Multimed Tools Appl (2014) 70:2139–2165
DOI 10.1007/s11042-012-1224-8

Using transcoding for hidden communication in IP telephony

Wojciech Mazurczyk · Paweł Szaga · Krzysztof Szczypiorski

Published online: 12 September 2012

Abstract The paper presents a new steganographic method for IP telephony called TranSteg (Transcoding Steganography). Typically, in steganographic communication it is advised for covert data to be compressed in order to limit its size. In TranSteg it is the *overt data* that is compressed to make space for the steganogram. The main innovation of TranSteg is to, for a chosen voice stream, find a codec that will result in a similar voice quality but smaller voice payload size than the originally selected. Then, the voice stream is transcoded. At this step the original voice payload size is intentionally unaltered and the change of the codec is not indicated. Instead, after placing the transcoded voice payload, the remaining free space is filled with hidden data. TranSteg proof of concept implementation was designed and developed. The obtained experimental results are enclosed in this paper. They prove that the proposed method is feasible and offers a high steganographic bandwidth while introducing small voice degradation. Moreover, TranSteg detection is difficult to perform when compared with existing VoIP steganography methods.

Keywords IP telephony · Network steganography · TranSteg · Information hiding

1 Introduction

Voice over IP (VoIP), or IP telephony, is one of the services of the IP world that is changing the entire telecommunication's landscape. It is a real-time service, which enables users to make phone calls through data networks that use an IP protocol. An IP telephony connection

W. Mazurczyk (✉) · P. Szaga · K. Szczypiorski
Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
00-665, Nowowiejska 15/19
e-mail: wmazurczyk@tele.pw.edu.pl

P. Szaga
e-mail: P.Szaga@stud.elka.pw.edu.pl

K. Szczypiorski
e-mail: ksz@tele.pw.edu.pl

consists of two phases, in which certain types of traffic are exchanged between the calling parties (Fig. 1). These are:

- *Signalling phase* – in this phase signalling protocol messages, e.g. SIP messages (Session Initiation Protocol) [38], are exchanged between the caller and callee. These messages are intended to setup and negotiate the connection parameters between the calling parties.
- *Conversation phase* – if the previous phase is successful then the conversation takes place, in the form of two audio streams which are sent bidirectionally. RTP (Real-Time Transport Protocol) [39] is most often utilised for voice data transport, thus packets that carry voice payload are called RTP packets (see Fig. 2). The consecutive RTP packets form an RTP stream.

Steganography encompasses various information hiding techniques, whose aim is to embed a secret message (steganogram) into a carrier. Network steganography, to perform hidden communication, utilizes network protocols and/or relationships between them as the carrier for steganograms. Because of its popularity, IP telephony is becoming a natural target for network steganography [26]. Steganographic methods are aimed at hiding of the very existence of the communication, therefore any third-party observers should remain unaware of the presence of the steganographic exchange [40].

In this paper we introduce a new steganographic method – TranSteg (Transcoding Steganography) – which is intended for a broad class of multimedia and real-time applications, but its main foreseen application is IP telephony. TranSteg can also be exploited in other applications or services (like video streaming), wherever a possibility exists to efficiently compress (in a lossy or lossless manner) the overt data. The typical approach to steganography is to compress the *covert data* in order to limit its size (it is reasonable in the context of a limited steganographic bandwidth). In TranSteg compression of the *overt data* is used to make space for the steganogram – the concept is similar like in invertible authentication watermark that was first proposed by Fridrich et al. [11] for JPEG images. TranSteg bases on the general idea of transcoding (lossy compression) of the voice data from a higher bit rate codec (and thus greater voice payload size) to a lower bit rate codec (with smaller voice payload size) with the least possible degradation in voice quality.

The general idea behind TranSteg is as follows (the detailed procedures for different hidden communication scenarios are described in Sec. 3). RTP packets carrying user voice

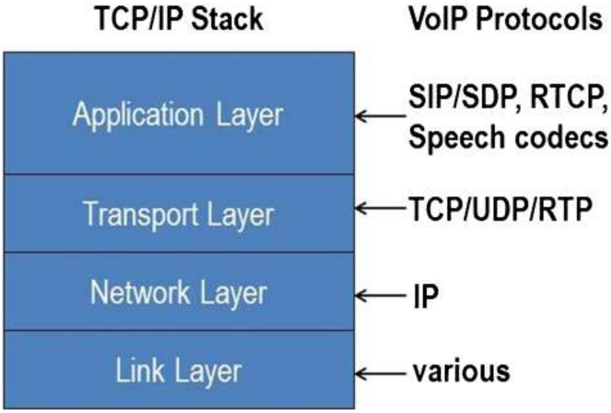

Fig. 1 Protocols for SIP-based VoIP

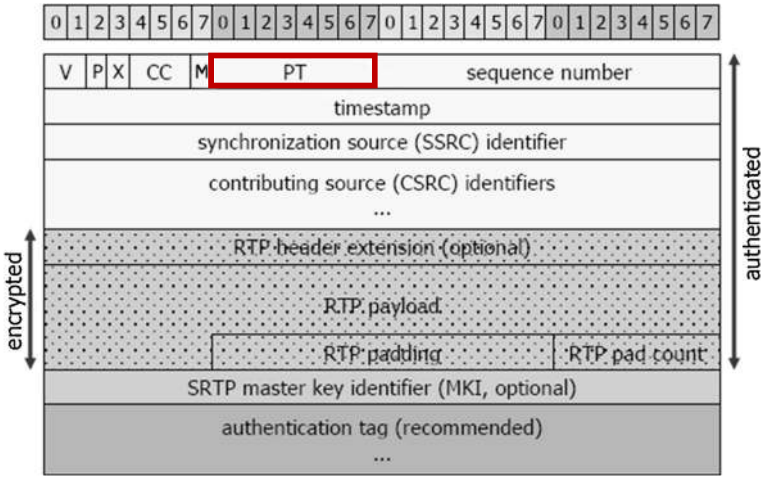
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
V
P
X
CC
M
PT
sequence number
timestamp
synchronization source (SSRC) identifier
contributing source (CSRC) identifiers
...
RTP header extension (optional)
RTP payload
RTP padding
RTP pad count
SRTP master key identifier (MKI, optional)
authentication tag (recommended)
...
encrypted
authenticated

Fig. 2 RTP packet secured with SRTP protocol

are inspected and the codec originally used for speech encoding (here called *overt codec*) is determined by analysing the PT (Payload Type) field in the RTP header (marked in Fig. 2).

Then, TranSteg finds an appropriate codec for the overt codec, called a *covert codec*. The application of the *covert codec* yields a comparable voice quality but a smaller voice payload size than originally. Next, the voice stream is transcoded, but the original, large, voice payload size and the codec type indicator are preserved, thus the PT field is left unchanged. Instead, after placing the transcoded voice of a smaller size inside the original payload field, the remaining free space is filled with hidden data (see Fig. 3).

It is worth noting that TranSteg can be utilized in different hidden communication scenarios – not only between the sender and the receiver of the RTP stream (i.e. in an end-to-end manner – see Sec. 3 for details). It can be also, in particular, utilized for secured VoIP streams e.g. using the most popular SRTP protocol [4] (Secure RTP) that provides confidentiality and authentication for RTP packets. Figure 2 illustrates the parts of RTP packets that are encrypted and authenticated.

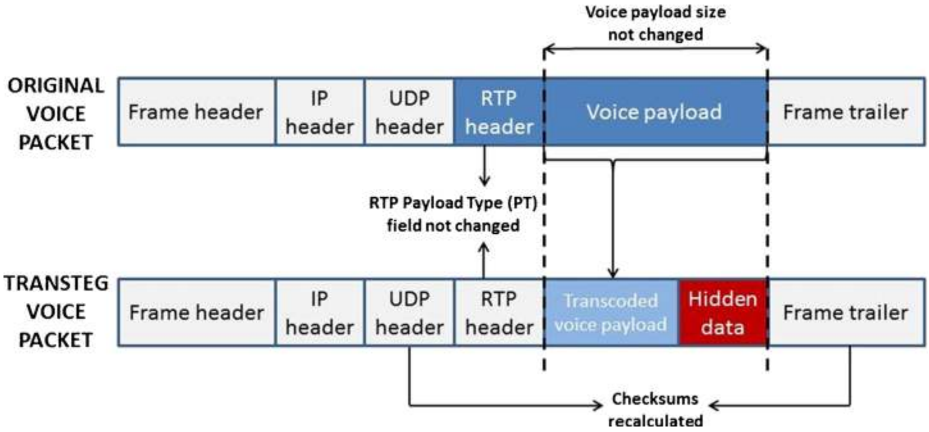

Fig. 3 Frame bearing voice payload: without (top) and with (bottom) hidden data inserted by TranSteg

TranSteg, like every steganographic method, can be described by the following set of characteristics: its steganographic bandwidth, its undetectability and the steganographic cost. The term – *steganographic bandwidth* – refers to the amount of secret data can be sent per time unit, when using a particular method. *Undetectability* is defined as the inability to detect a steganogram within a certain carrier. The most popular way to detect a steganogram is to analyse statistical properties of the captured data and compare them with the typical values for that carrier. Lastly, the *steganographic cost* characterises the degradation of the carrier caused by the application of the steganographic method. In the case of TranSteg, this cost can be expressed by means of providing a measure of conversation quality degradation, induced by transcoding and the introduction of an additional delay.

