Covert Channels through External Interference

Gaurav Shah and Matt Blaze
Department of Computer and Information Science
University of Pennsylvania
{gauravsh, blaze}@cis.upenn.edu

Abstract

This paper introduces *interference channels*, a new kind of covert channel that works by creating external interference on a shared communications medium (such as a wireless network). Unlike previous covert channels, here the covert sender does not need to compromise an authorized sender or require the ability to send messages on the network, but only needs the ability to jam traffic for short intervals. We describe an implementation of a wireless interference channel for 802.11 networks that can be used to superimpose low bandwidth messages over data streams, even when the network is encrypted or has other access controls. This channel is particularly well suited to watermarking VoIP flows, without compromising any routers or endpoint hosts.

1 Introduction

In this paper, we introduce *interference channels*, timing channels which can be induced without the need to compromise any trusted or untrusted part of the host or the network infrastructure. Interference channels are induced by causing external interference in a shared communication channel. The allows timing channels to be created without the aid of a compromised host, its peripherals, its OS and software and any intermediate routing nodes.

Although the idea of interference channels works for any shared medium communication network, we describe an implementation of a *wireless interference channel* where an external wireless jammer induces a timing channel on 802.11 network traffic by causing interference on the network.

The paper has three main contributions. First, we describe a new type of covert channel architecture where the covert sender exists in parallel to other legitimate senders on the network. Second, we show that such a channel architecture is practical by describing an implementation of it for 802.11 wireless networks. Finally, we show that the wireless interference channel can be used to watermark VoIP streams. Unlike previous schemes (e.g., [11]), this technique can perform watermarking without needing any help from the network infrastructure or communication end points.

2 Interference Channels

Broadcast based networks rely on multiple nodes communicating using a single shared medium. At any given time, only one node can be transmitting and other nodes have to wait for the medium to become idle again. A coordination function regulates access to the shared medium amongst multiple nodes.

Interference channels exploit the observation that the time at which a node can send a network packet on the shared network is affected by the activity of other nodes on the network and coordination function used. An external entity on the broadcast network modulates the availability of the shared medium, hence affecting network timings of other nodes on the network. The external entity on the network does not need the capability to send or receive packets, but only the ability go generate interference on it. The interference allows for the modulation of network timings of legitimate nodes on the network.

Interfering with a shared resource has previously been shown to be effective in performing host watermarking and detecting hidden services. For example, clock drift rates can be indirectly varied and observed by sending network traffic to a host causing a rise in its CPU load and heat output [5]. Here, network traffic is used to modulate a shared resource which can be observed remotely. However, we are interested in performing the opposite - interfering with a shared resource to modulate network traffic.

Although the effect of interference in shared medium communication systems has been speculated about in the security community, we are unaware of any concrete demonstrations of this technique in existing literature. Particularly restrictive from the covert sender's perspective is the requirement for having access to the shared medium. However, as we show in this paper, under certain conditions, this can be surprisingly practical vector for information leak-

age.

The mechanism of interference depends on the coordination function being used on the broadcast network. Coordination techniques on broadcast networks usually take the form of *Collision Avoidance* or *Collision Detection*. Collision avoidance generally uses a central controller which polls each node on the network for access to the medium in a round-robin manner. As the medium access is centrally coordinated, no contentions can occur. In contrast, collision detection is a decentralized mechanism for regulating access to the broadcast medium. Nodes transmit whenever they detect the medium as idle. If two nodes transmit at the same time, a collision occurs which is detected by both the nodes causing them to go into a collision resolution state. In this state, each node defers transmission by a random small amount of time before repeating the medium sensing and transmission attempt. Therefore, any detected collision delays the transmission of a network packet until the medium is idle and no collisions occur.

Interference channels are a threat in broadcast networks which use collision detection. The lack of a central coordinator makes it possible for any single node on the network to control access to the medium. For example, an interference agent can inhibit transmission on the network by making the other nodes detect the medium as being in use and busy. An interference channel relies on attacking the coordination function on a broadcast network to modulate the times at which hosts on the network can communicate.

