A New Data-Hiding Approach for IP Telephony Applications with Silence Suppression

Sabine S. Schmidt
FernUniversität in Hagen
Hagen, Germany
sabine.schmidt1@studium.fernuni-hagen.de

Wojciech Mazurczyk
Warsaw University of Technology
Warsaw, Poland
wmazurczyk@tele.pw.edu.pl

Jörg Keller
FernUniversität in Hagen
Hagen, Germany
joerg.keller@fernuni-hagen.de

Luca Caviglione
National Research Council of Italy
Genoa, Italy
luca.caviglione@ge.issia.cnr.it

ABSTRACT

Even if information hiding can be used for licit purposes, it is increasingly exploited by malware to exfiltrate data or to coordinate attacks in a stealthy manner. Therefore, investigating new methods for creating covert channels is fundamental to completely assess the security of the Internet. Since the popularity of the carrier plays a major role, this paper proposes to hide data within VoIP traffic. Specifically, we exploit Voice Activity Detection (VAD), which suspends the transmission during speech pauses to reduce bandwidth requirements. To create the covert channel, our method transforms a VAD-activated VoIP stream into a non-VAD one. Then, hidden information is injected into fake RTP packets generated during silence intervals. Results indicate that steganographically modified VAD-activated VoIP streams offer a good trade-off between stealthiness and steganographic bandwidth.

CCS CONCEPTS

•**Security and privacy** → *Security protocols;*

KEYWORDS

Information hiding, VoIP, Voice Activity Detection, covert channel, steganography.

ACM Reference format:
Sabine S. Schmidt, Wojciech Mazurczyk, Jörg Keller, and Luca Caviglione. 2017. A New Data-Hiding Approach for IP Telephony Applications with Silence Suppression. In *Proceedings of ARES '17, Reggio Calabria, Italy, August 29-September 01, 2017,* 6 pages.
DOI: 10.1145/3098954.3106066

1 INTRODUCTION

Steganography allows to hide secret information within an innocent looking carrier. For instance, it can be used to conceal data into image, video or audio files. The most recent trend is network steganography, which enables to transmit secret data over the network. To this aim, possible carriers are: unused fields available in the protocol headers, some statistics of the traffic such as the inter-packet time or the throughput, as well as the volume produced by a specific set of applications or protocols [1]. To not raise suspicions, the used carrier should not represent an anomaly itself, and its alteration should not reveal the presence of secrets. Such requirements limit the capacity of the hidden channel, which can be traded for undetectability or robustness according to the "magic triangle" rule [2]. For the case of network steganography, the presence of a covert channel could be revealed by anomalies such as incoherent protocol behaviors, inflated data volumes or uncommon timing statistics [1, 2]. Even if information hiding can be used for licit purposes, e.g., to prevent censorship attempts, nowadays it is primarily used by malware for remaining unnoticed while conducting large-scale attacks or exfiltrating stolen data [3, 4]. Therefore, investigating techniques to set up covert channels is mandatory to completely assess both the cybersecurity degree of the Internet and modern devices.

As regards the usage of steganographic techniques to create network covert channels, [1] discusses how the process mutated during the years, e.g., from secrets injected into unused fields of the IP header, to sophisticated cloud-based mechanisms. The work [2] surveys techniques targeting modern smartphones, especially those taking advantage of short-range connectivity, GPS-generated metadata and accelerometers. In the perspective of creating a network covert channel, VoIP is an excellent candidate as it is ubiquitously available, produces a huge amount of traffic and offers a rich set of features suitable for steganographic purposes [2, 5]. To this aim, the literature proposes different mechanisms to implement a network covert channel using IP telephony applications. However, for the sake of compactness, we limit the discussion to those manipulating the multimedia portion of the stream, which is the focus of this paper. Nevertheless, interested readers could refer to [6] surveying data hiding for VoIP-based applications in a more general and comprehensive manner. In more details, [7] proposes to inject secrets by using timestamps used by the Real-time Transport Protocol (RTP) for synchronizing the multimedia streams. A similar idea is also used in [8] and [9], where authors esploit different fields of the Real-time Control Protocol (RTCP) as carriers. In [10] authors further compress the voice of a conversation between two unaware parties to make room for secrets, while in [11] authors introduce a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ARES '17, Reggio Calabria, Italy
© 2017 ACM. 978-1-4503-5257-4/17/08…$15.00
DOI: 10.1145/3098954.3106066

steganographic method using intentionally delayed voice packets to form a covert channel. The investigation reported in [12] shows how to create a bidirectional covert channel by embedding secrets in the least significant bit of audio samples produced with the G.711 codec. The work [13] targets Skype, which is one of the most popular applications for VoIP communications. Since it does not use techniques for limiting the bandwidth consumption during silence periods, authors showcase how to embed secrets by replacing the payload of packets containing silence.

