

On Covert Acoustical Mesh Networks in Air

Michael Hanspach and Michael Goetz

Fraunhofer FKIE, Wachtberg, Germany

E-Mail: {michael.hanspach, michael.goetz}@fkie.fraunhofer.de

Abstract— Covert channels can be used to circumvent system and network policies by establishing communications that have not been considered in the design of the computing system. We construct a covert channel between different computing systems that utilizes audio modulation/demodulation to exchange data between the computing systems over the air medium. The underlying network stack is based on a communication system that was originally designed for robust underwater communication. We adapt the communication system to implement covert and stealthy communications by utilizing the ultrasonic frequency range. We further demonstrate how the scenario of covert acoustical communication over the air medium can be extended to multi-hop communications and even to wireless mesh networks. A covert acoustical mesh network can be conceived as a meshed botnet or malnet that is accessible via inaudible audio transmissions. Different applications of covert acoustical mesh networks are presented, including the use for remote keylogging over multiple hops. It is shown that the concept of a covert acoustical mesh network renders many conventional security concepts useless, as acoustical communications are usually not considered. Finally, countermeasures against covert acoustical mesh networks are discussed, including the use of lowpass filtering in computing systems and a host-based intrusion detection system for analyzing audio input and output in order to detect any irregularities.

Index Terms— malware, network covert channels, wireless mesh networks, ultrasonic communication

I. INTRODUCTION

If we want to exploit a rigorously hardened and tested type of computing system or networks of this type of computing system, we have to break new ground. Covert channels are communication channels utilizing means for communications that have not been designed for communication at all [3]. With a covert channel, we can circumvent system and network security policies by exploiting new, previously unregarded communication media. In operating systems, covert channels are usually established by exploiting shared resource access between different processes, establishing a covert storage channel by encoding data in parts of the operating system that were not considered for communication at all or establishing a covert timing channel by manipulating and analyzing the timing behavior of shared resources. In computer networks, covert storage channels can be established by utilizing normally unused parts of communication protocol headers and covert timing channels could be established over the timing behavior of network requests.

As a shared media for covert communications, not only shared computing resources or preexisting network interfaces could be used. One can imagine a covert network channel

between different computing systems, establishing a completely new network interface based on physical emanations (see also [4]). Alongside the established radio emanations frequently used in network communications, optical or acoustical emanations could also be used as means for communications. Although the existing radio communication standards are mature enough for regular wireless communications, they are not used in our scenario as they are already known to the computing system's operating system and actively managed by the operating system. For covert communications we specifically demand devices that are:

- 1) Usable as either a sending or a receiving device, i.e., they are able to output or input a physical emanation type.
- 2) Accessible to the sending or receiving process.
- 3) Not yet established as a communication device and, therefore, not subject to the system and network policies.
- 4) Able to support stealthy communication to prevent immediate detection of the covert channel.

In this article, we specifically target covert communications over acoustical emanations, utilizing speakers and microphones (available in commonly available computing systems) as sending and receiving devices (see 1) that are commonly accessible (see 2) to application partitions of the operating systems that might be in need to process audio (e.g., for video conferencing or IP phone communications). Speakers and microphones are also not established as means for communication and are not widely considered in security and network policies (see 3). Regarding 4, stealthy communication can be implemented in acoustical networks by utilizing inaudible frequencies, i.e., the ultrasonic and near-ultrasonic frequency ranges.

We do not only target covert communication between two computing systems or two isolated application partitions on a single computing system, but we establish means for multi-hop communications between different participants. As the underlying communication system, we utilize a specific network stack that has originally been developed for underwater communications. Preexisting TCP/IP stacks could not reasonably be utilized for acoustical networks due to their comparably large overhead, but a loosely related addressing scheme with less overhead is utilized (see Section III).

By providing multi-hop communication, we can significantly extend the communication range of the covert channel in order to interconnect scattered devices to a full-fledged wireless mesh network. With a covert acoustical mesh network, we can offer a whole range of *covert services* to the participating computing systems, including Internet access via

Originally appeared in Journal of Communications.
This version is derived from [1] and [2].

an IP proxy.

The remainder of this article is structured as follows. In Section II, we introduce the concept of covert networks and a scenario for covert and stealthy wireless communication between isolated computing systems. In Section III, we present the fundamental concepts of the underlying communication system that we are using for covert communications in the targeted scenario. In Section IV, we describe our experimental setup for covert communications and necessary adaptations to the utilized communication system. We construct an acoustical mesh network and describe the performance and reliability of the utilized communication system in the targeted scenario. In Section V, we present different applications for covert mesh networks. In Section VI, we explore potential countermeasures against participation in a covert acoustical mesh network. In Section VII, we discuss related work and how it differs from our work. In Section VIII, we conclude the article, discussing our results and giving an outlook at possible future research.

II. SCENARIO

The basic scenario for covert communications between two computers is depicted in Fig. 1.

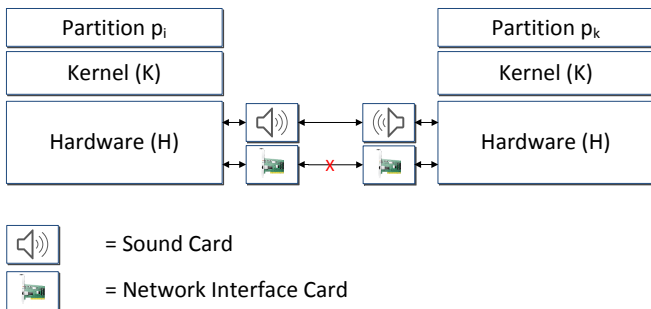


Figure 1. Scenario for two computers as part of a covert network.

Two computers that are not connected to each other via established types of network interfaces (e.g., IEEE 802.3 Ethernet [5] or IEEE 802.11 WLAN [6]) or that are prohibited to communicate with each other over these established types of network interfaces are, nevertheless, able to communicate with each other by using their audio input and output devices (microphones and speakers).

We assume a high-assurance setup where a component-based operating system (as described by Jaeger et al. [7] as an operating system that consists of a small trusted computing base and individual service components) is used to define fine-grained mandatory access control policies for communications between isolated application partitions. A component-based operating system is usually governed by a reference monitor in the kernel K (as introduced by Anderson [8]), which is an access control monitor that has always to be invoked in IPC (here: inter-partition communication) decisions.

