# Hide and Seek in Time — Robust Covert Timing Channels

Yali Liu[1], Dipak Ghosal[2], Frederik Armknecht[3], Ahmad-Reza Sadeghi[3],
Steffen Schulz[3], and Stefan Katzenbeisser[4]

[1] Department of Electrical and Computer Engineering and [2] Department of Computer Science,
University of California, Davis, USA
[3] Horst-Görtz Institute for IT-Security (HGI), Ruhr-University Bochum, Germany
[4] Department of Computer Science, Technische Universität Darmstadt, Germany

**Abstract.** Covert timing channels aim at transmitting hidden messages by controlling the time between transmissions of consecutive payload packets in overt network communication. Previous results used encoding mechanisms that are either easy to detect with statistical analysis, thus spoiling the purpose of a covert channel, and/or are highly sensitive to channel noise, rendering them useless in practice. In this paper, we introduce a novel covert timing channel which allows to balance undetectability and robustness: i) the encoded message is modulated in the inter-packet delay of the underlying overt communication channel such that the statistical properties of regular traffic can be closely approximated and ii) the underlying encoding employs spreading techniques to provide robustness. We experimentally validate the effectiveness of our approach by establishing covert channels over on-line gaming traffic. The experimental results show that our covert timing channel can achieve strong robustness and undetectability, by varying the data transmission rate.

## 1 Introduction

Covert channels aim to conceal the very existence of communication by hiding *covert traffic* in overt communication (*legitimate traffic*). In general, we can distinguish two types of covert channels in computer networks: *covert storage channels* and *covert timing channels* [1]. In covert storage channels, the sender transmits data to the receiver by modifying unused or "random" bits in the packet header [2, 3, 4]. However, many covert storage channels turned out to be easily detectable [5].

Covert timing channels on the other hand, modulate the message into temporal properties of the traffic. Instead of using the contents of packets, these channels convey information through the arrival pattern of packets at the receiver, such as individual inter-packet delays [6, 7, 8]. As we elaborate in Section 2, several methods have been proposed to detect or disrupt covert timing channels. Detection primarily uses statistical tests to distinguish covert from legitimate traffic. The modulation of timing patterns typically results in traffic with distinctive timing characteristics that deviate from legitimate traffic. It turns out that statistical tests that examine the shape and regularity of traffic [9, 7] are the most successful detection mechanisms known today. For disruption

M. Backes and P. Ning (Eds.): ESORICS 2009, LNCS 5789, pp. 120–135, 2009.

of covert timing channels, timing channel jammers have been designed that introduce additional noise by adding random delays to individual packets. To the best of our knowledge, no comprehensive approach for designing covert timing channels has been provided so far that achieves a highly robust covert timing channel that is undetectable by current statistical detection techniques.

**Contribution.** We systematically design a covert timing channel which is statistically undetectable by shape and regularity tests, while being robust against disruptions caused by active adversaries and/or noise in the network. We propose a method to mimic the distribution of inter-packet delays of legitimate traffic. This ensures that there is no first order statistic (e.g., shape difference) that can be applied to distinguish traffic modified by covert messages from legitimate traffic. Furthermore, by sharing a secret (a random number generator seed) between the sender and the receiver, encoding parameters that influence the high order statistics (i.e., correlations) of the modulated covert communication can be changed dynamically. To achieve robustness against intended and unintended channel noise, we apply spreading codes to the modulation of inter-packet delays. Our design features tunable encoding parameters that allow to trade-off the intended level of robustness and undetectability against the channel capacity.

We have validated our approach by testing our covert timing channel in an interactive online game environment. The results show that given certain undetectability requirements, the proposed method is able to generate covert traffic that closely mimics legitimate traffic. Additionally, we show that the proposed approach can achieve robustness against network noise due to packet loss, delay, jitter, and covert timing channel jammers.

## 2 Related Work

The first covert timing channel was proposed in [6], in which the sender either transmits or stays silent in a specific time interval. A similar idea was proposed in [10], where the authors limited the noise sensitivity by increasing the length of the inter-packet delays and reducing the channel capacity. Both approaches require synchronization between the sender and receiver in order to correctly decode a message. The study in [7] describes various ways to help maintain synchronization. However, as the authors note, these techniques still cannot completely solve the synchronization problem. Time-reply information has been used for creating a covert timing channel in [11]. A method to directly encode the covert message in the inter-packet delays was proposed in [9] in order to maximize the channel capacity. Finally, the keyboard jitterbug [8] aims at leaking typed information over the network but suffers from a very low channel capacity.

To defend against covert timing channels, researchers have proposed different solutions to detect and/or disrupt covert traffic. Many earlier works focused on the disruption of covert timing channels. For example, jammed timing channels have been investigated in [12]. By adding random delays to traffic, the rate at which covert information can be conveyed in the presence of a jamming device is made so low that further monitoring of the channel is not needed. However, this type of jamming method reduces the performance of legitimate traffic.

A different approach is to detect covert timing channels using statistical tests that differentiate covert traffic from legitimate traffic. Two classes of tests are considered in this paper. The *shape* of the traffic, which is described by its probability distribution, was adopted to detect binary and multi-symbol covert timing channels [7]; e.g., the statistical test proposed in [9] is based on the assumption that the inter-packet delays of covert traffic will center on limited numbers of distinct values instead of being randomly distributed. Another mechanism for detecting covert channels in network traffic is based on *regularity* testing. As described in [7], this technique exploits the fact that overt traffic packets can arrive at any time, resulting in a non-stationary process, where the variance of the inter-packet delays changes over time. This does not typically hold for covert traffic, especially if the encoding scheme does not change over time.

## 3 Problem Definition and Design Criteria

The goal of this work is to design a robust and high capacity covert timing channel by manipulating the delay between successive packets. At the same time, the covert channel should be undetectable by common statistical tests reported in the literature.

