec{C}_{xs_1}, \\ & \quad \dots, \\ & \quad x_{n-1}\vec{C}_{xs_{n-1}}\vec{C}_{xs_{n-1}} \\ & \quad + y_{n-1}\vec{C}_{ys_{n-1}}\vec{C}_{xs_{n-1}}) \end{aligned} \quad (24)$$

where $x_i\vec{C}_{xs_i}\vec{C}_{xs_i} + y_i\vec{C}_{ys_i}\vec{C}_{xs_i}$ represents one demodulated signal, $i \in [0, n-1]$, and $\vec{C}_{xs_i} \neq \vec{C}_{ys_i}$.

We can recover each original signal d_t from the target traffic flow with a simple and effective decision rule, based on Theorem 2. The detailed proof of this theorem is available in Appendix C.

Theorem 2: Donate r_x as one demodulated signal. we have

$$E(r_{x_i}) = \begin{cases} A, x_i = A \\ -A, x_i = -A \end{cases} \quad (25)$$

where A is the watermark amplitude and l is the length of each segment. The true value of r_{x_i} only has little difference from the above mean value of r_{x_i} .

From the Theorem 2, we give a decision rule as

$$d_t = \begin{cases} 1, r_x > 0 \\ -1, r_x < 0 \end{cases} \quad (26)$$

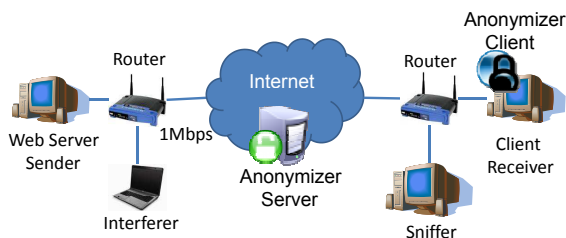


Fig. 7. Experiment Setup

We can recover an original signal bit easily by using the decision rule. Theorem 2 and the decision rule show that it is feasible to trace multiple flows in parallel for long PN code based DSSS watermarking technique in a simple way.

V. EVALUATION

We have conducted real-world experiments on Anonymizer to evaluate the performance of the long PN code based DSSS watermarking technique. In this section, we will first introduce the experiment setup. We will then present the experimental results of the detection rate, the false positive rate and the capability of the new traceback approach on tracing multiple flows. Finally, we demonstrate the long PN code based technique can defeat the MSAC based watermark detection.

A. Experiment Setup

Figure 7 illustrates the experiment setup. A web server (*sender*) running Windows 7 was located at a university campus. An off-campus computer (*receiver*) ran an Anonymizer client. By setting up an encrypted VPN tunnel between the off-campus computer and Anonymizer server on the Internet, the off-campus computer can surf the web without exposing its real IP address. In order to determine if the off-campus computer is downloading a file from the web server, we use a computer as an *interferer* to interfere with the outbound traffic from the web server, and use another computer as a *sniffer* to sniff the inbound traffic to the receiver. The interferer and the sender are connected by a router, as are the sniffer and the receiver. The interferer and the sender share a link, so that the interferer can interfere with the sender's traffic and modulate the outbound traffic with the long PN code based approach.

In our experiments, the interferer uses UDP constant bit rate (CBR) traffic to modulate the target flow. The CBR traffic packet size is fixed at 150 bytes. The CBR traffic is turned off when a chip within a signal modulated by the long PN code is +1. The CBR traffic is turned on when a chip is -1. The on-interval and off-interval are equal to the chip duration. Based on the TCP's loop control mechanism, when the CBR traffic rate increases, the TCP traffic rate decreases. When the CBR traffic rate decreases (e.g., no CBR traffic), the TCP traffic rate increases. To use the long PN code based DSSS watermarking technique and recover the original signal, we need to obtain a time series of the TCP flow rate. In order to recover the spread signal, the sampling period should be less than half of the chip duration based on the Nyquist sampling theory [21]. We use a sampling interval of 0.1s.

Because of dynamics of Internet traffic, in our experiments, we used a rough estimation of the delay to synchronize the

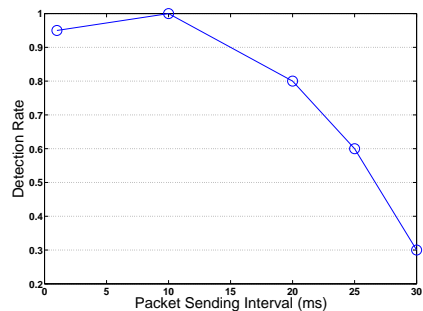


Fig. 8. Detection Rate v.s. Interfering Packet Sending Interval

intercepted traffic at the sniffer with the target traffic at the interferer in order to recover the PN code. We then used the matched filter based approach to search for the best match within a certain search range. We set the range as $[-1s, 1s]$ in our experiments.