Even in a component-based operating system that is governed by a reference monitor, acoustical communication is

possible between p_i and p_k , representing application partitions on different computing systems, as long as audio input and output devices as part of the underlying hardware H are accessible to both p_i and p_k (the utilized terminology is also described in [4]). While it would be an easy solution to just disallow access to the audio hardware, this is not possible, when p_i and p_k are in need of audio access (e.g., for IP phone conversations). Still there are applicable countermeasures against participation of a computing system in a covert acoustical network, as it will be presented in Section VI.

We will now extend the scenario of a covert acoustical network to a covert acoustical *mesh* network. A possible topology of a covert mesh network is depicted in Fig. 2.

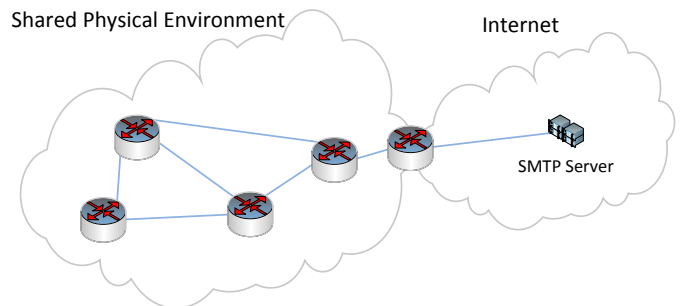


Figure 2. Topology of a covert mesh network that connects a shared physical environment to an SMTP server inside the Internet.

In a covert acoustical mesh network, more than two computing systems in a shared physical environment (i.e., within the physical communication range between two connected nodes) can be connected to the mesh network and computing systems are able to communicate indirectly by following routing paths over multiple hops. We distinguish between three types of participants in a covert mesh network:

Infected drone

An infected computing system that offers covert services or serves as a router in the covert mesh network.

Infected victim

An infected computing system that is targeted by the attacker to secretly leak information to other participants of the covert mesh network.

Attacker

The computing system controlling the covert mesh network, and the receiver of leaked information.

Each of these participants is configured as a sender or a receiver. In case of the infected drone, the computing system has to be configured both as a sender and a receiver, while the infected victim could just be used as a sender and the attacker could only be configured as a receiver. All participants must have installed a compatible acoustic communication system, either by infection of a malware or actively installed (on the attacker). The implementation details of the utilized acoustic communication system are described in the following section.

III. COMPOSITION AND ADAPTION OF THE UNDERLYING COMMUNICATION SYSTEM

A. A Network Stack for Acoustic Communication

Acoustical communication is seldom seen in terrestrial networks since radio offers much higher bit rates and communication ranges. However, acoustical communication is the method of choice in underwater networks, because electromagnetic waves are highly absorbed by sea water. We are, therefore, able to implement a terrestrial acoustical network on top of preliminary studies about robust underwater acoustical communication.

The utilized network stack is an adaption of an emulation system for underwater acoustical networks [9] from the Research Department for Underwater Acoustics and Marine Geophysics (FWG) of the Bundeswehr Technical Center WTD 71 in Kiel, Germany. It is split into four layers of connected applications (not to be confused with the TCP/IP stack):

- 1) Application Layer (APP)
- 2) Network Layer (NET)
- 3) Error Correction Layer (EC)
- 4) Physical Link Layer (PHY)

All layered applications of the network stack are independent and connected via internal TCP connections for IPC (inter-process communication), as shown in Fig. 3. This structure allows to replace any of the modules, e.g., the application layer module, without the need to touch one of the other layers. The error correction layer is optional and can be left out on devices with limited processing power or memory.

The left column in Fig. 3 lists the layers of the TCP/IP stack, while the right column shows the associated layers/applications within the acoustic emulation system. In the following subsections all three layers (application, network and physical link) are described in detail.

B. Application Layer

The application layer uses the frame format GUWAL (Generic Underwater Application Language) from FWG/FKIE [10]. GUWAL is an operational application language for tactical messaging in underwater networks with low bandwidth. It is based on 16 byte data frames, which include 2 bytes for a header in the beginning and 2 bytes for a CRC checksum at the end. The structure of the GUWMANET/GUWAL protocol is depicted in Fig. 4 (GUWMANET is described in the following section).

The GUWAL header contains a type field, a priority and an acknowledgement flag, two short operational addresses with 6 bit each, a source and a destination address. By only consuming 6 bit, source and destination addresses are designed smaller than regular 32 bit IPv4 addresses in order to save bandwidth. Nevertheless, the address range is sufficient for small acoustic mesh networks. Additionally, each GUWAL address can also be used as multicast address, if necessary.

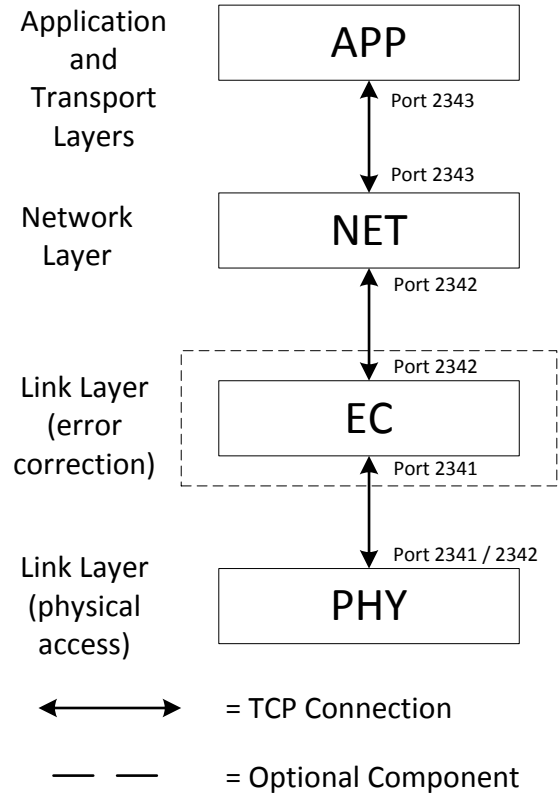


Figure 3. Communication system architecture for covert acoustical mesh networks.