For our model, we define the entities of the *sender* and the *receiver* of a covert communication and the *source* and the *destination* of the overt communication, i.e., the carrier signal. Sender and receiver are connected to the Internet; the sender has access to some sensitive information (covert message) that he wants to transmit to the receiver. To achieve this, the sender embeds the covert information into an overt packet stream that he generates himself. Our system considers both passive and active adversaries. A passive adversary aims at detecting the covert channel by monitoring the transmission between the sender and the receiver. On the other hand, an active adversary, e.g., a timing channel jammer, can disrupt the traffic information by manipulating the ongoing transmission.

We consider a binary channel, in which the covert message is coded as a binary sequence. First, the covert message $\{b_1, b_2, b_3, \ldots\}$, which we refer to as *information bits*, passes through an encoding process. In this step, we leverage a spreading code in order to deal with channel noise, including noise created by covert timing channel jammers. The resulting *code symbols* $\{s_1, s_2, s_3, \ldots\}$ are used to modulate the inter-packet delays $\{t_1, t_2, t_3, \ldots\}$ of a packet stream that is sent by the source to the destination. The receiver shares a code book and a secret random number seed that is used to determine code parameters at runtime. Knowledge of this shared secret enables the receiver to decode the received inter-packet delays $\{\hat{t}_0, \hat{t}_1, \hat{t}_2, \ldots\}$ and generate the received binary sequence $\{\hat{b}_1, \hat{b}_2, \hat{b}_3, \ldots\}$.

The two primary design goals of our covert timing channels are high channel capacity and undetectability.

### 3.1 Channel Capacity

As our carrier medium is the inter-packet delay of legitimate traffic, the channel capacity is the maximum number of bits per packet (bpp) that are passed through the carrier channel. In a generic Binary Symmetric Channel (BSC)[1], the channel capacity is determined by the transmission rate $R_t$ which measures the transmission efficiency of each bit by the number of packets and the bit error rate (BER) $P_e$. In order to achieve high channel capacity, we would like to have a high transmission rate $R_t$ while keeping a low BER $P_e$. Particularly, if $R_t$ approaches the maximum transmission rate for a given channel (i.e., 1 bpp in case of BSC) and the system can achieve any given error probability, we say the timing channel approaches the Shannon capacity limit.

### 3.2 Channel Undetectability

To make the channel undetectable, we need to ensure that the inter-packet delays of covert traffic are indistinguishable from that of legitimate traffic. As the adversary cannot observe legitimate and covert traffic at the same time, detection of covert timing channels can be formulated as a statistical significance testing problem. A covert channel is *undetectable* with respect to a certain test, if the test cannot distinguish between legitimate and covert traffic.

**Shape Test.** A passive adversary may employ many different statistical tests based on different statistical measures. In the most general case, the adversary may compare the distribution of the samples of the legitimate traffic with that of the monitored traffic. While there are a number of different methods to do this, one of the most well known approaches is the Kolmogorov-Smirnov test (KS-test). As the test is independent of the distribution, the KS-test is applicable to different types of traffic with different distributions and has already been successfully applied to detect watermarked inter-packet delays [13, 14].

Let $S(x)$ be the empirical distribution function based on the monitored inter-packet delay samples and let $F(x)$ be a given cumulative distribution function from the inter-packet delay samples of the legitimate traffic. Then the KS-test statistic $H_s$ is defined as

$$H_s = \text{sup}_x |F(x) - S(x)|, \tag{1}$$

which is the greatest distance between $S(x)$ and $F(x)$. One of the design goals of our covert timing channel is to provide tuning parameters that allow the user to select a specific level of $H_s$.

**Regularity Test.** As mentioned before, in most of the legitimate network traffic, the variance of the inter-packet delays changes over time. On the other hand, the variance of the inter-packet delays in a covert traffic may remain relatively constant if the encoding scheme does not change over time. Due to this feature, regularity tests can be employed to efficiently detect some covert timing channels [7].

A regularity test is used to measure the correlation in data. Mathematically, this can be achieved by taking samples of inter-packet delays and separating them into multiple sets with window size $w$. Then for each set $i$ the standard deviation $\sigma_i$ is computed. The regularity $H_r$ is defined as the standard deviation of the absolute difference between any pairs of $\sigma_i$ and $\sigma_j$ and is given by

[1] A BSC is a channel with binary input and binary output and same crossover probability for two inputs.

$$H_r = \text{std}\left(\frac{|\sigma_i - \sigma_j|}{\sigma_i}\right), \quad \forall i, j, \, i < j, \tag{2}$$

where std is the standard deviation operation. Another design criterion is thus to control tuning parameters to meet a given level of $H_r$.

## 4 Encoding with Spreading Codes

Routers or firewalls can incur processing delay and hence alter the inter-packet delays generated at the sender before reaching the receiver. In addition, timing channel jammers might induce additional noise into the channel. Therefore, it is important to design the inter-packet delay patterns to be robust to channel noise. Instead of adding additional bits before transmission to perform error correction, we introduce a spread encoding before the modulation process. Particularly, we borrow a concept from Code Division Multiple Access (CDMA) [15], which is a spread spectrum multiple access technique utilized in radio communication.

In the first step, each bit $b_k$ of the covert message $\{b_1, b_2, \ldots\}$ is encoded into $\tilde{\mathbf{c}}_k = b_k \cdot \mathbf{c}$, where $\mathbf{c} = (c_1, c_2, \ldots, c_N) \in \{\pm 1\}^N$ is a code word. Here, $b_k$ is a binary variable taking on values $-1$ and $+1$, and $N$ is called *spreading ratio*. Observe that $\langle \mathbf{c}, \mathbf{c} \rangle = N$. To decode a received vector $\tilde{\mathbf{c}}_k$, the sign of the inner product $\langle \tilde{\mathbf{c}}_k, \mathbf{c} \rangle$ is computed to recover an estimate $\hat{b}_k$ of the transmitted bit $b_k$. Note that the original bits can be recovered even if a limited number of bits flipped during transmission.