The contributions of this paper are:

- A detailed presentation of a new VoIP steganographic method,
- An analysis of the properties of TranSteg with the aid of a proof of concept implementation: the steganographic bandwidth, undetectability and the steganographic cost,
- Proposition of potential approaches for TranSteg detection.

The rest of the paper is structured as follows. Section 2 describes related work in VoIP steganography. Section 3 describes in detail the functioning of TranSteg and its hidden communication scenarios. Section 4 discusses the proof of concept implementation and presents the experimental results. Finally, Section 5 concludes our work.

2 Related work

IP telephony as a hidden data carrier can be considered a fairly recent discovery. The proposed steganographic methods stem from two distinctive research origins. Firstly, from the well-established image and audio file steganography [5] – these methods targeted the digital representation of voice as carrier for hidden data. The second sphere of influence are the so called covert channels, created in different network protocols [1], [34] (a good survey on covert channels, by Zander et al., can be found in [48]) – these solutions target specific VoIP protocol fields (e.g. signalling protocol – SIP, transport protocol – RTP or control protocol – RTCP) or their behaviour. Presently, steganographic methods that can be utilized in telecommunication networks are jointly described by the term *network steganography*, or, specifically, when applied to IP telephony, by the term *steganophony* [26].

The first VoIP steganographic methods that utilize the digital voice signal as a hidden data carrier were proposed by Dittmann et al. in 2005 [7]. Authors had evaluated the existing audio steganography techniques, with a special focus on the solutions which were suitable for IP telephony. This work was further extended and published in 2006 in [23]. In [17], an implementation of SteganRTP was described. This tool employed least significant bits (LSB) of the G.711 codec to carry steganograms. Wang and Wu, in [46], also suggested using the least significant bits of voice samples to carry secret communication but here, the bits of the steganogram were coded using a low rate voice codec, like Speex. In [41], Takahashi and Lee proposed a similar approach and presented its proof of concept implementation – Voice over VoIP (Vo^{2}IP), which can establish a hidden conversation by embedding compressed voice data into the regular, PCM-based, voice traffic. The authors had also considered other methods that can be utilized in VoIP steganography, like DSSS (Direct Sequence Spread Spectrum), FHSS (Frequency-Hopping Spread Spectrum) or Echo hiding. Aoki in [2] proposed a steganographic method based on the characteristics of PCMU

(Pulse Code Modulation), in which the 0-th speech sample can be represented by two codes due to the overlap. Another LSB-based method was proposed by Tian et al. in [45]. Authors had incorporated the m-sequence technique to eliminate the correlation among secret messages to resist statistical detection. A similar approach, also LSB-based, relying on adaptive VoIP steganography was presented by the same authors in [44]; a proof of concept tool - StegTalk – was also developed. In [32] Miao and Huang presented an adaptive steganography scheme that based on smoothness of the speech block. Such an approach proved to give better results in terms of voice quality than the LSB-based method. A high-capacity steganography technique based on the utilisation of inactive frames of G.723.1 speech codec was introduced by Huang et al., [16]. The authors prove that the inactive frames of VoIP streams are more suitable for data embedding than the active ones, thus more hidden data can be embedded in them with the same imperceptibility. They then proposed a steganographic algorithm in different speech parameters of the inactive frames for G.723.1 codec with 6.3 kbit/s bitrate. Experimental results show the solution is imperceptible and a high steganographic bandwidth up to 101 bits/frame is achieved.

Utilisation of the VoIP-specific protocols as a steganogram carrier was first proposed by Mazurczyk and Kotulski in 2006 [28]. The authors proposed using covert channels and watermarking to embed control information (expressed as different parameters) into VoIP streams. The unused bits in the header fields of IP, UDP and RTP protocols were utilized to carry the type of parameter and the actual parameter value is embedded as watermark into the voice data. The parameters are used to bound control information, including data authentication to the current VoIP data flow. In [30] and [31] Mazurczyk and Szczypiorski described network steganography methods that can be applied to VoIP: to its signalling protocol – SIP (with SDP), and to its RTP streams (also with RTCP). They discovered that a combination of information hiding solutions provides a capacity to covertly transfer about 2000 bits during the signalling phase of a connection and about 2.5 kbit/s during the conversation phase. In [31], a novel method called LACK (Lost Audio Packets Steganography) was introduced; it was later described and analysed in [29] and [26]. LACK relies on the modification of both: the content of the RTP packets, and their time dependencies. This method takes advantage of the fact that, in typical multimedia communication protocols, like RTP, excessively delayed packets are not used for the reconstruction of the transmitted data at the receiver, i.e. the packets are considered useless and discarded. Thus, hidden communication is possible by introducing intentional delays to selected RTP packets and substituting the original payload with a steganogram.

Bai et al. in [3] proposed a covert channel based on the jitter field of the RTCP header. This is performed two-stage: firstly, statistics of the value of the jitter field in the current network are calculated. Then, the secret message is modulated into the jitter field according to the previously calculated parameters. The utilization of such modulation guarantees that the characteristic of the covert channel is similar to that of the overt one. In [6], Forbes proposed a new RTP-based steganographic method that modifies the timestamp value of the RTP header to send steganograms. The method's theoretical maximum steganographic bandwidth is 350 bit/s.

The TranSteg technique presented in this paper is a development of the latter of the discussed groups of steganographic methods for VoIP, originating from covert channels. Compared to the existing solutions, its main advantages are: a high steganographic bandwidth, low steganographic cost and difficult detection. To authors' best knowledge TranSteg's steganographic bandwidth is the highest known in the state of the art while simultaneously maintaining the voice degradation at the safe level. Moreover, only TranSteg offers restoring voice data at steganogram receiver (by performing re-transcoding) that is practically the same as originally sent one. Thus, all evidences of steganogram are wiped out. For other steganographic methods presented above the steganogram can be extracted and removed but original data cannot be restored because it is being erased while performing steganogram embedding. That is why, TranSteg is harder to detect.

All of these features will be described and analysed in the context of TranSteg in the consecutive sections.

3 Communication scenarios, functioning and detection

The performance of TranSteg depends, most notably, on the characteristics of the pair of codecs (as mentioned in the Introduction): one used originally to encode user speech – the *overt codec*, and one utilized for transcoding – the *covert codec*. It is worth noting that, depending on the hidden communication scenario, TranSteg may or may not be able to influence the choice of this codec. It is assumed that it is always possible to find a covert codec for a given overt one. However, it must be noted, that for very low bit rate codecs, the steganographic bandwidth shall be limited. In the ideal conditions the covert codec should:

- not degrade considerably user voice quality (caused by the transcoding operation and the introduced delays), when compared to the quality of the overt codec.
- provide the smallest achievable voice payload size as it results in the most free space in an RTP packet to convey a steganogram.

If there is a possibility to influence the overt codec (see the hidden communication scenarios below), in an ideal situation it should:

- result in a largest possible voice payload size to provide, together with the covert codec, the highest achievable steganographic bandwidth,
- be commonly used to not to raise suspicion.

Taking the above into account, TranSteg's steganographic bandwidth (*SB*) can be expressed as:

$$SB = (PS_O - PS_C) \cdot PN_S[bit/s] \tag{3–1}$$

where PS_O denotes the overt codec's payload size, PS_C is the covert codec's payload size and PN_S describes the number of RTP packets sent during 1 s.

TranSteg can be utilized in four hidden communication scenarios (Fig. 4). The first scenario (S1 in Fig. 4) is the most common and typically the most desired: the sender and the receiver conduct a VoIP conversation while simultaneously exchanging steganograms (end-to-end). The conversation path is identical with the hidden data path. In the next three scenarios (marked S2-S4 in Fig. 4) only a part of the VoIP end-to-end path is used for hidden communication. As a result of actions undertaken by intermediate nodes, the sender and/or the receiver are, in principle, unaware of the steganographic data exchange. The application of TranSteg in IP telephony connections offers a chance to preserve users' conversation and simultaneously transfer steganograms. As noted previously, this is especially important for scenarios S2-S4.