Thus, differently from previous works, we instead exploit the Voice Activity Detection (VAD) features allowing to suspend the transmission of packets during speech pauses in order to save bandwidth. Our method (named stegVAD in the following) transforms a VAD-activated VoIP stream into a non-VAD one, and injects the hidden information into fake RTP packets generated during silence intervals. Compared to similar techniques, our method allows to have an improved channel capacity with a negligible impact on the quality of the conversation. Therefore, the main contributions of the paper are: the development of a novel covert channel exploiting VoIP, the evaluation of a real-world implementation as to measure its performance, and the identification of features for the development of proper countermeasures.

The rest of the paper is structured as follows: Section 2 introduces VoIP techniques and the reference implementations used in this work. Section 3 describes the stegVAD method, while Section 4 showcases the performance of the covert channel by means of a comprehensive measurement campaign. Finally, Section 5 concludes the paper.

2 VOIP: AN OVERVIEW

In this section, we briefly introduce some general concepts of Internet Telephony and VoIP applications, but limiting to aspects useful to create a covert channel. We also analyze the most popular VoIP User Agents (UAs) as to evaluate their suitability for implementing the information hiding technique used by stegVAD.

With the VoIP hyponym, we group applications for making phone calls through the Internet. The UA converts the voice into digital data, creates and manages the session with remote peer(s) by means of a suitable signaling protocol, and transmits the encoded audio through a proper media transport layer. As regards signaling, the standard approach uses the Session Initiation Protocol (SIP), which provides self-explanatory methods for managing the call, e.g., `REGISTER`, `INVITE`, `RINGING`, `ACK` or `BYE` [14]. Their detailed description is outside the scope of this paper, but we mention that the `INVITE` message also contains Session Description Protocol (SDP) parameters to initialize the media streaming. Even if SIP could also be used as a carrier for embedding secrets [2], in this paper we concentrate on the multimedia session flow transported by the RTP. To this aim, VoIP relies on the RTP allowing to detect losses, reorder packets and play samples at correct time intervals. To have a proper feedback on the quality of conversations, RTP cooperates with the RTCP, which provides statistics and information for synchronizing the streams [15]. Both protocols are transported via UDP, thus a relevant amount of the overall payload could be consumed by the header or the codec information. Increasing the size of the voice sample per packet is not an efficient workaround, as it worsens the effects of packetization delays and packet losses. As a consequence, further bandwidth optimizations are needed. One of the most effective and popular is VAD, which allows the sender side to stop the transmission during speech pauses [16]. This can provide major bandwidth saving, as typically VoIP calls are characterized by 35% to 70% of silence periods [17].

In general, the receiver recognizes the beginning of the talkspurt by means of the marker bit `M` in the header of RTP [15]. However, recognizing the voice activity is a challenging task, especially without discarding information that can cause distortion. Hence, when in the presence of a transition from speech to silence, a sort of guard period called "overhang" is considered, i.e., ~200 *ms* of additional transmission to prevent alterations of the voice due to many frequent and short pauses [16]. Unfortunately, complete silence can confuse the talker by creating the impression that the conversation has been shutdown. Thus, Comfort Noise Generation (CNG) mixes the silence with an ad-hoc synthetic background noise. Comfort noise can be generated in two modes: locally at the receiver side halting completely the transmission, i.e., Discontinuous Transmission (DTX), or remotely by sending packets containing a Silence Descriptor (SID) [18].

2.1 Analysis of UAs

The previously described VAD and CNG techniques are optional and heavily depend on the UA. Therefore, the RTP stream can look very different although the codec is the same. For the purpose of implementing and testing stegVAD, we performed preliminary traffic analysis as to compare different UAs: Ekiga[1], Linphone[2], Jitsi[3], Pjsua[4], and Twinkle[5].