As $N$ code symbols will be used to convey just one information bit, the transmission rate $R_t$ for the new system decreases to $\frac{1}{N}$ bpp. Hence, we aim at encoding multiple bits at once using careful code design. Specifically, to simultaneously transmit $K$ bits $b_1, \ldots, b_K$ over $K$ parallel channels, we transmit

$$\mathbf{s} = (s_1, s_2, ..., s_N) = \sum_{k=1}^{K} b_k \cdot \mathbf{c}_k, \tag{3}$$

using $K$ orthogonal code words $\mathbf{c}_1, \ldots, \mathbf{c}_K$. Walsh-Hadamard codes [15] are one of the popular orthogonal codes that can be used for this purpose. If $\mathbf{c}_i$ and $\mathbf{c}_j$ are two Walsh-Hadamard codes with length $N$, then it holds that $\langle \mathbf{c}_i, \mathbf{c}_j \rangle$ equals $N$ if $i = j$ and 0 otherwise. The receiver and sender must agree on the order of different channels and their codes before starting the covert communication to retrieve the bits correctly. Note that $K \leq N$, as $N$ is the length of the spreading code and the maximum number of orthogonal channels. Since the transmission rate is $R_t = \frac{K}{N}$, there is no transmission rate loss if we use all $N$ channels, i.e., $K = N$.

The orthogonality of the code words allows to decode each information bit $b_k$ separately:

$$\frac{1}{N}\langle \mathbf{s}, \mathbf{c}_k \rangle = \frac{1}{N}\langle \sum_{i=1}^{K} b_i \cdot \mathbf{c}_i, \mathbf{c}_k \rangle = \frac{1}{N}\sum_{i=1}^{K} b_i \cdot \langle \mathbf{c}_i, \mathbf{c}_k \rangle = \frac{1}{N} \cdot b_k \cdot N = b_k. \tag{4}$$

The robustness of the system is determined by the BER $P_e$, which is an inverse function of the Signal-to-Noise Ratio (SNR) $E_s/E_x$ [16], where $E_s$ is the signal power and $E_x$

is the noise power. Considering that the channel noise is arbitrarily distributed in the $N$-dimensional code space, the noise power in each channel after modulation will decrease to $E_x/N$ [15]. Consequently, the spreading code can reduce the power of the distortion by $N$ times and the system can achieve robustness against additive noise by increasing the spreading ratio $N$. Particularly, when $K = N$, the channel capacity approaches the Shannon limit with increasing $N$.

## 5 The Modulation/Demodulation Scheme

Next we investigate how to design the secure modem (modulator and demodulator). The function of the modem is to transfer coded symbols by modulating the inter-packet delays of overt communication and recover the original bits from the modulated delays at the receiver. Given *a priori* knowledge of the channel characteristics (which may be achieved by a training process before the covert communication begins), the security requirement is fulfilled by generating a modulated signal whose statistical properties are close to that of legitimate network traffic.

### 5.1 A Model-Based Modulation Scheme

The modulation process will modulate the inter-packet delays of overt communication depending on the code vector **s** as expressed by Eq. (3). We model the inter-packet delay $t$ as a random variable and let $f(t)$ and $\hat{f}(t)$ denote the probability density functions (PDFs) of the inter-packet delays of legitimate traffic and covert traffic, respectively.

To satisfy the requirement that the mapping of a code symbol to the inter-packet delay must be invertible and to consider implementation simplicity, we adopt a linear modulation:

$$t_n := \alpha + \beta s_n, \quad n = 1, \ldots, N, \tag{5}$$

where $\beta \in \mathbb{R}$ is a scaling parameter and $\alpha \in \mathbb{R}$ is a shift parameter. In the sequel, we show how to choose $\alpha$ and $\beta$. As discussed in the previous section, $N$ inter-packet delays will be used to encode $K$ bits. As these $K$ bits will be encoded at the same time, we will refer to them as a *modulation group* or *m-group*. The parameter $\beta$ will be chosen as a constant for one m-group but will change between different m-groups, following a deterministic (but secret) rule agreed between sender and receiver (more details will follow in Section 5.2). Thus, the value of $\beta$ does not need to be communicated explicitly. In contrast, $\alpha$ represents a random variable with PDF $f_\alpha(t)$. We use one of the $N$ channels and the code word $\mathbf{c}_0 = (1, \ldots, 1)$ from the spreading code (see Section 4) to carry the shift parameter $\alpha$. As long as the spreading code words $\mathbf{c}_1, \ldots, \mathbf{c}_K$ used for the $K$ information bits are orthogonal to $\mathbf{c}_0$, the receiver can successfully recover the information bits, even without knowing the value of $\alpha$ in advance.

As mentioned before, the encoded inter-packet delays $\mathbf{t}$ might be changed to $\hat{\mathbf{t}}$ due to some additive channel noise $\mathbf{x}$, that is $\hat{\mathbf{t}} = \mathbf{t} + \mathbf{x}$. For demodulation and decoding, we apply a threshold rule to the inner product of a scaled down version of the received inter-packet delays and the code words. As a result, we get $\hat{b}_k = \frac{1}{N}\langle\frac{1}{\beta}\hat{\mathbf{t}}, \mathbf{c}_k\rangle$. This recovers an estimate of $b_k$ resulting from the high spread spectrum ratio $N$, since

$$\hat{b}_k = \frac{1}{N}\langle\frac{1}{\beta}\hat{\mathbf{t}}, \mathbf{c}_k\rangle = \frac{1}{\beta \cdot N}\langle\mathbf{t}, \mathbf{c}_k\rangle + \frac{1}{\beta \cdot N}\langle\mathbf{x}, \mathbf{c}_k\rangle \tag{6}$$

$$= \underbrace{\frac{\alpha}{\beta \cdot N}\langle\mathbf{c}_0, \mathbf{c}_k\rangle}_{=0} + \underbrace{\sum_{i=1}^{K} \frac{\beta \cdot b_i}{\beta \cdot N}\langle\mathbf{c}_i, \mathbf{c}_k\rangle}_{=b_k} + \frac{1}{\beta \cdot N}\langle\mathbf{x}, \mathbf{c}_k\rangle = b_k + \frac{1}{\beta \cdot N}\langle\mathbf{x}, \mathbf{c}_k\rangle. \tag{7}$$