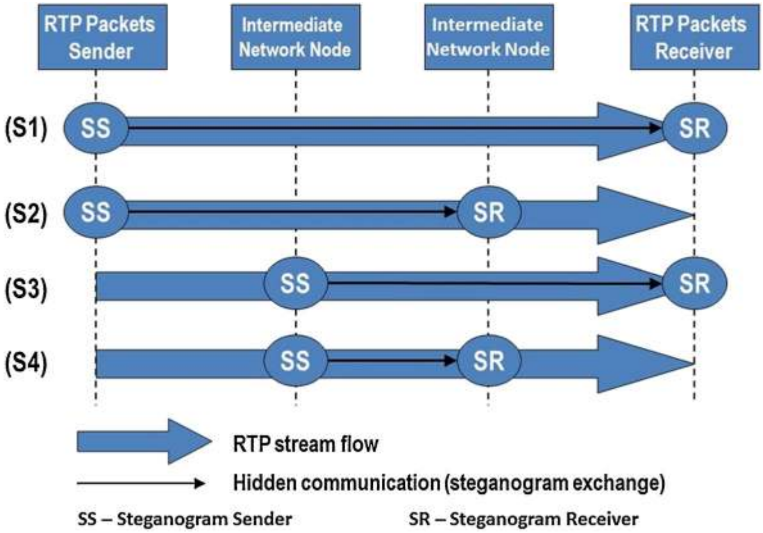
RTP Packets Sender
Intermediate Network Node
Intermediate Network Node
RTP Packets Receiver
(S1)
SS
SR
(S2)
SS
SR
(S3)
SS
SR
(S4)
SS
SR
RTP stream flow
Hidden communication (steganogram exchange)
SS – Steganogram Sender
SR – Steganogram Receiver

Fig. 4 Hidden communication scenarios for TranSteg

In the abovementioned scenarios it is assumed that potential detection (steganalysis), usually executed by a warden [9], is not able to audit the speech carried in RTP packets because of the privacy issues related with this matter. Thus, the presence of a steganogram inside RTP packet payload can remain undiscovered. Other possibilities of TranSteg detection will be discussed in detail in subsection 3.3.

In the following part of this section TranSteg will be described with reference to the abovementioned scenarios. The most important factor in this context is whether the Steganogram Sender (SS) is located on the same host as the RTP packets' issuer. Thus, it may be able to control the RTP stream transmitter. Otherwise, when located on some intermediate network node, it will not be capable of such control.

TranSteg may be also influenced by the utilization of SRTP protocol, which is used to provide the RTP stream with confidentiality and authentication. As mentioned in the Introduction, securing of the RTP stream does not necessarily impede the possibility of the exploitation of TranSteg. Such mode of operation may potentially even increase TranSteg's undetectability – this effect will be further investigated throughout the following subsections.

3.1 Steganogram sender controlling an RTP packet transmitter (scenarios S1 & S2)

In scenario S1 a steganogram is embedded into an RTP packet and travels along the entire path between the RTP stream sender and receiver. Thus, there is no need to execute the operation of transcoding. User voice can be directly encoded with the desired covert codec with the omission of the prior encoding with the overt codec and thus avoid the whole process of transcoding. Despite this, the RTP stream will appear to have been encoded with the aid of the overt codec. The voice payload size and PT field in the RTP header shall not be changed. It is assumed that the SS and SR had agreed prior on the choice of the covert codecs corresponding to different overt codecs. Such common mapping may, for example, bind an overt codec G.711 with the covert codec G.726, or Speex 24.6 kbit/s with Speex 8 kbit/s, etc.

Thus, the SS shall perform the following steps for the embedding of a steganogram (Fig. 5):

- **Step 1:** Set the RTP payload size and modify Payload Type in the RTP header according to the chosen overt codec. These changes will indicate usage of the overt coding algorithm that will not, actually, be utilized.
- **Step 2:** The voice transcoded with the covert codec is inserted into the overt codec's RTP payload field.
- **Step 3:** Remaining free space is allocated for the hidden data and filled with a steganogram.
- **Step 4:** RTP packet is sent to the receiver.

When the modified TranSteg RTP stream reaches the SR, it extracts the voice payload and steganogram from the consecutive packets. The voice payload is then used for speech reconstruction and the steganogram parts are concatenated. This preserves the conversation functionality between the SS and SR and simultaneously enables hidden communication. For a third party observer, even if he/she is able to physically monitor the activity of both users (e.g. wiretap both locations) it will look like a regular call taking place.

To further mask the presence of TranSteg, SS can utilize the SRTP protocol to perform RTP payload encryption of both: the voice coded with a covert codec and the steganogram; thus making the detection of steganography even harder to perform (see Fig. 2).

The hidden communication scenario S1 offers most flexibility, and is advantageous when compared with the remaining ones, because:

- SS can choose the overt codec and thus influence the resulting steganographic bandwidth.
- The delays introduced by TranSteg to the RTP stream are the smallest in this scenario as there is no time consumption related with the transcoding (the voice is directly encoded with the covert codec).
- This scenario does not assume any required path of communication that the RTP stream should follow.
- To capacitate the exploitation of TranSteg, it is only necessary to modify the IP telephony client. Notably, the RTP protocol is usually implemented in software, which

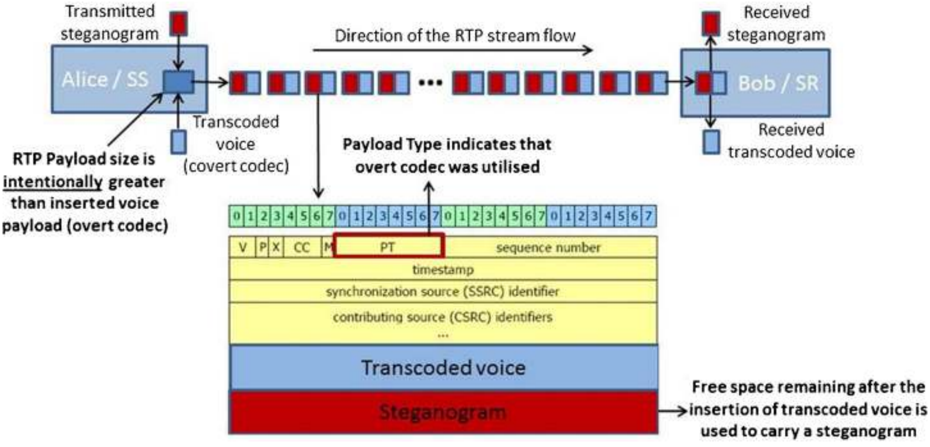
Transmitted steganogram
Alice / SS
Direction of the RTP stream flow
Received steganogram
Bob / SR
Transcoded voice (covert codec)
RTP Payload size is intentionally greater than inserted voice payload (overt codec)
Payload Type indicates that overt codec was utilised
Received transcoded voice
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
V P X CC M PT sequence number
timestamp
synchronization source (SSRC) identifier
contributing source (CSRC) identifiers
...
Transcoded voice
Steganogram
Free space remaining after the insertion of transcoded voice is used to carry a steganogram

Fig. 5 The TranSteg concept – scenario S1 (SS – Steganogram Sender; SR – Steganogram Receiver)

means it can be easily modified. No other protocol's modifications are required (i.e. UDP and frame checksums).
- RTP stream can be, additionally, secured with the aid of the SRTP protocol – this can be utilized to mask the contents of the transcoded voice data and the steganogram.

In scenario S2, the main difference when compared with S1, is that the SR is situated at some intermediate network node. Thus, the IP telephony conversation is performed between the SS (caller) and an unaware of the steganographic procedure callee. The assumption in this scenario is that the SR is able to intercept and analyze all RTP packets exchanged between the SS and the callee. The TranSteg procedure for SS remains the same as in scenario S1. What changes is the behaviour of the SR.

When the tampered RTP stream reaches the SR, it performs the following steps:

- **Step 1:** It extracts voice payload and the steganogram from the RTP packets.
- **Step 2:** The voice payload is transcoded from the covert to overt codec and placed once again in consecutive RTP packets. By performing this task the steganogram is overwritten with user voice data.
- **Step 3:** Checksums for the lower layer protocols (i.e. the UDP checksum and CRC at the data link if they had been utilized) are adjusted.
- **Step 4:** Modified frames with encapsulated RTP packets are sent to the receiver (callee).

If the IP telephony connection is required to be secured with the SRTP protocol it does not impede the possibility to utilize TranSteg. The session keys used for authentication and encryption are exchanged between the calling parties before the conversation phase of the call and will be known in advance to the SS. This means that when the SS initiates an RTP stream, the first RTP packets contain transcoded voice but are intentionally *not* encrypted. Instead of a steganogram, they carry cryptographic keys that where negotiated between the SS and callee. The cryptographic keys do not necessarily have to be carried inside the payload field as this can raise security issues. The better solution will be to use advance techniques like MLS (Multilevel Steganography) [10]. In MLS, at least two steganographic methods are utilised simultaneously in such a way that one method's (the upper-level) network traffic serves as a carrier for the second method (the lower-level). In such scenario TranSteg will be used as upper-level method and the lower-level method's steganographic bandwidth will be utilised to carry a cryptographic key. Upon their extraction, the SR is able to encrypt the transcoded voice payload prior to forwarding it to the RTP packets receiver. Thus, the receiving party will not be aware of the steganographic procedure. Secondly, the SR will be capable of performing bidirectional hidden communication.

