Behavioral Mimicry Covert Communication

Seyed Ali Ahmadzadeh and Gordon Agnew

Department of Electrical and Computer Engineering,
University of Waterloo,
Waterloo, ON, Canada, N2L 3G1
{ahmadzdh,gbagnew}@uwaterloo.ca

Abstract. In this paper, the use of structural behavior of communication protocols (e.g., CSMA) in designing new covert channels is investigated. In this way, the covert transmitter adopts the communication protocol architecture to control its overt traffic flow yet with different parameters that give it enough freedom to embed the covert message in its overt traffic. A salient feature of this scheme is that its rate increases in proportion with the overt capacity of the system. In addition, the paper presents a new covert channel for the wireless environment that mimics the structural behavior of CSMA protocol. The parameters of the proposed scheme are optimized in order to maximize the channel rate, stealthiness and robustness. Finally, the performance of the proposed scheme is analyzed from security, reliability and communication rate point of view.

Keywords: Covert communication, information hiding, wireless security.

1 Introduction

Covert communication often refers to the process of communicating data through a channel that is neither designed, nor intended to transfer information [13]. The primary use of covert channels was to allow information to be leaked to an unauthorized recipient by exploiting weaknesses in conventional communication systems.

In general, two major forms of covert channels are defined in the literature. One category involves direct or indirect storage of the covert message into certain portion of the network traffic (i.e., *storage channels*) [9]. The other category (i.e. *timing channels*) [7], targets some typical characteristics of the system (e.g., inter-packet delays) to exploit normal behavior of the system and open a covert channel. In this way, the receiver can interpret the covert transmitter's message by analyzing the system behavior. This classification can be extended by identifying new channels such as *counting channels* [8] in which the number and the frequency of events come into play instead of the occurrence of an isolated event.

Kemmer [12] identified three necessary conditions for existence of a covert channel. (i) a global resource that is shared between the receiver and the sender of the covert message, (ii) ability to modify the shared resource and, (iii) a method to achieve synchronization between the sender and the receiver. The wireless environment provides all three conditions making it a perfect medium for a wide range

M. Rajarajan et al. (Eds.): SecureComm 2011, LNICST 96, pp. 134–153, 2012.

of covert channels, some of them are yet to be found [3,14,19]. In [1] a covert channel based on jamming over slotted ALOHA was introduced in which the covert transmitter jams specific packets in the network. The receiver decodes the covert message through the packet loss pattern of the system. A covert channel based on splitting tree algorithm was introduce in [15]. This approach exploits splitting tree collision resolution algorithm by reconfiguring the covert transmitter to choose a particular path in the splitting tree according to the covert message. The receiver on the other hand, decodes the covert message through the relative position of the covert transmitter in the tree. Later, Wang *et al.* [20] extended the above approach into an anonymous covert channel in which the receiver decodes the covert message using a specific voting mechanism that considers the probabilistic decisions of multiple covert transmitters within the collision resolution algorithm. In [10] the authors investigate the application of the covert transmitter's inter-packet arrival time patters in order to synchronize the covert transmitter and the covert receiver in the DCF mode of the IEEE 802.11 protocol. Their scheme was based on a round of training where both the transmitter and the receiver adapt themselves with the network and generate a codebook in order to embed the covert message into the transmitter's inter-packet arrival time.

Although the above covert communication schemes provide secure and stealth communication channels, they trade the achievable rate of the channel in favor of reliability and secrecy of the channel. The synchronization between the covert transmitter and the covert receiver is also a challenging issue as covert channels are often one-way channels with no universal time reference available in the channel. Moreover, wide variety of covert communication schemes [4, 5, 6, 16], focus on long-term statistical properties of the covert transmitter and aim to keep the transmitter's statistical finger prints as close as possible to a legitimate transmitter. However, to achieve this goal, the covert transmitter has to deviate from typical short-term behaviors of a legitimate source. Therefore, a system observer may be able to detect the covert transmitter and uncover the existence of the covert channel.

In this paper, we presented a new approach that systematically exploits the probabilistic nature of access control protocols in communication networks to open a covert channel in the system. To this end, we turn our attention to the structural behavior of communication protocols and design a covert transmitter that mimics not only long-term statistical behaviors of a legitimate node but also reacts to the temporal changes in the system similar to a typical transmitter. A salient feature of the proposed covert channel is that its rate increases linearly with the overt channel rate of the system.

The rest of the paper is organized as follows. In Section 2, the principles of the system under study are reviewed followed by a detailed description of the proposed covert channel in Section 3. Section 4 contains a discussion on how the proposed covert transmitter and the covert receiver are tuned in order to mimic the behavior of legitimate nodes. The performance analysis and numerical results are presented in Section 5. Section 6 concludes the paper.

2 System Model

IEEE 802.11 [11] is one of the most popular wireless communication protocols. It uses the *carrier sense multiple access / collision avoidance* (CSMA/CA) technique in order to share the wireless channel among multiple users.

In CSMA/CA, the wireless channel is divided into small time periods called time slots. Users constantly check the channel to detect transmission activities. If the channel is busy, each user selects a backoff time (measured in slot times) randomly in the interval $[0, W)$, where W is the size of the contention window. This backoff timer is decreased any time that the channel is sensed idle for a specific period of time called DIFS (i.e., Distributed Inter-Frame Space). The timer stops if the channel gets busy again and when it reached zero, a packet is transmitted and the receiver acknowledges the packet after a period of SIFS (i.e., Short Inter-Frame Space). The size of the contention window is set to W_{min} following each successful transmission, and is doubled after each unsuccessful transmission. The expansion stops when the size of the contention window reaches W_{max} and remains constant until the transmission is successful.

Through the rest of the paper it is assumed that the covert receiver is aware of the covert transmitter's identity and shares a wireless channel with several users in the system (including the covert transmitter). The terms covert receiver and receiver and also covert transmitter and transmitter are used interchangeably. It is also assumed that each packet contains the identity of its transmitter (e.g., the source address field in IEEE 802.11).

3 Behavioral Mimicry Covert Communication

In principle, the behavioral mimicry covert communication is based on adopting the structure of the medium access control protocol, in use by ordinary users in the system, and modify it in such a way that gives the covert transmitter enough freedom to embed the covert message into its overt traffic. It is noted that most access control protocols in communication networks, bring some kind of randomness into the system in order to provide each user with a fair share of resources of the system. By adopting the structure of the communication protocol, one can benefit from the aforementioned random behavior and opens a new covert channel in the system.