B. Detection Rate of Long PN Code Based Flow Marking

For experiments in this section, the detection rate refers to the probability that a n -bit signal is correctly recognized. We can vary the parameters such as the long PN code length, watermark amplitude and chip duration to obtain high detection rate. In the experiments, we first generated a long PN code of $2^{15} - 1$ chips, then used the mask, $\{0,0,0,0,0,0,1,0,0,0,1,1,0,0,0\}$, to generate a long PN code with shift 20473. We use segments of the shifted long PN code to spread the signal $\{1 -1 1 1 -1 1 -1\}$, as discussed in Section III. The chip duration is fixed at 1 second.

We first examine the impact of the interfering CBR traffic rate (watermark amplitude) on detection rate. We change the CBR packet sending frequency from $1\text{packet}/1\text{ms}$ to $1\text{packet}/30\text{ms}$. In Figure 8, we can see that with the CBR traffic rate decreases (packet sending interval increases), the detection rate decreases. The reason is that the slow interfering traffic incurs a small watermark amplitude A in Equation (5). We then examine the impact of different long PN code lengths on detection rate. We used different long PN code segment lengths from 1 to 7 to spread a signal bit. Figure 9 shows that in general longer segment length achieves higher detection rate. This is the benefit of using spread spectrum spreading: we can use a long code to fight a noisy environment for better performance.

We also examine the impact of chip durations in detection rate. We varied the chip duration from 0.5s to 3.0s. From Figure 10, we can see that a longer chip duration produces better performance in terms of detection rate. A chip duration above 0.8s can achieve a detection rate of 90% or higher. The reason is a longer spreading process accumulates more signal energy.

C. False Positive Rate

Recall that the false positive rate $P_{F,n}$ for recognizing a n -bit original signal is $P_{F,n} = \frac{1}{2^n}$ [3]. In our experiments, we varied the signal length from 1 to 7. For each signal length we measured the false positive rates for the long PN code

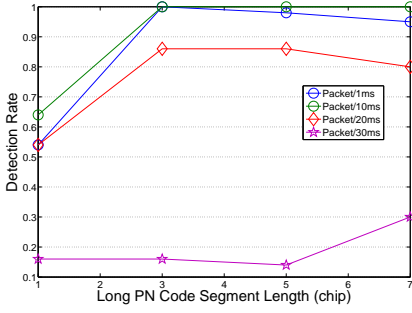


Fig. 9. Detection Rate v.s. Segment Length

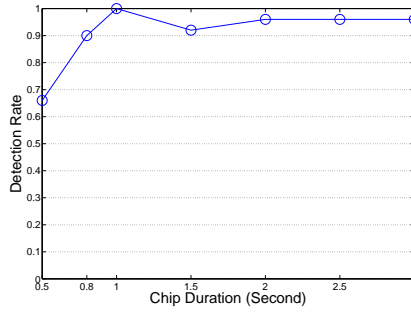


Fig. 10. Detection Rate v.s. Chip Duration

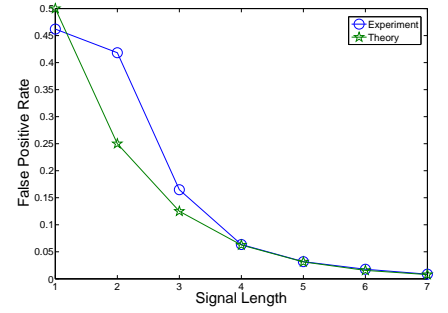


Fig. 11. False Positive Rate

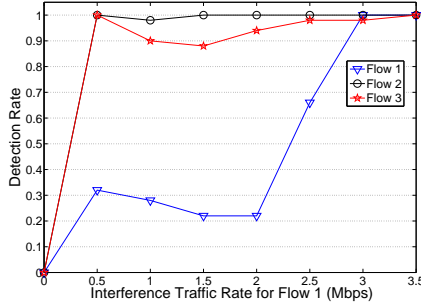


Fig. 12. Detection Rate v.s. Interference Traffic Rate for Tracing Multiple Flows