3 Wireless Interference Channel

Consider an internal wireless network which is also connected to an external public network like the Internet. In typical home and enterprise environments, wireless network access is protected using link encryption and access control protocols (e.g. WPA) to thwart unauthorized access and eavesdropping. In this scenario, the wireless interference channel can induce a timing channel in network traffic without needing to compromise any component of the hosts or the network.

The wireless interference channel works by using an external wireless jammer which jams the wireless medium for short periods of time preventing any network activity during that period. The jamming is performed at the physical medium layer and no actual access to the wireless network is required. This also means that the higher link and MAC layer security mechanisms do not protect against such interference. By preventing network transmission at appropriate times, the jammer can control network packet timings and induce a covert timing channel. From the perspective of the jammer, the network packets of interest are those that traverse the network boundary onto the public network where their timing can be observed by an eavesdropper.

Figure 1 shows a high level overview of the Wireless Interference Channel. The jammer interferes with communication between the host and the wireless router. The input to the jammer is any information that needs to be sent over the interference channel and can include sensor data about a host's local environment, watermarking bits, etc. The network eavesdropper lies on the public network where it is able to observe timings of network packets emanating from the wireless access point.

4 Wireless Jamming

The 802.11 wireless standard [3], defines specifications and protocols for wireless local area networks (WLAN). The 802.11 networks work in the ISM band of the radio spectrum which means that in most countries, if the devices follow maximum allowed output power guidelines, users need not to be licensed or coordinated by regulatory agencies (e.g. FCC in the United States). The 802.11 physical layer (PHY) standard defines various modulation techniques for encoding information for transmission at various speeds. These range from 1-11Mbps (for 802.11b networks) to 1-54Mbps (for 802.11g networks). Spread spectrum techniques are used for robustness and low interference. The lower speed networks typically use some form of frequency hopping (FH) whereas at higher speeds Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency Domain Multiplexing (OFDM) are used.

The 802.11 medium access control layer (MAC) is responsible for coordinating access to the wireless medium. Most 802.11 networks use the decentralized Distributed Coordination Function (DCF) where Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) regulates access to the shared medium. By using a combination of physical and virtual carrier sensing, multiple nodes can coordinate while talking to an access point.

4.1 802.11 Jamming

Because 802.11 networks use radio, they are inherently susceptible to jamming. Jamming in the 802.11 networks can be performed either at the PHY layer or at the MAC layer. As we assume that the interference agent has no access to the wireless network,

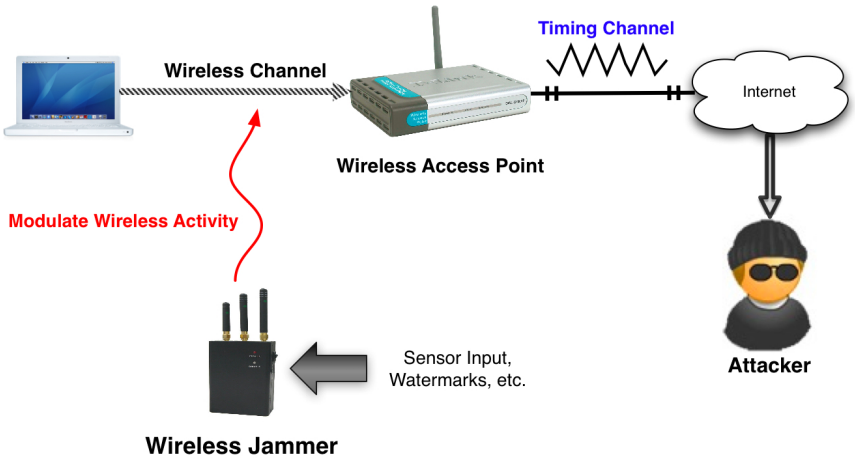

Figure 1: High-level overview of the Wireless Interference Channel

we focus on PHY layer jamming. PHY layer jamming techniques usually rely on sending a high powered narrowband signal corresponding to the 802.11 frequency band. This prevents nodes in a 802.11 network from communicating via an access point.

Most current 802.11 networks use spread spectrum techniques such as DSSS for encoding information. With DSSS, the signal is spread over the entire band using a spreading code (a random binary string). This mapping of the signal over the entire frequency band makes it easy to handle narrowband noise and hence is the scheme used for higher rate transmissions. In fact, DSSS causes a process gain of 10db to the signal [9] requiring more power to jam it. However, the rather low power output of 802.11 equipment makes it easy to jam DSSS based networks using cheap off-the-shelf equipment.