Inspired by this observation, in this section, a new covert communication scheme is presented which is based on mimicking the structural behaviors of CSMA/CA algorithm. The proposed scheme is primarily designed for wireless protocols such as IEEE 802.11, however it can be extended to other communication models that involve multiple access techniques and shared mediums. First, a fixed rate version of the proposed scheme is discussed in order to highlight main properties of the covert system. Then, an advanced modification of the proposed scheme is presented that adapts its rate according to the channel condition.

3.1 Fixed Rate Covert Communication (FRCC)

FRCC scheme is basically a timing covert channel that benefits from the channel activities of a subset of users in the system (i.e., the covert set) as clock ticks for a virtual clock called *covert clock*. The covert message is embedded into the covert transmitter's contention window which is controlled by the covert clock. The covert receiver also maintains its own covert clock by observing the same channel activities of members of the covert set. It is noted that due to the broadcast nature of the wireless environment, all users that share the same channel can overhear the transmitted packets. This only requires that both sides track the same set of users and are equipped with proper error correction methods in case a mismatch happens between the transmitter and the receiver. Hence, the transmitter and the receiver can observe the channel activities of the same set of users and increment their clock synchronously. In this way, upon receiving a packet from the covert transmitter, the receiver evaluates the value of its covert clock and decodes the embedded message.

Let S be the subset of network users in which their packet transmissions are considered as clock ticks. The covert set can be preset in the covert transmitter and the covert receiver or may be generated dynamically. For instance, it can be a set of users that their MAC addresses, or the hash values of their MAC addresses have certain properties. Also, a user can be included into S if its position is inside an acceptable region in the system (user position can be obtained from its signal power or using smart antenna techniques).

Figure 1 depicts how the transmitter embeds the covert message into its contention window and synchronizes the contention window with the covert clock. Let ω be a covert message from the covert message set Ω. Each covert message is associated to a unique state in the first stage of the transmitter's transmission window. Hence, the size of the message set is equal to the initial size of the transmitter's transmission window (i.e., T_0). For simplicity and without loss of generality, let's assume $\Omega = \{0, 1, ..., T_0 - 1\}$. Hence, the message $\omega \in \Omega$ corresponds to the state m_ω^0 in Figure 1. It is noted that for any message set Γ of size T_0, one can find a one-to-one mapping that transforms Γ to Ω.

For instance, suppose $T_0 = 4$ and the transmitter intends to transmit the binary sequence $\mathcal{B}$ over the covert channel. The transmitter generates the message set $\Gamma = \{00, 01, 10, 11\}$ and splits $\mathcal{B}$ into smaller items of size two, where each item is a member of Γ. It is noted that the binary to decimal conversion is a mapping that transforms Γ to Ω. In fact, one can recognize the similarity of the above example to the concept of modulation in digital communication [18].

Each communication block starts with a successful packet transmission by the covert transmitter. Then, the transmitter begins to monitor the channel to catch packets from members of S. For each packet, the transmitter's clock is incremented by one unit and it moves down one state (i.e., to the left in Figure 1) in its transmission window. The covert transmitter sends its next packet when it reaches to the last state of the transmission window to mark the value of its covert clock which is observed by the covert receiver. It is noted that the covert

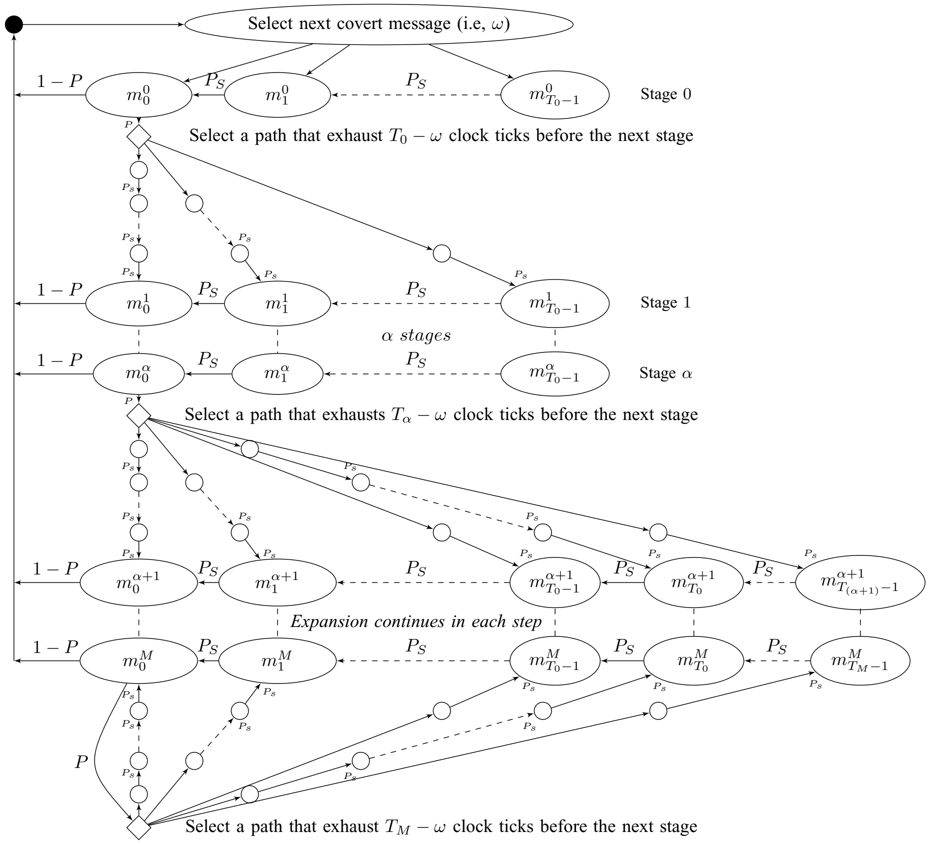
Select next covert message (i.e, ω)
$1-P$
m_0^0
P_S
m_1^0
P_S
$m_{T_0-1}^0$
Stage 0
P
Select a path that exhaust $T_0 - \omega$ clock ticks before the next stage
P_s
m_0^1
m_1^1
$m_{T_0-1}^1$
Stage 1
α stages
m_0^α
m_1^α
$m_{T_0-1}^\alpha$
Stage α
Select a path that exhausts $T_\alpha - \omega$ clock ticks before the next stage
$m_0^{\alpha+1}$
$m_1^{\alpha+1}$
$m_{T_0-1}^{\alpha+1}$
$m_{T_0}^{\alpha+1}$
$m_{T_{(\alpha+1)}-1}^{\alpha+1}$
Expansion continues in each step
m_0^M
m_1^M
$m_{T_0-1}^M$
$m_{T_0}^M$
$m_{T_M-1}^M$
P
Select a path that exhaust $T_M - \omega$ clock ticks before the next stage

Fig. 1. Covert message transmission. P_S is the probability of a successful transmission by members of S.

transmitter can recognize multiple transmissions of a single packet using the packet's sequence number field.