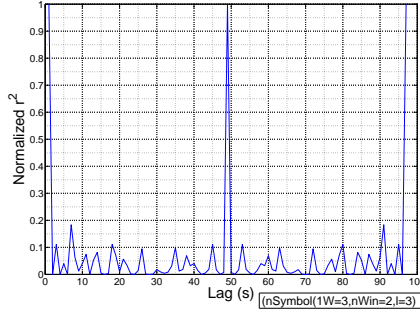


Fig. 13. Estimation of Mean-square Autocorrelation

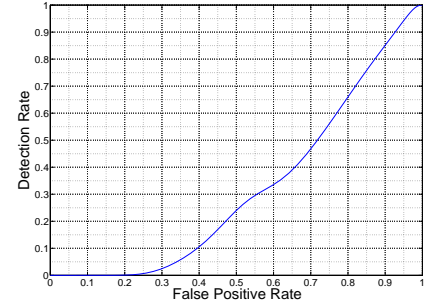


Fig. 14. ROC

segments of different lengths from 2 to 7. The false positive rate for each signal length is calculated as the average of the probabilities of detecting the signal with different long PN code segment lengths. From Figure 11, we can see that the false positive rate decreases with the increasing long PN code segment length. The theoretical curve matches the empirical curve very well.

D. Effectiveness of Tracing Multiple Flows

To demonstrate the effectiveness of tracing multiple flows by the long PN code based technique, we generated two other modulated flows. We first generated a long PN code of $2^{15} - 1$ chips. By using different masks below, we derived different shifted long PN code.

- mask1={0,0,0,0,0,0,1,0,0,0,1,1,0,0,0},
- mask2={0,0,0,0,0,0,1,0,0,0,1,1,0,0,1},
- mask3={0,0,0,0,0,0,1,0,0,0,1,1,0,1,0}.

The shifts for those long PN codes are: 20473, 4874, 25011.

In our experiments, the receiver downloaded files from three different servers. Three interferers then used different shifted long PN codes to modulate outbound traffic flows from different servers. The sniffer obtained the mixed traffic flow and applied different shifted long PN codes to demodulate the flow, and tried to recover the original signal. We set the narrow bandwidth links shared by servers and interferers as 4Mbps. We fixed the interfering traffic rate against flow 2 and flow 3 at 3Mbps. We then increased the interfering traffic rate for flow 1 from 0.5Mbps to 3.5Mbps. Figure 12 shows detection rate for those three flows. It can be observed that the long PN code based DSSS watermarking technique can effectively

trace multiple flows. The detection rate for all three flows can approach 100%.

E. Defeating MSAC Detection

In [3], the authors investigated the detection of watermarks generated by a short PN code, which is used to spread each signal bit. Through the *mean-square autocorrelation* (MSAC) analysis, periodic peaks show up due to self-similarity in the modulated traffic caused by homogeneous PN codes that are used in modulating a multiple-bit signal. Our strategy can defeat the MSAC analysis since we use different long PN code segments to spread different signal bits. Figure 13 shows the MSAC of a modulated flow. We can see there is no periodical peak. The authors also used detection rate P_D and false positive rate P_F as their evaluation metrics for evaluating MSAC's capability to detect short PN code generated DSSS watermarks. When they try to detect traffic containing DSSS watermarks, they need a high detection rate and a low false positive rate. Figure 14 shows Receiver Operating Characteristic (ROC) curve for our long PN code generated watermarks, which is a plot of P_D versus P_F . It can be observed that the false positive rate is as high as the detection rate. Therefore, it is hard to detect long PN code generated watermarks by the MSAC analysis.

VI. CONCLUSION

In this paper, we developed a long PN code based DSSS watermarking technique to trace suspect communication over anonymous venues on the Internet. This traceback technique has the following advantages. It has good invisibility. Since

different segments of a long PN code are used to modulate different signal bits, this technique removes regular patterns and self similarity from the generated watermarks. Therefore, it can defeat mean-square autocorrelation (MSAC) based detection of watermarks generated by a short PN code, which is used to repeatedly modulate each signal bit. We studied how to produce such long PN code and the correlation between different partial long PN codes. Because the abundance of the long PN code and the low cross correlation between different codes, this technique can be used to trace multiple flows in parallel without too much cross interference.

Through a combination of analytical modeling and an extensive set of experiments over Anonymizer, we demonstrated the effectiveness of the long PN code based DSSS watermarking technique. The long PN code based DSSS watermarking technique is a general one and can be used in other cyber crime scene investigations.

