

An Invisible Localization Attack to Internet Threat Monitors

Wei Yu, Xun Wang, Xinwen Fu, Dong Xuan, and Wei Zhao

Abstract—Internet threat monitoring (ITM) systems have been deployed to detect widespread attacks on the Internet in recent years. However, the effectiveness of ITM systems critically depends on the confidentiality of the location of their monitors. If adversaries learn the monitor locations of an ITM system, they can bypass the monitors and focus on the uncovered IP address space without being detected. In this paper, we study a new class of attacks, the invisible *LOCALization (iLOC)* attack. The *iLOC* attack can accurately and invisibly localize monitors of ITM systems. In the *iLOC* attack, the attacker launches low-rate port-scan traffic, encoded with a selected *pseudo-noise code (PN-code)*, to targeted networks. While the secret PN-code is invisible to others, the attacker can accurately determine the existence of monitors in the targeted networks based on whether the PN-code is embedded in the report data queried from the data center of the ITM system. We formally analyze the impact of various parameters on attack effectiveness. We implement the *iLOC* attack and conduct the performance evaluation on a real-world ITM system to demonstrate the possibility of such attacks. We also conduct extensive simulations on the *iLOC* attack using real-world traces. Our data show that the *iLOC* attack can accurately identify monitors while being invisible to ITM systems. Finally, we present a set of guidelines to counteract the *iLOC* attack.

Index Terms—Internet threat monitoring systems, Invisible localization attack, PN-code, Security

I. INTRODUCTION

In recent years, widespread attacks, such as worms [1], [2], [3] and distributed denial-of-service (DDoS) attacks [4], [5], have been dangerous threats to the Internet. Due to the widespread nature of these attacks, large-scale traffic monitoring across the Internet has become necessary in order to effectively detect and defend against them. Developing and deploying *Internet threat monitoring (ITM)* systems (or *motion sensor networks*) is a major effort in this direction.

An ITM system consists of a number of monitors and a data center. The monitors are distributed across the Internet and can be deployed at hosts, routers, and firewalls, etc. Each monitor is responsible for monitoring and collecting traffic addressed to a range of IP addresses within a sub-network. The range of IP addresses covered by a monitor is

also referred to as the *location* of the monitor. Periodically, the monitors send traffic logs to the data center. The data center analyzes the traffic logs and publishes reports to the public. Recall that in order to maximize the usage of such reports, most existing ITM systems publish the reports online and make them accessible to the public. The reports provide critical insights into widespread Internet attacks and are used in detecting and defending against such attacks. ITM systems have been successfully used to detect the outbreaks of worms [6] and DDoS attacks [7]. There have been many real-world developments and deployments of ITM systems. Examples include DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) [8], SANs ISC (Internet Storm Center) [6], Internet Sink [9], Network Telescope [10], CAIDA [11], MyNetWatchMan [12], and HoneyNet [13], [14].

However, the usability of ITM systems largely depends on the confidentiality of IP addresses covered by their monitors, i.e., the *locations* of monitors. If the locations of monitors are identified, the attacker can deliberately avoid these monitors and directly attack the uncovered IP address space. It is a known fact that the number of sub-networks covered by monitors is much smaller than the total number of sub-networks in the Internet [6], [9], [10]. In other words, the IP address space covered by monitors represents a very small portion of the entire IP address space [6]. Hence, bypassing IP address spaces covered by monitors will *significantly* degrade the accuracy of the traffic data collected by the ITM system in reflecting the real situation of attack traffic. Furthermore, the attacker may also poison ITM systems by manipulating the traffic towards and captured by disclosed monitors. For example, the attacker may launch high-rate port-scan traffic to disclosed monitors and feign a large-scale worm propagation. The attackers may even launch retaliation attacks (e.g., DDoS) against participants (i.e., monitor contributors) of ITM systems, thereby discouraging them from contributing to ITM systems. In summary, the attacker can significantly compromise the ITM system usability if locations of monitors are disclosed. It is important to have a thorough understanding of such attacks, in order to effectively protect ITM systems.

In this paper, we conduct a systematic investigation of a class of attacks which aim to localize monitors *accurately* and *invisibly*. Accuracy is very important for an attacker in identifying monitor locations. Meanwhile, invisibility is also vital to a successful attack. If the attack attempts are identified by the defender (such as the ITM administrators), countermeasures can be applied by the defender to reduce or eliminate the effects of the attack by filtering suspicious traffic (so that the attacker will not be able to identify monitors

Wei Yu is with the Dept. of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: weiyu@cs.tamu.edu. Xun Wang and Dong Xuan are with the Dept. of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210. E-mail: {wangxu, xuan}@cse.ohio-state.edu. Xinwen Fu is with the College of Business and Information Systems, Dakota State University, Madison, SD 57042. Email: Xinwen.Fu@dsu.edu. Wei Zhao is with the School of Science, Rensselaer Polytechnic Institute, Troy, NY 12180. E-mail: zhaow3@rpi.edu.

A short conference version appears in the *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM) (mini-conference)*, Phoenix, AZ, April 13-18, 2008.

through traffic analysis [15]), confusing attackers (to make the attacker obtain wrong monitor location information [13]), and even tracking an attacker to its origin (so that attackers can be held accountable for their malicious acts [16], [17]).

Several attack schemes to discover the location of monitors have been investigated [18], [19]. However, our work is the first to address an attack aiming to achieve the objectives of both accuracy and invisibility. It is challenging for the attacker to achieve these two objectives simultaneously. Intuitively, the attacker can use the high-rate attack traffic, as in [18], [19], to achieve high attack accuracy as follows. The attacker can launch high-rate port-scan traffic to a target network. The attacker then queries the data center for the report on recent port-scan activities. If there is a traffic *spike* in the report data reflecting the high-rate port-scan traffic sent by the attack, the attacker can determine that the target network is deployed with monitor(s) which sends traffic report to the data center. However, the drawback of this scheme is its high visibility, since the launched high-rate traffic makes it highly visible to the defender.

In this paper, we investigate a new class of attacks called invisible *LOC*calization (*iLOC*) attack. In the *iLOC* attack, the attacker launches low-rate port-scan traffic (also referred to as *attack traffic*) to target networks. The scan traffic is encoded with a carefully selected *pseudo-noise code* (PN-code), known by only the attacker. The PN-code embedded in traffic can be accurately recognized by the attacker even with the interference from background traffic aggregated by the data center but not generated by *iLOC*. Thus, the attacker is able to *accurately* determine the existence of monitors in the target networks based on whether the same PN-code is embedded in the report data queried from the data center of the ITM system. The PN-code modulated/embedded scan traffic will appear as innocent noise in both the time and frequency domains, rendering it *invisible* to others who do not know the PN-code. Only those aware of the original PN-code can correctly recover the encoded PN-code and identify the monitor locations. Therefore, using the *iLOC* technique, the attacker can accurately localize monitors while evading detection.

We conduct both theoretical analysis and experimental evaluation on the *iLOC* attack. We derive formulas for both the accuracy and invisibility of the attack. We analyze and discuss the impacts of various attack parameters (e.g., PN-code length, attack traffic rate etc.) on the effectiveness of attack. Based on the analytical results, we discuss how the attacker can select the attack parameters in order to achieve both attack accuracy and invisibility. We implement the *iLOC* attack and perform the performance evaluation on a real-world ITM system, which demonstrates the possibility of the *iLOC* attack. We also conduct extensive performance evaluations on the *iLOC* attack in a simulated environment. Our evaluations are based on replaying a large set of real-world Internet traffic traces collected by a real-world ITM system. The evaluation data demonstrate that the attack can accurately identify the locations of monitors, while evading detection by those who do not know the PN-code used by the attacker. Furthermore, we present a set of guidelines on how to counteract the *iLOC*

attack.

The remainder of the paper is organized as follows. In Section II, we describe the *iLOC* attack in detail. In Section III, a formal analysis of attack accuracy and invisibility, and the impacts of various parameters on the performance of *iLOC* attack are presented. In Section IV, we introduce our implementation of the *iLOC* attack and the validation in the real-world experiments. In Section V, we report our performance evaluation results on the *iLOC* attack. In Section VI, we discuss some preliminary countermeasures against the *iLOC* attack. In Section VII, we review the related work. Finally, we conclude the paper in Section VIII.

II. *iLOC* ATTACK

In this section, we will present the *iLOC* attack in detail. We will first give an overview of the *iLOC* attack, and then introduce the detailed procedures of the attack, followed by discussions. Table I summarizes the notations used in this paper.

A. Overview

Fig. 1 shows the basic workflow of the *iLOC* attack and the basic idea of the ITM system. In the ITM system, monitors deployed at various networks record their observed port-scan traffic and continuously update their traffic logs to the data center. The data center first summarizes the volume of port-scan traffic towards (and reported by) all monitors, and then publishes the report data to the public in a timely fashion. In this paper, *background traffic* refers to aggregate traffic collected by the data center but not generated by *iLOC* attacks.

As shown in Fig. 1 (a) and (b) respectively, the *iLOC* attack consists of the following two stages: (i) *Attack Traffic Generation*: In this stage, as shown in Fig. 1 (a), the attacker first selects a code and encodes the attack traffic by *embedding* a selected code. The attacker then launches the attack traffic towards a target network (e.g., network *A* in Fig. 1 (a)). We denote such an *embedded code pattern* in the attack traffic as the *attack mark* of the *iLOC* attack, and denote the attack traffic encoded by the code as *attack mark traffic*. (ii) *Attack Traffic Decoding*: In this stage, as shown in Fig. 1 (b), the attacker first queries the data center for the traffic report data. Such report data consist of both attack traffic and background traffic. Given the report data, the attacker tries to recognize the attack mark (i.e., the code embedded in the *iLOC* attack traffic) by decoding the report data. If the attack mark is recognized, the report data must include the attack traffic, which means the target network is deployed with monitors and the monitors are sending traffic reports to the data center of ITM systems.

Code-based Attack: The *iLOC* attack adopts a code-based approach to generate the attack traffic. Coding techniques have been widely implemented in secured communications; for example, *Morse code* is one such example. Without knowledge of Morse code, it is impossible for the receiver to interpret the carried information [20]. In the *iLOC* attack, the PN-code-based approach we adopt has three advantages. First, the code is embedded in traffic and can be correctly recognized by the attacker even with the interference from background traffic. This favors the attack accuracy. Second, the code of sufficient

length provides enough privacy. That is, the code is only known by the attacker, and thereby the code pattern embedded in attack traffic can only be recognized by the attacker. Lastly, the code is able to carry information. A longer code is more immune to interference, and requires comparatively lower-rate attack traffic as the carrier, which is harder to detect. All these characteristics contribute help to achieve the objectives of attack accuracy and invisibility.