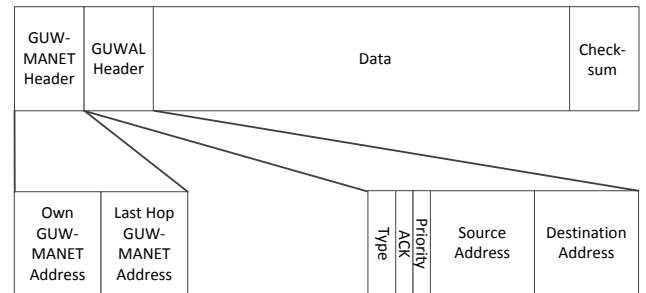


Figure 4. The GUWMANET/GUWAL protocol.

Some potential applications for covert acoustical mesh networks based on GUWAL are described in detail in Section V. For our experiments we implemented an example application that parses UTF-8 characters from STDIN. After the GUWAL header one additional byte is used at the application layer to specify the type of payload. Each data frame can include a payload of up to eleven characters. Frames are sent via a TCP connection to the network layer where they are further processed for output.

C. Network Layer

In the network layer we use the ad-hoc routing protocol GUEMANET (Gossiping in Underwater Mobile Ad-hoc NETWORKS) from FKIE/FWG [11] as the counterpart to GUAL. GUEMANET reuses the two 6 bit operational source and destination addresses of GUAL for network routing. Due to the fact that these addresses can be used as multicast addresses they do not have to be unique in the network. GUEMANET introduces a new 5 bit network address, which should be unique in the local 2 hop neighborhood. This allows to distinguish between all neighbors, which is important to build up routes. In the current version of GUEMANET these network addresses must be set statically. It is intended by FWG/FKIE to implement an automatic network configuration procedure in a future version.

The network header of GUEMANET consists of two address fields, as can be seen in Fig. 4. The first address field contains the network address of the current transmitter. The second field contains the network address of the previous node which forwarded the message before (last hop). While the source and destination addresses of the GUAL header are end-to-end addresses and static during the complete forwarding process, the transmitter and last hop address in the GUEMANET header changes at each hop.

The route establishment operates on the principle of reactive routing protocols. The first message is flooded through the network. Each node repeats the message and sets the last hop field to the network address of the first node it received the message from. A node generates a temporary routing entry if it overhears that it was selected as last hop in a forwarding from one of its neighbors. After the message reaches one of the destination nodes, it replies with an acknowledgement that is routed back with the temporary routing entries to the source. On the way back, the route is persisted for all further packets. If a route breaks down, the route establishment is initiated again. Further details on the routing protocol are provided by Goetz and Nissen [11].

D. Link Layer

1) *General Considerations:* At the link layer, we provide the PHY application, which includes the modem of the Adaptive Communication System (ACS), originally developed by FWG and based on GNU Radio [12], an open-source development toolkit for signal processing.

2) *Error Correction Layer:* Both modems can benefit from the optional EC (error correction) layer from FWG/FKIE. The EC layer uses the 16 bit checksum included at the end of each packet to restore packets with one or two bit errors. If a node receives multiple error-prone versions of the message, the EC layer tries to merge them into a correct one. More details on the error correction will be included in a future article.

3) *Physical access using ACS modem:* The ACS modem uses JANUS [13], a robust signaling method for underwater communications. It is based on FHSS (frequency-hopping spread spectrum) with 48 carriers (20 carriers are used for pure ultrasonic communication in this article). Fig. 5 shows

the spectrogram that reveals the frequency hopping within a transmission of the ACS modem.

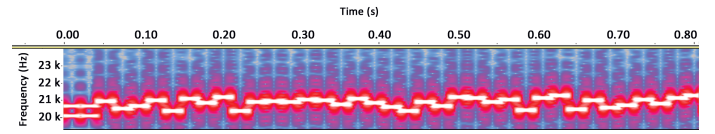


Figure 5. Signal spectrogram—The results of the utilized frequency hopping algorithm are made visible.

The center frequency was modified from 4,200 Hz to 21,000 Hz (18,600 Hz with the originally presented waveform [1]), completely placing the signal into the ultrasonic frequency range. In order to avoid undersampling, the sampling frequency was modified from 12,800 Hz to 48,000 Hz and the sample size for each bit value was modified from 256 to 1024. It could be observed that the ACS modem (running on top of a Lenovo T400) was not capable of calculating the IFFT/FFT of size 1024 fast enough in software. Therefore, for any bit position j within a output frame with the associated output frequency $\lambda(j)$ within the frequency hopping sequence λ and for each sample position k , the sample s_k is calculated by Eq. (1).

$$s_k = \frac{1}{1024} \cdot e^{2\pi i \cdot \frac{\lambda(j)k}{1024}} \cdot w_k \quad (1)$$

To obtain a better differentiation between each bit, we multiply the sample with the window function w_k , which implements a trapezoid window and the square root of a Hamming window, which is defined by Eq. (2) for each index i .

$$\sqrt{0.54 + 0.46 \cdot \cos\left(\frac{2\pi(i - 1024/2)}{1024}\right)} \quad (2)$$

The resulting waveform is depicted in Fig. 6.

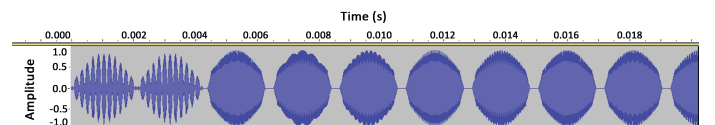


Figure 6. Signal waveform—The preamble and 6 transmitted bits are made visible.

In the first 42 ms a preamble for packet detection and synchronization is sent. We added an adapted bandpass filter at the receiver of the original ACS modem in order to filter for the transmitted frequency range (Fig. 7).

The bandpass FIR (finite impulse response) filter is a Blackman-Harris window with a frequency range between 20,400 and 23,000 Hz.