Similarly, at the other end of the channel, the covert receiver monitors the channel for successful transmissions of members of the covert set and maintains the same covert clock value as the covert transmitter. In this way, when the covert receiver detects a packet from the covert transmitter, it reads the value of its covert clock and decodes the corresponding covert message. The receiver then resets its clock and monitors the channel in order to receive the next covert message. It is worth noting that this scheme does not affect the usual packet transmission of the receiver or other nodes in the system. The transmitter also is able to maintain its overt communication except that its contention window is controlled by the covert clock.

However, if the covert transmitter fails to transmit the packet on the proper time slot (e.g., due to collision with other nodes), the covert transmitter expands its contention window and selects another transmission slot to send its packet.

Algorithm 1. FRCC transmission sequence.

```
/* i is the covert transmitter's current stage index.
function $Success = sendpkt(T_i, message)$
$wait(message)$;
$Success = transmit\_pkt()$;
if $Success$ then
  return true
else
  $wait(T_i - message)$;
  return false
end if
```

The reason for the covert transmitter to expand its contention window is two fold. First of all, it is essential for the covert transmitter to achieve maximum stealthiness and does not behave differently as compared to other nodes in the system. In the traditional CSMA/CA protocol, each node expands its contention window and waits for a random amount of time before retransmitting its packets. The covert transmitter should not be exempted from this rule, otherwise it would be easy to detect the covert channel. Thus, the size of the covert transmitter's contention window in the i^{th} transmission stage is calculated as follows:

$$T_i = \begin{cases} T_0 & 0 \leq i \leq \alpha \\ 2T_{i-1} & \alpha < i \leq M \\ T_M & i > M \end{cases} \tag{1}$$

Where M is the index of the last stage in which the transmitter expands its contention window. The parameter α is a design parameter of the proposed scheme and will be explained in the Section 4. Algorithm (1) shows how the covert clock is used in order to transmit a covert message. Here, the $wait()$ function exhausts certain number of clock ticks before it returns the control to the main process, and the $transmit\_pkt()$ routine transmits the packet through the channel and returns true if the actual receiver of the packet acknowledges the reception of the packet.

In addition to achieve stealthiness, expanding the transmitter's contention window plays a major role in synchronizing end peers of the covert channel. In fact, as the covert message is embedded in the covert clock, both nodes require accurate knowledge on the current state of the covert clock if they are about to communicate effectively. To this end, following each unsuccessful packet transmission attempt, the transmitter waits for exactly $T_i - \omega$ clock ticks before expanding its contention window and moving to the next stage. Hence, the covert clock is equal to $\sum_{j=0}^{i-1} T_j$ at the beginning of the $(i)^{th}$ stage regardless of the value of the covert message. The receiver removes this offset from the value of its covert clock (i.e., C_r) in order to decode the covert message. Thus,

$$\omega = C_r \ \ mod \ \ T_0. \tag{2}$$

Algorithm 2. Covert transmitter function of FRCC.

```
/* $PRNG(n)$ generates a random bit stream of length $n$.
$i = 0$; $\omega_s = \omega$; $T_{-1} = T_0$;
repeat
  if $\alpha < i < M$ then
    $b_i = PRNG(i - \alpha)$;
    $T_i = 2^{i-\alpha}.T_0$;
    $\omega_s = (b_i||\omega)$;
  else
    $T_i = T_{i-1}$;
  end if
  $Success = sendpkt(T_i, \omega_s)$;
  $i = i + 1$;
until $Success$
```

Algorithm 3. Covert transmitter function of ARCC.

```
/* $getbit()$ returns extra information bits to be concatenated to the original message.

$i = 0$; $\omega_s = \omega$; $T_{-1} = T_0$;
repeat
  if $\alpha < i < M$ then
    $b_i = getbit(i)$;
    $T_i = 2^{i-\alpha}.T_0$;
    $\omega_s = (b_i||\omega_s)$;
  else
    $T_i = T_{i-1}$;
  end if
  $Success = sendpkt(T_i, \omega_s)$;
  $i = i + 1$;
until $Success$
```

It is worth nothing that exhausting $T_0 - \omega$ clock ticks (instead of $T_i - \omega$) at the end of each stage also satisfies the synchronization criteria, however it deviates the covert transmitter's behavior from an ordinary user in the system. Hence, it is not an option for a stealth channel design.

As the size of the transmitter's contention window increases, there are more states in each stage that correspond to a particular message. For instance, if $T_i = kT_0$, there exist k states in the stage i that corresponds to the message ω (i.e., m^i_{ω}, $m^i_{T_0+\omega}$,..., $m^i_{(k-1)T_0+\omega}$). In FRCC scheme, the transmitter randomly picks one of the aforementioned states and moves to the next stage. Algorithm (2) depicts the transmitter's function of FRCC.

3.2 Adaptive Rate Covert Communication (ARCC)

In order to achieve stealthiness, the covert transmitter has to expand its contention window after each unsuccessful transmission attempt. In principle, by doubling the contention window in each expansion, the covert transmitter may add an additional information bit to the original message and increase the covert channel rate. ARCC scheme is designed to exploit the extra capacity and increase the covert communication rate. However, the rate increase comes at the price of reducing the reliability of the covert channel especially for the extra bits that are added to the original message. In other words, in order to decode the covert message, the receiver has to account not only for the value of the covert clock, but also it has to keep track of how many bits of information is added into the original message. Algorithm 3 depicts the transmission procedure of ARCC.

To decode the message, the covert receiver first decodes the original message according to Equation (2). Then, it checks the existence of extra information bits by removing the effect of first α stages from its covert clock. If C_r is still positive, it means that the transmitter had more than α unsuccessful re-transmission attempts, and it had to expand its contention window. The receiver counts the number of expansions by removing multiples of T_i from the value of the covert clock and decodes extra bits. Algorithm (4) depicts ARCC decoding process.