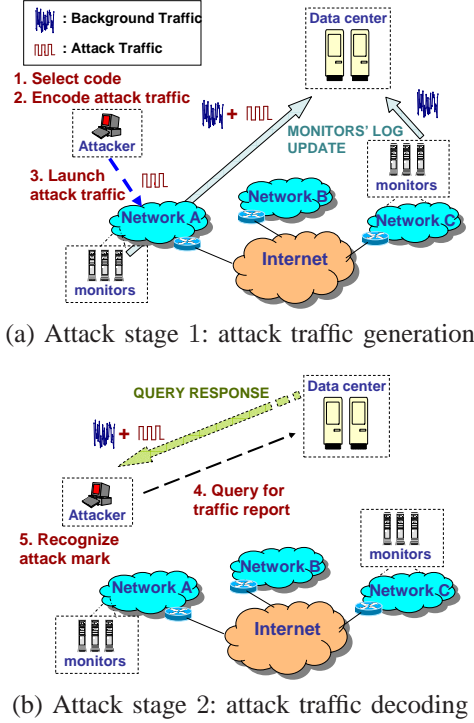


Fig. 1. Workflow of the *iLOC* Attack

Parallel Attack Capacity: Intuitively, one simple way to achieve this parallel attack is to launch port-scan/attack traffic towards multiple target networks simultaneously by scanning a different port for each target network. For example, if the data center publishes traffic reports of 1000 (TCP/UDP) ports, then the attacker can identify all these 1000 networks simultaneously, attacking each network with a different port. Since attack traffic on different ports are summarized separately at the data center, the attacker can still separate, and thus decode, the traffic towards different targets. The attacker, therefore, can localize monitors in multiple networks simultaneously and accurately; however, can the attacker further improve the attack efficiency? Assuming that the data center only publishes reports of 1000 ports, can the attacker fingerprint 10,000 target networks simultaneously, for example, by attacking 10 different networks using *one* port? A high-rate port-scan traffic cannot achieve this as it is indiscernible whether a spike in the traffic report is caused by traffic logs from one network or the other 9 networks. In order to achieve this goal in the code-based attack, the selected code and corresponding encoded attack traffic towards multiple networks for the same port should not interfere with each other (i.e., each of them can be decoded *individually* and *accurately* by the attacker, although they are integrated/summarized in the traffic report from the ITM data center). The PN-code used by the *iLOC* attack can

target multiple networks by launching probing traffic on the same port simultaneously. This unique feature can improve the attack efficiency significantly. The details of how to select the PN-code will be discussed in the following sections.

B. Attack Traffic Generation Stage

In this attack stage, the attacker: (i) selects the code, a *PN-code* in our case; (ii) encodes the attack traffic using the selected PN-code; and (iii) sends the encoded attack traffic towards the target network. In the third step, the attacker can coordinate a large number of compromised bots to generate the attack traffic [21]; however, this is not the focus of this paper. In the remaining sections, we will focus on the first and second steps.

1) **Code Selection:** To evade detection, the attack traffic should be similar to the background traffic. From a large set of real-world traffic traces obtained from SANs ISC [6], [22], we conclude that the background traffic shows random patterns in both the time and frequency domains. The attack objectives of both accuracy and invisibility, and an attacker's desire for parallel attacks require that: (i) the encoded attack traffic should blend in with background traffic, i.e., be random in both the time and frequency domains, (ii) the code embedded in the attack traffic should be easily recognizable to the attacker alone, and (iii) the code should support parallel attacks on the same port.

To meet the above requirements, we choose the PN-code to encode the attack traffic. The PN-code in the *iLOC* attack is a sequence of -1 or $+1$ with the following features [23], [24], [25]: (i) The PN-code is random and "balanced". The -1 and $+1$ are randomly distributed and the occurrence frequencies of -1 and $+1$ are nearly equal. This feature contributes to good spectral density properties (i.e., equally spreading the energy over all frequency-bands). It makes the attack traffic appear as noise and blend in with background traffic in both time and frequency domains. In Appendix A, we show that the traffic encoded by a PN-code is much similar to the traffic without including the attack traffic. (ii) The PN-code has a high correlation to itself and a low correlation to others (such as random noise), where the correlation is a mathematical utility for finding repeating patterns in a signal [25]. This makes it feasible for the attacker to accurately recognize attack traffic (encoded by the PN-code) from the traffic report data, even under the interference of background traffic. (iii) The PN-code has a low cross-correlation value among different PN-code instances. The lower this cross-correlation value, the less interference among multiple attack sessions in parallel attack. This makes it feasible for the attacker to conduct parallel attacks towards multiple target networks on the same port.

There are mature PN-code generators such as m-sequences code, Barker code, gold codes and Hadamard-Walsh codes [23], [24]. In this paper, we use the m-sequence code, which has the best autocorrelation (it only highly correlates to itself with a sharp autocorrelation peak) [23], [26]. We use the *feedback shift register* to repeatedly generate the M-sequence PN-code due to its popularity and ease of implementation [26]. In particular, a feedback shift register consists of two parts.

One is an ordinary shift register consisting of a number of flip-flops (two state memory devices). The other is a feedback module to form a multi-loop feedback logic.

TABLE I
NOTATIONS

Notation	Definition
L	Number of bits in the PN-code
$C_i = \langle C_{i,1}, \dots, C_{i,L} \rangle$	Vector to present i^{th} PN-code of length L in which each bit $C_{i,j}$ is either -1 or $+1$
T_q	Time unit of each queried data lasts for
T_s	Mark-bit duration as a unit time which is mapped to a single bit of PN-code in encoded attack traffic
V	Mark traffic rate as the high rate of encoded attack traffic in T_s
$P \in [0, 1]$	Attack traffic rate defined as ratio of attack traffic rate over the background traffic rate
(E_x, σ_x)	Statistical profile of background traffic rate (e.g., mean E_x and standard deviation σ_x) for port x
$\psi_i = \langle \psi_{i,1}, \dots, \psi_{i,L} \rangle$	Vector to present the encoded attack traffic
$\omega_i = \langle \omega_{i,1}, \dots, \omega_{i,L} \rangle$	Vector to present the background traffic
$\lambda_i = \langle \lambda_{i,1}, \dots, \lambda_{i,L} \rangle$	Vector to present the traffic report data which are queried from the data center
$\psi'_i = \langle \psi'_{i,1}, \dots, \psi'_{i,L} \rangle$	Vector to present the shifted vector which is generated by subtracting $E(\psi_{i,j})$ from ψ_i
$\omega'_i = \langle \omega'_{i,1}, \dots, \omega'_{i,L} \rangle$	Vector to present the shifted vector which is generated by subtracting $E(\omega_{i,j})$ from ω_i
$\lambda'_i = \langle \lambda'_{i,1}, \dots, \lambda'_{i,L} \rangle$	Vector to present the shifted vector which is generated by subtracting $E(\lambda_{i,j})$ from λ_i
$\Gamma(X, Y) = X \odot Y$	Correlation degree between vector X and vector Y

2) *Attack Traffic Encoding*: During the attack traffic encoding process, each bit of the selected PN-code is mapped to a unit time period T_s , denoted as *mark-bit duration*. The entire duration of launched attack traffic (referred to as *traffic launch session*) is $T_s L$, where L is the length of the PN-code. After the attacker launches port-scans to target networks, he/she also queries the data center for the traffic report periodically. For brevity, this query interval is set to T_s . The detailed discussion of determining these parameters will be presented in Section III.

The encoding is conducted based on the following rules: each bit of the PN-code maps to a mark-bit duration T_s ; when the PN-code bit is $+1$, port-scan traffic with a high rate, denoted as *mark traffic rate* V , is generated in the corresponding mark-bit duration; when the code bit is -1 , no port-scan traffic is generated in the corresponding mark-bit duration. Thus, the attacker embeds the attack traffic with a special pattern, i.e., the *original PN-code*. Recall that, after this encoding process, the PN-code pattern *embedded* in traffic is denoted as the *attack mark*. If we use $C_i = \langle C_{i,1}, C_{i,2}, \dots, C_{i,L} \rangle \in \{-1, +1\}^L$ to represent the PN-code and use $\eta_i = \langle \eta_{i,1}, \eta_{i,2}, \dots, \eta_{i,L} \rangle$ to represent the attack traffic rate, then we have $\eta_{i,j} = \frac{V}{2} C_{i,j} + \frac{V}{2}$. That is, $\eta_{i,j} = V$ if $C_{i,j} = +1$ and $\eta_{i,j} = 0$ if $C_{i,j} = -1$ ($j = [1, L]$). Fig. 2 shows one example of the PN-code and the corresponding attack traffic encoded with the PN-code.

C. Attack Traffic Decoding Stage

In this stage, the attacker takes the following two steps: (i) The attacker queries the data center for the traffic report data,

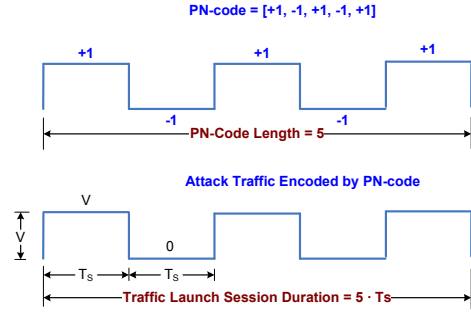


Fig. 2. PN-code and Encoded Attack Traffic

which consist of both the attack traffic and the background traffic. (ii) From the report data, the attacker attempts to recognize the embedded attack mark. The existence of the attack mark determines whether the targeted network is deployed with monitors or not. As the query of traffic report data is relatively straightforward, here we only detail the second step, i.e., attack mark recognition, as follows.

In the report data queried from the data center, the attack traffic encoded with the attack mark is mixed with the background traffic which is aggregated by the data center but not generated by *iLOC*. It is critical for the *iLOC* attack to accurately recognize the attack mark from the traffic report data. To address this, we develop a correlation-based scheme. This scheme is motivated by the fact that the original PN-code (used to encode attack traffic) and its corresponding attack mark (embedded in the traffic report data) are highly correlated: in fact, they are sharing the same pattern.