To summarize scenario S2:

- SS can still choose the overt codec and thus influence the resulting steganographic bandwidth.
- The delays introduced by TranSteg to the RTP stream depend on one transcoding operation.
- There is an assumption that SR is on the communication path between the calling parties and is able to oversee the whole RTP stream.
- TranSteg requires certain protocol modifications in the SR: the RTP and other network protocols (e.g. the UDP or data link layer protocols).
- Utilization of SRTP between the calling parties is not an obstacle for TranSteg. Analogically to scenario S1, it can be viewed as means to further mask hidden communication.

3.2 Steganogram sender located at an intermediate network node (scenarios S3 & S4)

In scenario S3, the assumption is that SS is able to intercept and analyse all RTP packets exchanged between caller and the callee. SS does not control the RTP packet's transmitter, thus it cannot pick a suitable overt codec. However, SR is a legitimate (overt) receiver of the RTP stream. Thus it is able to influence the choice of overt codec by negotiating it during the signalling phase of the call, with the calling party remaining unaware of the steganographic procedure. The behaviour of the SS is similar to the behaviour of SR in scenario S2 (see Sec. 3.1). The only difference is that SS is responsible for the transcoding from the overt to covert codec and for embedding of the steganogram – the remaining steps are the same. Thus, SS behaves as follows:

- **Step 1:** For an incoming RTP stream it transcodes the user's voice data from the overt to covert codec.
- **Step 2:** Transcoded voice payload is placed once again in an RTP packet.
- **Step 3:** The remaining free space of the RTP payload field is filled with steganogram's bits (thus the original voice payload is erased).
- **Step 4:** Checksums in lower layer protocols (UDP checksum and CRC at the data link) are adjusted.
- **Step 5:** Modified frames with encapsulated RTP packets are sent to the receiver (SR).

SR's operation is solely limited to extraction and analysis of the voice payload and steganogram from consecutive RTP packets (it is the same behaviour as in scenario S1, see Sec. 3.1).

In the presence of SRTP, in this scenario, the use of the TranSteg is not compromised – the conditions and the solution (cryptographic key's sharing between the SR and SS by means of TranSteg) is similar like in scenario 2 (see Sec. 3.1). The only difference is that SR after establishing the cryptographic key for SRTP purposes sends it to SS in the first RTP packets. These packets are intentionally *not* encrypted and they carry transcoded voice and SRTP cryptographic key. SR retrieves the key and is responsible to perform re-transcoding and ciphering of resulting voice payload.

To summarize, in scenario S3:

- SR is responsible of influencing the choice of the overt codec by negotiating it with the calling party (unaware of the steganographic procedure).
- The delays introduced to the RTP stream by TranSteg depend on one transcoding operation.
- There is an assumption that SS is on the communication path between the calling parties and is able to oversee the whole RTP stream.
- TranSteg requires certain protocol modifications in the SS: the RTP and other network protocols (e.g. the UDP or data link layer protocols).
- Utilization of SRTP between the calling parties is not an obstacle for TranSteg. Analogically to scenario S1, it can be viewed as means to further mask hidden communication.

In scenario S4 it is assumed that both: SS and SR, are able to intercept and analyze all RTP packets exchanged between the calling parties. Thus, SS and SR cannot at all influence the choice of the overt codec, because they are both located at some intermediate network node (Fig. 6). Due to this fact they are bound to rely on the codec chosen by the overt, non-steganographic, calling parties. This, in particular, can result in low steganographic bandwidth as the hidden communication parties must adjust the covert codec to the negotiated

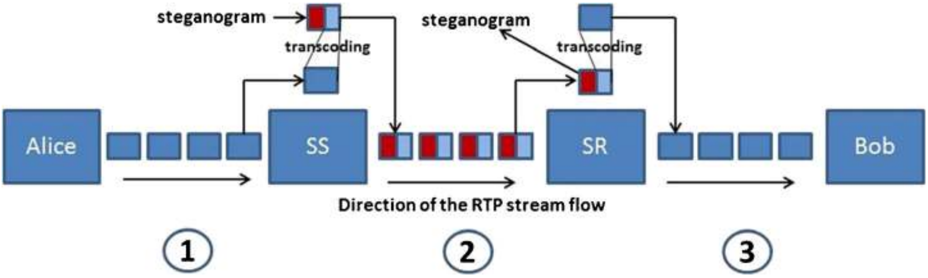

Fig. 6 The TranSteg concept – scenario S4 (SS – Steganogram Sender; SR – Steganogram Receiver)

overt codec. The most significant advantage of this TranSteg scenario is its potential use of aggregated IP telephony traffic to transfer steganograms. If both SS and SR have access to more than one VoIP call then the achievable steganographic bandwidth can be significantly increased, which can compensate for the loss in steganographic bandwidth caused by the inability to influence the choice of the overt codec.

The behavior of SS and SR is similar – they both perform transcoding: SS from overt to covert, and SR from covert to overt codecs. Steganogram is exchanged only along the part of the communication path where RTP stream travels “inside” the network – it never reaches the endpoints. The steps of the TranSteg scenario for SS are exactly the same as in scenario S3 (see above) and SR follows the logic presented in scenario S2 (see Sec. 3.1). It is also worth noting that, in this scenario, the utilization of SRTP protocol for conversation security entirely incapacitates the usage of TranSteg.

To summarize scenario S4:

- There is an assumption that both: SS and SR, are on the communication path between the calling parties and are able to oversee the whole RTP stream.
- Neither SS nor SR can influence the choice of the overt codec, which potentially leads to a decrease in the steganographic bandwidth.
- There is a possibility to use aggregated VoIP traffic at the path between SS and SR, and thus significantly increase TranSteg’s steganographic bandwidth.
- Neither SS nor SR are involved in the IP telephony conversation as overt calling parties. Thus, it is harder to detect hidden communication between the SS and SR comparing to the previously described scenarios (since neither is an initiator of the overt traffic).
- The delays introduced to the RTP stream are the highest compared with the other presented scenarios (due to the two transcoding operations).
- TranSteg requires certain protocol modifications in both: SS and SR; these involve modifications to the RTP and other network protocols (e.g. UDP or data link layer protocols).
- Utilization of SRTP to secure the conversation makes the use of TranSteg impossible.

In all of the presented scenarios S1-4, RTP packet losses which are a natural phenomenon in IP networks can make the successful extraction of the steganogram at SR impossible. That is why an additional protocol in a hidden channel may be required to provide reliability. One solution is to use an approach proposed by Hamdaqa and Tahvildari [15] because it can be easily incorporated for TranSteg purposes. It provides a reliability and fault tolerance mechanism based on a modified (k, n) threshold based on Lagrange Interpolation and results demonstrated in that paper proves that the complexity of steganalysis is increased. Of course the “cost” for the extra reliability is always a loss of some fraction of the steganographic bandwidth.

3.3 TranSteg detection

As mentioned at the beginning of Section 3, we assume that during the TranSteg-based hidden communication there is a warden executing detection (steganalysis) methods. We further assume that the warden will not be able to "physically listen" to the speech carried in RTP packets because of the privacy issues related with this matter. This means that the warden will be capable of capturing and analysing the payload of each RTP packet but not capable of replaying the call's conversation (its content).

However, it must be emphasised that, if the SRTP protocol had been used for securing a TranSteg conversation, the warden will fail to detect the presence of steganograms in the RTP stream, in any of the below-mentioned scenarios (with the exception for S4).

To perform hidden communication, TranSteg utilizes modifications to the PDUs (Protocol Data Units) as a carrier – more precisely to the RTP payload field. When compared with other steganographic VoIP methods that, e.g. influence the order of the RTP packets or the delays between them, TranSteg does not introduce any irregularities to the RTP stream.

The successful detection of TranSteg mainly depends on:

- the location(s) at which the warden is able to monitor the modified RTP stream.
- the utilized TranSteg scenario (S1-4).

If the warden is capable of inspecting traffic solely in a single network, e.g. in the LAN (Local Area Network) of the overt transmitter or receiver, then the detection is hard to accomplish. The reason for the above is due to the fact that an RTP stream at a single traffic inspection point resembles legitimate streams. The remaining cases will be discussed below.

In scenario S1, there is no change of format of voice payloads during the traversing of the network. Thus, even if the warden would monitor traffic in different networks – the result would always be the same. Thus, the chances of TranSteg detection are very limited.

In scenarios S2 and S3 there is one transcoding operation, therefore the modification of the RTP packets' payload can be detected if the warden is able to probe and compare traffic from two localizations: prior and post the transcoding. However, it must be emphasized that the same happens to other existing VoIP steganographic methods i.e. in these scenarios all data hiding is easier to be detected.

In scenario S4, there are three possible locations where the warden can inspect RTP traffic. They are marked as 1, 2 and 3 in Fig. 6. If a warden can monitor traffic in networks: 1 and 2 or 2 and 3 the detection capabilities are the same for scenarios S2 and S3.

In the case when the warden is able to inspect the RTP stream in networks 1 and 3, where the voice format should be the same, then, due to the transcoding operation, some slight differences can be noted. This case is further investigated in Section 4. It must be noted that this scenario potentially can induce the largest voice quality degradation due to the necessary two transcoding operations. However even in these circumstances TranSteg is superior to other VoIP steganography methods because it offers restoring voice data (by performing re-transcoding at SR) that is practically the same as originally sent one. Thus, all evidences of steganogram are wiped out. For other steganographic methods the steganogram can be extracted and removed but original data cannot be restored because it was erased while performing steganogram embedding. That is why, TranSteg is harder to detect.