Algorithm 4. ARCC decoding at the receiver.

```
function $message = decode(C_r)$
/* First decode the original message $\omega_o$
$\omega_o = C_r \ mod \ T_0$;
$C_r = C_r - \omega_o$;
/* Decode the additional message $\omega_a$
if $C_r < \alpha T_0$ then
    return $\omega_o$
else
    $C_r = C_r - \alpha T_0$;
    for $i = 1 \ to \ M - \alpha$ do
        if $C_r < 2^i T_i$ then
            $\omega_a = \frac{C_r}{T_0}$;
            return $(\omega_a || \omega_o)$
        else
            $C_r = C_r - 2^i T_i$;
        end if
    end for
    $C_r = C_r \ mod \ 2^M T_0$;
    $\omega_a = \frac{C_r}{T_0}$;
    return $(\omega_a || \omega_o)$
end if
```

4 System Parameters

In this section, different parameters of the proposed covert communication scheme are derived. The main idea is to optimize these parameters to achieve maximum stealthiness (i.e., similar characteristics as compared to other nodes in the system) and maximum channel rate. From Figure 1, it can be observed that the proposed scheme mimics the same principles as a regular CSMA/CA system. Hence, to harmonies the behavioral fingerprints of these systems, it is

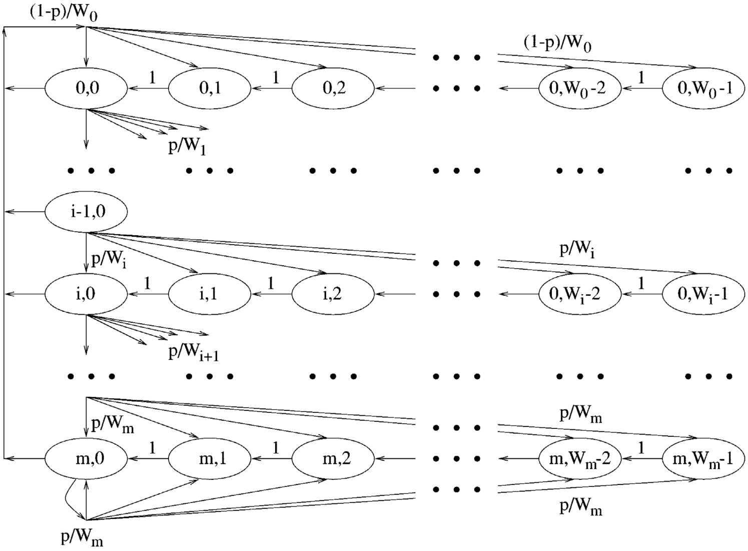

Fig. 2. Two-dimensional Markov model of the binary backoff scheme in CSMA/CA. In each stage W_i is the size of the contention window where $W_i = 2^i W_{min}$ [2]

important to derive the parameters of a regular CSMA transmitter and then adapt the covert transmitter to resemble the same characteristics.

Figure 2 depicts the two-dimensional Markov chain model for the binary backoff algorithm which is widely used for performance analysis of IEEE 802.11 MAC architecture [2]. Each state of this Markov process is represented by an ordered pair (s, b), where b represents the current state of the backoff counter and s represents the backoff stage of a given station. Using the above model, it is easy to verify that the collision probability p for a given user and the probability of transmitting a packet (i.e., q) can be written as [2]:

$$p = 1 - (1 - \sigma)(1 - q)^{N-2}, \tag{3}$$

$$q = \frac{2}{1 + W_{min} + pW_{min} \sum_{k=0}^{n-1} (2p)^k}. \tag{4}$$

Where, N is the number of users, σ is the transmission probability of the covert transmitter, and W_{min} is the minimum size of the contention window for regular users in the system. It is noted that the very first property of a transmitter is its transmission rate. Hence, if the covert transmitter has a different transmission rate as compared to a regular user in the system, it can be easily detected by a system observer. Therefore, the transmission rate of the covert transmitter is restricted to be the same as the transmission rate of regular users in the system (i.e., $\sigma = q$). Thus, one can rewrite Equation (3) as follows:

$$p = 1 - (1 - q)^{N-1}. \tag{5}$$

Let d_r^0 be the average number of time slots that each regular user has to wait before its first transmission (i.e., $i = 0$). Thus,

$$d_r^0 = \frac{W_{min} - 1}{2}. \tag{6}$$

On the other hand, the covert transmitter needs $\omega \in \{0, 1, ..., T_0 - 1\}$ successful packet transmissions from members of the covert set before it can send a packet to mark the value of the covert clock. Let π be the probability of a successful packet transmission of a user in the system. Hence, $\pi = q(1-q)^{N-1}$ and the probability of a successful transmission by members of the covert set can be derived as:

$$P_S = |S| \times \pi. \tag{7}$$

Therefore, the average number of slots that the transmitter has to wait before its first transmission attempt (i.e., d_c^0) can be written as the average number of slots in which the transmitter observes ω packets from members of S. Thus,

$$\begin{aligned}
d_c^0 &= \frac{1}{T_0}\sum_{\omega=0}^{T_0-1}\sum_{n=\omega}^{\infty} n.\binom{n-1}{\omega-1}P_S^{\omega-1}.(1-P_S)^{n-\omega}.P_S \\
&\overset{(1)}{=} \frac{1}{T_0}\sum_{\omega=0}^{T_0-1}\sum_{x=0}^{\infty}(x+\omega).\binom{x+\omega-1}{\omega-1}P_S^{\omega}.(1-P_S)^{x} \\
&= \frac{1}{T_0}\sum_{\omega=0}^{T_0-1}\Big[\sum_{x=0}^{\infty}x.\binom{x+\omega-1}{\omega-1}P_S^{\omega}.(1-P_S)^{x}+\omega.\sum_{x=0}^{\infty}\binom{x+\omega-1}{\omega-1}P_S^{\omega}.(1-P_S)^{x}\Big] \\
&\overset{(2)}{=} \frac{1}{T_0}\sum_{\omega=0}^{T_0-1}\Big[\sum_{x=0}^{\infty}x.\binom{x+\omega-1}{\omega-1}P_S^{\omega}.(1-P_S)^{x}+\omega\Big] \\
&\overset{(3)}{=} \frac{1}{T_0}\sum_{\omega=0}^{T_0-1}\Big[\frac{\omega(1-P_S)}{P_S}+\omega\Big] \\
&= \frac{T_0-1}{2P_S}.
\end{aligned} \tag{8}$$

Where (1) comes from $x = n - \omega$, and (2) and (3) are based on the definition of the negative binomial distribution [17].