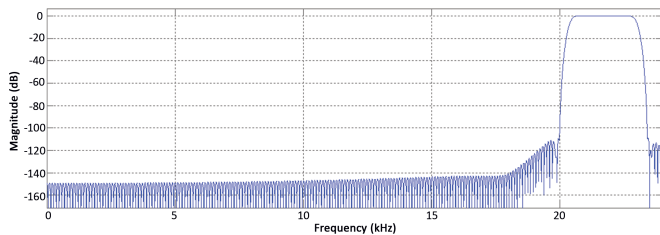


Figure 7. The bandpass FIR filter designed (Blackman-Harris window).

IV. EXPERIMENTS AND MEASUREMENTS

A. Experiment Setup

For the experimental setup we are using five laptops (model: Lenovo T400) as the mesh network participants. As operating system for each node, we installed Debian 7.1 (Wheezy) on each laptop. All experiments were performed at FKIE, building 3, and without any acoustical preparations made.

B. Output Frequency Measurement

The Lenovo T400 features an Intel Corporation 828011 (ICH9 Family) HD Audio Controller. The frequency range that the audio processor is able to output and input is determined by directly connecting the line in and line out jacks and recording an increasing signal from 0 to 35,000 Hz. The resulting graph is depicted in Fig. 8 where the output sound pressure levels are correlated with the output signal frequency. The recorded sound pressure levels are relative values that describe the attenuation of the original audio signal.

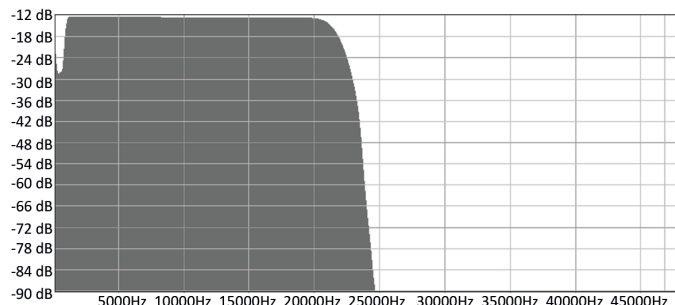


Figure 8. The recorded output/input frequency response of a Lenovo T400 Laptop (Hanning window, FFT, size 512).

From the recorded frequency range it can be discovered that we are able to process frequencies in the low ultrasonic range around 20,000 Hz. Previously performed tests with a Lenovo T410 Laptop featuring the Conexant 20585 audio codec (with 192 kHz DAC / 96 kHz ADC) [14] have shown very similar results. The results lead us to the conclusion that ultrasonic or near-ultrasonic communication with computing systems of the Lenovo T400 series is possible.

Fig. 9 shows the output frequency spectrum of a pure ultrasonic transmission with the ACS modem.

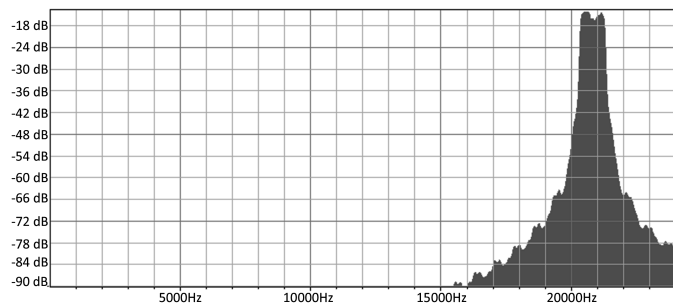


Figure 9. The recorded frequency range of an ACS modem transmission (Hanning window, FFT, size 1024).

The original preamble of the ACS modem has been replaced by a 0.042 s long preamble (see also waveform, Fig. 6), which is also placed in the ultrasonic frequency range. The transmission was found to be inaudible to the observers of the experiment at the configured volume levels.

C. Range Experiment

In another experiment we interconnect two laptops in order to gain an understanding of the achievable range in a covert acoustical mesh network. Acoustic waves are attenuated during transmissions in air (or water) due to scattering, absorption and reflection, leading to lower sound pressure levels in the received signal. The degree of attenuation is specifically depending on the distance between transmitter and receiver and on the frequencies used in data transmission. Therefore, we should be able to communicate over longer distances with sub-ultrasonic frequencies in comparison to ultrasonic frequencies. Fig. 10 shows the setup of the range experiment.

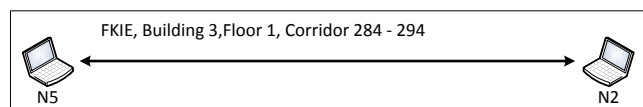


Figure 10. Schematic view of the range experiment.

As depicted, the range experiment is performed in an approximately 25 m long corridor at FKIE. Both nodes are placed in direct line of sight to each other with the displays directed at each other so that output of the internal speakers (built-in alongside the keyboard) is loosely directed at the corresponding node. The achievable range is determined by repeatedly transmitting messages and gradually increasing the distance between the nodes with each successful transmission.

1) *Measurements with ACS Modem:* With the ACS modem, transmissions over a distance of up to 8.2 m could be observed with a transmission rate of approximately 20 bit/s. Moreover, distances of 19.7 m have been overcome with the originally presented waveform that is placed slightly below the ultrasonic frequency range [1]. As it was already presented in Fig. 8, the frequency response of the utilized audio hardware gradually decreases for frequencies $\geq 20,000$ Hz, and this is

one of the reasons that the near-ultrasonic frequency range below 20,000 Hz (which might be considered less stealthy in comparison to the ultrasonic frequency range $\geq 20,000$ Hz) is more stable in covert acoustical transmissions. No bit errors could be observed in transmissions over these ranges. While much higher transmission rates are generally possible with acoustical communication (using a less narrow band), tests with a configured transmission rate of 20 bit/s have been the most successful in transmissions over distances ≥ 5 m. Thus, we conclude that the achievable transmission rate is highly dependent on the desired transmission distance.

D. Interconnection Experiment with ACS Modem

Fig. 11 shows the experiment setup for a covert network of 5 participants with the adapted ACS modem.