ACKNOWLEDGMENT

This work was supported in part by USA NSF grants 0942113, 0958477, 0943479, and 0907964. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor agencies.

REFERENCES

- [1] Anonymizer, Inc., <http://www.anonymizer.com/>, 2010.
- [2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [3] 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.
- [4] W. Jia, F. TSO, Z. Ling, X. Fu, D. Xuan, and W. Yu, "Blind detection of spread spectrum flow watermarks," in *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.
- [5] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: design of a type iii anonymous remailer protocol," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy (S&P)*, May 2003.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] L. Overlier and P. Syverson, "Locating hidden servers," in *Proceedings of the IEEE Security and Privacy Symposium (S&P)*, May 2006.
- [11] X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework," in *Proceedings of 16th International Conference on Information Security (IFIP/Sec)*, June 2001.
- [12] 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.
- [13] 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.

- [14] 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.
- [15] N. Kiyavash, A. Houmansadr, and N. Borisov, "Multi-flow attacks against network flow watermarking schemes," in *Proceedings of the 17th USENIX Security Symposium*, July/August 2008.
- [16] Y. J. Pyun, Y. H. Park, X. Wang, D. S. Reeves, and P. Ning, "Tracing traffic through intermediate hosts that repacketize flows," in *Proceedings of IEEE INFOCOM*, May 2007.
- [17] S. C. X. Wang and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, May 2007.
- [18] X. Luo, J. Zhang, R. Perdisci, and W. Lee, "On the secrecy of spread-spectrum flow watermarks," in *Proceedings of European Symposium on Research in Computer Security ESORICS*, 2010.
- [19] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes, 2nd Edition*. Cambridge, MA: The MIT Press, 1972.
- [20] S. Lee, *Spread Spectrum CDMA: IS-95 and IS-2000 for RF Communications*. Chicago, IL: McGraw-Hill Professional, August 2002.
- [21] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and systems*, 2nd ed. Upper Saddle River, NJ 07458, USA: Prentice-Hall, 1997.

APPENDIX

APPENDIX A: PROOF OF LEMMA 1

In this Appendix, we provide the proof of Lemma 1.

Case 1. $\gamma = 0$, from (10) we have:

$$E\{r_{C_s}(0)\} = E\left\{\sum_{i=0}^{l-1} (C_s \cdot C_s)\right\} \quad (27)$$

$$= \sum_{i=0}^{l-1} (1 \times P_r(1) - 1 \times P_r(-1)) \quad (28)$$

$$= \sum_{i=0}^{l-1} (1 \times 1 - 1 \times 0) \quad (29)$$

$$= l \quad (30)$$

In a long PN code with period P , the number of -1 s differs from the number of 1 s by at most 1. That is, there are $\frac{P-1}{2}$ 1 s and $\frac{P-1}{2} + 1$ -1 s. Therefore, $P_r(1) = \frac{P-1}{P}$, $P_r(-1) = \frac{\frac{P-1}{2} + 1}{P}$.

Case 2. $\gamma \neq 0$, we have:

$$E\{r_{C_s}(\gamma)\} = \sum_{i=0}^{l-\gamma-1} (c_{i+s}c_{i+s+\gamma}) \quad (31)$$

Based on the closure property of PN sequences, the sequence formed by the product $c_{i+s}c_{i+s+\gamma}$ is a different phase shift of the sequence, denoted $c_{i+s+\gamma'}$. So we have:

$$E\{r_{C_s}(\gamma)\} = \sum_{i=0}^{l-\gamma-1} c_{i+s+\gamma'} \quad (32)$$

$$= \sum_{i=0}^{l-\gamma-1} (1 \times P_r(1) - 1 \times P_r(-1)) \quad (33)$$

$$= \sum_{i=0}^{l-\gamma-1} \left(\frac{P-1}{2} - \frac{P-1}{2} + 1\right) \quad (34)$$

$$= \sum_{i=0}^{l-\gamma-1} -\frac{1}{P} \quad (35)$$

$$= -\frac{l-\gamma}{P} \quad (36)$$

In case 2, $\gamma \neq 0$ means there are two different PN codes which are generated by different phase shift from the same long PN code. The period P is very long and $l \ll P$. Therefore, approximately, we have:

$$E\{r_{C_s}(\gamma)\} = -\frac{l-\gamma}{P} \approx 0, l \ll P, \text{ and } l-\gamma \ll P \quad (37)$$

APPENDIX B: PROOF OF THEOREM 1

We proof the Theorem 1 in this Appendix.