In order to emulate an ordinary user in the system, the covert transmitter has to spend, on average, the same number of slots before its first transmission attempt as compared to any regular user in the system. To this end, by combining Equations (6) and (8), the proper value of the initial transmission window of the covert transmitter can be derived as:

$$T_0 = P_S(W_{min} - 1) + 1. \tag{9}$$

Hence, the covert transmitter and regular users in the system, on average, wait for the same number of slots prior to their first packet transmission attempt.

On the other hand, in order to maintain synchronization with the covert receiver, following each unsuccessful transmission, the transmitter resets the covert clock to $\sum_{j=0}^{i-1} T_j$ where i is the number of unsuccessful transmission attempts for the current packet (Figure 1). This re-synchronization task accounts for additional delay for the transmitter as compared to regular users in the system. Thus, if the transmitter doubles the size of its contention window after each unsuccessful transmission, the number of slots that the covert transmitter waits between consecutive packet transmissions may deviate from the same parameter of ordinary users. This difference in behavior can be detected by a system observer exposing the existence of the covert channel.

To combat this problem, the transmitter postpones expanding its transmission window for α stages. Where α is selected such that *the average number of slots between the last successful packet transmission and the α^{th} re-transmission attempt to send a new packet converges for both ordinary users and the covert transmitter*. Hence, the covert transmitter controls the delay between retransmission attempts in order to compensate for the additional delay due to the synchronization process.

Since the transmitter does not expand its transmission window up to the stage α, the average number of slots between the last successful transmission and the i^{th} re-transmission attempt for a new packet ($i \leq \alpha$) can be written as the average number of slots to observe $iT_0 + \omega$ packets from members of S. Thus, similar to the calculation of Equation (8):

$$\begin{aligned} d_c^i &= \frac{1}{T_0} \sum_{t=iT_0}^{(i+1)T_0-1} \sum_{n=t}^{\infty} n.\binom{n-1}{t-1} P_S^{t-1}.(1-P_S)^{n-t}.P_S \\ &= \frac{(2i+1)T_0 - 1}{2P_S}. \end{aligned} \tag{10}$$

Similarly, a regular user, on average, spends $\frac{W_i-1}{2}$ slots in the stage i before retransmitting the packet. It also spends one slot trying to transmit the packet at the end of each stage. Hence, the average number of slots between the last successful transmission and the i^{th} retransmission attempt for a new packet is:

$$d_r^i = \sum_{j=0}^{i} \frac{W_j - 1}{2} + i = \frac{W_{min}(2^{i+1} - 1) + i - 1}{2}. \tag{11}$$

Therefore, α can be derived as the index of the last stage in which the transmitter spends more slots, on average, trying to transmit a packet as compared to a regular user. Thus,

$$\begin{aligned} \alpha &= \max_{i>0} \ \{i| \ d_r^i \ \leq \ d_c^i\} \\ &= \max_{i>0} \ \{i|(\frac{2^{i+1} - 2i - 2}{i} \leq \frac{2 - 3P_S}{P_S W_{min}})\} \end{aligned} \tag{12}$$

Table 1. System PHY Parameters

Parameters	Selected Values
Slot Time	20 μs
SIFS	10 μs
DIFS	50 μs
Transmission Rate	1 $Mbps$
Payload Size	1500 $Bytes$

5 Performance Analysis

In this section the performance of the proposed covert communication scheme is analyzed from security, reliability and achievable rate point of view. Table 1 contains the basic parameters of the system under study in this section. The performance analysis is performed on four distinct scenarios to cover networks with different sizes and system parameters. For each scenario, the parameters of the covert system are calculated based on the discussion in Section 4. Table 2 contains the corresponding parameters of each scenario.

Table 2. Covert Communication Simulation Scenarios

Parameter	SC1	SC2	SC3	SC4
Number of users (N)	25	35	50	15
Size of the covert set ($\|S\|$)	16	21	33	10
Covert transmitter min window size (T_0)	4	7	8	4
Expansion postpone parameter (α)	1	1	1	1
Regular user min window size (W_{min})	16	32	32	16
Number of back off stages (M)	6	5	5	6

5.1 Detection and Stealthiness of the Covert Channel

As the covert receiver is a complete passive entity in the proposed scheme, it is undetectable even if the covert channel is detected. Indeed, since the transmitter does not need to know the receiver's identity, the receiver is safe in case that the transmitter's information is revealed to a system observer.

There are several different statistical tests to detect a covert channel and distinguish abnormal behaviors of a covert transmitter. One of the most well known approaches is the Kolmogorov-Smirnov test (KS-test) [17]. This test has been used in detecting the watermarked inter-packet delays and is a major tool in detecting timing covert channels [6].

Let $S(x)$ and $F(x)$ be the distribution of inter-packet delays of the covert transmitter's traffic and the legitimate traffic of the same system respectively. The KS-test is defined as:

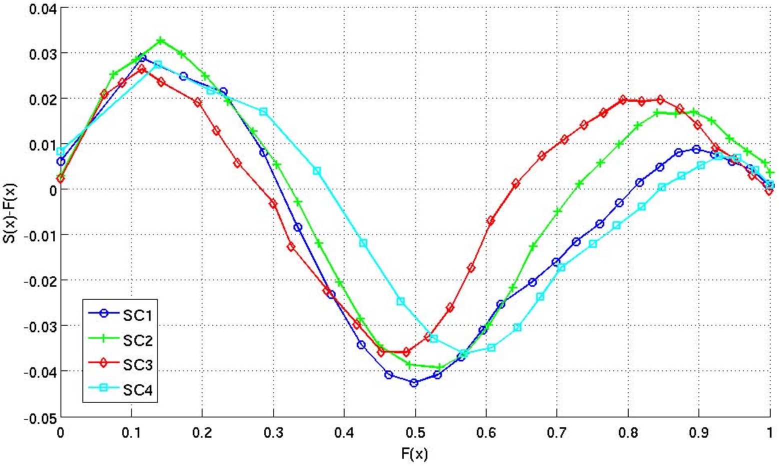

Fig. 3. Kolmogorov-Smirnov test for different scenarios

$$H_s = \sup_x |S(x) - F(x)|. \tag{13}$$

The difference between the distribution of inter-packet delays of the traffic originated from a regular user and the covert transmitter is depicted in Figure 3. According to the graph, the difference between two parameters is less than 5% at its peak which is an acceptable margin for the KS-test [16]. Such a small difference, makes it extremely difficult for an observer to detect any abnormal behavior in the system based on first level statistical tests such as the KS-test.