Ekiga supports VAD and, when enabled, the UA does not send SID packets, but it just stops sending any RTP traffic. Talkspurts can be identified by means of the marker bit `M` and silence patterns are constant and provided for G.711(a), G.711(μ), GSM, Speex/8000, Speex/16000 and iLBC. Unfortunately, the throughput observed during our tests exhibited a very high variability, since RTP packets were always transmitted in bursts interleaved by large pauses. Linphone has no VAD enable/disable option, but it does not seem to use VAD as claimed. Hence, it was not possible to capture any RTP stream continuation (even by using v3.10.0, which is the last available version at the time of writing). A similar behavior characterized Jitsi, which has not any configuration for VAD (de)activation. These UAs have not been considered for testing the proof-of-concept implementation of stegVAD.

Instead, Pjsua uses the VAD engine of Speex by default. Besides, it exhibits constant silence patterns for G.711(a), G.711(μ), GSM, Speex/8000, Speex/16000, Speex/32000 and iLBC. It sends SID packets when using Speex in VAD mode. All packets are marked and generated with the smallest payload possible [19]. Instead, when using iLBC, G.711(a), G.711(μ) and GSM, Pjsua sends SID packets as well, but their size is equal to all other voice packets. Twinkle also uses VAD and has constant silence patterns for G.711(a), G.711(μ), GSM,

[1] https://ekiga.im
[2] http://www.linphone.org
[3] https://jitsi.org
[4] http://www.pjsip.org
[5] http://www.twinklephone.com (and the other fork of the project available at: http://twinkle.dolezel.info)

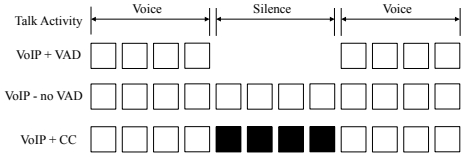

Figure 1: Reference traffic flows for the VoIP with VAD-based covert channel.

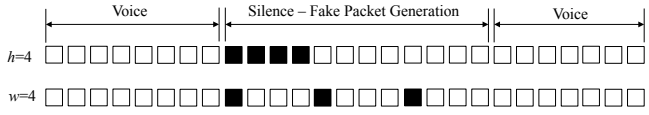

Figure 2: Secret injections in fake packets during silence periods. Black squares denote packets with steganographic data.

Speex/8000, Speex/16000, Speex/32000, iLBC, G726/24, G726/32, and G726/40. If VAD and support for Variable Bit Rate (VBR) are both activated, Twinkle does not switch to Constant Bit Rate (CBR), thus violating the RFC 5574 [19]. Therefore, this configuration has not been taken into account for testing stegVAD.

Summing up, in this work to create a prototype implementation of stegVAD, we only considered Pjsua and Twinkle, while the support of the remaining UAs is left as a possible future development.

3 DESIGN OF THE COVERT CHANNEL

The general communication model considered in this work is composed by a secret sender and a secret receiver wanting to communicate through the Internet in a stealthy manner. To this aim, we establish a covert channel by exploiting a VAD-enabled VoIP connection. To have a carrier where to inject secrets, we generate fake packets during the silence periods as to produce a packet flow resembling a VoIP conversation without VAD. Figure 1 depicts the traffic streams considered in this paper.

In our communication model we also assume the existence of a warden. According to [20], its aim is to reveal or prevent the covert communication: this can be done in several manners, for instance by inspecting or altering the carrier.

Even if the covert channel is bidirectional, for the sake of simplicity we consider a flow of information generated by the UA of the caller towards the UA of the callee. The architecture of stegVAD is implemented through a client interface able to embed and extract data, as well as to revert the hijacked VoIP stream to its original form. This could be made in different points of the network connecting the caller and the callee. In this paper, we consider the caller and the callee as the secret sender and the secret receiver, respectively. Therefore, to create an end-to-end covert channel, client interfaces implementing stegVAD must be co-located within the hosts/devices of the VoIP caller and callee. Other use cases, where the sender and receiver are "innocent" and involved in a normal VoIP call are possible. In this case, stegVAD client interfaces could modify traffic on its way, e.g., to overcome a firewall or a warden. Investigating such scenarios is subject to future research.