**Determining the Model Parameters.** The goal is to determine $\alpha$ and $\beta$ such that the inter-packet delay distribution of the covert traffic $\hat{f}(t)$ can emulate a given distribution of legitimate traffic $f(t)$. From Eq. (5), the modulated inter-packet delay $t$ is the sum of two independent random variables: the shift parameter $\alpha$ and the code symbol $s_n$. Thus, the PDF of $t$ is given by

$$\hat{f}(t) = \frac{1}{\beta}\int_{-\infty}^{\infty} f_\alpha(\tau) f_s\left(\frac{t-\tau}{\beta}\right) d\tau, \tag{8}$$

where $f_s(t)$ and $f_\alpha(t)$ are the PDFs of $s_n$ and $\alpha$, respectively. The amplitude of the code symbol $s_n$ is a discrete random variable taking on values between $-K$ and $K$. We denote its probability mass function (PMF) by $P_s(k)$; it can be shown that the PMF of $P_s(k)$ is an up-sampled Binomial distribution (see derivation in Appendix A). Thus, the PDF of $s_n$ can be expressed as

$$f_s(t) = \sum_{k=-K}^{K} P_s(k)\delta(t-k), \tag{9}$$

where $\delta(t)$ is the Dirac-delta function. As illustrated in Figure 1, $P_s(k)$ is a symmetric function with a roll-off shape and can be approximated by $\text{sinc}(t) = \sin(\pi t)/(\pi t)$.

We can apply here the Nyquist-Shannon sampling theorem [17] which states that if a function $f(t)$ is sampled using a sampling interval $T \leq \frac{1}{2W}$, where $W$ is the bandwidth of $f(t)$, then the function can be completely recovered from the discrete samples. Mathematically, this is represented by

$$f(t) = \int_{-\infty}^{\infty} f_T(\tau)\text{sinc}(\frac{t-\tau}{T})d\tau, \tag{10}$$

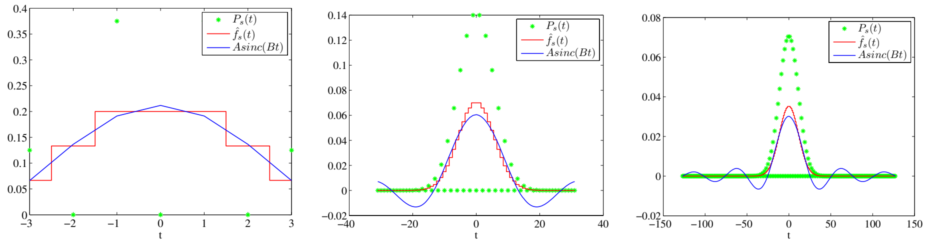
where

$$f_T(t) = \sum_{n=-\infty}^{\infty} f(nT)\delta(t-nT). \tag{11}$$

If $T > \frac{1}{2W}$, the reconstruction (10) will cause aliasing and thus the continuous function $f(t)$ cannot completely be recovered from discrete samples.

Eqs. (8) and (10) show that if we can approximate $f_s(t)$ by a sinc function and approximate the PDF of $f_\alpha(t)$ by $f_T(t)$, then the PDF of the covert traffic $\hat{f}(t)$ approximates the PDF of the legitimate traffic $f(t)$. For this purpose, we first approximate $f_s(t)$ by a continuous function $\hat{f}_s(t)$, which is constructed from $P_s(k)$ by

$$\hat{f}_s(t) = \begin{cases} P_s(k) & \text{if } k - 0.5 < t \leq k + 0.5 \text{ and } -K \leq k \leq K \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$



(a) (b) (c)

**Fig. 1.** Approximating $\hat{f}_s(t)$ by a sinc function for a fixed $T$ and: (a) $K = 3$; (b) $K = 31$; (c) $K = 127$. (Note that $P_s$ is a discrete function; it only has non-zero value when $t = k$.).

Then $\hat{f}_s(t)$ resembles the envelope of $P_s(k)$. Since half of the points in $P_s(k)$ are zeros (Appendix A), we use an interpolated version $P_s'(k)$ to replace $P_s(k)$ in Eq. (12) to achieve a smoother approximation of $f_s(t)$. This is given by

$$P_s'(k) = \begin{cases} qP_s(k) & \text{when K - k even} \quad (13a) \\ q\dfrac{P_s(k-1)+P_s(k+1)}{2} & \text{otherwise,} \quad (13b) \end{cases}$$

where $q$ is chosen so that $\int_{-\infty}^{\infty} \hat{f}_s(t)\,dt = 1$. Then, we approximate the right hand side of Eq. (8) by

$$\hat{f}(t) \approx \frac{1}{\beta}\int_{-\infty}^{\infty} f_\alpha(\tau)\hat{f}_s\left(\frac{t-\tau}{\beta}\right)d\tau. \tag{14}$$

Next, we aim for approximating $\text{sinc}(\frac{t}{T})$ by $\frac{\gamma}{\beta}\hat{f}_s(\frac{t}{\beta})$, where $\gamma$ is an auxiliary constant. Note that this approximation is just a scaled version of $\hat{f}_s(t) \approx A \cdot \text{sinc}(Bt)$. We solve for $A$ and $B$ by curve fitting, and then solve for $\gamma$ and $\beta$, which are given by

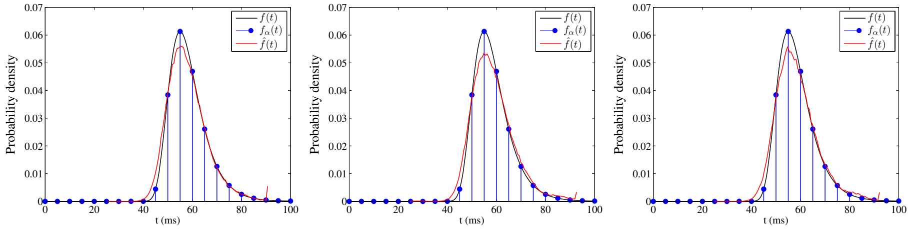
$$\beta = TB, \quad \gamma = \frac{TB}{A}. \tag{15}$$

For any fixed $K$, the PMF $P_s(k)$ is given. Therefore, for different $T$, we only need to perform the approximation once at the baseline case and the parameters $\gamma$ and $\beta$ can be obtained by (15). The accuracy of the approximation is shown in Figure 1.