In addition, Figure 3 highlights how the transmitter systematically adapts its behavior in order to emulate the transmission pattern of ordinary users in the system. It is noted that the transmitter postpones its transmission window expansion for α stages in order to compensate for the extra delay caused by the synchronization process. Hence, the transmitter experiences lower inter-packet delays during the first α stages of the transmission process (i.e., the first peak of the graph). As the transmitter begins expanding its transmission window, the inter-packet delay of the transmitter's traffic increases faster than the delays of the regular traffic of the system (i.e., the second extreme point of the graph). This ends when the contention window of ordinary users are large enough such that the waiting times of the transmitter and regular users converge.

Another widely used statistical measure to detect timing covert channels is the *regularity test* [4] . In principle, the variance of the inter-packet delays changes over time due to different conditions of the network. In fact, regular users in the system have the same reaction to sudden events in the system such as packet loss or collision. However, as the covert transmitter is committed to transmit a particular covert message, it may not be able to react to network events similar to other nodes in the system. The regularity test is meant to detect such a behavior and track covert activities. To calculate the regularity test score, samples of the inter-packet delays are collected and then spread into multiple sets of size γ. The regularity score (i.e., H_r) is derived as follows:

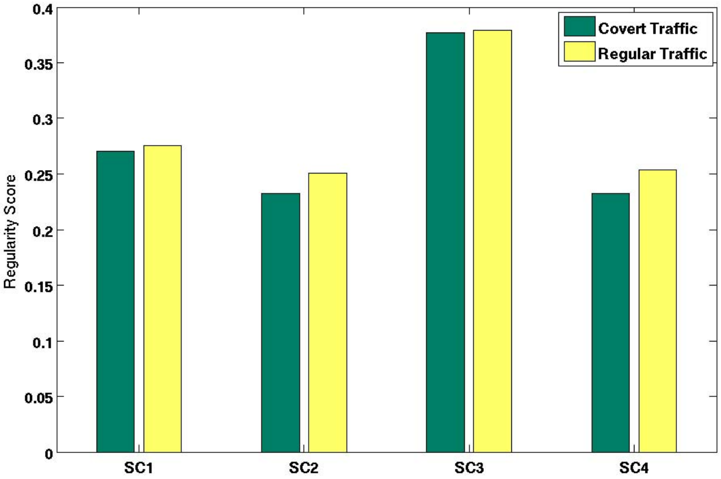

Fig. 4. Regularity test for different scenarios

$$H_r = std(\frac{|\sigma_i - \sigma_j|}{\sigma_i}), \quad \forall i, j, i < j. \tag{14}$$

Where std is the standard deviation operation and σ_i is the standard deviation of the i^{th} set of inter-packet delays. High regularity scores means large variance in inter-packet delays of each set while the low value of H_r depicts a set of regular inter-packet delays that is likely to carry covert information.

Figure 4 shows the regularity score of the covert transmitter and also ordinary users for $\gamma = 50$. According to the graph, the covert transmitter's regularity score is extremely close to the regular users' score in all four scenarios. In other words, the covert transmitter has managed to blend itself into the crowd well enough that a simple regularity test can not detect the existence of the covert channel.

The key in maintaining the regularity score is the packet transmission mechanism of the covert transmitter and the covert clock. It is noted that the covert clock increases based on activities of other nodes in the system. Hence, if the channel condition changes in a way that other users have to wait for a longer period of time between consecutive transmissions (e.g., reduction of channel capacity, high error rates, etc), the covert clock advances with a slower paste leading to larger inter-packet delays for the covert transmitter as well. Such an adaptive behavior is an advantage of the proposed scheme as compared to other similar techniques that aim to artificially increase their regularity score by switching the transmission mode after a certain amount of time [16] or replaying a part of previously sampled legitimate traffic and switch from one sample to another periodically [4].

5.2 Reliability

Two independent packet loss events are considered in order to evaluate the robustness of the proposed scheme. The first error event is due to failure in detecting packets from members of the covert set. Such an event directly affects the value of the covert clock in one side of the channel leading to erroneous decoding

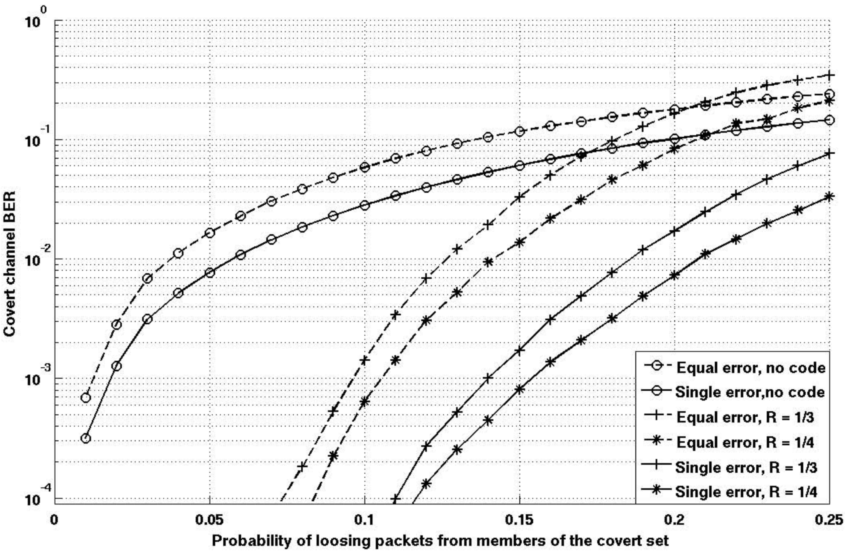

Fig. 5. Covert channel bit error rate based on the packet loss ratio of the packets from members of the covert set. Results are presented for the first scenario in Table 2, using FRCC scheme.

at the receiver. The second error event is caused by loosing the transmitter's packet at the covert receiver. If this happens, the covert receiver continues on incrementing its clock value while the transmitter resets its covert clock for the next round of transmission. Hence, the transmitter and the receiver loose synchronization and the covert channel becomes noisy.

There are several methods to reduce the effect of the aforementioned packet loss events on the performance of the proposed scheme. First of all, in most communication protocols, each packet is acknowledged by the actual receiver of the packet where the Ack message contains the sequence number and the source of the original packet. Thus, the receiver or the transmitter can learn about a packet by detecting either the packet or the corresponding Ack message.