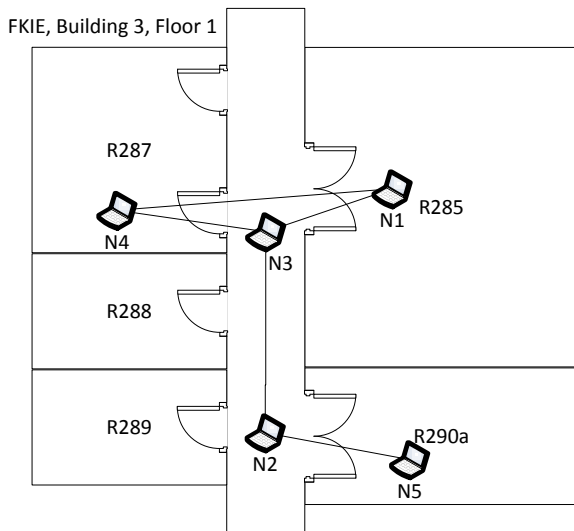


Figure 11. Schematic view of the interconnection experiment.

The nodes are placed in a mixed office/lab environment at FKIE. Every node is placed in direct line of sight to another node in order to establish connectivity. While indirect communication over reflections from walls and doors has also been tested, connectivity could only be established over direct communications. Tab. I shows the distances between connected nodes in the performed experiment.

As one GUWMANET/GUWAL packet is transmitted over a duration of approximately 6 s, a three-hop transmission (using 4 computers) would have a latency of 18 s, which could be confirmed in the experiment setup. Every sent packet could be delivered successfully (no bit errors at destination) in the 5-nodes-network, either in the first transmission or in one of the three automatic retransmissions. With the equiripple bandpass filter in place (see Fig. 7), common sources of noise (such as human speech) were not found to have a considerable effect on the transmissions. In another test, the absorption of acoustic

TABLE I.
DISTANCES OVERCOME BETWEEN CONNECTED NODES IN THE INTERCONNECTION EXPERIMENT.

Connected Nodes	Distance
N5 \leftrightarrow N2	3.4 m
N2 \leftrightarrow N3	5.4 m
N3 \leftrightarrow N4	2.8 m
N3 \leftrightarrow N1	3.3 m
N4 \leftrightarrow N1	6.2 m

waves by humans, walking through the experiment setup and, therefore, blocking the line of sight between two nodes, was found to have an adverse effect on connectivity. However, this does not affect the operation of a covert acoustical mesh network, as transmissions without acknowledgments are automatically retransmitted. Now that we have shown the feasibility of implementing covert acoustical mesh networks, we will have a look at potential applications for these covert networks.

V. APPLICATIONS OF COVERT ACOUSTICAL MESH NETWORKS

A. A Multi-Hop Acoustical Keylogger

We use the keylogging software logkeys [15] for our experiment on acoustical multi-hop keylogging. The considered scenario is depicted in Fig. 12.

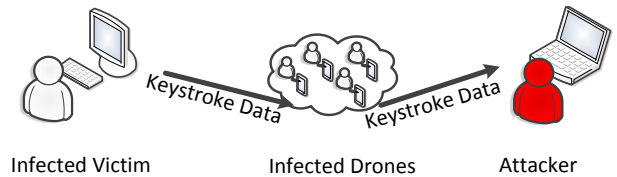


Figure 12. Scenario for a multi-hop acoustical keylogger.

The infected victim sends all recorded keystrokes to the covert acoustical mesh network. Infected drones forward the keystroke information inside the covert network till the attacker is reached who is now able to read the current keyboard input of the infected victim from a distant place. To implement the described scenario, we propose a multi-hop acoustical keylogger (Fig. 13).

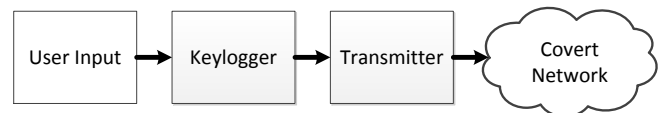


Figure 13. A multi-hop acoustical keylogger that is maliciously placed at the infected victim.

User input is recorded by the keylogger *logkeys* that must be in place and started with root privileges at the infected victim. The keylogger is configured to write any keystrokes to a named pipe that is read out by the acoustic transmitter. The transmitter becomes active when a line feed symbol is reached and sends the keystroke information out to the covert network. The multi-hop acoustical keylogger has been successfully tested in the previously defined experimental setup.

B. Connecting to and Tunneling over the Internet

In a more advanced experimental setup, we connect the attacker to an SMTP server that is connected to the Internet, collecting frames or lines of keystrokes and sending them out as email. For this purpose, we propose an SMTP/TCP/IP proxy (Fig. 14) that encapsulates the data from these frames into the TCP/IP world.

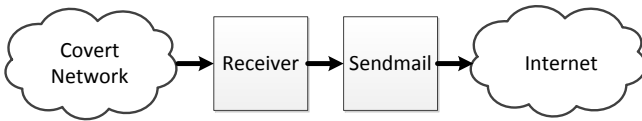


Figure 14. An SMTP/TCP/IP proxy that is used by the attacker to forward the keystroke information into the Internet.

The acoustical receiver gathers input from the covert network and forwards it to a named pipe. From there, a local mail server (i.e., *sendmail*, but a remote mail server could also be used) is called upon to send an SMTP message out to an arbitrary email address. This message could just contain the recorded keystrokes, but it is also possible to include the GUWMANET/GUWAL headers in order to tunnel the protocol over HTTP/TCP/IP (circumventing corporate firewalls), and to extend the covert acoustical mesh network to another covert network at any place in the world. The attacker has to be (directly or indirectly) connected to the Internet in order to perform this procedure, but messages could also be gathered and sent out as soon as Internet access is established. Sending out gathered keystrokes to a local SMTP server and forwarding them to a remote email server has been successfully performed in the previously defined experimental setup.

C. Other Applications for Covert Networks

Alongside the presented proof-of-concept, even more applications of covert networks are conceivable. Besides keystroke information it would also be possible to forward other security critical data such as private encryption keys or small-sized text files with classified information from the infected victim to the covert network. This data could be sent out periodically to maximize the likelihood of data extraction from the host and it could also be spread to different environments when the computing system is carried around.

Finally, an infected drone might not only serve as a router in the covert network, but it might also offer *covert services*

to the network. A covert service is a network service that could, for instance, provide access to further networks (e.g., the Internet, if the attacker is not connected to the Internet itself).

Against malicious participation in a covert acoustical mesh network, countermeasures might be applied as described in the following section.