Communication via TranSteg can be thwarted by certain actions undertaken by the wardens. The method can be defeated by applying random transcoding to every non-encrypted VoIP connection, to which the warden has access. Alternatively, only suspicious connections may be subject to transcoding. However, such an approach would lead to a deterioration of the quality of conversations. What must be emphasised, not only steganographic calls would be affected – the non-steganographic calls could also be "punished". In section 4 we provide guidelines for pinpointing suspicious IP telephony connections: we investigate RTP payload byte values' distribution as a possible indication of TranSteg utilization. It is worth noting that this approach will fail to succeed if SRTP protocol is applied.

Due to the above, it is necessary to explore other possibilities, which could facilitate the development of an efficient detection method fulfilling the constraints dictated by the VoIP's real-time operation constraints. One promising research direction worth pursuing is the adoption of the method proposed by Wright et al. in [47], which can be utilized for SRTP encrypted payload. However, this technique is only applicable for variable bit rate codecs. The authors of this work discovered that the lengths of encrypted RTP packets can be used to identify phrases spoken within a call. Therefore, if extended, this approach can be applied to deduce the characteristics of the employed speech codec, which would increase the probability of detection of covert communication.

The summary and comparison of hidden communication scenarios with respect to TranSteg functioning and detection (described in Sections 3.1–3.3) is presented in Table 1.

Intentional attacks to remove steganogram when TranSteg is applied to VoIP call will likely be unsuccessful because steganogram can be spread across the payload field and "mixed" with voice data. However, one thing can be done to limit the TranSteg utilization i.e. performing intentional, real transcoding inside the network. This will lead to destroying the steganogram while it has little impact on non-steganographic VoIP users.

4 TranSteg implementation and experimental results

TranSteg implementation was developed for the hidden communication scenario presented in Fig. 6, i.e. when both SS and SR are located at intermediate network nodes. This scenario

Table 1 Comparison of hidden communication scenarios (S1-4)

Scenario	S1	S2	S3	S4
SS/SR must be on the communication path to capture RTP stream	-	SR	SS	SS and SR
SS/SR influence the choice of the overt codec	SS and SR	SS	SR	-
Number of necessary TranSteg transcoding operations	0	1	1	2
Possibility to use aggregate VoIP traffic	-	-	-	+
Neither SS nor SR is a VoIP calling party (harder detection)	-	-	-	+
Necessary modifications to lower layer protocols, i.e. UDP checksum, frame CRC	-	+	+	+
SRTP utilization	Masks TranSteg	Masks TranSteg	Masks TranSteg	Prevents TranSteg
Number of necessary network monitoring/probing localizations for TranSteg detection	-	2	2	2

was chosen because, from the perspective of the delays introduced by TranSteg, it is a worst case scenario (due to the two transcoding operations - first at SS and then at SR). Thus, with the aid of the prototype, we want to find out to what extent TranSteg can degrade voice quality. For any other hidden communication scenario from Fig. 4, the introduced delays will be lower.

4.1 TranSteg proof of concept implementation

TranSteg proof of concept implementation was developed in C++ on the Linux Ubuntu 10.10 (kernel 2.6.35) platform. Figure 7 presents the functional architecture of the TranSteg prototype implementation. It is based on GTK+2.24.4 [13] threads which are used for GUI support, packet capture control and steganogram exchange. Netfilter [35] framework was also utilized for IP packet manipulation – modules: *iptables 1.4.10* and *libnetfilter_queue 0.0.17* were used.

All packets passing through a host are traversing *iptables* chains (see Fig. 8). There are three main types of these chains:

- INPUT – chain for incoming (received) packets that are intended for a process running on the local machine;
- OUTPUT – chain for packets that are being sent from a process running on the local machine;
- FORWARD – chain for packets that are being forwarded through the host (from one network interface to another).

The developed TranSteg implementation enables controlling the *iptables* module. It is therefore possible to select from which chain the packets will be taken for modifications. This permits for the recreation of all of the previously discussed hidden communication scenarios (from S1 to S4) with the aid of the created prototype. Table 2 presents which chains should be used for each scenario S1-4.

While traversing the *iptables* chains, packets are in the kernel space, which is inaccessible from the user space. Because of that, to perform modifications, packets from selected chains are put to a special QUEUE chain. This chain does not appear in the normal/usual packet traversing paths (see Fig. 8). Subsequently, the *libnetfilter_queue* module is used to send packets one by one to the user space, where TranSteg modifications take place. This process

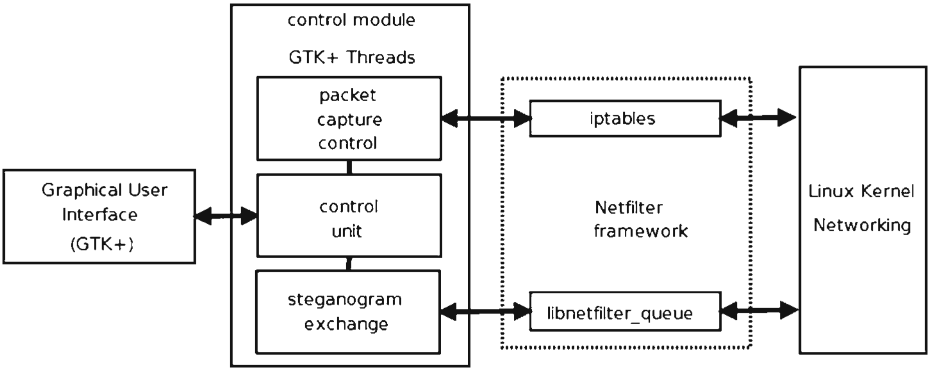

Fig. 7 TranSteg implementation architecture

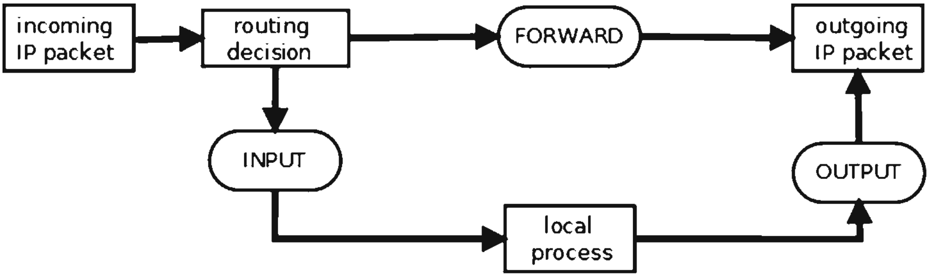

Fig. 8 IP packet traversing *iptables* chains

will be described further. Following the modifications, packets are returned to the QUEUE and continue traversing the *iptables* chains.

The TranSteg implementation bases on the G.711 (64 kbit/s) as an overt codec and G.726 (32 kbit/s) as the covert one. Thus, the transcoding is from G.711 to G.726 at the SS and the inverse is performed at the SR. The choice of the covert codec is entirely controlled by the SS and SR. Therefore, once the overt codec is known, a proper covert codec can be selected (e.g. one that does not degrade the voice quality and offers high steganographic bandwidth). In this TranSteg implementation G.711 codec was chosen as an overt codec because:

- It is the most popular speech codec that is widely used in IP telephony endpoints (both soft- and hard-phones) - it is simple to implement and does not require any license to be used.
- Generally, most common hard-phones and VoIP hardware devices, like gateways, (e.g. Cisco IP Phones, Sipura SPA-841, Vonage Phone Adapter, Linksys PAP2 or WRT54GP2) utilize codecs such as G.711, G.723 or G.729, while most of the popular softphones (e.g. SJPhone, Google Talk or X-lite) often offer, besides G.711, popular codecs like free Speex or iLBC (a list of VoIP clients and supported codecs can be found at: http://www.ozvoip.com/voip-codecs/devices/). In the majority of the endpoints G.711 is chosen as a default codec as it offers high quality – above 4 in MOS (Mean Opinion Score) scale – and is most likely to guarantee a successful speech codec negotiation during the signalling phase of a connection. For example, if we want to make a call between a Cisco IP Phone 7960 (which supports G.711 and G.729) and a popular, free softphone X-lite (which supports G.711, GSM, iLBC and Speex) successful connection negotiation is only possible with the G.711 codec.
- Whenever it is necessary to setup a call to PSTN (Public Switched Telephone Network) then, to avoid unnecessary transcoding and provide interoperability, the G.711 codec is frequently utilized.