Recalling that the covert channel lies within the RTP traffic of VoIP UAs, to avoid detection, stegVAD must not disrupt the flow or alter the PDUs in a too aggressive manner. Then, it should guarantee the following properties:

- *coherence*: stegVAD adds packets to the stream generated by the UA of the caller. This alters several fields of the header of RTP packets, which must be properly restored before being delivered to the UA of the callee. Besides, a warden (e.g., a network analysis tool) could find incorrect details about the amount of exchanged data or the number of dropped packets. Hence, the client interface must adjust RTCP reports accordingly.
- *quality*: stegVAD does not delete/alter the payload of RTP packets carrying vocal samples. Rather, it only injects PDUs during silence periods. However, this could affect the timing of the stream and reducing the quality of the conversation. Hence, possible "time warps" or additional delays should be properly addressed as to not introduce audible artifacts reducing the stealthiness of the covert channel.
- *silence encoding*: to cloak the carrier, stegVAD transforms a VAD-enabled VoIP conversation into a non-VAD one. Thus, RTP packets carrying the secret data (depicted in black in Figure 1) should not lead to a silence period appearing as an anomaly.

3.1 Embedding and Extraction of Secrets

As a prerequisite, the sender and the receiver share a secret, which will be used to compute proper hashes to recognize RTP packets carrying steganographic information. The original data to be transmitted through the covert channel is divided into chunks to fit into PDUs produced by the RTP layer. Since the payload depends on the codec, the size of chunks is evaluated at runtime by inspecting the stream. Modified PDUs are then identified by using a 4-byte long hash of the shared secret, which is placed as a preamble in the payload of RTP packets. This ensures enough space for the content, even if RTP limits the payload to 20 bytes.

To maintain the coherence of the stream, stegVAD uses as a template the last observed legitimate RTP packet produced by the UA before a silence period. In more details, it produces forged packets by reusing the `SSRC` and `PT` fields and by properly incrementing the `sequence number` and the `timestamp`. As regards the network layer, the source and destination IP addresses as well as the source and destination TCP ports are reused, while the `Identification` field has to be properly computed. Since the RFC 4413 does not specify how to assign these values, we used the sequential jump procedure [21], which has been observed in all the hosts used for our tests. To achieve an accurate simulation of a classical non-VAD VoIP connection, stegVAD does not use the entire silence interval to inject steganographic packets. Rather, it provides two mechanisms for the injection of the secret, as depicted in Figure 2. Specifically: *i*) the secret data is injected in the first h PDUs carrying silence, and *ii*) the secret data is injected every w PDUs carrying silence.

Otherwise, stegVAD generates fake packets containing silence in adherence to the used codec.

To extract the steganographic payload at the receiving end, the application probes every incoming RTP packet for the hash. If

present, the secret is extracted and substituted with the proper sequence of silence. The RTP fields, such as the `sequence number`, are adjusted before delivering PDUs to the UA of the callee.

Lastly, to guarantee a proper degree of stealthiness, stegVAD also intercepts RTCP messages, which periodically appear in the VoIP stream according to the UA implementation. In this case, the client interface spoofs RTCP sender reports and adjusts the `packet count`, `octet count` and `current RTP timestamp` by "compensating" the alterations in terms of data volumes and delays produced by the additional non-VAD RTP packets.

3.2 Silence Generation

As said, stegVAD transforms a VAD-enabled VoIP stream into a non-VAD one. Thus, generating fake RTP packets containing silence is critical to guarantee a suitable undetectability of the covert channel. Unfortunately, a one-fits-all solution does not exist since each codec produces silence with given patterns of bits, and has different working modes influencing the size and rates of the PDUs. Therefore, stegVAD has to: *i*) recognize the codec and *ii*) generate silence accordingly. As regards *i*) RTP does not provide any information about the used codec besides the payload type `PT` field, which for many codecs is defined dynamically and also differs among the UAs. Thus, along the lines of [22] stegVAD tries to guess the codec by evaluating the size of the payload and differences in the values of the `timestamp` field. However, this method produced collisions during our tests. For instance, Speex/16000 in mode seven has a payload size of 60 bytes and a timestamp difference of 320, which is also used by G722.1. As a workaround, StegVAD also considers known `PT` numbers. We point out that more sophisticated fingerprinting could be made, but at the price of using a greater amount of computing resources. This could make the detection of a running instance of stegVAD easier, hence reducing the effectiveness its integration within a malware for implementing a stealthy communication service [23]. Besides, part of our future works aims at enriching stegVAD by a minimal SIP parser in order to extract additional information contained in the SDP message (where details about used codec are present). Concerning *ii*) stegVAD does not "run" any codec-dependent functionalities. Rather, it uses a catalog of recorded silence patterns for Pjsua and Twinkle. The current proof-of-concept implementation supports only constant silence encoding, e.g., it supports G722.1 but not AMR-WB. Supporting a wider population of UAs/configurations is part of future development.