VI. COUNTERMEASURES AGAINST INFORMATION LEAKS FROM COVERT NETWORKS

A. Implementing Audio Filtering Options

If audio input and output devices cannot be switched off, implementation of audio filtering options may be an alternative approach to counter maliciously triggered participation in covert networks. In Linux-based operating systems, a software-defined audio filter can be implemented with ALSA (Advanced Linux Sound Architecture) in conjunction with the LADSPA (Linux Audio Developer's Simple Plugin API) as mentioned by Phillips [16].

The specific frequencies used by the presented acoustic modem could be filtered out with a bandpass or lowpass filter. We tested a 4-pole lowpass filter with LADSPA and a configured cutoff frequency of 18,000 Hz in the presented experimental setup that effectively prevented inaudible communications in any ALSA-based application.

A general approach for audio filtering in a component-based operating system might be to implement a trusted audio filtering component, i.e., an audio filtering guard. The audio filtering guard would be connected to the audio input and output devices and to the operating system partitions p_i and p_j that need access to the audio devices. For all audio input and output operations, the audio guard would apply input and output filters. To ensure that audio filtering is always-invoked in audio input and output, the access control policy of the component-based operating system needs to enforce that audio input and output devices can only be accessed via the audio guard. More details on the concept of an audio filtering guard are provided by Hanspach and Keller [17].

A more advanced approach might be to implement an audio intrusion detection system (IDS) that does not only filter along predefined settings but supports methods for detection of modulated audio signals and handling input and output based on the signal characteristics (e.g., as recently presented by us [2]). This audio IDS might also be implemented as an operating system guard, as will be described in the following section.

B. A Host-Based Audio IDS Designed as an Operating System Guard

The proposed architecture of a host-based audio intrusion detection guard (Fig. 15) is similar to the described audio filtering guard, but an IDS state may be included for stateful inspection of audio input and output.

The audio intrusion detection guard would forward audio input and output signals to their destination and simultaneously

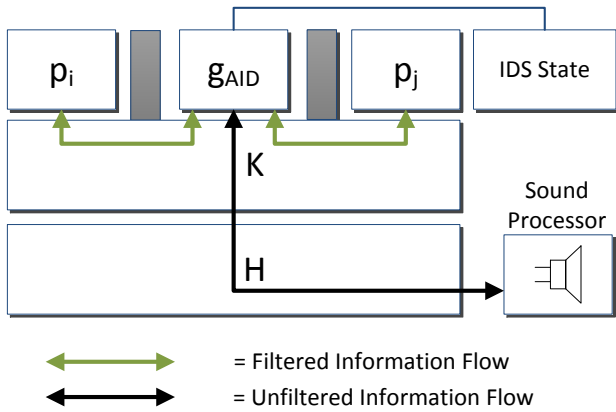


Figure 15. Design concept of an audio intrusion detection guard.

store them inside the guard’s internal state where they are subject to further analyses. Signal analyses might include frequency, amplitude and phase measurement in order to detect digital signal processing modes. A decision routine inside the guard would apply filters to the audio interfaces so that any audio input from and output to operating system partitions is filtered. For instance, it is possible to pitch highly suspicious ultrasonic signals down in order to inform the user of a hidden audio transmission (we have also demonstrated this with a mobile detection device, see also [2]). Further information on the captured signal would be included in the operation system’s log files in order to enable in-depth assessment of a possible attack.

The concept of an intrusion detection guard analyzing device input and output might also be adapted to other types of physical emanations such as modulated optical transmissions that have been described by Frankland [18], Loughry and Umphress [19], and Hasan, Saxena, Tzipora, Shams and Dustin [20]. Considerations on the systematic identification of these covert physical channels will be presented in an upcoming publication [21]. Related studies are further discussed in the following section.

VII. RELATED WORK

As White et al. [22] pointed out, mobile cyber-physical system applications might be present on common sensory devices (e.g., on smartphones). Malware infected devices might be connected to form a botnet (as described by Dagon, Gu and Lee [23]). With smartphones and similar devices, such a botnet might be built upon one of many network interfaces present in such devices and even on input and output sensors as we describe in this article, although we do not specifically experiment with smartphones and focus at interconnecting laptops. A near-field communication botnet approach based on bluetooth is presented by Singh, Jain, Traynor and Lee [24]. We, instead, use audio communications for our botnet-like approach in near-field communications to construct a covert acoustical mesh network consisting of malware infected drones.

Hasan, Saxena, Tzipora, Shams and Dustin [20] present a botnet approach where different types of physical emanations are used for command-and-control in botnets. In contrast to these authors, we are not only looking at command-and-control messages, but we build a whole covert mesh network from acoustical emanations.

Different types of physical emanations have been discussed as means for information leaking. Van Eck [25] describes the risk from electromagnetic radiations, while Loughry and Umphress [19] present information leakage from optical emanations. Shamir and Tromer [26], and LeMay and Tan [27] describe acoustical emanations from the computing system (e.g., supply capacitors from the computer’s mainboard) that might leak critical information. We are also using acoustical emanations for information leakage, but instead of passively monitoring the computing system in order to exploit a side channel, we actively produce acoustical emanations to construct a covert channel between different computing systems and interconnect them into a covert acoustical mesh network.

For eavesdropping on a computer user’s keystrokes by using a computers’ physical emanations, several techniques have been proposed. Halevi and Saxena [28] present a study on acoustical emanations from keyboards where the keyboard input is recovered from the sound of a keystroke. Raguram, White, Goswami, Monroe and Frahm [29] discuss a setup where typed input is recovered from environmental reflections, while Frankland [18] describes information leakage over keyboard LEDs that are manipulated to carry an optical signal. Moreover, Balzarotti, Cova and Vigna [30] show that keystrokes can be automatically captured by recording the keyboard with a camera and analyzing the video stream. We also gather keystrokes from a computer user, but we record the keystrokes directly on the infected victim (i.e., the user’s computer) where a malware prototype needs to be placed in preparation of an attack. This malware prototype contains both a keylogger software and a network stack for covert acoustic communication in order to spread the recorded keystrokes over the covert network.