Based on these results, we approximate the PDF $f_\alpha(t)$ by $\gamma f_T(t)$. Since (11) is just the PDF of a discrete random variable, we have $\text{Prob}(\alpha = nT) = \gamma f(nT)$. More precisely, as this may not define a valid probability measure, we apply normalization

$$\text{Prob}(\alpha = nT) = f(nT)/P_0, \quad P_0 = \sum_{n=-\infty}^{\infty} f(nT). \tag{16}$$

Note that modulation and demodulation is fully determined by $\alpha$ and $\beta$, the helper constant $\gamma$ does not actually need to be computed.



(a) (b) (c)

**Fig. 2.** Synthesizing a given inter-packet delay distribution $f(t)$ with (a) $K = 3$, (b) $K = 31$ and (c) $K = 127$. (Here the sampling time is $T = 5$.).

In summary, the process described above determines the distribution of $\alpha$, which is the PDF of the samples of $f(t)$, sampled with an interval $T$. The parameter $\beta$ is given by the number of channels $K$ and sample interval $T$. Although the sinc function is a very coarse approximation of $f_s(t)$, a combination of Dirac delta functions, Figure 2 shows that a given inter-packet delay distribution can be emulated very well using our encoding scheme.

### 5.2 Removing Regularity

As typical network traffic is non-stationary[2] [18], the statistics of the generated inter-packet delays should vary with time. In our proposal, this can be realized by adjusting the encoder and modulator parameters dynamically. Particularly, for each m-group $g$, the variance is given by $\sigma_g^2 = \beta^2\sigma_s^2$, where $\sigma_s^2$ is the variance of the code symbol $s_n$. As shown in Section 5.1, $\beta$ and the distribution of $s_n$ are determined by $K$ and $T$, so we can adjust $\sigma_g^2$ by changing these two parameters for each m-group.

For each m-group $g$, a random $\alpha$ is generated according to Section 5.1 to emulate the given inter-packet delay distribution. We denote it by $\alpha_g$. Considering that $\alpha$, $\beta$ and $s_n$ are independent, the correlation coefficient of the modulated inter-packet delay $t$ is given by

$$R(t_i, t_{i+\tau}) = \frac{\operatorname{cov}(\alpha_{g(i)}, \alpha_{g(i+\tau)})}{\sqrt{\sigma_\alpha^2 + \beta_{g(i)}^2\sigma_{g(i)}^2} \cdot \sqrt{\sigma_\alpha^2 + \beta_{g(i+\tau)}^2\sigma_{g(i+\tau)}^2}}, \qquad (17)$$

where $i$ is the index of the generated inter-packet delay and $g(i)$ is the group index that contains packet $i$. Also, $\sigma_\alpha^2$ and $\operatorname{cov}(\alpha_{g(i)}, \alpha_{g(i+\tau)})$ are the variance and the covariance of the parameter $\alpha$, respectively.

Therefore, the correlation of the inter-packet delays of the covert traffic can dynamically change by appropriately controlling the generation of $\alpha$ and $\beta$, which are determined by parameter $T$ and $K$. Considering that $T$ controls the system robustness and undetectability, in our proposed system, we fix $T$ and use a cryptographically secure pseudo-random number generator to choose a pseudo-random sequence of values for $K$ which is uniformly distributed in $[1, K_{max}]$. The seed for the sequence is secretly shared between the sender and the receiver of the covert channel.

[2] A non-stationary traffic means that its statistical properties may vary with time.

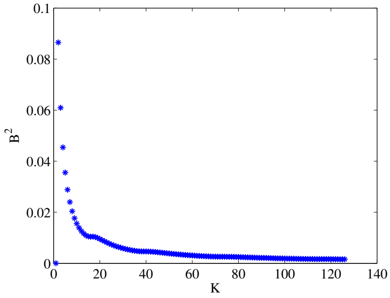


**Fig. 3.** The impact of $K$ on parameter $B^2$

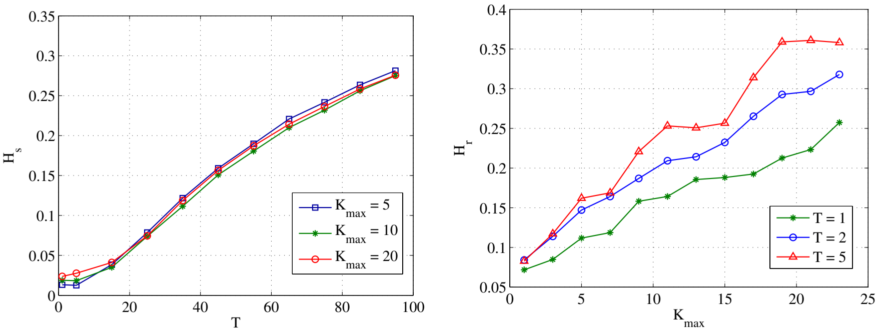
### 5.3 Evaluation Trade-Off

In this subsection, we discuss the system evaluation trade-off in terms of transmission rate, robustness, and undetectability, by varying the number of channels $K$, the sampling interval $T$, and the spreading ratio $N$.

**Transmission Rate.** The transmission rate $R_t$ is only determined by the ratio of $\frac{K}{N}$. Considering that we need at least one channel to transmit $\alpha$, for a given spreading ratio $N$, the maximum transmission rate is $\frac{N-1}{N}$.

**Robustness.** According to Eq. (9), after performing encoding and modulation, the SNR of the new system will increase by $G = \beta^2 N$, which we denote as *robustness gain*. Specifically, the larger the value of $\beta^2 N$, the more robust is the system. Note that $\beta = TB$ and $B$ is determined by the sinc approximation for a given $K$. With $T$ fixed, Figure 3 shows the variation of $B^2$ for various $K$. Apparently, a larger $K$ will lead to a smaller $B^2$ and thus a smaller $\beta^2$. On the other hand, for a given $K$, Eq. (15) shows that $\beta$ is proportional to $T$. This implies that a smaller $T$ leads to a smaller $\beta$. Hence, one can achieve a higher robustness by decreasing $K$ and increasing $N$ and $T$.