Table 2 SS and SR *iptables* chains for different hidden communication scenarios S1-S4

Scenario	SS *iptables* chain	SR *iptables* chain
S1	OUTPUT	INPUT
S2	OUTPUT	FORWARD
S3	FORWARD	INPUT
S4	FORWARD	FORWARD

4.2 Experiment methodology

The experimental setup used to evaluate TranSteg's performance is presented in Fig. 9. The experimental environment was a controlled LAN network, so that no RTP packets were lost or excessively delayed. There were no network-related or endpoint-related interferences. Two hosts (A and B) took part in this experiment, both of them working under Linux Ubuntu 10.10 (kernel 2.6.35). On host A a soft-phone and TranSteg SS were launched, while on host B a soft-phone and TranSteg SR (Fig. 9).

RTP packets were exchanged between the soft-phones (Ekiga 3.2.7 [8] and Linphone 3.3.2 [25] soft-phones were used in the tests). Both of them were configured to encode users' voice with the G.711 codec. After the generation of RTP packets at the originating user's soft-phone, they are taken by Netfilter module to the TranSteg SS application. Here they are being transcoded to G.726 and a steganogram is added. The transcoding is performed with the aid of Sun's CCITT implementation published on General Public License. RTP packet payload coded with G.711 is of the size 160 bytes. Post the transcoding to G.726, voice payload size decreases to 80 bytes. As mentioned earlier, TranSteg intentionally does not change the RTP payload field size after the transcoding. That is why the rest of the unused 80 bytes can be allocated for a steganogram. In the developed implementation, the steganogram is inserted into RTP packets from a user selected file. After changing the RTP payload, the application recalculates UDP checksum. Then the modified packet is returned to the Netfilter module and sent through outgoing network interface. Ethernet frame recalculation is done automatically before RTP packets are sent via the network interface. Then, after traversing the network, RTP packets reach host B. Next, they are redirected by the Netfilter module to TranSteg SR responsible for the extraction of the steganogram and transcoding the voice payload from G.726 back to G.711. It also inserts the transcoded voice into RTP payload fields and recalculates UDP checksums. Afterwards, the RTP packets are sent to soft-phone application for user conversation reconstruction.

Four TranSteg characteristics were measured with the aid of the above testbed (Fig. 10):

- Steganographic bandwidth (expressed in kbit/s),
- Influence on the call quality (expressed in MOS scale),
- Introduced delays (expressed in ms),
- Distribution of the byte values in RTP packets' payload.

From abovementioned characteristics TranSteg's call quality degradation and introduced delays express the steganographic cost it introduces. Distribution of byte values allows to assess TranSteg undetectability.

It was experimentally verified that the average call duration for IP telephony falls in the range of 7–11 min [14]. Thus, the tests for obtaining the values of steganographic bandwidth and latency were obtained for call duration of 9 min. The 9 min representation was chosen to

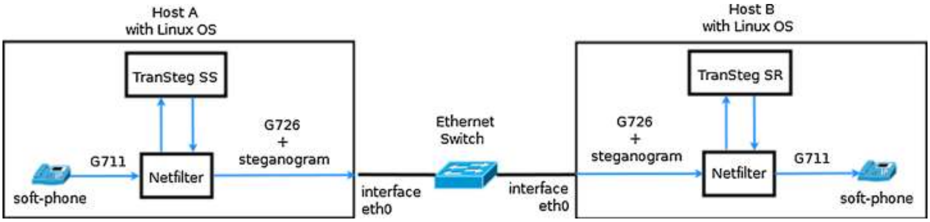

Fig. 9 TranSteg experimental test-bed

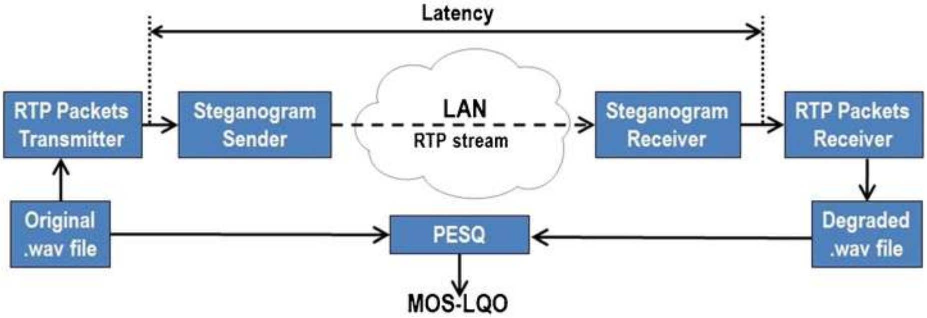

Fig. 10 TranSteg experimental setup

show how much secret data can be sent in one direction during a typical IP telephony call. Each experiment was repeated 20 times and the average results are presented.

Call quality may be expressed in terms of subjective and objective quality measures. Objective measures are usually based on algorithms such as the E-Model [19], PAMS or PESQ [21] or others [24], [49]. In our analysis we shall use the subjective measure MOS (Mean Opinion Score) [20] calculated with the PESQ method. In our experiments, we used audio recordings from the TIMIT [12] continuous speech corpus - one of the most widely used corpora in the speech recognition community. Based on these recordings the voice packet payload was compiled into seven .wav files. Both male and female voices speaking English were used. Each resultant .wav input file was about 30 s long. The adopted coding involved PCM, 8000 Hz sampling, 16 bit sample representation of monophonic signal. It was ensured that the setup conformed to ITU-T P.862.3 recommendation [22] requirements which guarantees proper functioning of the PESQ method. Every .wav input file part was then inserted into the payloads of consecutive RTP packets and sent to the receiver where reassembling into a .wav file was performed. Then, the original and degraded files were compared with the use of PESQ and the resultant MOS-LQO (MOS-Listening Quality, Objective) was returned.

Besides MOS values the TranSteg imperceptibility was also verified by computing the DSNR (Difference in Signal to Noise Ratio) values. The DSNR is defined after Huang et al. [16] as the difference in SNR between the original speech (without steganography) and when the TranSteg was applied to VoIP call.

Latency was measured with the aid of the MGEN 5.02 tool [42]. To achieve this goal MGEN utilizes NTP (Network Time Protocol) 4.2.6 [33], implementation v. 4.2.6 [36]. After the synchronization of two hosts: A and B (Fig. 11) latency measurements were performed.

MGEN client and server were running on host A and B respectively. MGEN enables latency measurements by saving packet sending time in the packet payload. The MGEN client was sending a UDP stream, whose characteristic was chosen to exactly match RTP stream with G.711 payload (50 packets per second, 160 bytes of voice payload). The MGEN-generated packets were sent to the Netfilter module and then to TranSteg Steganogram Sender. In this module all of the TranSteg-related operations i.e. the transcoding of voice from G.711 do G.726 and the insertion of steganogram were performed, but packet payload was intentionally *not* changed. It was necessary to leave the original UDP payload because otherwise it would be impossible to extract packets' sending time. Next, packets were sent to host B. There they were transferred by the Netfilter module to the TranSteg

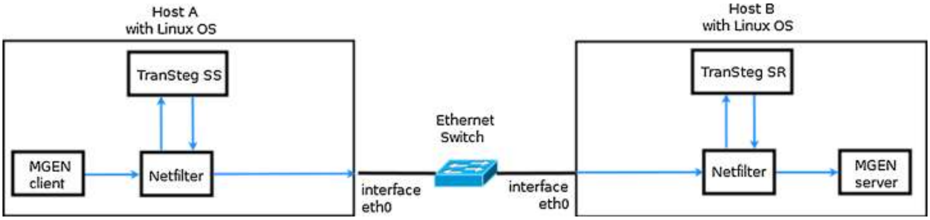

Fig. 11 Latency measurement

Steganogram Receiver. In the SR all necessary operations i.e. transcoding from G.726 to G.711 and steganogram extraction were performed, and once again the payload was not changed. Then, when packets reached the MGEN server it would then generate a report about their sending and receiving time. Basing on the information from the MGEN report TRPR (TRace Plot Real-time) 2.1b2 tool [43] latency of each packet was calculated. Additionally the delay introduced by SS and SR were measured independently and utilized similar methodology as for latency measurement.

4.3 Experimental results

The obtained experimental results are presented in Tables 3 and 4 and in Figs. 12, 13, 14, 15 and 16. As mentioned in the previous subsection, TranSteg had been investigated for the hidden communication scenario S4 from Fig. 6.

4.3.1 Steganographic bandwidth

Steganographic bandwidth for every performed call was identical and equalled 32 kbit/s. The explanation of this lies in the utilized codecs. For the overt codec – G.711 – the resulting payload is 160 bytes and for the covert codec – G.726 (32 kbit/s) – it is 80 bytes. Thus, when using 80 bytes for steganogram in every RTP packet (and 50 packets are generated every second of the call), then, during the whole, typical IP telephony connection it is possible to transfer about 2.2 MB (in each direction). This must be considered as a high steganographic bandwidth when compared with other existing VoIP steganographic methods.