Using audio as a networking technology has been described by Madhavapeddy, Scott, Tse and Sharp [31], by Yan, Zhou, Shi and Li [32] and by Lopes and Aguiar [33]. Very recently, Nandakumar, Chintalapudi, Padmanabhan and Venkatesan [34] have presented a study on acoustic peer-to-peer near-field-communication (up to 15-20 cm), where eavesdropping by third parties is prevented by the means of signal jamming. Based on the concept of audio networking, we construct a wireless mesh network, not with the ambition to compete with wireless communication standards, but to prove that covert and stealthy communication is possible with audio networking and that critical information can be leaked over a covert acoustical mesh network.

Furthermore, acoustic wave propagation is regularly used in underwater setups as described by Otnes et al. [35] and many more authors. We, however, are using a setup for covert air communications while utilizing parts of an underwater communication system [9].

Preliminary work on ultrasonic communication has been performed by Hosman, Yearly, Antonio and Hobbs [36],

by Tofsted, O'Brien, D'Arcy, Creegan and Elliott [37], by Altman, Antebi, Atsmon, Cohen and Lev [38], and by Li, Hutchins and Green [39]. In contrast to these authors, we show how ultrasonic communication can be placed in the context of information security and how security critical data can be leaked over multiple hops of infected drones.

On behalf of the UCAC Consortium, Leus and van Walree [40] present a setup for stealthy communication with unmanned underwater vehicles where audio messages are transmitted "at a low signal-to-noise ratio (SNR), thereby hiding messages in the ambient noise and reducing the probability of detection by third parties". We also propose a stealthy audio-based communication system, but we primarily aim at providing stealthiness in audio transmissions over the air and from human computer users that might not anticipate information leaks from audio subsystems. With this scenario in mind, we build a mesh network upon audio transmissions that are inaudible to human computer users by utilizing the ultrasonic frequency range.

VIII. CONCLUSION

We have shown that the establishment of covert acoustical mesh networks in air is feasible in setups with commonly available business laptops. By reutilizing an underwater communication system, we take advantage of a network stack that was built with robust acoustical communication in mind. The presented approach to covert acoustical mesh networks allows to transmit messages with a rate of approximately 20 bit/s up to a range of 19.7 m between two connected nodes, but much higher transmission rates would be possible for low-distance transmissions. The complete path of a single frame from the infected victim at the first hop to the attacker at the last hop over two additional infected drones as intermediate hops took 18 s.

Acoustical networking as a covert communication technology is a considerable threat to computer security and might even break the security goals of high-assurance computing systems that did not consider acoustical networking in their security concept.

We did not specifically address the problem of how to infect a computing system with the malware, but this problem exists with any covert channel technology.

For the prevention of participation in a covert acoustical network, it might not always be acceptable to switch off the audio input and output devices, as they might be needed for IP phone communications and other audio applications. For these cases it is possible to prevent inaudible communication of audio input and output devices by application of a software-defined lowpass filter. An audio filtering guard can be used to control any audio-based information flow in a component-based operating system. Furthermore, a host-based audio intrusion detection system can be utilized, as we already presented in a different article. The audio IDS analyzes audio input and output to detect modulated signals or hidden messages in audio playback, and it might also be implemented as an operating system guard.

ACKNOWLEDGMENT

We would like to thank Jörg Keller, Jochen F. Giese, Henning Rogge and Tobias Ginzler for their helpful comments. We also would like to acknowledge Ivor Nissen at the Research Department for Underwater Acoustics and Marine Geophysics (FWG) of the Bundeswehr Technical Center WTD 71, Kiel, Germany, for providing us with access to the source code of an underwater networking software (based on the open JANUS waveform protocol) that we adapted for setting up a covert acoustical mesh network in air.

REFERENCES

- [1] M. Hanspach and M. Goetz, "On Covert Acoustical Mesh Networks in Air," *Journal of Communications*, vol. 8, no. 11, pp. 758–767, Nov. 2013.
- [2] —, "Recent Developments in Covert Acoustical Communications," in *Sicherheit 2014*. Bonn, Germany: GI, Mar. 2014, pp. 243–254.
- [3] B. W. Lampson, "A note on the confinement problem," *Commun. ACM*, vol. 16, no. 10, pp. 613–615, Oct. 1973.
- [4] M. Hanspach and J. Keller, "A Taxonomy for Attack Patterns on Information Flows in Component-Based Operating Systems," in *Proceedings of the 7th Layered Assurance Workshop*, New Orleans, LA, USA, Dec. 2013.
- [5] IEEE Standards Association, "IEEE 802.3 Ethernet," 1985. [Online]. Available: <http://standards.ieee.org/about/get/802/802.3.html> [Accessed: 2013-06-15]
- [6] IEEE Standards Association, "IEEE 802.11 Wireless LAN," 1997. [Online]. Available: <http://standards.ieee.org/about/get/802/802.11.html> [Accessed: 2013-06-15]
- [7] T. Jaeger, J. Liedtke, V. Panteleenko, Y. Park, and N. Islam, "Security architecture for component-based operating systems," in *Proc. 8th ACM SIGOPS European workshop on Support for composing distributed applications*, ser. EW 8. New York, NY, USA: ACM, 1998, pp. 222–228.
- [8] J. P. Anderson, "Computer Security Technology Planning Study," *Tech. Rep. ESD-TR-73-51, Volume II. Electronic Systems Division, AFSC*, Oct. 1972.
- [9] R. Odugoudar and I. Nissen, "Ad-hoc Network Emulation Framework for Underwater Communication Applications," in *Proc. 5th ACM International Conference on Underwater Networks & Systems*. Woods Hole, MA: ACM, Sept. 2010.
- [10] I. Nissen and M. Goetz, "Generic UnderWater Application Language (GUWAL) - Specification of Tactical Instant Messaging in Underwater Networks," Research Department for Underwater Acoustics and Marine Geophysics, Kiel, Germany, Tech. Rep. WTD71 - 0070/2012 FR, Dec. 2012.
- [11] M. Goetz and I. Nissen, "GUWMANET - Multicast Routing in Underwater Acoustic Networks," in *Military Communications and Information Systems Conference*, ser. MCC '12. IEEE, Oct. 2012, pp. 1–8.
- [12] E. Blossom, "GNU radio: tools for exploring the radio frequency spectrum," *Linux J.*, vol. 2004, no. 122, pp. 4–, June 2004.
- [13] G. Z. K McCoy, B Tomasi, "JANUS: the genesis, propagation and use of an underwater standard," in *European Conference on Underwater Acoustics*, July 2010.
- [14] Conexant Systems, Inc., "HD-Audio CODEC with 4DAC, 6ADC, Mono Class D, 2 Capless Headphone Driver, SPDIF Out/In." [Online]. Available: <http://www.conexant.com/Product/Audio/pchdaudio/CX2058X> [Accessed: 2013-06-15]
- [15] Google Code, "logkeys Linux keylogger," Jan. 2010. [Online]. Available: <http://code.google.com/p/logkeys/> [Accessed: 2013-06-15]