The attack mark in the traffic report data is the *embedded form* of the original PN-code. The attack mark is similar to its original PN-code, although the background traffic may introduce interference and distortion into the attack mark. We adopt the following correlation degree to measure their similarity. Mathematically, the *correlation degree* is defined as the inner product of two vectors. For two vectors $X = \langle X_1, X_2, \dots, X_L \rangle$ and $Y = \langle Y_1, Y_2, \dots, Y_L \rangle$ of length L , the correlation degree of vector X and Y is

$$\Gamma(X, Y) = \frac{\sum_{i=1}^L X_i Y_i}{L}, \quad (1)$$

where $\Gamma(\cdot)$ represents the operator for the inner product of two vectors. Based on the above definition, we have $\Gamma(X, X) = \Gamma(Y, Y) = 1, \forall X, Y \in \{-1, +1\}^L$.

We use two vectors, $\eta_i = \langle \eta_{i,1}, \eta_{i,2}, \dots, \eta_{i,L} \rangle$ and $\omega_i = \langle \omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,L} \rangle$ to represent the attack traffic (embedded with the attack mark) and the background traffic, respectively. We *shift* the above two vectors by subtracting the mean value from the original data, resulting in two new vectors, $\eta'_i = \langle \eta'_{i,1}, \eta'_{i,2}, \dots, \eta'_{i,L} \rangle$ and $\omega'_i = \langle \omega'_{i,1}, \omega'_{i,2}, \dots, \omega'_{i,L} \rangle$. We continue to use a vector $C_i = \langle C_{i,1}, C_{i,2}, \dots, C_{i,L} \rangle \in \{-1, +1\}^L$ to represent the PN-code. Thus, the correlation degree between the PN-code and the (shifted) attack traffic can be obtained. Similarly, we can also obtain the correlation degree between the PN-code and the (shifted) background traffic as follows.

According to the rules of encoding attack traffic discussed in Section II-B.2, $\eta_i = \frac{V}{2} C_i + \frac{V}{2}$. Thus, $\eta'_i = \eta_i -$

$E(\eta_i) = \eta_i - \frac{V}{2} = \frac{V}{2}C_i$. Hence, the correlation degree between the original PN-code and the (shifted) attack traffic is $\Gamma(C_i, \eta'_i) = \frac{V}{2}\Gamma(C_i, C_i) = \frac{V}{2}$. Furthermore, we can also derive the correlation degree between the PN-code and the (shifted) background traffic, i.e., $\Gamma(C_i, \omega'_i)$. The mean of such a correlation degree is close to 0, since the PN-code has low correlation with the (shifted) background traffic (i.e., $E[\Gamma(C_i, \omega'_i)] = \frac{1}{L}E[\sum_{j=1}^L(\omega'_{i,j}C_{i,j})] \approx 0$). If the standard deviation of the background traffic rate is σ_x , the variance of such a correlation degree is

$$\text{Var}[\Gamma(C_i, \omega'_i)] = E[(\Gamma(C_i, \omega'_i) - 0)^2] \quad (2)$$

$$= \frac{1}{L^2}E[\sum_{j=1}^L C_{i,j}^2 \omega'_{i,j}{}^2] \quad (3)$$

$$\approx \frac{1}{L^2}E[\sum_{j=1}^L \omega'_{i,j}{}^2] = \frac{\sigma_x^2}{L}. \quad (4)$$

Thus, the correlation degree between the PN-code and the (shifted) background traffic is $\Gamma(C_i, \omega'_i) \approx \frac{\sigma_x}{\sqrt{L}}$. Based on the above discussion, the attacker can choose appropriate attack parameters (e.g., PN-code length L and mark traffic rate V) to make the correlation degree ($\frac{V}{2}$) (between the PN-code and the attack mark traffic) much larger than the correlation degree ($\frac{\sigma_x}{\sqrt{L}}$) (between the PN-code and the background traffic). As such, the attacker can accurately distinguish the attack mark traffic from the background traffic.

In the attack mark recognition, vector λ_i is used to represent the queried report data, and vector λ'_i is used to represent the *shifted* report data (by subtracting $E(\lambda_{i,j})$ from λ_i). According to the above discussion, $\lambda'_i = \eta'_i + \omega'_i$ (i.e., report data include the attack traffic and the background traffic) or $\lambda'_i = \omega'_i$ (i.e., report data include only the background traffic). The attacker uses the correlation degree between λ'_i and the PN-code C_i , i.e., $\Gamma(C_i, \lambda'_i)$, to distinguish the above two cases and determine the existence of a PN-code in the report data. If $\Gamma(C_i, \lambda'_i)$ is larger than a threshold T_a ¹, which is referred to as the *mark decoding threshold*, then the attacker determines that the report contains attack traffic as well as the PN-code C_i , and decides whether the target network is deployed with monitors or not. The accuracy of the correlation degree-based recognition scheme is analyzed and evaluated in Sections III, IV and V.

Using real-world traces provided by SANs ISC [6], we show the results of correlation degrees in Fig. 3 and the PDF of the correlation degrees in Fig. 4. We consider three types of correlation degrees here. The first type is the correlation degree between the PN-code and the queried report data (probe mark embedded) from the data center. This type of correlation degree has a comparatively large value. We use a PN-code of length 20 and the probe mark traffic rate is equal to $0.7\sigma_x$, where σ_x is the standard deviation of the background traffic rate. The second type is the correlation degree between the PN-code and the background traffic. The correlation degree in this type is much smaller in comparison with the one in the first type. The third type is the correlation degree between

a randomly generated PN-code and the queried traffic report data from the data center. This simulates the case that the defender uses a guessed PN-code and attempts to recognize the probe mark generated by an attacker. We randomly generate 120 PN-codes with length 20 (instead of the original PN-code used to encode the attack traffic). The results show that these randomly generated codes achieve a much smaller correlation degree with the probe mark in comparison with the original PN-code. Thus, we know that the probe mark can be accurately recognized only by the attacker who knows the original PN-code. Notice that for the PN-code of length 20, the defender has a very small probability of $1/2^{20} \approx 10^{-7}$ to correctly guess the PN-code used by an attacker.

D. Discussion

In order to accurately and effectively recognize the attack mark (PN-code) from the report data, we need to find the segment of the report data containing the PN-code (i.e., we need to fulfill the synchronization between the port-scan traffic report data and the PN-code). For this purpose, we introduce a sliding window based scheme. The basic idea is to let the attacker obtain enough report data with small granularity. Then, a sliding window iteratively moves forward to capture a segment of the report data. For each segment, we apply the correlation-based scheme discussed in Section II.C to recognize whether the attack mark exists or not. The details of this synchronization is presented as follows.

As shown in Fig. 5, the attacker iteratively moves the sliding window forward. The attacker first sends a sequence of queries to the data center and each query requests a portion of report data which lasts for a given unit of time, known as query duration T_q . To guarantee good synchronization and capture each bit in the PN-code, T_q should be smaller than the mark-bit duration T_s . Also, the attacker must send enough queries to ensure that the queried report data contains the entire attack mark and attack mark traffic. The attacker iteratively conducts a correlation test on the report data using a sliding window. For example, in the i^{th} round, the attacker selects t_i as the starting time for the sliding window. In $i + 1^{\text{th}}$ round, the attacker moves the sliding window one step (T_q) further, and the start time of the sliding window becomes $t_i + T_q$, and so on. In the i^{th} round, a sequence of data (length of L) is obtained in the sliding window. The first data in the sequence is the traffic data in time duration $[t_i, t_i + T_s]$, and the second data in the sequence is the traffic data in time duration $[t_i + T_s, t_i + 2T_s]$, and so on. With this series of data, the attacker conducts the attack mark recognition procedure discussed earlier. The attacker repeats the attack mark recognition after each time, moving the sliding window forward until the attack mark is recognized, or the sliding window has gone through all the report data. According to Equation (1), the computation complexity of one round of correlation test is $O(L)$, where L is the PN-code length. Therefore, the computation complexity for performing the correlation test is $O(T_s/T_q L)$. Given such low complexity, the correlation test can be carried out in real time.

There is a trade-off in selecting the query duration T_q . On one hand, if such a duration is smaller, although the better

¹The selection of T_a is impacted by not only the values of η'_i and ω'_i , but also the desired attack accuracy, which is analyzed in Section III.

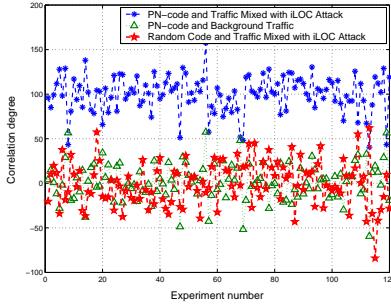


Fig. 3. Correlation Degree

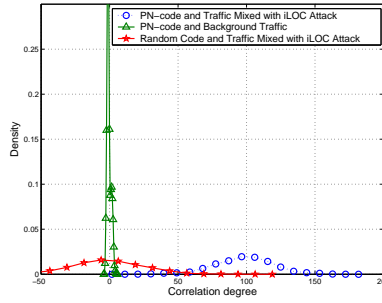


Fig. 4. PDF of Correlation Degree

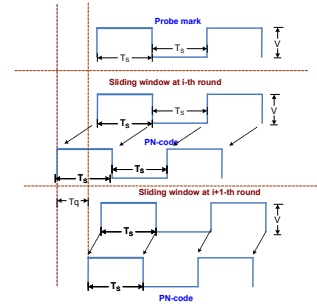


Fig. 5. Sliding Window based synchronization

synchronization accuracy can be achieved, the attacker needs to generate more queries to the data center and more iterations for the synchronization process is needed. On the other hand, if the duration is too large, the attacker might not be able to correctly synchronize the probe mark, and the probe mark recognition accuracy will be reduced. The impact of query duration on attack accuracy is shown in Section III.

III. *iLOC* ATTACK ANALYSIS AND PARAMETER DETERMINATION

Recall that there are some important parameters in *iLOC* attack, including the mark traffic rate V , the mark decoding threshold T_a , the length of PN-code L , and the mark-bit duration T_s . In this section, we first present our formal analysis of the impact of different parameters on attack objectives. The analytical results are validated by empirical results presented in Section V. Then based on such analytical results, we further discuss how to determine attack parameters.