4.3.2 Steganographic cost

The voice quality results (Table 3) show that the average obtained result 3.83 (in MOS scale), is similar to the voice quality offered by G.726 (it is about 3.85) and

Table 3 TranSteg voice quality results

Wav file number	1	2	3	4	5	6	7
Reference MOS				4.46			
MOS-LQO	4.013	3.908	3.687	3.657	3.709	3.715	4.149
Average MOS-LQO	3.834 (with standard deviation 0.18)						
DSNR [dB]	0.21	0.25	0.45	0.46	0.34	0.33	0.15
Average DSNR [dB]	0.31 (with standard deviation 0.11)						

Table 4 TranSteg latency results

	With TranSteg	Without TranSteg	Difference
Average Latency [ms]	1.24	0.85	0.39
Standard Deviation [ms]	0.32	0.07	-

resulting DSNR so small (about 0.3 dB on average) that the influence of TranSteg on VoIP call is hard to perceive. Thus, the resulting voice quality is lower than originally but it is still considered as good – the change is almost imperceptible for an average IP telephony user.

Table 4 presents delays introduced by TranSteg. The measured latency difference between calls with and without TranSteg turned out to be about 0.4 ms. This signifies that TranSteg does not introduce significant delays that could seriously affect voice quality. The exemplary latency results for VoIP calls with and without TranSteg are presented in Fig. 12. It is worth noting that latency was measured for the worst case scenario (from the point of view of the introduced delays) where two transcoding operations took place (first at SS, and then the second at SR). For the other hidden communication scenarios from Fig. 4 the introduced delays would be even lower. However, it must be emphasised that for a different pair of codecs the results would differ as both codecs: G.711 and G.726, are of low computational complexity.

Additionally, delays introduced solely at SS and SR were measured. Obtained experimental results show that SS is responsible for about 0.24 ms (with standard deviation 0.1 ms) and SR for about 0.13 ms (with standard deviation 0.05 ms).

Maximum acceptable one-way latency recommended by ITU-T G.114 is 150 ms [18]. It means that applying TranSteg to VoIP call adds only 0.3 % to total latency. Thus TranSteg allows making real-time processing possible.

4.3.3 TranSteg detection

Distribution of the byte values in RTP packet's payload was also investigated. This was done to verify how much the transcoding operations and the addition of the steganogram change the voice payload. This knowledge can be later utilized to aid the development of TranSteg detection method (see Sec. 3.3).

Figure 13 presents the byte values' distribution for G.711 encoded speech before it reaches SS, i.e. prior to transcoding to G. 726, and after this operation (after leaving SS).

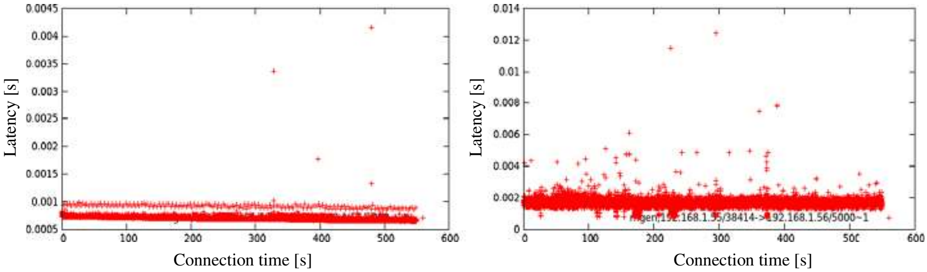

Fig. 12 Latency results for one exemplary IP telephony connection without (left) and with (right) TranSteg

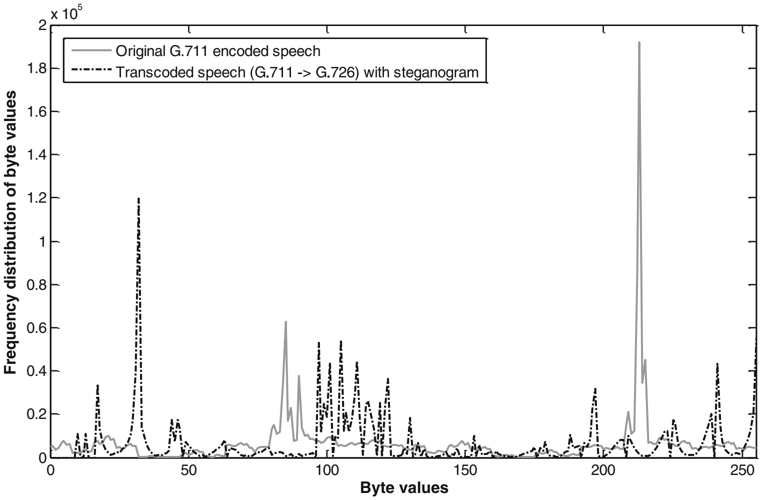

Fig. 13 Frequency distribution of byte values for G.711 encoded speech (before reaching SS) and transcoded to G.726 (after leaving SS)

As expected, there is a significant difference between the two presented curves. Thus, if the warden is able to monitor RTP traffic in the two networks (1 and 2 in Fig. 6), then the suspicious IP telephony connections can be discovered.

However, it must be emphasised, that in the other hidden communication scenarios from Fig. 4 (S1-3), when the SRTP protocol is utilized for conversation security, the byte frequency distribution in RTP payloads before and after transcoding will be similar (due to SRTP encryption) and, thus, TranSteg will be difficult to detect.

Figure 14 illustrates frequency distribution of byte values for the case where speech is transcoded at SS to G.726 – prior and post to the embedding of the steganogram. This

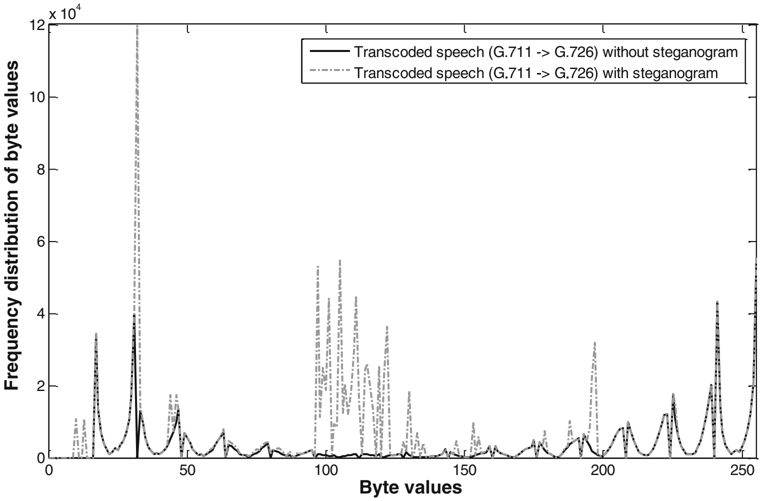

Fig. 14 Frequency distribution of byte values for speech transcoded to G.726 (at SS) without a steganogram and with a steganogram (after leaving SS)

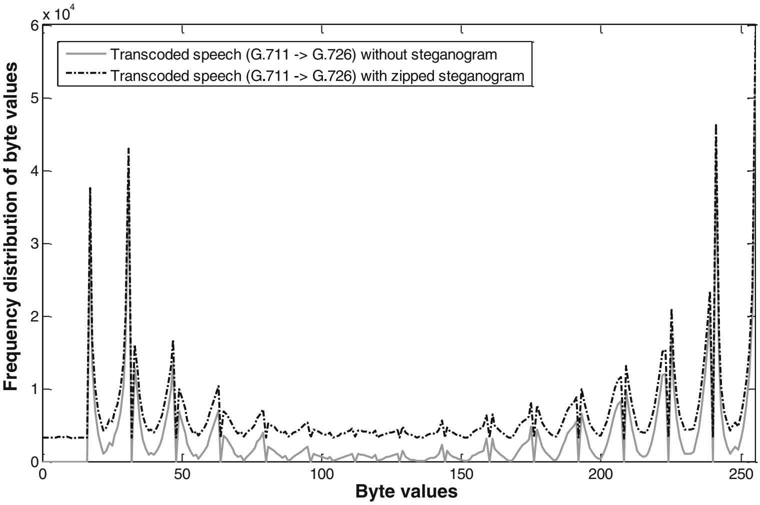

Fig. 15 Frequency distribution of byte values for speech transcoded to G.726 (at SS) without steganogram and with zipped steganogram (after leaving SS)

diagram shows the sole influence of this operation, without the transcoding, of the voice payload. As can be seen, the two curve shapes are significantly different.

In Fig. 15 it is presented how this situation can be changed when the same steganogram is compressed using zip algorithm. In this case, the two curve shapes are very similar. The compression operation leads to the randomization of the steganogram's bytes. This leads to an even shift in the observed frequency distribution of byte values.

Finally, let us analyze for an originally G.711 encoded speech i.e. at a point prior to reaching SS and how frequency distribution of byte values changes past the inverse transcoding to G.711, i.e. after leaving the SR. The obtained results are illustrated in Fig. 16.