**Undetectability.** The undetectability of covert communication is measured by shape and regularity tests. Figure 4 illustrates the influence of the parameters $K$ and $T$ on the undetectability. For illustrative purposes, we use a theoretical distribution function of the inter-packet delays obtained from legitimate traffic of online game [19]. As discussed in Section 5.2, $K$ is randomized to circumvent regularity detection. Consequently, the undetectability performance is determined by $K_{max}$, the dynamic range of $K$, and thus we use $K_{max}$ instead of a certain value of $K$ in the following discussions. As mentioned in Section 3, the KS-test statistic $H_s$ is used to measure the distance of the distribution functions of covert traffic and legitimate traffic. If $H_s$ is small, it implies that the distribution of the covert inter-packet delays is close to that of the legitimate traffic. Figure 4(a) clearly shows that the parameter $K_{max}$ has little impact on the shape test while the system can achieve the given shape requirement by selecting an appropriate $T$. Regarding the regularity test, we considered the variation of the standard deviation among sets of 100 packets, which is a typical value used in existing detection schemes. If the regularity score is low, the covert traffic is highly regular, indicating the possible existence of a covert timing channel. The effects of $K_{max}$ and $T$ on the regularity test are shown in Figure 4(b). A larger dynamic range of $K$ or a greater sampling



(a) (b)

**Fig. 4.** The influence of $K_{max}$ and $T$ on the (a) shape and (b) regularity statistics

time $T$ results in a higher regularity score, making detection less probable. Therefore, for a given undetectability requirement $H_s$, we can find the maximum sampling interval $T$ based on the shape requirement. Then by increasing $K_{max}$, the system regularity requirement $H_r$ can also be fulfilled.

**Trade-off.** In conclusion, the number of channels $K_{max}$, the spreading ratio $N$, and the sampling interval $T$ together achieve a trade-off among the three evaluation criteria. To achieve a better channel capacity, $K_{max}$ must approach $N$. The robustness is controlled by all three parameters together: larger $N$ and $T$ with a smaller $K_{max}$ will lead to a more robust system. As for the undetectability, a more accurate shape approximation can be achieved with a smaller $T$ and on the contrary, a better regularity performance can be achieved will a bigger $T$ or $K_{max}$.

### 5.4 Algorithm Summary

The function *CovertInterPacketDelayGenerator*$(H_s, H_r, G, f)$ depicts how to generate the covert inter-packet delays $\mathbf{t}$ under given undetectability and robustness requirements. Here the function *ParameterEstimate* is used to determine the system parameters $T$ and $K_{max}$ with given shape and regularity statistics, as elaborated in Section 5.3.

## 6 Experimental Results

We have developed a covert timing channel testbed that consists of a server and a client which act as the sender and the receiver of both the covert and the overt communication, respectively. The sender controls the TCP/UDP inter-packet transmission delays to modulate the hidden message. The receiver passively collects the packet inter-arrival delays and decodes them with the shared code book and a shared seed.

**Testing Scenarios.** We have considered two testing scenarios for our experimental evaluation. The first scenario is in a LAN environment in a medium-size campus network; the client and the server functions are implemented in hosts that are located in two different departments. The second scenario is in the WAN environment. The sender and the

**Algorithm 5.1.** CovertInterPacketDelayGenerator($H_s, H_r, G, f$)

```
Input : Undetectability requirements $(H_s, H_r)$, robust gain $G$,
the legitimate inter-packet delay distribution $f(t)$.

Output : covert inter-packet delays $\mathbf{t}$

// estimate parameters with given shape and regularity statistics
$(T, K_{max}) \leftarrow$ ParameterEstimate$(H_s, H_r, f)$

for each m-group
do {
    generate $\alpha$ following the distribution Prob$(\alpha = nT)$    // Eq. (16)
    generate $K$ following Uniform$(1, K_{max})$
    solve $B$ by curve fitting $\hat{f}_s(t) \approx A \cdot \text{sinc}(Bt),\ \beta \leftarrow T \cdot B$
    $N \leftarrow \lceil G/\beta^2 \rceil$    // find the minimum $N$ satisfying $G$
    $(s_1, \ldots, s_N) \leftarrow \sum_{k=1}^{K} b_k \cdot \mathbf{c}_k$    // encoding
    $t_n \leftarrow \alpha + \beta s_n,$ for $1 \le n \le N$    // modulation
    $\mathbf{t} := (t_1, t_2, \ldots, t_N)$
}
```

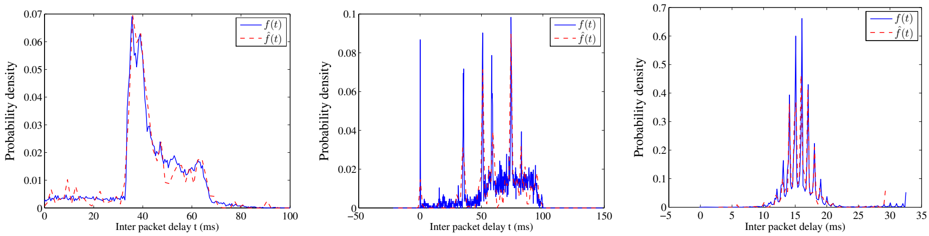
**Table 1.** The network conditions for each test scenario

| | LAN | WAN |
|---|---|---|
| Packet loss rate (%) | 0 | 0.0024 |
| Physical distance (miles) | 1.5 | 5352 |
| Jitter(std) (ms) | 0.43 | 0.6316 |
| Jitter(mean)(ms) | 0.0283 | 0.0768 |

receiver are located in United States and Germany, respectively. The network attributes for the two experimental scenarios are summarized in Table 1.

**Dataset.** A significant amount of today's Internet traffic is generated by multimedia applications (e.g., network gaming, video streaming or Voice over IP). As a result, multimedia traffic is a promising medium for covert communications. In this study, we consider network gaming traffic using the User Datagram Protocol (UDP) as the medium for the covert timing channel. Note that our covert timing channel, like most existing encoding schemes [20], require packet order information to align the encoded traffic for correct decoding. We assume that this ordering is available as a side information. This is not a critical limitation since such information is often contained in the user transport or application layer protocol, like in RTP over UDP.