A. *iLOC* Attack Analysis

1) *Attack Accuracy Analysis*: In order to measure attack accuracy in terms of how correctly the attacker is able to recognize the probe mark and identify monitor location, we introduce the following two metrics. The first one is the *attack success rate*, PA_D , the probability that an attacker correctly determines that a selected target network is deployed with monitors. From the attacker's perspective, the higher PA_D , the better the attack accuracy. The second metrics is the *attack false positive rate* PA_F , the probability that the attacker mistakenly determines a target network as one with monitors. From the attacker's perspective, the lower PA_F , the better the attack accuracy.

Recall that $\Gamma(\cdot)$ represents the correlation degree operator between two vectors of the same length L . Vector $C_i = \langle C_{i,1}, C_{i,2}, \dots, C_{i,L} \rangle \in \{-1, +1\}^L$ represents the PN-code. Vectors $\eta_i = \langle \eta_{i,1}, \eta_{i,2}, \dots, \eta_{i,L} \rangle$ and $\omega_i = \langle \omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,L} \rangle$ represent (probe mark embedded) attack traffic and background traffic, respectively. After subtracting mean value from the original data, the two *shifted* vectors are $\eta'_i = \langle \eta'_{i,1}, \eta'_{i,2}, \dots, \eta'_{i,L} \rangle$ and $\omega'_i = \langle \omega'_{i,1}, \omega'_{i,2}, \dots, \omega'_{i,L} \rangle$. Similarly, we use vector λ_i to represent the queried report data, and vector λ'_i to represent the *shifted* report data (by subtracting $E(\lambda'_{i,j})$ from λ_i). Assume random variables $\omega'_{i,1}, \dots, \omega'_{i,L}$ are independent and identically distributed (i.i.d) and are drawn from a Gaussian random distribution with standard deviation σ_x . Note that real Internet port-scan traffic may

not follow the Gaussian distribution. In fact, to the best of our knowledge, the traffic distribution of Internet port-scans is still an open problem and requires careful investigation. Here, we use Gaussian white-noise as an example in our theoretical analysis to provide insights into the effectiveness of *iLOC* attacks. Our simulation data based on real-world traces validate our theoretical findings well. Recall that T_a is the mark decoding threshold and V is the mark traffic rate. We have the following theorem for the *iLOC* attack accuracy. The detailed proof of this theorem can be found in Appendix B.

Theorem 1: In an *iLOC* attack, the attack success rate PA_D is

$$PA_D = 1 - Pr[\Gamma(\lambda'_i, C_i) \leq T_a | (\lambda'_i = \eta'_i + \omega'_i)] \quad (5)$$

$$= 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(T_a - T_a)}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (6)$$

where $\Gamma(\lambda'_i, C_i) = \lambda'_i \odot C_i$. The attack false positive rate PA_F is

$$PA_F = Pr[\Gamma(\lambda'_i, C_i) \geq T_a | (\lambda'_i = \omega'_i)] \quad (7)$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{\sqrt{L}T_a}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (8)$$

Notice that given the background noise ω' drawn from the Gaussian distribution, $\Gamma(\lambda', C_i)$ can be approximated by a Gaussian distribution as well. This can be reasoned as follows. Based on Equation (1), we have $\Gamma(\lambda', C_i) = \Gamma(\eta' + \omega', C_i) = \Gamma(VC_i + \omega' C_i) = V + \Gamma(\omega', C_i)$ and $\Gamma(\omega', C_i)$ can be approximated by a Gaussian distribution.

We have a few observations from the Theorem 1. First, the attack success rate, PA_D , increases and the attack false positive rate, PA_F , decreases with the increasing PN-code length L . Thus, a better attack accuracy is achieved. Second, with the increasing mark traffic rate V , a better attack success rate can be achieved as well.

2) *Attack Invisibility Analysis*: Here, *attack invisibility* refers to how invisible the *iLOC* attack is from the detection of the defender. In order to analyze invisibility, we need to consider the detection algorithms. While there have been many different algorithms proposed to detect anomalies in port-scan traffic, here we use a representative and generic algorithm which has no specific requirement on detection systems and has been widely adopted by many systems [2], [6], [27], [28]. In this algorithm, if the traffic rate (volume in a given time duration) is larger than a pre-determined threshold T_d , the *defender detection threshold*, the defender issues threat alerts and initiates reactions [6]. Such a detection

threshold is usually obtained through statistical analysis of the background traffic. Note that the threshold T_d must be chosen for anomaly detection, maintaining both the high detection rate (the probability that an ongoing attack is detected) and the low false positive rate (the probability that an alarm is triggered when no attack is occurring).

To measure attack invisibility in terms of how well the *iLOC* attack can evade detection by the defender, we use the following two metrics. The first one is the *defender detection rate* PD_D , the probability that the defender correctly detects the attack traffic introduced by the *iLOC* attack. The second one is the *defender false positive rate* PD_F , the probability that the defender mistakenly identifies the attack traffic.

Similar to our approach in Section II-B.2, we use random variable ω' to represent the *shifted* background traffic, and random variable λ' to represent the *shifted* traffic data reported by the ITM system. Note that if no *iLOC* attack exists, $\lambda' = \omega'$. If we assume that values of ω' at different time units are independent and identically distributed (i.i.d) and follow a Gaussian random distribution with standard deviation σ_x (i.e., ω' follows $N(0, \sigma_x^2)$), then we have the following theorem for attack invisibility.

Theorem 2: In the *iLOC* attack, the defender detection rate PD_D is

$$PD_D = 1 - Pr[\lambda' \leq T_d | (\lambda' = V + \omega')] \quad (9)$$

$$= 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(V-T_d)}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (10)$$

The defender false positive rate PD_F is

$$PD_F = Pr[\lambda' \geq T_d | (\lambda' = \omega')] \quad (11)$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{T_d}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (12)$$

The proof of Theorem 2 is similar to that of Theorem 1, therefore, we will skip it here due to space limitation. Notice that in Equations (9)-(12), our analysis for detection algorithm assumes that λ' is measured and compared to T_d every T_s , where T_s is the duration of 1 bit of a PN-code (also called the mark-bit duration). In reality, the defender may not have the knowledge of T_s , and this assumption helps the worst-case attack analysis in terms of the attack invisibility. Note that as researchers assume that the encryption algorithms are known to attackers in cryptanalysis [29], we assume that the strategy of mounting PN-code modulated low-rate port-scan traffic and its parameters such as mark-bit duration T_s are known to the defender. This creates the worst-case security analysis in our study. Even without knowledge of T_s , the defender can still develop adaptive strategies to carry out anomaly detection. For example, based on historical traffic logs, the defender may build the traffic statistics profile on different time durations. Then the defender measures traffic on different time durations and compares them to the traffic statistic profile on the corresponding time duration.

We have the following observations from Theorem 2. First, with the increasing mark traffic rate V , the defender detection rate PD_D increases. Thus, the attack invisibility will be worsened. Second, the mark traffic rate V does not affect the

defender false positive rate PD_F , which is only determined by the threshold T_d configured by the defender.

As we mentioned earlier, the query duration T_q will also affect attack accuracy. Recall that the recognition of probing mark is based on a sliding window as discussed in Section II-C. The maximum synchronization error between the PN-code and corresponding probe mark will be one query duration T_q as shown in Fig. 5. We know that the correlation degree between the attack traffic and PN-code is $\Gamma(C_i, \eta'_i) = C_i \odot C'_i \cdot \frac{V}{2}$, where C'_i is the result of shifting C_i by time unit T_q as shown in Fig. 5. Notice that T_q controls the maximal synchronization error. Based on the correlation degree defined in Equation (1), we have $C_i \odot C'_i = 1/L \sum_{j=1}^L C_{ij} C'_{ij}$. Since the overlapped area between C_{ij} and C'_{ij} is $T_s - T_q$, we have $C_{ij} C'_{ij} = \frac{T_s - T_q}{T_s} = 1 - \frac{T_q}{T_s}$. Then $\Gamma(C_i, C'_i) = \frac{1}{L} L (1 - \frac{T_q}{T_s})$ and $\Gamma(C_i, \eta'_i) = \frac{V}{2} (1 - \frac{T_q}{T_s})$. Similar to the proof of Theorem 1 as shown in Appendix B, the attack success rate PA_D becomes

$$PA_D = 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(\frac{V}{2}(1-\frac{T_q}{T_s})-T_r)\sqrt{L}}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (13)$$

B. Determination of *iLOC* Attack Parameters

1) *Determine V , T_a and L :* An attacker can use the above analytical results to determine attack parameters. First, the attacker can determine the mark traffic rate V . The reasons are: (i) V is only related to the attack invisibility metric (defender detection rate PD_D), and (ii) V impacts the determination of other parameters. Given the expected false alarm rate, the attacker can also determine the mark decoding threshold T_a and the PN-code length L . Notice that the parameter for the background traffic σ_x can be obtained through analyzing historical traffic data published by the data center of ITM systems.

We give the details of determining attack parameters as follows: (i) *Mark traffic rate:* Using Equation (12), the attacker can first estimate the defender threshold T_d , given a reasonable upper-bound of defender false positive rate PD_F . Notice that the T_d should be selected to be larger than the background traffic σ_x . For example, using central limitation theory, we know that $T_d = 3\sigma_x$ achieves a reasonable defender false positive rate PD_F (1.7%). Thus, we can use $3\sigma_x$ as a reasonable estimation of T_d . Given the defender detection rate PD_D , defender threshold T_d , and background traffic σ_x , the attacker can determine the mark traffic rate V by resolving Equation (10). (ii) *Mark recognition threshold and Length of PN-code:* Given the mark traffic rate V (determined previously) and attack false positive rate PA_F , and attack success rate PA_D , the attacker can further determine the mark decoding threshold T_a and L by resolving Equations (6) and (7) in Theorem 1.

Based on the above discussion, we show the determination results of attack parameters in Table II. We determine the mark decoding threshold T_a and the defender threshold T_d in order to derive a reasonable attack false positive rate PA_F and defender false positive rate PD_F (below 1%). For instance, to achieve a 95% attack success rate PA_D and 5% defender detection rate PD_D , we can use a PN-code of length $L = 20$

and a probe mark traffic rate of $V = 0.6\sigma_x$. In Section V, these numerical results are validated by our empirical evaluations.