4 PERFORMANCE EVALUATION

In this section, we present the performance evaluation of stegVAD when used in a realistic network scenario. Since the client interface implementing stegVAD has to capture, spoof and inject time-sensitive traffic flows, we developed a software prototype written in C and optimized to reduce context switches between kernel and user space. Our proof-of-concept implementation relies upon several third-party libraries: `iptables`-related functions taken from I)ruid [12], `libfindrtp` to identify RTP endpoints and `lookup3` for producing hashes for performing lookups. To conduct tests, we used Kali Linux (SMP Debian 4.6.4-1kali1 kernel) hosts running VoIP UAs Pjsua and Twinkle, as discussed in Section 2.1. Trials have been conducted both on a controlled LAN set-up and through the Internet as to evaluate the impact of uncontrollable errors and delays. To have a proper statistical relevance, each test has been repeated 10 times. To perform a sort of sensitivity analysis on stegVAD, we varied its "aggressiveness" in order to establish how quickly PDUs with secrets are generated when the UA stops sending traffic owing to VAD. To this aim, we introduced a threshold value t indicating the silence interval before starting the injection of secrets into the fake RTP packets. We underline that this should not be confused with thresholds used by the speech activity detection algorithm implemented within the UA to discriminate between voice and silence.

Table 1: Experimental PESQ MOS values.

StegVAD setting	t = 30 ms h = 3	t = 40 ms w = 4	none
Mean PESQ MOS value	3.894	3.859	3.895
Standard deviation	0.226	0.270	0.245

As it is usually done in the literature (see, e.g., [1, 2]), in the following, we characterize stegVAD in terms of steganographic bandwidth, undetectability, and robustness. As a prerequisite, we assess its steganographic cost in terms of degradation of the sound quality due to the manipulation of the stream.

4.1 Sound Quality

As discussed, preserving the sound quality is important to not reveal the presence of the covert channel. This is especially true for the case of VoIP applications, which deal with human-to-human communication. In addition, the availability of a rich literature on multimedia analysis for detecting artifacts (see, e.g., [1, 2] and references therein) requires to completely preserve the overall quality of the conversation. Therefore, to assess the impact of the injected hidden data within the VoIP conversation, we performed tests in a controlled LAN environment.

To model the voice of the talker, we used recordings from the TIMIT continuous speech corpus [24] containing both male and female talkers speaking in English. To assess the quality of the conversation, we compared the original voice with the one received. The resulting quality has been evaluated by using the Perceptual Evaluation of Speech Quality (PESQ) - Mean Opinion Score (MOS) P.862 standard. We considered three different scenarios:

- Sc. 1: VoIP and stegVAD with $t = 30$ ms and $h = 3$;
- Sc. 2: VoIP and stegVAD with $t = 40$ ms and $w = 4$;
- Sc. 3: VoIP without steganography.

In all cases, the UAs used the G.711(μ) codec. Table 1 reports the collected results. Unsurprisingly, Scenario 3 had the highest mean value of PESQ MOS. Yet, similar results have been obtained in other scenarios. This is due to the fact that stegVAD does not alter packets containing the voice. Instead, it only produces negligible alterations to the stream due to additional jitter caused by the client interface "snooping" the VoIP traffic. Moreover, differently to [10] - [13], a higher steganographic bandwidth does not affect the voice quality as stegVAD will just highjack more PDUs carrying silence.

Table 2: Steganographic bandwidths with different injection techniques during the silent periods and different codecs.

Codec	Steg. payload size [bytes]	Bandwidth with h=3 [kbit/s]	h=4 [kbit/s]	w=4 [kbit/s]	w=3 [kbit/s]
G.711(μ)	156	0.57	0.76	5.32	7.09
GSM	29	0.11	0.14	0.99	1.31
Speex/8000	34	0.12	0.17	1.16	1.55
Speex/16000	66	0.24	0.32	2.26	3.00
Speex/32000	70	0.26	0.34	2.39	3.18

4.2 Steganographic Bandwidth and Robustness

To have a realistic reference scenario, we performed measurements by injecting secrets during pauses of a vocal conversation generated starting from [24]. Besides, to stress the covert channel, we also considered a model of the talker making a phone conversation with sentences interleaved with periods of silence with a 50-50 ratio [17].