In our experiments, two popular on-line games, "Counter Strike" and "Starcraft" are adopted as the carrier application. The legitimate samples that we use for our experiments are from two datasets: 1) two four hours traffic traces for both games were collected on LAN environment and consist of 1000000 packets and 2) a two hours traffic trace for "Counter Strike" was collected in a WAN environment which consists of 500000 packets.



(a) (b) (c)

**Fig. 5.** The probability density function of the inter-packet delay of covert traffic and legitimate traffic for (a) Counter Strike in LAN ($H_s = 0.032$, $H_r = 1.23$), (b) Starcraft in LAN ($H_s = 0.028$, $H_r = 0.78$) and (c) Counter Strike in WAN environment ($H_s = 0.026$, $H_r = 1.45$)

**Undetectability.** Figure 5 shows the distribution of the inter-packet delays for the covert traffic generated by our proposed method along with the legitimate traffic observed from the two on-line games. As shown in these figures, our covert traffic emulates the given distribution very closely. The shape statistic parameter $H_s$ between the covert traffic and the legitimate traffic was set to $0.035$, which is the minimum score obtained from legitimate game traffic samples with a total of $1500000$ inter-packet delays. The regularity criterion $H_r$ was set to the same as that of legitimate traffic. These results indicate that the covert traffic distribution is nearly identical to that of legitimate traffic.

**Robustness.** We have also evaluated the robustness of the proposed algorithm by considering different types of noise during the transmission process. Specifically, covert inter-packet delays are generated with the given undetectability requirements (here we use the same shape and regularity requirement as the ones in the previous section). The robustness gain $G$ is set to be $40$ and $15$ in LAN and WAN tests, respectively. The resulting transmission rates for the covert communication are $0.23$ bpp and $0.98$ bpp, respectively.

Three types of channel noise are considered in our study. The first type corresponds to noise that is inherent in the network due to packet loss, delay, and jitter. The second and the third types of noise are the jamming noises which may be injected by an active adversary. Specifically, the second type is a theoretical noise model that has a normal distribution with zero mean and variance $\sigma^2$ to simulate noise within certain constraints. Considering that a uniformly distributed noise represents the worst case scenario in terms of channel capacity [20], the third type of noise is uniformly distributed in the range $[0, \Delta]$. Note that, similar to adding a random $\alpha$ during the modulation process, the mean of the noise does not impact the demodulation and decoding accuracy as it is orthogonal to all effective channels carrying the covert message. Using the Linux IPFilter suite, we introduced the noise directly into the network stack the sender.

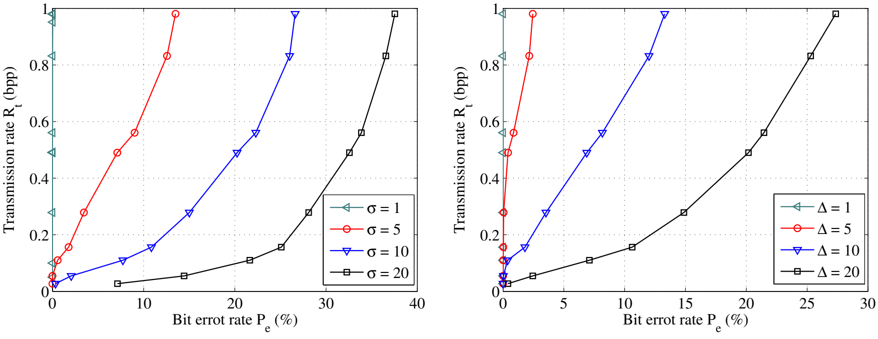
Table 2 and Table 3 summarize the results of these experiments. In these tables, we provide the BER $P_e$, which is the average fraction of incorrectly received bits for both the LAN and the WAN tests. The throughput $\bar{C}$, which is the correctly received bits per packet (bpp), is given by $\bar{C} = R_t(1 - P_e)$. The results clearly show that where there is no jamming noise, there are no bit errors in the LAN scenario. When we add noise uniformly distributed between $[0, 5]$ ms, the correct bit rate $(1 - P_e)$ achieved by our

**Table 2.** Summary of the bit error rate $P_e$ for the timing channel experiments in the LAN

| Game | LAN | Gaussian $\sigma$ | | | | Uniform $\Delta$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | noise | 1 | 5 | 10 | 20 | 1 | 5 | 10 | 20 |
| Counter Strike $P_e(\%)$ | 0 | 0.15 | 3.28 | 15.28 | 31.30 | 0.034 | 0.15 | 4.15 | 17.36 |
| Starcraft $P_e(\%)$ | 0 | 0 | 4.30 | 14.90 | 29.54 | 0 | 0.19 | 3.92 | 16.63 |

**Table 3.** Summary of the bit error rate $P_e$ and the throughput $\bar{C}$ for the timing channel experiments in the WAN for Counter Strike

| Performance | WAN | Gaussian $\sigma$ | | | | Uniform $\Delta$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | noise | 1 | 3 | 5 | 10 | 1 | 3 | 5 | 10 |
| $P_e(\%)$ | 0.10 | 0.32 | 5.98 | 16.34 | 32.91 | 0.24 | 4.72 | 5.80 | 20.24 |
| $\bar{C}(bpp)$ | 0.9641 | 0.9620 | 0.9074 | 0.8073 | 0.7075 | 0.9628 | 0.9195 | 0.9091 | 0.7697 |



(a) (b)

**Fig. 6.** Trade-off among the transmission rate $R_t$ and the bit error rate $P_e$ under jammed (a) Gaussian and (b) Uniform noise ($H_s$ is set to 0.03 and $H_r$ is set to 0.68)

proposed algorithm is more than $99.8\%$ for both gaming traffic. Even when the upper limit of noise is increased to 20 ms, we can still correctly transmit more than $83\%$ of the total bits. Note that the average inter-packet delays in game traffic is around 50 ms. This clearly shows that our system can achieve a high robustness (i.e., reliability) even in a highly noisy channel. In the WAN environment, the throughput of our covert timing channel for Counter Strike is 0.9 bpp for jamming noise range of $[0, 5]$ ms and $\sigma = 5$ ms for additive Gaussian. Even for the higher noise range of 10 ms the throughput is still more than 0.7 bpp.