- [16] D. Phillips, "Linux Audio Plug-Ins: A Look Into LADSPA," Feb. 2001. [Online]. Available: <http://www.linuxdevcenter.com/pub/a/linux/2001/02/02/ladspa.html> [Accessed: 2013-06-15]
- [17] M. Hanspach and J. Keller, "In Guards we trust: Security and Privacy in Operating Systems revisited," in *Proc. 5th ASE/IEEE International Conference on Information Privacy, Security, Risk and Trust*, ser. PASSAT '13. Washington D.C., USA: IEEE, Sept. 2013.
- [18] R. Frankland, "Side Channels, Compromising Emanations and Surveillance: Current and future technologies," Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, Tech. Rep. RHUL-MA-2011-07, Mar. 2011.
- [19] J. Loughry and D. A. Umphress, "Information leakage from optical emanations," *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 3, pp. 262–289, Aug. 2002.
- [20] R. Hasan, N. Saxena, T. Haleviz, S. Zawoad, and D. Rinehart, "Sensing-enabled channels for hard-to-detect command and control of mobile devices," in *Proc. 8th ACM SIGSAC symposium on Information, computer and communications security*, ser. ASIA CCS '13. New York, NY, USA: ACM, 2013, pp. 469–480.
- [21] M. Hanspach and J. Keller, "The Upcoming Threat of Covert Communication Technologies," in *Future Security 2014 (to appear)*, Berlin, Germany, Sept. 2014.
- [22] J. White, S. Clarke, C. Groba, B. Dougherty, C. Thompson, and D. Schmidt, "R&D challenges and solutions for mobile cyber-physical applications and supporting Internet services," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 45–56, 2010.
- [23] D. Dagon, G. Gu, and C. Lee, "A Taxonomy of Botnet Structures," in *Botnet Detection*, ser. Advances in Information Security, W. Lee, C. Wang, and D. Dagon, Eds. Springer US, 2008, vol. 36, pp. 143–164.
- [24] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating Bluetooth as a Medium for Botnet Command and Control," in *Proc. 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA '10*. Springer, July 2010, pp. 61–80.
- [25] W. van Eck, "Electromagnetic radiation from video display units: an eavesdropping risk?" *Comput. Secur.*, vol. 4, no. 4, pp. 269–286, Dec. 1985.
- [26] A. Shamir and E. Tromer, "Acoustic cryptanalysis: On nosy people and noisy machines. The Blavatnik School of Computer Science, Tel Aviv University." [Online]. Available: <http://cs.tau.ac.il/~tromer/acoustic/> [Accessed: 2013-06-15]
- [27] M. D. LeMay and J. Tan, "Acoustic Surveillance of Physically Unmodified PCs," in *Proc. 2006 International Conference on Security & Management*, June 2006.
- [28] T. Halevi and N. Saxena, "A closer look at keyboard acoustic emanations: random passwords, typing styles and decoding techniques," in *Proc. 7th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 89–90.
- [29] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, "iSpy: automatic reconstruction of typed input from compromising reflections," in *Proc. 18th ACM conference on Computer and communications security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 527–536.
- [30] D. Balzarotti, M. Cova, and G. Vigna, "ClearShot: Eavesdropping on Keyboard Input from Video," *IEEE Symposium on Security and Privacy*, vol. 0, pp. 170–183, 2008.
- [31] A. Madhavapeddy, D. Scott, A. Tse, and R. Sharp, "Audio Networking: The Forgotten Wireless Technology," *IEEE Pervasive Computing*, vol. 4, no. 3, pp. 55–60, July 2005.
- [32] H. Yan, S. Zhou, Z. J. Shi, and B. Li, "A DSP implementation of OFDM acoustic modem," in *Proc. second workshop on Underwater networks*, ser. WuWNet '07. New York, NY, USA: ACM, 2007, pp. 89–92.
- [33] C. V. Lopes and P. M. Aguiar, "Acoustic Modems for Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 62–71, 2003.
- [34] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: secure peer-to-peer acoustic NFC," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, ser. SIGCOMM '13. New York, NY, USA: ACM, Aug. 2013, pp. 63–74.
- [35] R. Otne, A. Asterjadhi, P. Casari, M. Goetz, T. Husøy, I. Nissen, K. Rimstad, P. van Walree, and M. Zorzi, *Underwater Acoustic Networking Techniques*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer, 2012.
- [36] T. Hosman, M. Yearly, J. K. Antonio, and B. Hobbs, "Multi-tone FSK for ultrasonic communication," in *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, 2010, pp. 1424–1429.
- [37] D. Tofsted, S. O'Brien, S. D'Arcy, E. Creegan, and S. Elliott, "An Examination of the Feasibility of Ultrasonic Communications Links," Army Research Laboratory, White Sands Missile Range, NM, USA, Tech. Rep. ARL-TR-5200, June 2010. [Online]. Available: <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA522291> [Accessed: 2013-07-29]
- [38] N. Altman, A. Antebi, A. Atsmon, M. Cohen, and Z. Lev, "Computer communications using acoustic signals," U.S. Patent 7,480,692 B2, Jan. 2009.
- [39] C. Li, D. Hutchins, and R. Green, "Short-range ultrasonic digital communications in air," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 55, no. 4, pp. 908–918, 2008.
- [40] G. Leus and P. van Walree, "Multiband OFDM for Covert Acoustic Communications," *IEEE J.Sel. A. Commun.*, vol. 26, no. 9, pp. 1662–1673, Dec. 2008.