Channel coding [18] is an alternative approach to combat the mismatch in the covert clocks of the transmitter and the receiver. In this way, the covert transmitter encodes the original message (e.g., a binary sequence $\mathcal{B}$) into a coded message which is more resilient against channel errors. Then, the coded message is modulated into covert messages based on the approach explained in section 3 to be transmitted over the covert channel using FRCC or ARCC schemes. In this section, a rate 1/3 convolutional code with the generator matrix [47; 53; 75] and a rate 1/4 convolutional code with the generator matrix [17; 13; 13; 15] are used in order to analyze the effect of channel coding on the performance of the proposed scheme. Due to the space limitation, the numerical results are presented for the first scenario in Table 2 using FRCC scheme.

Figure 5 depicts the covert channel bit error rate (BER) based on the ratio of the packet loss from members of the covert set. For each coding scheme, the graph depicts two extreme scenarios from packet loss view point. (i) the transmitter

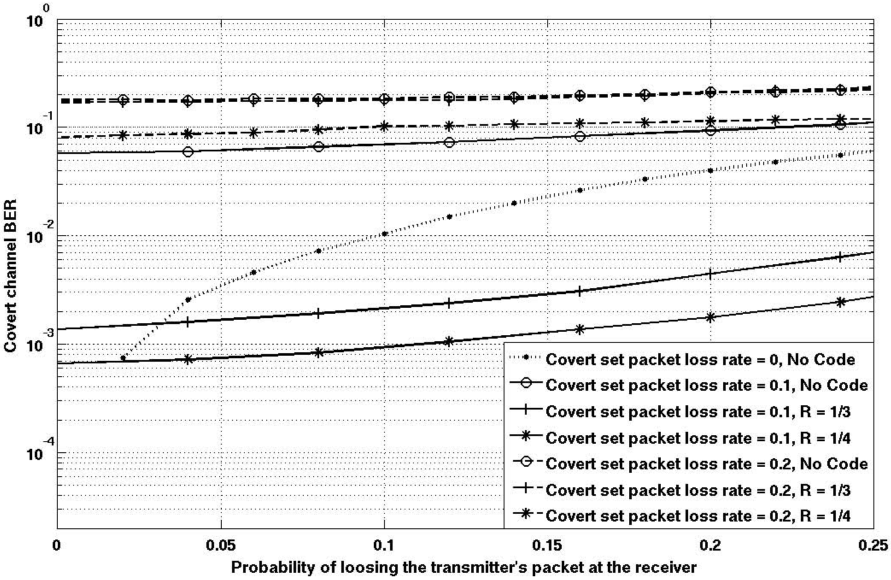

Fig. 6. Covert channel bit error rate due to the loss of transmitter's packets at the receiver. Packets from members of the covert set are subjected to error with packet loss rate 0 for dotted lines, 0.1 for solid lines, and 0.2 for dashed lines. Results are presented for the first scenario in Table 2, using FRCC scheme.

and the receiver have the same chance of loosing a packet from members of the covert set (i.e., equal error case). It is noted that the packet loss events for the receiver and the transmitter are assumed to be independent. (ii) only one of the receiver or the transmitter experiences packet loss from members of S (i.e., single error case). It is easy to verify that picking either the receiver or the transmitter in this case, does not change BER of the covert channel. In this way, all other possible scenarios are covered as their corresponding BER is bounded by the aforementioned extreme cases. The graph also illustrates that the channel coding can effectively improve the robustness of the proposed scheme against packet loss from members of the covert set. For instance, if both the receiver and the transmitter experience 15% packet loss from members of the covert set (i.e., equal error case), the BER of the covert channel drops from 0.1 (for the uncoded scenario) down to 0.035 for the rate 1/3 channel code and even further to 0.015 if the rate 1/4 code is used.

Figure 6 depicts the BER of the covert channel when the covert transmitter's traffic is also subject to error. Loosing the transmitter's packet at the receiver affects the synchronization between the two end points of the covert channel. It is noted that the covert receiver can learn about lost packets and re-synchronize itself with the transmitter when it gets a new packet from the transmitter (e.g., using the sequence number field of the new packet). However, the covert messages that were transmitted between the last received transmitter's packet and the new packet from the covert transmitter would be lost.

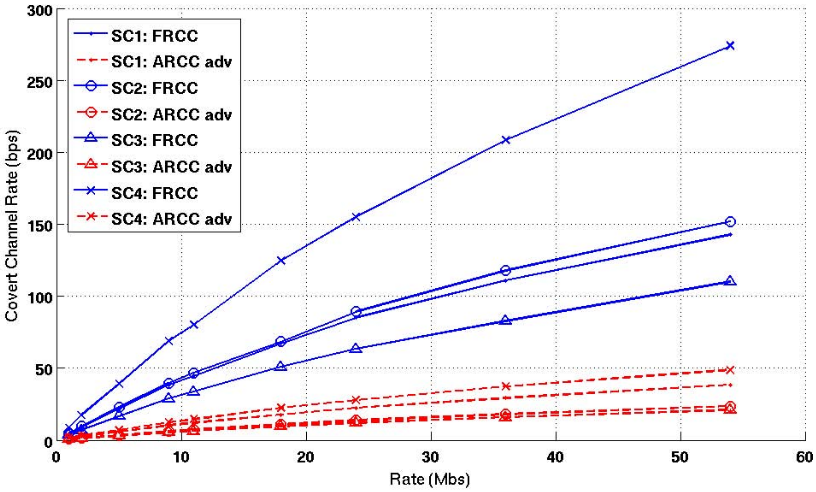

Fig. 7. Achievable rate of the proposed covert channel. Dashed lines show the extra capacity of ARCC scheme.

Plots in Figure 6 are grouped based on the ratio of the packet loss from members of the covert set. In all groups, it is assumed that the covert transmitter and the covert receiver experience the same packet loss ratio from members of the covert set (equal error case). Each group consists of three plots (one for the uncoded scenario, and two for the rate 1/3 and the rate 1/4 convolutional codes). Remarkably, if the receiver and the transmitter enjoy lossless channels from members of the covert set, the selected channel codes are strong enough to correct all errors caused by missing the transmitter's packets at the receiver. Hence, those plots are not reported in the graph.

By comparing the plots in Figure 5 and Figure 6, it can be observed that the BER of the covert channel increases much faster with the error from members of the covert set as compared to the error caused by loosing the transmitter's packet at the reviver. The flat plots of Figure 6 confirms this observation proving that the dominant factor in the reliability of the proposed scheme is in fact the packet loss from members of the covert set. This is due to the fact that the covert receiver can learn about the exact position and the number of lost packets from the covert transmitter (e.g., using the sequence number field of the transmitter packets), and use this information in order to improve the performance of the channel coding schemes that are being used in the system

5.3 Communication Rate

Figure 7 shows the achievable rate of the proposed scheme. The graph also illustrates that the channel rate increases linearly with the capacity of the communication channel. In fact, the covert channel is capable of achieving relatively high covert rates without compromising the security of the channel. The graph also depicts that it is possible to boost the capacity of the covert channel even further using ARCC scheme.