4.2 Jammer Implementation

Our prototype jammer is a narrow band jammer implemented using the GnuRadio [2] software defined radio (SDR) framework on a Universal Software Radio Peripheral (USRP) [10]. It transmits a continuous wave signal at the 802.11 network channel's center frequency.

The Universal Software Radio Peripheral (USRP) is a programmable radio hardware device that allows general purpose computers to digitally control and implement highly configurable software radios. The USRP interacts with the host computer using the USB interface and can be controlled using the GNU Radio library.

The USRP narrow band jammer uses the XCVR2450 daughterboard with an attached vertical antenna. This board is a RF transceiver for the 2.4-2.5 GHz and 4.9 to 5.85 GHz ISM band. The frequency of interest for 802.11 networks is 2.4-2.5 GHz. To interfere with a wireless network, the USRP is programmed to send a narrowband high power signal on the center channel frequency of the network. This center channel frequency is determined by the channel number of the wireless network of interest. By sending a sufficiently powerful narrow band signal, it is possible to block access to the wireless medium for certain periods of time. The XCVR2450 provides a stated power output of around 100mW which is sufficient to jam wireless communication on most consumer 802.11 access point based networks. Our jammer implementation provides a peak power output between 18.5-20 dBm (72-100 mW) as measured using a Rohde & Schwarz URV5 RF Powermeter.

The implementation is controlled by the host computer which interacts with the USRP to activate the narrow band jam signal at specific times to create interference on the wireless network.

In general, an 802.11 jammer can also be created without using such specialized hardware. Certain commodity off-the-shelf 802.11 hardware can also be configured to perform it instead.

4.3 Jamming Strategies

We considered 3 different strategies for a jammer to influence packet timings on a wireless network.

In *whole medium jamming*, the jammer blocks the whole medium between the wireless sender (client) and receiver (access point). This has the effect of preventing the sending of a wireless frame either due to the medium being detected as busy by the sender or due to frame corruption if the interference is caused while the frame is being transmitted.

In *sender-only jamming*, the wireless jammer always jams the medium near the sender. If the jam-

ming is only performed when the wireless medium is detected as idle, this has the advantage of the 802.11 MAC layer deferring transmission and not causing a random backoff (because of no collisions). However, this requires that the jammer actively carrier-sense to ensure that the medium is free before it causes interference.

Finally, the most sophisticated strategy for jamming is for the interference agent to act as a Layer 1 switch. In *receiver-only jamming*, the jammer prevents the access point from receiving a frame at the time it is sent by the sender. The sender is kept totally oblivious of this failed delivery. This can be done by only generating the jam signal in the vicinity of the access point (for example, by using a directional antenna). To induce the timing channel, the jammer receives the frame itself while blocking the access point, adds the necessary amount of delay and resends it to the access point essentially doing man-in-the-middle to act as a Layer 1 delay switch.

In our implementation, we always use the simplest strategy of jamming the whole medium without explicitly targeting either the sender (wireless client) or the receiver (access point). The modulation precision this simple strategy affords is sufficient for sending short sequences of information like that needed for applications such as flow watermarking.

5 Results

In this section, we first describe our encoding scheme (Section 5.1). We show the effect of this scheme on constant high rate network traffic in Section 5.2. In Section 5.3, we show that this simple encoding scheme allows information leakage over VoIP calls.

5.1 Encoding Scheme

Our encoding scheme assumes that the wireless network is moderately saturated such that the average inter-packet delay is below a certain threshold. For the encoding parameters used in our implementation, this requires the inter-packet delays to be <100ms. The wireless jammer waits for a fixed *wait time* before jamming the network for an *interference duration*. Each waiting and interference period constitutes one *jam cycle*. A bit is encoded by using multiple jam cycles and varying the interference duration for each of these cycles. The jammer also waits between sending each bit of information. This *recovery time* ensures that the wireless medium is available for normal transmission and helps prevent aggressive backoff by the 802.11 MAC layer which may lead to noise on the timing channel.