Concerning the network environment, the client interfaces were used to create a covert channel over the Internet, as to consider the uncontrollable impairments of real-world cases. Table 2 depicts the collected results averaged over the entire dataset of repeated trials to have a proper statistical relevance.

Results confirm that the steganographic bandwidth (or channel capacity) for stegVAD highly depends on the selected codec, as this impacts the payload available for injecting information. Fortunately, as discussed in Section 4.1 this does not affect the sound quality, thus it does not account for additional fragilities in terms of detectability. On the contrary, the number of RTP PDUs used during silence periods allows to increase the capacity, but at the price of an alteration of the related network fingerprint. This, as it will be detailed later on, could account for a reduced undetectability. Therefore, when deploying stegVAD a proper trade-off has to be searched for. Concerning the usage of stegVAD for implementing a steganographic communication layer for a malware architecture, it offers performances suitable for long-lasting data exfiltration or to be incorporated within Advanced Persistent Threats (APTs) architectures (see, e.g., [3, 4] and references therein).

We point out that, compared to similar mechanisms using silence for data-hiding purposes in IP telephony, stegVAD achieves a higher steganographic bandwidth with a reduced impact on the voice quality. For instance, when the method of [13] uses the 100% of packets carrying silence, the resulting bandwidth is 2.78 kbit/s at the price of a 1.54 drop in voice quality. Besides, when 30% of the silence packets are used, the steganographic bandwidth decreases to 1.83 kbit/s at the cost of a 0.14 reduction in the quality. Instead, when using stegVAD with the 25% of packets with silence, we obtained a hidden channel with a capacity of ~5 kbit/s, but only paying 0.036 in terms of a decrease in the voice quality.

With respect to robustness, when UDP is used to transport data generated by the RTP, stegVAD inherits such additional fragilities. In this case, to counteract to packet losses and errors, an additional layer implementing some form of error detection and recovery is needed (see, e.g., [11]). This is a part of our future developments.

Table 3: Main traffic statistics characterizing the RTP flow.

Traffic statistics	call with StegVAD	call without StegVAD
Max. delta (ms)	93.15	107.74
Std. deviation (σ)	14.42	114.61
Max. jitter (ms)	8.75	8.95
Std. deviation (σ)	0.89	3.26
Mean jitter (ms)	2.64	2.3
Std. deviation (σ)	3.12	0.78
Max. skew (ms)	290.50	165.02
Std. deviation (σ)	43.85	1.18
packet loss	2.2	1.4
Std. deviation (σ)	1.24	1.56
Clock drift (ms)	135.5	8.7
Std. deviation (σ)	25.40	0.64

On the contrary, if the TCP is used, stegVAD could omit such functionalities. In both cases, the stream could be also manipulated by a warden to impair the performance of the covert channel, for instance by using tools like network pumps [2]. Since the hidden communication lies within the VoIP channel, the warden cannot alter the stream too much as it will result in a very poor conversation quality. Thus, the real-time nature of audio communication represents a sort of protection against aggressive network-based countermeasures, such as deep packet inspection tools or wardens limiting the performance of the covert channel through traffic normalization approaches [25].

4.3 Undetectability

Transforming a VAD-enabled VoIP conversation into a non-VAD one leads to alterations in the produced RTP stream. This impacts on the produced network traffic, which can be used to detect the presence of the covert channel. Therefore, we compared different traces produced by the clean UAs against those altered by stegVAD. Table 3 shows the average values computed over the entire dataset.