Case 1 $\gamma = 0$: From (13) and (14), we have

$$r(0) = \frac{1}{N} \sum_{i=0}^{N-1} (x_i c_{il+j} x_i c_{il+j}) \quad (38)$$

$$= \frac{1}{N} \sum_{i=0}^{N-1} (x_i^2 c_{il+j}^2). \quad (39)$$

Since $x_i^2 = A^2$ and $c_i^2 = 1$, then

$$r(0) = A^2, \quad (40)$$

and

$$E(r(0)) = A^4. \quad (41)$$

Case 2 $\gamma \neq 0$: From (13) and (14), we have

$$r(\gamma) = \frac{1}{N-\gamma} \sum_{i=0}^{N-1-\gamma} (a_{i_j} a_{i_j+\gamma}) \quad (42)$$

$$= \frac{1}{N-\gamma} (x_0 \vec{C}_0 x_\gamma \vec{C}_\gamma + \dots + x_{N-1-\gamma} \vec{C}_{N-1-\gamma} x_{N-1} \vec{C}_{N-1}). \quad (43)$$

C_i in (43) is the i^{th} segment from the long PN code to modulate the i^{th} signal bit. Therefore, there are no same segments, that is, $C_i \neq C_j$, if $i \neq j$. And the long PN code we use in this paper is an best noise-like autocorrelation function among popular PN codes. That means: when the lag $\gamma \neq 0$, based on the analysis of PN code's correlation in Appendix A, we have $E(C_i C_{i+\gamma}) = -\frac{l-\gamma}{P} \approx 0$. Therefore, approximately, we have

$$E(x_i C_{il+j} x_{i+\gamma} C_{il+j+\gamma}) = E(x_i x_{i+\gamma}) E(C_{il+j} C_{il+j+\gamma}) \quad (44)$$

$$= 0 \quad (45)$$

$$r(\gamma) = 0, \gamma \neq 0. \quad (46)$$

Therefore,

$$E(r^2(\gamma)) = 0, \gamma \neq 0. \quad (47)$$

APPENDIX C: PROOF OF THEOREM 2

A demodulated signal is as

$$r_x = x_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} + y_i \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}} \quad (48)$$

Recall that the length of each segment $\vec{C}_{x_{s_i}}$ or $\vec{C}_{y_{s_i}}$ is l , the full period of the whole long PN code is P . From (10) and (36), we have

$$E\{\vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}\} = l \quad (49)$$

Since $l \ll P$, $l-\gamma \ll P$, we have

$$\frac{(l-\gamma)}{P} \approx 0 \quad (50)$$

Therefore

$$E\{\vec{C}_{x_{s_i}} \vec{C}_{y_{s_i}}\} = -\frac{l-\gamma}{P} \approx 0 \quad (51)$$

Case 1 $x_i = A$: From (49) and (51), we have

$$E(r_x) = E(x_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) + E(y_i \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) \quad (52)$$

$$= A \cdot E(\vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) + 0 \quad (53)$$

$$= Al \quad (54)$$

Case 2 $x_i = -A$: From (49) and (51), we have

$$E(r_x) = E(x_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) + E(y_i \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) \quad (55)$$

$$= -A \cdot E(\vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) + 0 \quad (56)$$

$$= -Al \quad (57)$$

$E(r_x)$ means the average value of r_x . To prove the variation of r_x from $E(r_x)$ is very small, we need to calculate variance of the random variable r_x .

$$E((r_x)^2) = E((x_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} + y_i \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}})^2) \quad (58)$$

$$= E(x_i^2 \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) + E(y_i^2 \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) + E(2 * x_i y_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) \quad (59)$$

From (49) and (51), we have

$$E(x_i^2 \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}}) = A^2 l^2 \quad (60)$$

$$E(y_i^2 \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) \approx 0 \quad (61)$$

$$E(2 * x_i y_i \vec{C}_{x_{s_i}} \vec{C}_{x_{s_i}} \vec{C}_{y_{s_i}} \vec{C}_{x_{s_i}}) = 0 \quad (62)$$

Therefore, from (60), (61) and (62)

$$E((r_x)^2) \approx A^2 l^2 \quad (63)$$

From (54), (57) and (63), we can get variance as

$$\text{var}(r_x) = E((r_x)^2) - (E(r_x))^2 \quad (64)$$

$$\approx A^2 l^2 - (Al)^2 \quad (65)$$

$$= 0 \quad (66)$$