TABLE II
DETERMINATION OF PN-CODE LENGTH AND MARK TRAFFIC RATE
(DENOTED BY (L, V))

PA_D vs. PD_D	$PD_D=5\%$	$PD_D=4\%$	$PD_D=3\%$	$PD_D=2\%$
$PA_D=94\%$	$(15, \sigma_x)$	$(24, 0.9\sigma_x)$	$(32, 0.8\sigma_x)$	$(50, 0.5\sigma_x)$
$PA_D=95\%$	$(20, \sigma_x)$	$(29, 0.9\sigma_x)$	$(42, 0.8\sigma_x)$	$(62, 0.5\sigma_x)$
$PA_D=96\%$	$(32, \sigma_x)$	$(42, 0.9\sigma_x)$	$(53, 0.8\sigma_x)$	$(76, 0.5\sigma_x)$
$PA_D=97\%$	$(55, \sigma_x)$	$(63, 0.9\sigma_x)$	$(71, 0.8\sigma_x)$	$(94, 0.5\sigma_x)$

2) *Determine T_s* : To determine the mark-bit duration T_s , the attacker needs to estimate the possible delay from the moment when attack traffic is recorded by monitors to the moment when such attack traffic is published by the data center. To make the *iLOC* attack effective, the mark-bit duration needs to be at least as large as such a delay. Otherwise, the traffic in different bit durations (each lasts T_s) may interfere with each other and make it hard to recognize the probe mark. Several possible ways can be used to estimate the delay. For example, the attacker may obtain such information through publicly available resources. Some ITM systems may publish such information on their websites. The attacker may also actively conduct experiments on ITM systems and measure the delay. For example, the attacker may install monitors and connect them to the targeted ITM system. The attacker can simply use such monitors to report logs embedded with special patterns (e.g., PN-code) and keep querying the data center until the embedded traffic patterns are recognized. After repeating the above process for a number of times, the attacker is able to derive the statistics profile of delay and then determine the mark-bit duration T_s . We use this method in our implementation of *iLOC* attack described in Section IV.

IV. IMPLEMENTATION AND VALIDATION

In this section, we first introduce our implementation of an *iLOC* attack. Then we report validation results of the *iLOC* design and implementation on a real-world ITM system.

A. Implementation of *iLOC* Attack

We implement an *iLOC* attack prototype based on the design in Section II. Recall that an attacker has two objectives: attack accuracy and invisibility. This prototype works against any ITM system with a web-based user interface. There are five independent and important components in our *iLOC* implementation as shown in Fig. 6. Our *iLOC* is implemented in Microsoft MFC and Matlab on Windows XP OS.

(1) *Data Center Querist*: This component interacts with the data center of ITM systems. Its main tasks are to send queries to the data center and to retrieve responses from the data center. The inputs to this component are the URL or IP address of the data center and port number to query. This component provides basic services to other components.

(2) *Background Traffic Analyzer*: This component receives the data of background port-scan traffic on given ports via *Data Center Querist*. With such data, this component obtains

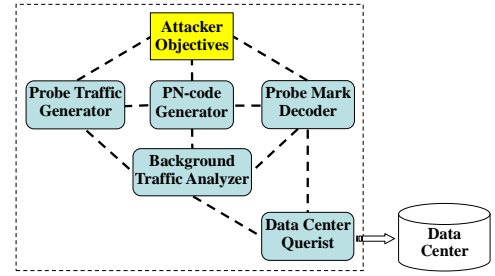


Fig. 6. *iLOC* Implementation Components

the statistics profile of background traffic, e.g., standard deviation σ_x . The profile is used to determine attack parameters for other components.

(3) *PN-code Generator*: This component generates and stores a PN-code. The PN-code length is determined according to the attacker's objectives and the background traffic profile in the way discussed in Section III-B. Recall that we use the feedback shift register to generate the PN-code as discussed in Section II-B. The feedback shift register repeatedly generates a PN-code of length L .

(4) *Probe Traffic Generator*: This component generates attack traffic based on the PN-code and the statistics profile of background traffic. With the profile of the background traffic, the attack traffic rate is determined based on the method shown in Section III-B. Then, the PN-code encoded traffic is generated in a way as discussed in Section II-B.2. Inputs to this component are the IP addresses of target network, port number, and transportation protocol (TCP or UDP).

(5) *Probe Mark Decoder*: This component obtains the port-scan report data through *Data Center Querist*, and decides whether the probe mark exists in the way discussed in Section II-C or not. The PN-code used in the decoding process is the same one used in encoding attack traffic and stored in the *PN-code Generator*. The decoding threshold is determined by this component based on the attack accuracy requirement and the background traffic profile, as explained in Section III.A.

B. Validation of *iLOC* Attack

The evaluation should be carried out over a real ITM system in an ideal situation. Since an extensive experiment on a real ITM system will affect its usability (e.g., generating skewed reports of the actual Internet traffic), in our evaluation, we considered both experiments with a real-world ITM system and simulations using off-line traffic traces. In order to validate our *iLOC* implementation, we carried out experiments with a real-world threat monitoring system, SANs ISC shown in Fig. 7. We deployed several monitors that collect port-scan logs of the monitored networks and report data to the data center of SANs ISC periodically (every half hour). We launched the probing traffic addressed to these target networks deployed with monitors and derived the fine-grained report by periodically requesting the port report from the data center. Notice that if the monitors we targeted report logs more frequently and the data center of ITM systems collects and publishes logs more frequently, we can obtain port-scan report with finer granularity.

Fig. 7 illustrates our experimental setup. For the purposes of this research, we requested information about locations of experimental monitors in this figure. We were provided with the identities of two networks A and B . There are some monitors in network A and there is no monitor in network B . The monitors in network A monitor a set of IP addresses and log the port-scans. We (the attacker) execute the *iLOC* attack to decide whether monitors exist in network A and B , respectively.

In our experiment, we use a PN-code of length 15. The mark-bit duration is set at 1 hour. We use two machines. On one machine, the *Encoding Process* sends attack traffic to network A and B , respectively. On the other machine, the *Decoding Process* sends a query to the data center every 20 minutes. With report data, we find the *Decoding Process* can correctly determine that network A is deployed with monitors and network B is not deployed with monitors. Fig. 8 shows the traffic rate in time-domain. Based on the data shown in the Fig. 8, we calculate the correlation degree. The correlation degree between the mixed traffic and the PN-code is around 28, while the correlation value between the background traffic and the PN-code is around 8. Given the detection threshold as 14, we know that the attack traffic and background traffic can be easily distinguished. Therefore, the attacker can accurately identify that network A is deployed with monitor and network B is not deployed with monitor. Fig. 9 shows the traffic rate in frequency-domain in terms of *Power Spectrum Density (PSD)*. The *PSD* describes how the power of a time series data is distributed in frequency domain. Mathematically, it is equal to the *Fourier* transform of the auto-correlation of a time series data [30]. From these two figures, we observe that it is hard for the defender to detect an *iLOC* attack, since the overall traffic with an *iLOC* attack is very similar to the traffic without *iLOC* attack traffic embedded. That is, such experiments demonstrate that the *iLOC* attack can effectively and stealthily identify monitors in reality.

V. PERFORMANCE EVALUATION

In this section, we conduct the performance evaluation by merging simulated *iLOC* attack traffic into replayed real-world traffic traces.

A. Evaluation Methodology

1) *Experiment Setup*: In our evaluation, we use the real-world port-scan traces from SANs ISC including the detail logs from 01/01/2005 to 01/15/2005 [6], [22]². The traces used in our study contain over 80 million records and the overall data volume exceeds 80 GB. We use these real-world traces as the background traffic. We merge records of simulated *iLOC* attack traffic into these traces and replay the merged data to simulate the *iLOC* attack traffic. We evaluate different attack scenarios by varying attack parameters such as the mark traffic rate V , the length of PN-code L , and the number of parallel attack sessions N (on the same port). We report the results for the cases where attacks are launched to port 4321 (representing an unpopular port with low traffic

rate), port 135 and port 25 (representing popular ports with high traffic rate). Experiments on other ports result in similar observations.

2) *Evaluation Metrics*: We explore both attack accuracy and invisibility to evaluate attack performance. For attack accuracy, we use two metrics: one is the *attack success rate* PA_D and the other is the *attack false positive rate* PA_F , which are defined in Section III-A.1. For attack invisibility, we use two metrics: one is the *defender detection rate* PD_D and the other is *defender false positive rate* PD_F , which are defined in Section III-A.2.

3) *Evaluation Schemes*: We evaluate the *iLOC* attack in comparison to two other baseline attack schemes. The first one is the attack that launches a significantly high-rate of port-scan traffic to target networks as introduced in [18], [19]. We denote this attack as a *volume-based attack*. Notice that the technique used in our simulation for the *volume-based attack* is similar to the noise cancellation technique in [18]. However, the work in [18] did not provide much detail on how to choose the noise cancellation factor. In contrast, given the random and burst nature of data report as shown in Fig. 17, our scheme is more general. The second baseline scheme embeds the attack traffic with a unique frequency pattern. In this attack, the attack traffic rate changes periodically. Then the attacker expects that the report data from the data center show such a unique frequency pattern if the selected target network is deployed with monitors. We denote this attack scheme as a *frequency-based attack*. All three evaluated attack schemes are listed in Table III.

For fairness, we adjust the detection thresholds in all schemes so that the desired *attack false positive rate* PA_F and *defender false positive rate* PD_F (below 1%) are achieved. For the *iLOC* attack, we generate different attack traffic based on a variant PN-code length L (i.e., 15, 30, 45). The default PN-code length is set to 30. To better quantify the attack traffic rate for the *iLOC* attack and other attack schemes, we use the normalized attack traffic rate P , which is defined as $P = V/\sigma_x$ for an *iLOC* attack, where σ_x is the standard variation for background traffic rate. The default value of $T_q = 0.1T_s$.

TABLE III
PROBE ATTACK SCHEMES

Notation	Description
<i>iLOC</i>	Our camouflaged probe attack scheme
Frequency-based probe	Frequency-based probe attack scheme [31]
Volume-based probe	Volume-based probe attack scheme [18][19]

B. Evaluation Results

1) *Attack Accuracy*: To compare the attack accuracy of the *iLOC* attack with that of volume and frequency-based probe schemes, we plot the attack success rate PA_D under different attack traffic rates (e.g., $P = [0.01, 3]$). Fig. 10, Fig. 11 and Fig. 12 show the results on different ports. From these figures, we observe that both *iLOC* and frequency-based attacks consistently achieve a much higher attack success rate PA_D than the volume-based scheme. This performance improvement is more significant when the attack traffic rate is lower. The reason can be explained as follows. For the

²We thank the ISC for providing us valuable traces in this research.