As discussed, stegVAD does not degrade the voice quality, but alters some statistical values of the network traffic that can be used as markers for detecting the covert channel. In more details, the interarrival time (i.e., max. delta as reported in Table 3) is not significantly affected by stegVAD. Similarly, no major alterations can be found in the maximal and mean values for jitter. This is also confirmed by the negligible alteration of the sound quality as shown in Section 4.1. Instead, the skew value, i.e., the deviation from the absolute expected packet arriving time, differs in a relevant manner. To explain this behavior, let us consider the following example. If the codec generates PDUs with a rate r equal to 50 pkt/s and the RTP payload carries 20 ms of voice, then a packet should arrive every 20 ms. If the packet arrives after 19 ms, it has a skew value equal to 1 ms. Such a negative deviation indicates that stegVAD is sending packets too quick. Thus, according to the Table 3, stegVAD has the tendency to shift the packet generation value, i.e., it cannot adjust permanently the rate to match r. By performing additional

investigations, we discovered that the main limit of stegVAD is its inability to handle the presence of too frequent talkspurts interrupting the silence period in a very abrupt manner and causing an increase of the overall skew. This produces a sort of statistical signature that can be exploited to identify the traffic.

As regards the impact of the steganographic bandwidth on the stealthiness of the channel, this varies with the strategy used to inject secrets into PDUs carrying silence. Specifically, when injecting secrets in the first h packets, if the silence period is too short or the bandwidth is too high (i.e., h increases), stegVAD fails to accurately simulate the appearance of a "normal" non-VAD VoIP conversation. Thus, it would preferable to use the approach based on skipping packets, and trading a more bursty behavior for a reduced detectability.

4.4 Development of a Warden: Detecting stegVAD

As shown, stegVAD does not produce a significant impact on the quality of voice, thus mechanisms such as the Mel-cepstrum [26] applied to the audio stream could not be effective. Instead, the development of a warden able to detect stegVAD should focus on the generation of fake RTP packets and alteration to the network footprint. In fact, depending on the method, the hidden data produced by stegVAD could be located in the first h PDUs or regularly spread over n packets, where n is the duration of the silence burst. Even if longer interleavings would make the detection harder (e.g., they reduce the effectiveness of inspecting PDUs by using a random-sampled approach), both mechanisms make stegVAD vulnerable to some form of machine learning-based steganalysis (see, e.g., [2]). Hence, part of our ongoing research aims at investigating a proper "scramble" scheme to prevent such fragility.

A similar issue is given by the alteration of the skew, which can be used as indicator to detect stegVAD. In this case, a proper signature capturing the deviation of some traffic statistics could be developed and used jointly with network analysis or anomaly detection tools to reveal the presence of the covert channel. Again, future research aims at developing proper techniques to better adjust the injection rate of fake RTP packets into the original VoIP stream.

5 CONCLUSIONS AND FUTURE WORK

In this paper we presented a technique to create a covert channel by transforming a VAD-activated VoIP stream into a non-VAD one, and then by using additional RTP packets as the carrier for embedding the secret information. Results indicate the feasibility of the approach and a proper degree of stealthiness, especially as stegVAD does not impact the quality of the VoIP conversations.

Future works aim at refining the approach, especially to increase the robustness of the covert channel and its undetectability. Another relevant amount of work will be done for enriching the implementation of stegVAD. In this respect, we plan to support also the parsing of some SIP methods as to gather more detailed information on the used codec, as well as to enrich the number of supported UAs. Lastly, we are also working towards the development of a dedicated warden able to detect stegVAD-like covert channels, i.e., those manipulating the silence of Internet Telephony conversations.

REFERENCES

[1] S. Zander, G. Armitage, P. Branch, "A survey of Covert Channels and Countermeasures in Computer Network Protocols", IEEE Communications Surveys & Tutorials, Vol. 9, No. 3, pp. 44–57, Third Quarter 2007.

[2] W. Mazurczyk, L. Caviglione, "Steganography in Modern Smartphones and Mitigation Techniques," IEEE Communications Surveys & Tutorials, Vol. 17, No. 1, pp. 334–357, First Quarter 2015.

[3] W. Mazurczyk, L. Caviglione, "Information Hiding as a Challenge for Malware Detection", IEEE Security & Privacy, Vol. 13, No. 2, pp. 89–93, Mar.-Apr. 2015.

[4] S. Wendzel, W. Mazurczyk, L. Caviglione, M. Meier, "Hidden and Uncontrolled - On the Emergence of Network Steganographic Threats", in H. Reimer, N. Pohlmann, W. Schneider (Ed.s), ISSE 2014 Securing Electronic Business Processes, pp. 123–133, Springer, 2014.

[5] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, "Detailed Analysis of Skype Traffic", IEEE Transactions on Multimedia, Vol. 11, No. 1, pp. 117–127, January 2009.