For sending bit *zero* the jammer uses increasingly higher interference durations for subsequent jam cycles. For example, if the wait time is 0.5 seconds and 3 jam cycles are used for sending each bit, the interference durations could be picked as 0.2, 0.3 and 0.4 seconds. For sending *zero*, the encoder would jam the network for 0.2 seconds, wait for 0.5 seconds, jam the network again for 0.3 seconds, wait for 0.5 seconds and finally jam the network again 0.4 seconds, wait for 0.5 seconds before sending out the next bit. For sending *one*, the jammer does the opposite - it uses decreasing interference durations for each of the subsequent jam cycles.

These incremental jam patterns shape the distribution of inter-packet delays as observed by the covert channel receiver. An interference duration of x seconds will cause at least one observed inter-packet delay to be $> x$. If this interference duration is much larger than the average inter-packet latency of the network traffic, then the covert channel receiver can filter out these *interesting* inter-arrival times and search for interference patterns corresponding to the bit encodings within them.

As the encoding scheme relies on using inter-arrival times for performing bit encodings, both the interference agent (wireless jammer) and channel receiver (network eavesdropper) can use the real time clock as a reference for measuring time durations while performing the encoding and decoding. No explicitly shared and synchronized clocks are required.

5.2 Effect on periodic network traffic

To test if the above encoding scheme works in practice, we setup an experimental testbed by creating an instance of a wireless network using an 802.11g access point. This access point is connected to our lab local area network. A wireless node connects to the access point and generates network traffic directed towards one of the hosts on the LAN. Our jammer implementation is used to create wireless interference and send a sequence of alternating zeros and ones. The covert channel receiver is a network eavesdropper running at the destination of the network traffic.

We use a wait time of 0.5 seconds. Each bit is sent using 3 jam cycles. The interference duration patterns of $\{0.1, 0.2, 0.3\}$ and $\{0.3, 0.2, 0.1\}$ are used for sending *zero* and *one* respectively.

We use two different types of traffic profiles. The first consists of a constant stream of UDP network packets generated every 10ms closely resembling applications with real time constraints like streaming media and VoIP. The second profile uses a TCP based network workload. File transfer is initiated using `scp` over the wireless network to the LAN host. Figure 3 shows the effect of wireless interference on the observed inter-arrival time trace for TCP traffic. In

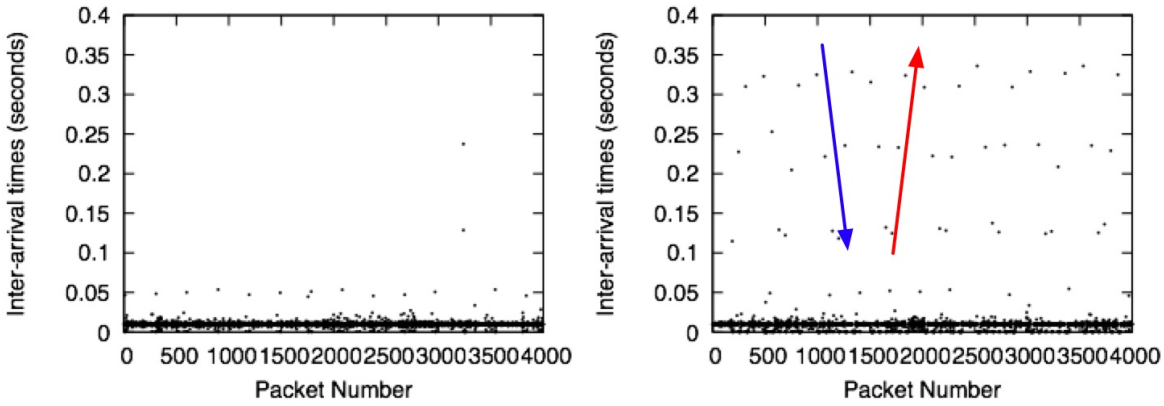

Figure 2: Inter-arrival times without and with wireless interference for a constant rate UDP transmission. With interference, the bit encoding patterns (marked by arrows) are clearly visible.