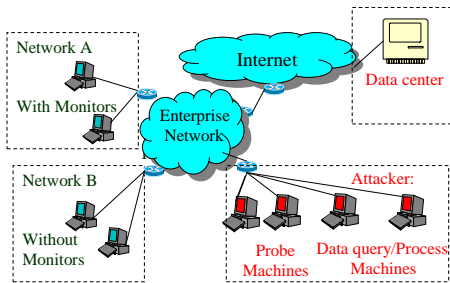


Fig. 7. Experiment Setup

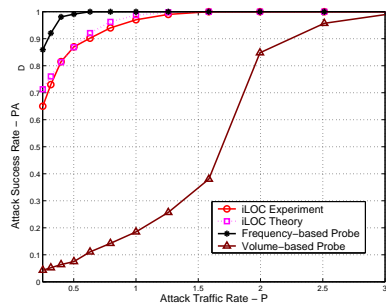


Fig. 10. Attack Success Rate (Port 4321)

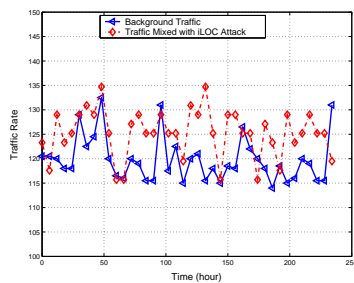
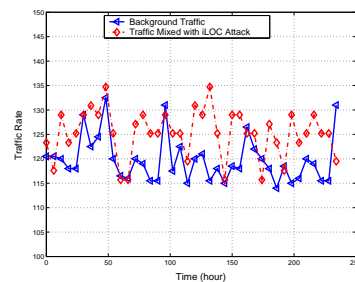
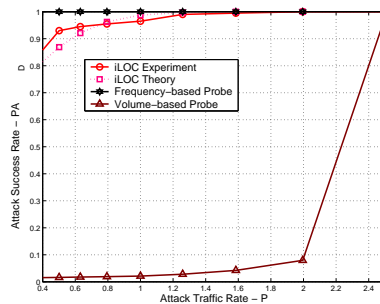
Fig. 8. Background Traffic vs. Traffic Mixed with *iLOC* AttackFig. 9. PSD for Background Traffic vs. Traffic Mixed with *iLOC* Attack

Fig. 11. Attack Success Rate (Port 135)

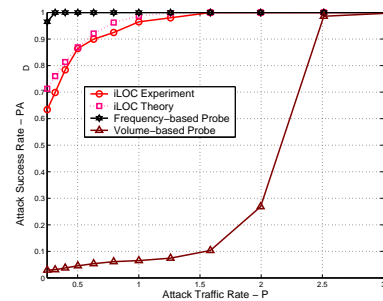


Fig. 12. Attack Success Rate (Port 25)

iLOC scheme, the PN-code based encoding/decoding makes the recognition of probe marks robust to interference from the background traffic. For the frequency-based scheme, the invariant frequency in the attack traffic is also robust to the interference from the background traffic. But the volume-based scheme relies on a high rate of attack traffic.

2) *Attack Invisibility*: To compare the attack invisibility of the *iLOC* attack with that of the other two attack schemes, we show the defender detection rate PD_D on different ports (e.g., 4321, 135, and 25) in Table IV, V, and VI. Each table shows the defender detection rate PD_D , given an attack success rate PA_D (90%, 95%, and 98%). Recall that the defender sets the detection threshold to make the defender false positive rate PD_F below 1%. In all tables, “(Time)” and “(Freq)” mean that the defender adopts the *time-domain* and *frequency-domain* analytical techniques to detect attacks, respectively. It is observed that our *iLOC* scheme consistently achieves a much lower defender detection rate PD_D than that of the other two schemes. Therefore, the *iLOC* attack achieves the best attack invisibility performance. As expected, the defender can easily detect the frequency-based attack, as a unique frequency pattern exists in attack traffic.

3) *Impact of the Length of PN-code*: To investigate the impact of the PN-code length on the performance of the *iLOC* attack, we show the attack success rate PA_D for a PN-code of different lengths (e.g., 15, 30, 45) in Fig. 13. Data are also collected for various attack traffic rates. In the legend, $iLOC(L = x)$ means that the PN-code length is x . This figure shows that the attack success rate PA_D increases with the increasing PN-code length because a long PN-code reduces the interference from the background traffic on recognizing the probe mark, and thereby improves attack accuracy.

4) *Impact of the Number of Parallel Localization Attacks*: To evaluate the impact of the number of parallel localization attacks on attack accuracy, we show the attack success rate

TABLE IV

DEFENDER DETECTION RATE PD_D (PORT 4325)					
PA_D	<i>iLOC</i> (Time)	<i>iLOC</i> (Freq)	Volume probe(Time)	Freq probe(Freq)	Freq probe(Time)
90%	2.2%	2.3%	90%	90%	2.6%
95%	2.7%	2.6%	95%	95%	2.7%
98%	3%	2.9%	98%	98%	2.9%

TABLE V

DEFENDER DETECTION RATE PD_D (PORT 135)					
PA_D	<i>iLOC</i> (Time)	<i>iLOC</i> (Freq)	Volume probe(Time)	Freq probe(Freq)	Freq probe(Time)
90%	2.5%	2.2%	90%	90%	2.9%
95%	2.8%	2.4%	95%	95%	3.1%
98%	3.1%	2.8%	98%	98%	3.3%

PA_D for a variety of parallel attack sessions on the same port in Fig. 14. In the legend, $iLOC(N = x)$ means that there are x parallel attack sessions. This figure shows that in terms of the attack success rate PA_D , the *iLOC* attack scheme is not sensitive to the number of parallel attack sessions. The attack success rate PA_D only slightly decreases with the increasing number of parallel attack sessions. This is because the traffic for different attack sessions is encoded by PN-codes which are low cross-correlated (described in Section II-B), and thereby have little interference. Fig. 15 shows the impact of the number of parallel attack sessions on attack invisibility. It can be observed that the increasing number of parallel attack sessions results in a slight increase in the defender detection rate PD_D .

TABLE VI

DEFENDER DETECTION RATE PD_D (PORT 25)					
PA_D	<i>iLOC</i> (Time)	<i>iLOC</i> (Freq)	Volume probe(Time)	Freq probe(Freq)	Freq probe(Time)
90%	2.1%	1.9%	93%	94%	2.2%
95%	2.4%	2.1%	95%	97%	2.8%
98%	3.0%	2.3%	96%	98%	3.1%

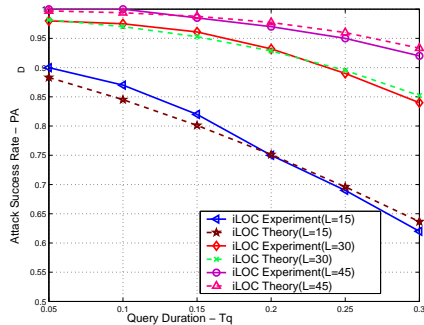


Fig. 16. Attack Accuracy vs. Query Duration T_q

Therefore, parallel attack capability can significantly improve the attack efficiency without compromising the effectiveness.

The *iLOC* attack achieves invisibility by using the PN-code, which causes a longer period for the *iLOC* attack than the ones in [18], [19]. Nevertheless, parallel features of *iLOC* attack can significantly improve attack efficiency. In the following, we provide one example to compare the efficiency of our attack with the one in [18], [19]. This example demonstrates that our attack is slower than the one in [18], [19] and the parallel feature of our attack can effectively reduce the performance gap between our attack and the one in [18], [19]. Assume that a system that consists of 1200 networks is attacked. Using one port, the volume-based attack needs 1200 time units to perform the attack task. To fulfill the same attack task, *iLOC* with 4 attack sessions in parallel using a code length of 15 can achieve the desired performance of attack accuracy and invisibility as shown in Fig. 14. In this case, the total time for *iLOC* attack is $1200 \times 15/4 = 4500$ units, which is around four times of the volume-based attack in [18], [19].

5) *Impact of Query Duration on Attack Accuracy*: To investigate the impact of the query duration T_q on the *iLOC* attack accuracy, we show the attack success rate PA_D under different query durations ($T_q = [0.05T_s, 0.3T_s]$) in Fig. 16. In the legend, *iLOC*($L = x$) refers to a PN-code of length x . From this figure, we observe that, with the decreasing query duration T_q , the attack success rate PA_D increases. The reason is that a smaller query duration improves synchronization granularity and thus the attack has a better chance to recognize the probe mark. Hence the attack accuracy will be improved. However, the smaller query duration T_q will also increase the number of queries sent to the data center and the synchronization time for recognizing the attack mark.

VI. GUIDELINES OF COUNTERMEASURE

We have demonstrated the *iLOC* attack against ITM systems. Let us discuss possible countermeasures against such an attack. It is relatively easy to defend against the volume-based and frequency-based localization attacks which embed either a spike pattern (using a high-rate scan traffic) [18], [19] or an invariable frequency pattern (using the attack embedded with a certain frequency pattern), since these two attack schemes show strong signatures in the attack traffic (either in time domain or in frequency domain). However, in order to defend against the *iLOC* attack, the defender needs the insightful understanding of the attack. We provide several

general guidelines for counteracting the *iLOC* attack from the following aspects.

1) *Limiting the Information Access Rate*: Recall that in the *iLOC* attack, the attacker must generate a significant amount of queries to the data center of ITM systems in order to accurately recognize the encoded attack traffic. We may explore such knowledge to reduce the effectiveness of an *iLOC* attack. To do so, the data center may throttle the query request rate. One possible way is to enforce human/system interaction for each query, and thereby eliminate the automatic query in the *iLOC* attack. This can be conducted through authenticated registration, e.g., one authenticated registration is only valid for a certain number of queries. However, these limitations on the information access rate may also reduce the usability of ITM systems.

2) *Perturbing the Information*: Recall that in the *iLOC* attack, the attacker needs to recognize the encoded attack traffic. Thus, the quality of reports plays an important role in such a recognition process. To reduce the effectiveness of an *iLOC* attack, we may perturb the published report data by adding some random noise or randomizing the data publishing delay. This scheme is similar to the data perturbation in the private data sharing realm [32], [33], [34]. By perturbing report data, the attack accuracy of an *iLOC* attack will be degraded. However, adding random noise and randomizing the delay in publishing report data will also affect the data accuracy and usability of ITM systems. Studying such a trade-off will be one aspect future work.