[6] W. Mazurczyk, "VoIP Steganography and Its Detection - A Survey", ACM Computing Surveys, Vol. 46, No. 2, Nov. 2013.

[7] C. R. Forbes, "A New Covert Channel over RTP", MSc Thesis, Rochester Institute of Technology, 2009. Available on-line: https://ritdml.rit.edu/bitstream/handle/1850/12883/CForbesThesis8-21-2009.pdf?sequence=1 [Last Accessed: April 2017].

[8] L. Y. Bai, Y. Huang, G. Hou, B. Xiao, "Covert channels based on jitter field of the RTCP Header", in Proc. of Int Conf. Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, pp. 1388–391, 2008.

[9] Y. Huang, J. Yuan, M. Chen, B. Xiao, "Key Distribution over the Covert Communication Based on VoIP", Chinese Journal of Electronics, Vol. 20, No. 2, pp. 357–360, 2011.

[10] W. Mazurczyk, P. Szaga, K. Szczypiorski, "Using Transcoding for Hidden Communication in IP Telephony", Multimedia Tools and Applications, Vol. 70, No. 3, pp. 2139–2165, 2012.

[11] M. Hamdaqa, L. Tahvildari, "ReLACK: A Reliable VoIP Steganography Approach", in Proc. of the Fifth International Conference on Secure Software Integration and Reliability Improvement, pp. 189–197, Jun. 2011.

[12] I)ruid, "Real-time Steganography with RTP", DefCon, Sept. 2007. Available Online: https://www.defcon.org/images/defcon-15/dc15-presentations/dc-15-druid.pdf [Last Accessed: April 2017].

[13] W. Mazurczyk, M. Karas, K. Szczypiorski, "Skyde: A Skype-based Steganography Method", International Journal of Computers, Communications and Control, Vol. 8, No. 3, pp. 389–400, 2013.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, Network Working Group, RFC Editor, June 2002.

[15] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, Network Working Group, RFC Editor, July 2003.

[16] A. Sangwan, M. C. Chiranth, H. S. Jamadagni, R. Sah, R. V. Prasad, V. Gaurav, "VAD Techniques for Real-Time Speech Transmission on the Internet", in Proc. of the 5th IEEE International Conference on High Speed Networks and Multimedia Communication, pp. 46–50, July 2002.

[17] J. Berger, A. Hellenbart, B. Weiss, S. Moller, J. Gustafsson, G. Heikkila, "Estimation of 'quality per call' in Modelled Telephone Conversations", in Proc. of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4809–4812, 2008.

[18] R. Zopf, "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, Network Working Group, RFC Editor, Sept. 2002.

[19] G. Herlein, J. Valin, A. Heggestad, A. Moizard, "RTP Payload Format for the Speex Codec", RFC 5574, Network Working Group, RFC Editor, June 2009.

[20] G. J. Simmons, "The Prisoner's Problem and the Subliminal Channel", in Proceedings of CRYPTO, pp. 51–67, 1983.

[21] M. West, S. McCan, "TCP/IP Field Behavior", RFC 4413, Network Working Group, RFC Editor, March 2006.

[22] P. Matousek, O. Rysavy, M. Kmet, "Fast RTP Detection and Codecs Classification in Internet Traffic", Journal of Digital Forensics, Security and Law, Vol. 9, No. 2, pp. 101–112, 2014.

[23] L. Caviglione, M. Gaggero, J. F. Lalande, W. Mazurczyk, M. Urbański, "Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence", IEEE Transactions on Information Forensics and Security, Vol. 11, No. 4, pp. 799–810, April 2016.

[24] V. Zue, S. Seneff, J. Glass, "Speech Database Development at MIT: TIMIT and beyond", Speech Communication, Vol. 9, No. 4, pp. 351–356, Aug. 1990.

[25] M. Handley, V. Paxson, C. Kreibich, "Network Intrusion Detection: Evasion, Traffic Normalization, End-to-end Protocol Semantics", in Proc. of the 10th USENIX Security Symposium, vol. 10, pp. 115–131, Aug. 2001.

[26] C. Kraetzer, J. Dittmann, "Mel-cepstrum-based Steganalysis for VoIP Steganography", in Proc. SPIE 6505, Security, Steganography and Watermarking of Multimedia Contents IX, Feb. 2007.