**Tradeoff.** From the results obtained in the LAN and WAN scenarios, we have observed that there is a tradeoff between the transmission rate $R_t$, the robustness, and the undetectability. In particular, different transmission rates yield different robustness performance with the given undetectability requirement. We thus address the more interesting question: if the undetectability requirement is fixed, how does the robustness performance change with the transmission rate? With predefined settings of $K_{max}$ and $T$ satisfying the undetectability requirement, Figure 6 depicts the relationship between the transmission rate and $P_e$ under different amounts of noise in the LAN environment. It is apparent that the bit error rate increases monotonically with the transmission rate. This property can easily be verified by examining the definition of $R_t$, which is $K/N$, and the measure of robustness gain $\beta^2 N$.

## 7 Conclusions

In this paper, we proposed a comprehensive method for establishing a covert timing channel in computer networks, which allows to balance undetectability against the most common detection methods (shape and regularity) with robustness against network noise. Robustness is achieved by encoding the message using a spreading code scheme. Undetectability is fulfilled by using a model-based modulation scheme that allows us to approximate any legitimate traffic distribution. We have implemented our scheme and have conducted extensive experiments and found that our system can achieve the requirements.

## Acknowledgements

This research was funded in part by NSF grant 0551654, US and SPEED grant, EU.

## References

1. Deparment of Defense Standard: Trusted computer system evaluation criteria. Tech. Rep. DOD 5200.28-STD (1985)
2. Handel, T.G., Sandford, M.T.: Hiding data in the OSI network model. In: Proceedings of the First International Workshop on Information Hiding, London, UK, pp. 23–38 (1996)
3. Rowland, C.H.: Covert channels in the TCP/IP protocol suite. Tech. Rep. 5, First Monday, Peer Reviewed Journal on the Internet (1997)
4. Giffin, J., Greenstadt, R., Litwack, P., Tibbetts, R.: Covert messaging through TCP timestamps. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 194–208. Springer, Heidelberg (2003)
5. Murdoch, S.J., Lewis, S.: Embedding covert channels into TCP/IP. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 247–261. Springer, Heidelberg (2005)
6. Padlipsky, M., Snow, D., Karger, P.: Limitations of end-to-end encryption in secure computer networks. Tech. Rep. ESD TR-78-158, Mitre Corporation (1978)
7. Cabuk, S., Brodley, C.E., Shields, C.: IP covert timing channels: design and detection. In: CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security, New York, pp. 178–187 (2004)
8. Shah, G., Molina, A., Blaze, M.: Keyboards and covert channels. In: USENIX-SS 2006: Proceedings of the 15th Conference on USENIX Security Symposium, pp. 59–75 (2006)
9. Berk, V., Giant, A., Cybenko, G.: Detection of covert channel encoding in network packet delays. Tech. Rep. Darthmouth College (2005)
10. Girling, C.G.: Covert Channels in LAN's. IEEE Transactions on Software Engineering 13(2), 292–296 (1987)

11. Cabuk, S.: Network covert channels: Design, analysis, detection, and elimination. PhD thesis (2006)
12. Giles, J., Hajek, B.: An information-theoretic and game-theoretic study of timing channels. IEEE Transactions on Information Theory 48(9), 2455–2477 (2002)
13. Peng, P., Ning, P., Reeves, D.S.: On the secrecy of timing-based active watermarking traceback techniques. In: SP 2006: Proceedings of the 2006 IEEE Symposium on Security and Privacy, Washington, DC, pp. 334–349 (2006)
14. Gianvecchio, S., Wang, H.: Detecting covert timing channels: an entropy-based approach. In: CCS 2007: Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp. 307–316 (2007)
15. Prasad, R., Hara, S.: An overview of multi-carrier CDMA. In: IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings, vol. 1, pp. 107–114 (1996)
16. Proakis, J.: Digital Communications (1995)
17. Shannon, C.E.: Communication in the presence of noise. Proceedings of the IEEE 72(9), 1192–1201 (1984)
18. Cao, J., Cleveland, W.S., Lin, D., Sun, D.X.: On the nonstationarity of internet traffic. In: SIGMETRICS 2001: Proceedings of the International Conference on Measurement and Modeling of Computer Systems, Cambridge, Massachusetts, United States, pp. 102–112 (2001)
19. Färber, J.: Traffic modelling for fast action network games. Multimedia Tools and Applications 23(1), 31–46 (2004)
20. Sellke, S.H., Wang, C., Shroff, N., Bagchi, S.: Capacity bounds on timing channels with bounded service times. In: IEEE International Symposium on Information Theory, pp. 981–985 (2007)

## A Derivation of $P_s(k)$

Following Eq. (3), each code symbol $s_n$ can be expressed as $s_n = \sum_{k=0}^{K} b_k c_{n,k}$, where $c_{n,k}$ denotes the $n$-th entry of $\mathbf{c}_k$. Due to the random code and the input binary bits with equal probability, we have Prob$(b_k c_{n,k} = 1) =$ Prob$(b_k c_{n,k} = -1) = 1/2$. Let $k_1$ be the number of channels with the code value $b_k c_{n,k} = 1$ and $k_2$ be the one with the code value $b_k c_{n,k} = -1$. We have $K = k_1 + k_2$ and $s_n = k_1 - k_2$, where $0 \leq k_1 \leq K$ and $0 \leq k_2 \leq K$. Then

$$P_s(k) = \begin{cases} \binom{K}{\frac{K-k}{2}} (\frac{1}{2})^K & \text{when } K - k \text{ even} \qquad (18a) \\ 0 & \text{otherwise,} \qquad (18b) \end{cases}$$

where $-K \leq k \leq K$. The distribution of $s_n$ resembles an up-sampled version of the PDF of a binomial distribution $B(K, 1/2)$.