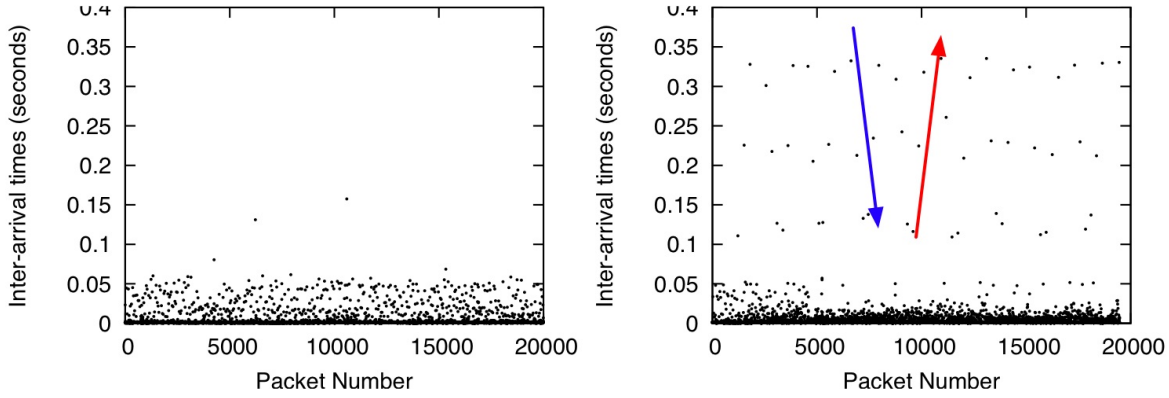

Figure 3: Inter-arrival times without and with wireless interference for a `scp` file transfer. With interference, the bit encoding patterns (marked by arrows) are clearly visible.

particular, with wireless interference, the inter-arrival time patterns (shown using arrows in the figure) for bits zero and one are clearly visible. In fact, the covert channel receiver is able to decode these bits with almost 100% accuracy. The results for the synthetic UDP workload (Figure 2) are similar.

Note however that an interference duration of 0.1s does not translate to packet inter-arrival times of exactly 0.1s. Even with UDP, the induced delays are slightly larger than interference duration. This is likely caused either due to 802.11 MAC layer effects or is a property of the jammer implementation. However, if the input inter-packet latency of legitimate traffic is less than that 0.1s, then these encoding inter-arrival times can be filtered by the receiver.

5.3 Embedding Tracking Information in VoIP streams

To demonstrate the capability of the wireless interference channel in adding timing based watermarks to a VoIP stream, we use Skype [8] to setup a call between a host located in the Distributed Systems Lab at Penn connected to the Internet via a 802.11g access point and a node located on the Emulab [13] network housed in the University of Utah. The node is 20 hops away with an average round trip time of 80ms. Although Skype doesn't seem to use silent intervals where call audio packets stop being generated whenever silence is detected, we emulate an actual call by setting up audio loopback devices with speech audio files playing on either endpoints of the call.

The network eavesdropper (watermark detector) is located at the other endpoint of the call at the node on the Emulab network. The eavesdropper looks at the timing network packets corresponding to the Skype call and extracts the watermark from their timing. We use the encoding scheme from Section 5.1 to test watermark decoding accuracy for different encoding parameters.

Jam Sequence(j_1, j_2, j_3)	**Wait Time**(t_w)	**Recovery Time**(t_r)	**Bit Time**(t_b)	**Error Rate**
(0.08,0.1,0.12)	0.4s	0.5s	2.0s	28.7%
(0.08,0.1,0.12)	0.7s	0.5s	2.9s	25.6%
(0.08,0.1,0.12)	1.0s	0.5s	3.8s	35%
(0.08,0.1,0.12)	0.4s	1.0s	2.5s	26%
(0.08,0.1,0.12)	0.7s	1.0s	3.4s	28%
(0.08,0.1,0.12)	1.0s	1.0s	4.3s	38.2%
(0.1,0.2,0.3)	0.4s	0.5s	2.3s	93.5%
(0.1,0.2,0.3)	0.7s	0.5s	3.2s	87%
(0.1,0.2,0.3)	1.0s	0.5s	4.1s	91.5%
(0.1,0.2,0.3)	0.4s	1.0s	2.8s	94.5%
(0.1,0.2,0.3)	0.7s	1.0s	3.7s	88%
(0.1,0.2,0.3)	1.0s	1.0s	4.6s	91%
(0.2,0.4,0.6)	0.4s	0.5s	2.9s	82%
(0.2,0.4,0.6)	0.7s	0.5s	3.8s	78.7%
(0.2,0.4,0.6)	1.0s	0.5s	4.7s	83.1%
(0.2,0.4,0.6)	0.4s	1.0s	4.0s	85%
(0.2,0.4,0.6)	0.7s	1.0s	4.9s	79.5%
(0.2,0.4,0.6)	1.0s	1.0s	5.8s	81%