3) *Investigating Advanced Detection Schemes*: Recall that in the *iLOC* attack, in order to effectively evade detection of monitors in ITM systems, the attacker has to continuously launch port-scan attack traffic to different target networks to localize as many monitors as possible. Consequently, the target IP addresses of attack traffic may exhibit a widely dispersed distribution [35]. Thus, analyzing the distribution of IP addresses may provide one possible method of detection. Additionally, in [36], we proposed an information-theoretic framework to analyze *iLOC* attacks. In particular, we modeled the *iLOC* attack based on a communication channel and derived closed formulae for the capacity of *iLOC* attacks. Based on this framework, we studied two different kinds of *iLOC* attacks, which encode the probing traffic in either the temporal domain (the scheme studied in this paper) or the spatial domain (on multiple monitors). We also investigated the effectiveness of possible detection strategies, including centralized, distributed, and hybrid detection.

VII. RELATED WORK

Many ITM systems have been developed and deployed since CAIDA initiated the network telescope project to monitor background traffic in 2001 [37]. The ITM system is similar to the knowledge sharing of distributed intrusion detection [38]. Although the IP addresses of monitors themselves can be protected by mechanisms, such as encryption and Bloom filters [39], the public data reported by these ITM systems could be used to disclose the IP address space covered by monitors. Existing attack approaches achieve this by launching

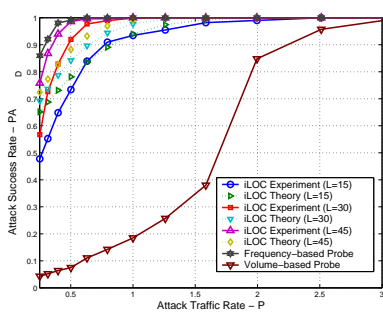


Fig. 13. Attack Success Rate vs. Code Length

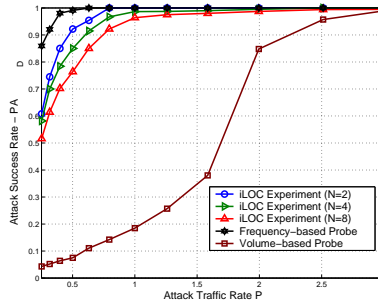


Fig. 14. Attack Success Rate vs. Number of Parallel Attack Sessions on the Same Port

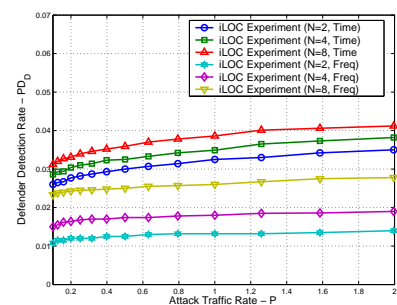


Fig. 15. Defender Detection Rate vs. Number of Parallel Attack Sessions on the Same Port

high-rate port-scan traffic [18], [19]. However, these kinds of attacks do not consider the invisibility of attacks, since the high-rate attack traffic exposes the attack.

The invisibility techniques in our work uses the camouflage principle, as illustrated by nature and the military. In nature, an animal can disguise itself as the object on which it stands in order to fool its predators or prey [40]. In military, soldiers wear camouflage clothing designed to blend into the surrounding terrain [41]. As an invisibility technique, our work leverages the PN-code technology and extends it to a new Internet cybersecurity realm. The PN-code was initially used in military communication systems to provide anti-jamming and secured communication [23]. In wireless communication, the PN-code has been widely used to improve the communication efficiency [24]. In addition, the PN-code has other broad applications, such as cryptography [42], secured data storage and retrieving [43], and image processing [44].

Our work is related to robust watermarking. There are some researches on how to design robust watermarking for specific applications. For example, Li *et al.* in [45] developed a watermarking scheme that allows the owner to publish a large number of media files; provides the owner the ability to detect watermarks; and prevents the owner from cheating by ambiguity attacks. Some research has focused on breaking digital watermarks and developing countermeasures. For example, Arnold in [46] presented a classification of attacks against digital watermarks along with countermeasures. They categorized attacks into different categories, such as removing, desynchronization, and noise-embedding. Briassouli and Mouline in [47] evaluated the effects of a desynchronizing warp attack (e.g., time-varying delay) on the performance of detecting watermarks. Li *et al.* in [48] proved that the worst-case additive attack (deliberately adding noise to degrade the watermark detection) against a watermark is a $3 - \delta$ function (δ is the distortion compensation factor). There is other work related to the digit steganography [49], [50], which intends to hide the presence of information despite its practical relevance for digital content (e.g., image, video) protection using watermarking and fingerprinting schemes.

Our work is also related to the covert channel. Various covert channels have been studied [51], [52], [53]. For example, JitterBugs is a class of inline interception mechanisms that covertly transmit data by perturbing the timing of input events in order to affect externally observable network traffic [52]. Takahshi *et al.* in [54] assessed VoIP covert channel

threats that utilize an IP phone conversation to illicitly transfer information across the network. In our study, the sequence of attack traffic to the monitor, transmission of log information to the data center, and the transmission of query data back to the attacker forms a covert channel for the attacker to discover the location of monitors. Our work presents a deep study of PN-code based localization attacks, addressing both accuracy and secrecy.

In this paper, we study techniques in applying the PN-code in the *iLOC* attack. Work in [16] also studied how to use PN-code to effectively track anonymous flows through mix networks. Since it is applied to a different problem domain, the solution in [16] is significantly different from the one in this paper, including the use of the PN-code, designed algorithms, decision rule, and theoretical analysis.

VIII. CONCLUSION

In this paper, we investigated a new class of attacks, i.e., the *invisible LOCALization (iLOC)* attack. It can accurately and invisibly localize monitors of ITM systems. Its effectiveness is demonstrated by theoretical analysis, simulations, and experiments with an implemented prototype. We believe that this paper lays the foundation for ongoing studies of attacks that intelligently adapt attack traffic to avoid the detection by defense systems. Our study is critical for securing ITM systems. Since the attacker has a large space to improve the secrecy of the attack, the detection of such an invisible attack remains a challenging task. Comprehensive study of other methods to protect the location of monitors is a part of our future work.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their invaluable feedback. This work was supported in part by the National Science Foundation under grants 0808419, 0324988, 0546668 and 0721766. This work was also supported in part by Army Research Office (ARO) under grant no. AMSRD-ACC-R50521-CI. 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 National Science Foundation and Army Research Office. The authors would like to acknowledge Ms. Larisa Archer for her dedicated editorial help to improve the paper.