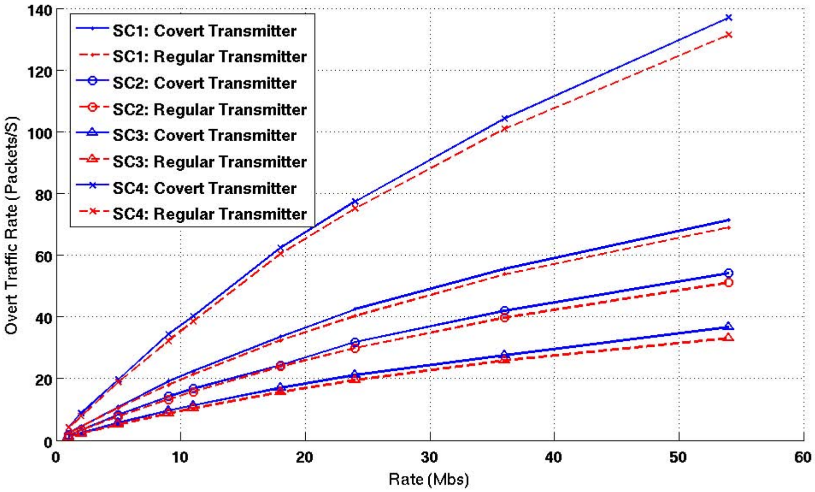

Fig. 8. Overt traffic communication rate of the system

Figure 8 depicts the overt communication rate of the transmitter and regular users in the system. According to the discussion in Section 4, the parameters of the proposed scheme are calculated based on the assumption that the covert transmitter has the same transmission probability as other users in the system. This assumption is crucial in order to prevent a system observer from tracking the transmitter based on its overt traffic rate. Figure 8 confirms the validity of this assumption as in all scenarios, the transmitter conveys the same overt transmission rate than other nodes in the system.

6 Conclusion

In this paper the concept of behavioral mimicry covert communication was introduced. In this way, it is possible to adopt a communication protocol and modify it in such a way that gives the covert transmitter enough freedom to embed a covert message into its overt traffic with minimum deviation from characteristics of a regular user of the same protocol. The paper also presents a new covert channel which is based on mimicking the structural behavior of CSMA/CA algorithm in the wireless environment.

The covert transmitter adopts CSMA/CA protocol so that the transmitter's contention window is controlled by a virtual clock called the *covert clock*. The covert clock is linked to the channel activities of all or a subset of regular nodes in the system using the broadcast nature of the wireless environment. These activities are observed by the covert transmitter and the covert receiver in order to synchronize their covert clocks and communicate through the covert channel. One important feature of the proposed scheme is that its rate linearly increases with the overt rate of the communication channel. Moreover, the covert transmitter and the covert receiver can maintain their overt channel rate like typical users in the system.

The performance of the proposed covert communication scheme is analyzed from stealthiness, reliability and communication capacity aspects showing that the proposed scheme has similar long term (statistical) and short term (temporal) characteristics as compared to legitimate traffic of the network. The numerical results also verify that the proposed scheme achieves relatively high communication rates with outstanding security and reliability scores.

References

1. Bhadra, S., Bodas, S., Shakkottai, S., Vishwanath, S.: Communication Through Jamming Over a Slotted ALOHA Channel. IEEE Transactions on Information Theory 54(11), 5257 (2008)
2. Bianchi, G., et al.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)
3. Butti, L., Veysset, F.: Wi-Fi Advanced Stealth. In: Proc. Black Hat, US (August 2006)
4. Cabuk, S., Brodley, C., Shields, C.: IP covert timing channels: design and detection. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 178–187. ACM (2004)
5. Calhoun Jr., T., Cao, X., Li, Y., Beyah, R.: An 802.11 MAC layer covert channel. Wireless Communications and Mobile Computing
6. Gianvecchio, S., Wang, H.: Detecting covert timing channels: an entropy-based approach. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 307–316. ACM (2007)
7. Girling, C.: Covert channels in LAN's. IEEE Transactions on Software Engineering 13(2), 292–296 (1987)
8. Gray III, J.: Countermeasures and tradeoffs for a class of covert timing channel. Hong Kong University of Science and Technology Technical report (1994)
9. Handel, T., Sandford, M.: Hiding Data in the OSI network Model. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 23–38. Springer, Heidelberg (1996)
10. Holloway, R.: Covert DCF - A DCF Based Covert Timing Channe. 802.11 Networks. Master's thesis, Georgia State University, Atlanta, Georgia (2010)
11. IEEE: IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1997)
12. Kemmerer, R.: Shared resource matrix methodology: An approach to identifying storage and timing channels. ACM Transactions on Computer Systems (TOCS) 1(3), 277 (1983)
13. Lampson, B.: A note on the confinement problem. Communications of the ACM 16(10), 613–615 (1973)
14. Li, S., Ephremides, A.: A Network Layer Covert Channel in Adhoc Wireless Networks. In: 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), pp. 88–96 (2004)
15. Li, S., Ephremides, A.: A covert channel in MAC protocols based on splitting algorithms. In: 2005 IEEE Wireless Communications and Networking Conference, pp. 1168–1173 (2005)

16. Liu, Y., Ghosal, D., Armknecht, F., Sadeghi, A.-R., Schulz, S., Katzenbeisser, S.: Hide and Seek in Time — Robust Covert Timing Channels. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 120–135. Springer, Heidelberg (2009)
17. Papoulis, A., Pillai, S., Unnikrishna, S.: Probability, random variables, and stochastic processes. McGraw-Hill, New York (2002)
18. Proakis, J., Salehi, M.: Digital communications. McGraw-Hill, New York (2001)
19. Szczypiorski, K.: HICCUPS: Hidden communication system for corrupted networks. In: International Multi-Conference on Advanced Computer Systems, pp. 31–40 (2003)
20. Wang, Z., Deng, J., Lee, R., Princeton, P.: Mutual anonymous communications: a new covert channel based on splitting tree MAC. In: 26th IEEE International Conference on Computer Communications, IEEE INFOCOM 2007, pp. 2531–2535 (2007)