Table 1: **Measured Bit Error Rates for sending 200 watermarking bits over a Skype Call**

Table 1 lists the measured watermark accuracy rate for different parameters of the encoding scheme. The bit time t_b is the amount of time taken to send one bit. It is equal to $j_1 + j_2 + j_3 + 3t_w + t_r$, where t_w is the wait time, t_r the recovery time and (j_1, j_2, j_3) are the interference duration sequences for sending each bit.

We tested the interference channel with three different jamming sequences, each more aggressive than the previous one. From the measured error rates, it is clear that both weak interference (smaller jam times) and strong interference (larger jam times) cause the channel to become noisy and increase the error rate.

The encoding scheme relies on the eavesdropper's ability to distinguish inter-arrival times caused due to interference from legitimate ones. Smaller jam times in the jam sequence make it more difficult to distinguish encoding intervals from legitimate ones. On the other hand, more aggressive larger jam times also cause the accuracy to drop. A higher amount of interference on the network causes interactions with the 802.11 MAC layer where legitimate nodes backoff on transmissions because the medium is detected as busy or noisy. This causes further noise in the jammer induced encoding inter-arrival times leading to a higher error rate. Therefore, both jam sequences of $(0.08, 0.1, 0.12)$ and $(0.2, 0.4, 0.6)$ do not perform as well than the jam sequence of $(0.1, 0.2, 0.3)$.

Finally, our experiments indicate that both the *wait time* and *recovery time* do not influence the accuracy of the channel significantly. The channel performance stays similar. However, because of the parasitic nature of the wireless interference channel, we want these to be high enough so that the actual available bandwidth of legitimate senders is not affected significantly (at the cost of decreased interference channel bandwidth).

Although these results correspond to a direct Skype connection without the use of an anonymizing network, we expect a low-latency anonymizing network to not affect the results in a major way. In fact, our earlier work [7] used inter-arrival time based encoding schemes and demonstrated that observed network jitter on most paths on the Internet are low enough to make the use of such schemes practical. Here, we are using inter-arrival times of the order of 100ms, which is much less than the approximately 10ms average observed jitter on most Internet paths [1].

6 Applications

The ability to superimpose a timing channel on network traffic without the cooperation of hosts or network traffic has many interesting applications. The interference channel acts as a subliminal out-of-band channel which can be augmented with information about the network traffic, its source or originating network.

Information about the local host and network environment The interference modulator will generally be placed in proximity to the local host and

network environment where it can generate the interference required to induce the channel. This has many malicious sensing applications. For example, the modulator can be outfitted with a GPS-like device which can generate location tracking information that can then be sent over the channel. This attack is particular effective if the modulator is part of a mobile host or surreptitiously installed within it.

Similarly, the modulator can also be augmented with other types of environment sensors. The type of information that can be collected and sent over the channel depends on its bandwidth. For the wireless interference channel, with a bandwidth of ≈ 0.3 bits/s, sensor data which is small in length can be sent, for example, temperature. With higher capacity interference channels, collected information requiring higher bandwidths like environmental audio and images are also a possibility.

Note that although the bandwidth of the timing channel restricts the types of information that can be sent over it, a smart attacker can offset this disadvantage by moving the processing of the raw sensor information to the modulator. If the available covert bandwidth is much smaller than the actual sensor bandwidth, the interference modulator can choose to only send summary information or elements that have been determined to be sensitive and of interest to the attacker.