REFERENCES

- [1] D. Moore, C. Shannon, and J. Brown, "Code-red: a case study on the spread and victims of an internet worm," in *Proceedings of the 2nd Internet Measurement Workshop (IMW)*, Marseille, France, November 2002.
- [2] D. Moore, V. Paxson, and S. Savage, "Inside the slammer worm," *IEEE Magazine of Security and Privacy*, vol. 1, no. 4, pp. 33–39, 2003.
- [3] *W32/MyDoom.B Virus*, <http://www.us-cert.gov/cas/techalerts/TA04-028A.html>.
- [4] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–54, 2004.
- [5] *Internet Security News*, <http://www.landfield.com/isn/mail-archive/2001/Feb/0037.html>.
- [6] SANS, *Internet Storm Center*, <http://isc.sans.org/>.
- [7] D. Moore, G. M. Voelker, and S. Savage, "Inferring internet deny-of-service activity," in *Proceedings of the 10th USENIX Security Symposium (SECURITY)*, Washington, DC, August 2001.
- [8] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the domino overlay system," in *Proceedings of the 11th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2004.
- [9] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and utility of internet sinks for network abuse monitoring," in *Proceeding of Symposium on Recent Advances in Intrusion Detection (RAID)*, Pittsburgh, PA, September 2003.
- [10] D. Moore, "Network telescopes: Observing small or distant security events," in *Invited Presentation at the 11th USENIX Security Symposium (SECURITY)*, San Francisco, CA, August 2002.
- [11] *Dynamic Graphs of the Nimda Worm*, <http://www.caida.org/dynamic/analysis/security/nimda>.
- [12] myNetWatchman, *myNetWatchman Project*, <http://www.mynetwatchman.com>.
- [13] L. Spitzner, *Know Your Enemy: Honeynets*, HoneyNet Project, <http://project.honeynet.org/papers/honeynet>.
- [14] N. Provos, "Honeyd - a virtual honeypot daemon," in *Proceedings of the 10th DFN-CERT Workshop*, Hamburg, Germany, February 2003.
- [15] J. Twocrps and M. M. Williamson, "Implementing and testing a virus throttling," in *Proceedings of the 12th USENIX Security Symposium (SECURITY)*, Washington, DC, August 2003.
- [16] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2007.
- [17] V. Sekar, Y. Xie, D. Maltz, M. Reiter, and H. Zhang, "Toward a framework for internet forensic analysis," in *Proceeding of the 3rd Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, November 2004.
- [18] J. Bethencourt, J. Frankin, and M. Vernon, "Mapping internet sensors with probe response attacks," in *Proceedings of the 14th USENIX Security Symposium (SECURITY)*, Baltimore, MD, July-August 2005.
- [19] Y. Shinoda, K. Ikai, and M. Itoh, "Vulnerabilities of passive internet threat monitors," in *Proceedings of the 14th USENIX Security Symposium (SECURITY)*, Baltimore, MD, July-August 2005.
- [20] L. Y. Chuang, C. H. Yang, C. H. Yang, and S. L. Lin, "An interactive training system for morse code users," in *Proceedings of Internet and Multimedia Systems and Applications*, Honolulu, Hawaii, August 2002.
- [21] R. Naraine, *Botnet Hunters Search for Command and Control Servers*, <http://www.eweek.com/article2/0,1759,1829347,00.asp>.
- [22] Dshield, *Distributed Intrusion Detection System*, <http://www.dshield.org/>.
- [23] R. K. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communication - tutorial," *IEEE Transaction on Communication*, vol. 30, no. 5, pp. 855–884, 1982.
- [24] E. J. Crusellers, M. Soriano, and J. L. Melus, "Spreading codes generator for wireless cdma network," *International Journal of Wireless Personal Communications*, vol. 7, no. 1, 1998.
- [25] Robert Dixon, *Spread Spectrum Systems, 2nd Edition*, John Wiley & Sons, 1984.
- [26] Nova Engineering, *Linear Feedback Register Shift*, <http://www.sss-mag.com/pdf/lfsr.pdf>.
- [27] S. Venkataraman, D. Song, P. Gibbons, and A. Blum, "New streaming algorithms for superspreader detection," in *Proceedings of the 12th IEEE Network and Distributed Systems Security Symposium (NDSS)*, San Diego, CA, February 2005.
- [28] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium (SECURITY)*, San Francisco, CA, August 2002.
- [29] "Cryptanalysis," <http://en.wikipedia.org/wiki/Cryptanalysis>.
- [30] R. L. Allen and D. W. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*, Wiley and Sons, 2004.
- [31] X. Fu, Y. Zhu, B. Graham, R. Bettati, and W. Zhao, "On flow marking attacks in wireless anonymous communication networks," in *Proceeding of the 24th International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, March 2004.
- [32] N. Zhang, S. Wang, and W. Zhao, "A new scheme on privacy preserving association rule mining," in *Proceeding of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Pisa, Italy, September 2004.
- [33] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private database," in *Proceeding of the 22th SIGMOD International Conference on Management of Data*, San Diego, CA, July 2003.
- [34] N. Zhang and W. Zhao, "Privacy-preserving data-mining systems," *IEEE Computer*, vol. 40, no. 4, April 2007.
- [35] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distribution," in *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.
- [36] W. Yu, N. Zhang, X. Fu, R. Bettati, and W. Zhao, "On localization attacks to internet threat monitors: An information-theoretic framework," in *Proceedings of IEEE International Conference on Dependable Systems and Networks (DSN) (Performance and Dependability Symposium - PDS)*, Anchorage, Alaska, June 2008.
- [37] CAIDA, *Telescope Analysis*, <http://www.caida.org/analysis/security/telescope>.
- [38] H. Debar and A. Wespi, "Aggregation and correlation of intrusion-detection alerts," in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, Davis, CA, October 2001.
- [39] P. Gross, J. Parekh, and G. Kaiser, "Secure selectcast for collaborative intrusion detection systems," in *Proceedings of the 3rd International Workshop on Distributed Event-based Systems (DEBS)*, May 2004.
- [40] A. Anderson, A. Johnston, and P. McOwan, *Motion Illusions and Active Camouflaging*, http://www.ucl.ac.uk/ucbplrd/motion/motion_middle.html.
- [41] Chief of Engineers, *United States Army: Army facilities components system user guide*, <http://www.usace.army.mil/inet/usace-docs/armymtm/tm5-304/>, October 1990.
- [42] M. Bellare, S. Goldwasser, and D. Miccianciom, "Pseudo-random number generation within cryptographic algorithms: the dss case," in *Proceedings of advances in cryptology'97, Lecture Notes in Computer Science*, Springer-Verlag, May 1997.
- [43] L. Wang and B. B. Hirsbrunner, "Pn-based security design for data storage," in *Proceedings of Databases and Applications*, Innsbruck, Austria, February 2004.
- [44] X. G. Xia, C. G. Boncele, and G. R. Arce, "A multiresolution watermark for digital images," in *Proceedings of International Conference on Image Processing (ICIP)*, Washington, DC, October 1997.
- [45] Q. M. Li and E. C. Chang, "Zero-knowledge watermark detection resistant to ambiguity attacks," in *Proceedings of the 8th ACM workshop on Multimedia and Security (MMSEC)*, NY, September 2006.
- [46] M. Arnold, "Attacks on digital audio watermarks and countermeasures," in *Proceedings of the 3rd IEEE International Symposium on Web Delivering of Music*, Florence, Italy, September 2003.
- [47] Alexia Briassouli and Pierre Moulin, "Detection-theoretic analysis of warping attacks in spread-spectrum watermarking," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, P.R.China, April 2003.
- [48] N. Liu and K. P. Subbalakshmi, "Worst case attack on quantization based data hiding," in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM)*, San Diego, CA, December 2006.
- [49] C. Cachin, "Digital steganography," <http://www.zurich.ibm.com/~cca/papers/encyc.pdf>, 2005.
- [50] N. Provos, "Defending against statistical steganalysis," in *Proceedings of the 10th USENIX Security Symposium (SECURITY)*, Washington, D.C., August 2001.
- [51] S. Cabuk, C. Brodley, and C. Shields, "Ip covert timing channels: design and detection," in *Proceedings of the 2004 ACM Conference on Computer and Communications Security (CCS)*, Washington D.C., October 2004.
- [52] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in *Proceedings of the 15th USENIX Security Symposium (SECURITY)*, San Diego, CA, July-August 2006.

- [53] D. Bailey, D. Boneh, E.-J. Goh, and A. Juels, "Covert channels in privacy-preserving identification systems," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, November 2007.
- [54] T. Takahashi and W. Lee, "An assessment of voip covert channel threats," in *Proceedings of the IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, Nice, France, September 2007.

APPENDIX B. PROOF OF THEOREM 1

i) Derivation of attack success rate PA_D . The attack success rate PA_D is the probability that an attacker correctly recognizes the fact whether a selected target network is deployed with monitors. Following this definition, we have

$$PA_D = 1 - Pr[\Gamma(\lambda'_i, C_i) \leq T_a | (\lambda'_i \eta'_i + \omega'_i)] \quad (14)$$

$$= 1 - Pr[\Gamma(\lambda'_i, C_i) \leq T_a - \frac{V}{2} | (\lambda'_i = \omega'_i)]. \quad (15)$$

Then PA_D can be represented by

$$PA_D = 1 - \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{T_a - \frac{V}{2}} e^{-\frac{x^2 L}{2\sigma_x^2}} dx. \quad (16)$$

Let $y^2 = \frac{x^2 L}{2\sigma_x^2}$ and $y = \frac{x\sqrt{L}}{\sqrt{2}\sigma_x}$. Then we have

$$PA_D = 1 - \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{\frac{(T_a - \frac{V}{2})\sqrt{L}}{\sqrt{2}\sigma_x}} \frac{\sqrt{2}\sigma_x}{\sqrt{L}} e^{-y^2} dy \quad (17)$$

$$= 1 - \frac{1}{\sqrt{\pi}} \int_{\frac{(\frac{V}{2} - T_a)\sqrt{L}}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (18)$$

ii) Derivation of attack false positive rate PA_F . The attack false positive rate PA_F is the probability that an attacker mistakenly identifies a selected target network as being deployed with monitors. If $\Gamma(\lambda'_i, C_i)$ follows a Gaussian distribution $N(0, \sigma_x^2/L)$ (for detail see Equation (2) in Section II-C), we have $PA_F = Pr[\Gamma(\lambda'_i, C_i) \geq T_a | (\lambda'_i = \omega'_i)]$. Thus, the PA_F can be presented by

$$PA_F = \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{T_a}^{\infty} e^{-\frac{x^2 L}{2\sigma_x^2}} dx. \quad (19)$$

Let $y^2 = \frac{x^2 L}{2\sigma_x^2}$ and $y = \frac{\sqrt{L}x}{\sqrt{2}\sigma_x}$, we have

$$PA_F = \frac{\sqrt{L}}{\sqrt{2\pi}\sigma_x} \int_{\frac{\sqrt{L}T_a}{\sqrt{2}\sigma_x}}^{\infty} (e^{-y^2} \frac{\sqrt{2}\sigma_x}{\sqrt{L}}) dy \quad (20)$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{\sqrt{L}T_a}{\sqrt{2}\sigma_x}}^{\infty} e^{-y^2} dy. \quad (21)$$



Wei Yu Dr. Wei Yu received the BS degree in Electrical Engineering from Nanjing University of Technology in 1992, the MS degree in Electrical Engineering from Tongji University in 1995, and the PhD degree in computer engineering from Texas A&M University in 2008. He has been working for Cisco Systems Inc. since 2001. His research interests include cyber space security, computer network, and distributed systems.



Xun Wang Dr. Xun Wang received the BS and MS in computer engineering from The East China Normal University, Shanghai, China, in 1999 and 2002, and the PhD degree in Computer Science and Engineering from The Ohio State University in 2007. He has been working for Cisco Systems, Inc. since 2007. His research interests include network security, overlay networks, and wireless sensor networks.



Xinwen Fu Dr. Xinwen Fu is an assistant professor in the Department of Computer Science, University of Massachusetts Lowell. He received his BS (1995) and MS (1998) in Electrical Engineering from Xi'an Jiaotong University, China and University of Science and Technology of China respectively. He obtained his PhD (2005) in Computer Engineering from Texas A&M University. From 2005 to 2008, he was an assistant professor with the College of Business and Information Systems at Dakota State University. In summer 2008, he joined University of Massachusetts Lowell as a faculty member. Dr. Fu has been publishing papers in prestigious conferences such as S&P, INFOCOM and ICDCS, journals such as TPDS, and book chapters. His group won the best paper award at International Conference on Communications (ICC) 2008. His current research interests are in network security and privacy.



Dong Xuan Dr. Dong Xuan received the BS and MS degrees in electronic engineering from Shanghai Jiao Tong University (SJTU), China, in 1990 and 1993, and the PhD degree in computer engineering from Texas A&M University in 2001. Currently, he is an associate professor in the Department of Computer Science and Engineering, The Ohio State University. He was on the faculty of Electronic Engineering at SJTU from 1993 to 1997. In 1997, he worked as a visiting research scholar in the Department of Computer Science, City University of Hong Kong. From 1998 to 2001, he was a research assistant/associate in Real-Time Systems Group of the Department of Computer Science, Texas A&M University. He is a recipient of the US National Science Foundation (NSF) CAREER award. His research interests include distributed computing, computer networks and cyber space security.



Wei Zhao Dr. Wei Zhao is currently a full professor of Computer Science and the Dean for the School of Science at Rensselaer Polytechnic Institute. Between 2005 and 2006, he served as the director for the Division of Computer and Network Systems in the US National Science Foundation when he was on leave from Texas A&M University, where he served as Senior Associate Vice President for Research and Professor of Computer Science. He was the founding director of the Texas A&M Center for Information Security and Assurance, which has been recognized as a Center of Academic Excellence in Information Assurance Education by the National Science Agency. Dr. Zhao completed his undergraduate program in physics at Shaanxi Normal University, Xian, China, in 1977. He received the MS and PhD degrees in Computer and Information Sciences at the University of Massachusetts at Amherst in 1983 and 1986, respectively. Since then, he has served as a faculty member at Amherst College, the University of Adelaide, and Texas A&M University. As an elected IEEE fellow, Wei Zhao has made significant contributions in distributed computing, real-time systems, computer networks, and cyber space security.