Tagging network streams The problem of VoIP tracking is that of determining if two observed VoIP streams at different tap points within a network correspond to the same call. When call signalling and content information is cryptographically protected, and the call is setup through an anonymizing network, this correspondence can not be determined by examining and comparing the contents and metadata of observed network packets. Moreover, passive timing based correlation techniques (e.g., [4]) do not work since VoIP flows exhibit similar and periodic timing characteristics. Wang et al. [11] were the first to propose an active timing-based watermarking approach to perform VoIP tracking. A short unique bit sequence (watermark) is embedded into an outgoing call flow by modifying the timing of network packets at the caller's end. Incoming VoIP flow timings can then be observed by a watermark detector at the callee's end to extract the watermark and perform correlation between the caller and callee.

Such active watermarking approaches (e.g., [6, 12, 14]) require the help of an intermediate router (e.g., by the ISP) or the call endpoint (e.g., through a compromised host) to perturb flow timings to embed the watermark. However, with an interference channel, watermarking can be performed without any help from the host or network infrastructure. In particular, this makes possible the *parking lot attack*. Wireless network access points typically have a much longer range than the usual proximity in which they are used. A wireless jammer can be placed quite further away, say in the parking lot of a building, from the actual physical location of the access point of interest and still have the capability to interfere with the activity on the wireless network. In spite of the spatial gap between them, the interference channel modulator can perform watermarking of network streams (including VoIP calls) that are being routed by the access point.

In general, interference channels can aid in the watermarking of any type of network traffic which is highly regular or indistinguishable across different flows, or where passive comparison of timing characteristics is not sufficient in correlating flows.

7 Conclusion

We introduced and analyzed a new type of covert timing channel architecture where the covert sender exists in parallel to other legitimate senders on the network. We showed that such channels are practical by describing an implementation of a wireless interference channel for 802.11 networks. Finally, we demonstrated the practical utility of the wireless interference channel by showing its use for solving a practical problem, that of watermarking VoIP flows without the need for any modifications to the network infrastructure or the communicating hosts. A simple encoding scheme allows encoding one watermark bit for every 2.5 seconds of network traffic which can then be decoded by an eavesdropper with around 90% accuracy. The wireless interference channel is not only useful for watermarking. By placing the wireless interference agent near the wireless host, and augmented with environmental sensors, it can leak small but sensitive information about a host's local environment.

Partial support for this work was provided by NSF grants CNS-0627579 and CNS-0831375.

References

[1] Acharya, A., and Salz, J. A Study of Internet Round-Trip Delay. Tech. Rep. CS-TR-3736, University of Maryland, 1996.

[2] GNURadio. GNU Radio: the GNU software radio. *http://www.gnuradio.org*.

[3] IEEE. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std 802.11-2007, 2007.

[4] Levine, B., Reiter, M., Wang, C., and Wright, M. Timing Attacks in Low-Latency Mix Systems. In *Proceedings of Financial Cryptography: 8th International Conference (FC 2004): LNCS-3110* (2004).

[5] Murdoch, S. J. Hot or not: revealing hidden services by their clock skew. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security* (New York, NY, USA, 2006), ACM, pp. 27–36.

[6] Pyun, Y. J., Park, Y. H., Wang, X., Reeves, D., and Ning, P. Tracing Traffic through Intermediate Hosts that Repacketize Flows. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (May 2007), 634–642.

[7] Shah, G., Molina, A., and Blaze, M. Keyboards and Covert Channels. *Proceedings of the 15th Usenix Security Symposium* (August 2006).

[8] Skype. *http://www.skype.com* (2008).

[9] Ståhlberg, M. Radio Jamming Attacks Against Two Popular Mobile Networks. In *Helsinki University of Technology Seminar on Network Security* (2000).

[10] USRP. USRP: Universal Software Radio Peripheral. *http://www.ettus.com/*.

[11] Wang, X., Chen, S., and Jajodia, S. Tracking anonymous peer-to-peer VoIP calls on the internet. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security* (New York, NY, USA, 2005), ACM Press, pp. 81–91.

[12] Wang, X., Chen, S., and Jajodia, S. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. *Security and Privacy, 2007. SP '07. IEEE Symposium on* (May 2007), 116–130.

[13] White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., and Joglekar, A. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation* (Boston, MA, Dec. 2002), USENIX Association, pp. 255–270.

[14] Yu, W., Fu, X., Graham, S., Xuan, D., and Zhao, W. DSSS-Based Flow Marking Technique for Invisible Traceback. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 18–32.