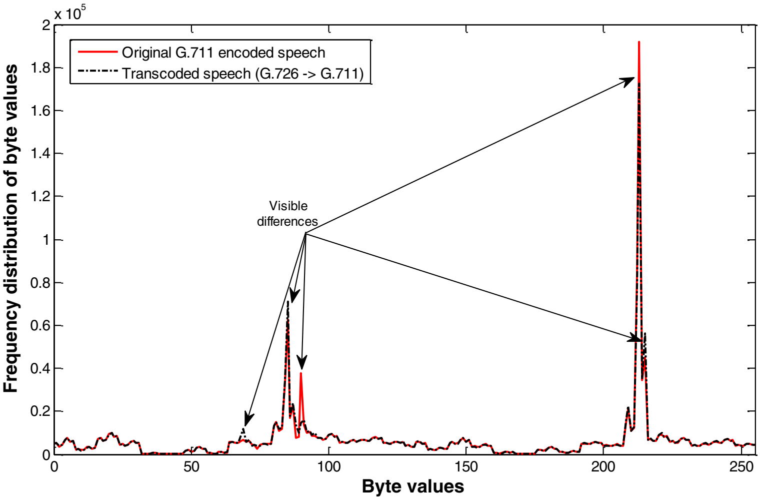

Fig. 16 Frequency distribution of byte values for G.711 encoded speech (before reaching SS) and transcoded to G.711 (after leaving SR)

Careful analysis of this figure leads to the conclusion that there are only slight differences between the G.711 encoded speech at the transmitter as compared with the received one (marked with arrows in Fig. 16). This means that, if the warden is able to monitor RTP traffic in two distinct localizations: in the overt transmitter's and the overt receiver's LANs (Local Area Networks), then TranSteg utilization could still remain unnoticed. The differences in shapes of the presented curves can be explained, to some extent, as introduced by the network, e.g. due to packet losses or transmission errors. The latter are present as the underlying transport protocol in the TCP/IP stack that is used is the unreliable UDP protocol. However, the presence of major differences can indicate TranSteg utilization.

The presented experimental results confirm the conclusions from Sec. 3.3. A warden capable of monitoring traffic in more than single network localization is very likely to detect the presence of TranSteg. However, one thing must be emphasised: detection of *all* of the existing VoIP steganographic methods presented in related work section is easier if used in scenario S4 and the warden is capable of monitoring traffic in more than single network localization. In this case some of the methods are even trivial to detect. In this context TranSteg is superior to other methods which will be proved below.

It must be also noted, that if any other scenario is utilized, e.g. the S1 scenario, together with SRTP encryption, then the disclosure of TranSteg can be very hard to attain.

However, even when considering the scenario S4, there is a simple way to obstruct the detection process, especially for the situation illustrated in Fig. 16. The proposed solution is to encode steganogram's bits until the original and transcoded speech's curves, at the receiver, are the same. Such an approach will surely limit the potential available steganographic bandwidth. At the same time, with the original and transcoded byte values' frequency distribution curves looking exactly the same, the disclosure of hidden communication shall be impossible.

To summarize, the detection of the TranSteg method is not trivial, especially for the hidden communication scenarios S1-3. Even for scenario S4 some simple measures can be taken to improve undetectability. Simple analysis of the frequency distribution of byte values in RTP payload as shown above may not be sufficient. Thus, as stated in Sec. 3.3, other possibilities leading to the development of an efficient detection method (one that would fulfil VoIP's real-time constraints) must be investigated.

Analysis of TranSteg experimental results presented above indicates that it is an efficient steganographic method. Comparison of TranSteg with state of the art data hiding approaches for VoIP like LSB (Least Significant Bits) [41] or LACK (Lost Audio Packets Steganography) [27] is showed in Table 5. It turns out that TranSteg achieves impressive steganographic bandwidth of 32 kbit/s. To authors' best knowledge this is the highest steganographic bandwidth available to be utilized in VoIP environment. The difference in bandwidth for these three steganographic methods results from the fact that in TranSteg every packet's payload from RTP stream is utilized to insert secret data while LACK uses only a certain number of them. More RTP packets could be used for LACK, however, it

Table 5 TranSteg comparison with existing VoIP steganography methods for G.711 speech codec

	Steganographic bandwidth [bit/s]	Voice degradation [MOS]	Detection
LSB [41]	3 567.2	0.75	Easy
LACK [27]	1 236.5	0.83	Medium
TranSteg	32 000	0.63	Hard

would degrade the call quality. For the same reason LSB could not take more least significant bits to increase its steganographic bandwidth.

TranSteg also introduces the least voice degradation while limiting the chance of disclosure. For hidden communication scenario S4 TranSteg will be the least detectable method. It is because after the steganogram reaches SR it is extracted and speech data practically the same to originally sent ones is restored (by performing transcoding). Thus, all evidences of steganogram are wiped out. However, when LSB or LACK are utilized the steganogram can be extracted and removed but original data cannot be restored because it was erased at the SS. That is why when the warden inspects traffic in two different locations such methods are easier to detect.

5 Conclusions and future work

In this paper, a new IP telephony steganographic method, named TranSteg, was introduced. It was described basing on the possible hidden communication scenarios (S1-4 from Fig. 4). It was shown that the scenario where steganogram sender and receiver are the original source and final destination of the RTP traffic, respectively, is the most advantageous from the point of view of the achievable steganographic bandwidth, introduced steganographic cost and the undetectability of the method.

TranSteg proof of concept implementation was also designed and developed for a worst case scenario, where the introduced delays are largest. The obtained experimental results, for G.711 as an overt and G.726 as a covert codec, proved that the proposed method is feasible and offers a high steganographic bandwidth up to 32 kbit/s while introducing delays lower than 1 ms, and still retaining good voice quality (about 3.8 in MOS scale). To authors' best knowledge this is the highest steganographic bandwidth available to be utilized in VoIP environment.

Detection of TranSteg strongly depends on the realized hidden communication scenario and the capabilities of a warden responsible for network steganography detection (e.g. the locations where it can monitor VoIP traffic). Generally, TranSteg detection can be difficult to perform, especially, if the SRTP protocol is utilized for securing RTP streams. Detection is also impeded when the warden is able to inspect traffic only in a single network localization.

Moreover, it must be emphasized that in TranSteg after the steganogram reaches the receiver and the hidden information is extracted and the speech data is practically restored to the originally sent one (by means of transcoding). Thus, all evidences of steganogram are wiped out. However, when other VoIP steganographic methods are utilized then the hidden data can be extracted and removed but original data cannot be restored because it was previously erased due to hidden data insertion process.

Future work should involve an in depth analysis of speech codec pairs (overt and covert) that would be most advantageous for TranSteg. The algorithm for the selection of the covert codec will be developed, with the consideration of assuring an acceptable voice quality, low introduced delays and different VoIP codecs' characteristic features. Moreover, a prototype implementation should be developed for an end-to-end hidden communication scenario (S1 from Fig. 4) with the SRTP capability. This will allow the pursuing of an efficient, real-time, TranSteg detection method. On the other hand, to enhance the undetectability of TranSteg the different mechanisms of spreading the steganogram over the voice data instead of filling them in the end of the payload will be analysed in more detail. Additionally, detection methods that proved to be successful for digital images based on identifying double-compression [37] or compression artefacts should be considered and evaluated in TranSteg context. However, it must be noted that when for TranSteg scenario S1 (Fig. 4) is utilized together with SRTP voice stream encryption then utilization of such techniques for TranSteg detection will likely fail.

Acknowledgments This work was supported by the Polish Ministry of Science and Higher Education and Polish National Science Centre under grants: 0349/IP2/2011/71 and 2011/01/D/ST7/05054.

The authors would like to thank Elżbieta Zielińska from Warsaw University of Technology (Poland) for helpful comments and remarks.

References

1. Ahsan K, Kundur D (2002) Practical Data Hiding in TCP/IP. In: Proc. of: Workshop on Multimedia Security at ACM Multimedia 2002, Juan-les-Pins, France
2. Aoki N (2008) A Technique of Lossless Steganography for G.711 Telephony Speech, International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2008), pp. 608 – 611, Harbin, China, 15–17
3. Bai LY, Huang Y, Hou G, Xiao B (2008) Covert Channels Based on Jitter Field of the RTCP Header, International Conference on Intelligent Information Hiding and Multimedia Signal Processing
4. Baugher M, Casner S, Frederick R, Jacobson V (2004) The Secure Real-time Transport Protocol (SRTP), RFC 3711
5. Bender W, Gruhl D, Morimoto N, Lu A (1996) Techniques for Data Hiding. IBM System J 35(3,4):313–336
6. Christopher R. Forbes, A New Covert Channel over RTP, MSc thesis, Rochester Institute of Technology, URL: https://ritdml.rit.edu/bitstream/handle/1850/12883/CForbesThesis8-21-2009.pdf?sequence=1
7. Dittmann J, Hesse D, Hillert R (2005) Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set. Proc SPIE 5681:607–618, Security, Steganography, and Watermarking of Multimedia Contents VII, San Jose
8. Ekiga soft-phone, URL: http://ekiga.org/
9. Fisk G, Fisk M, Papadopoulos C, Neil J (2002) Eliminating steganography in Internet traffic with active wardens, 5th International Workshop on Information Hiding. Lecture Notes in Computer Sci 2578:18–35
10. Frączek W, Mazurczyk W, Szczypiorski K, Multi-Level Steganography: Improving Hidden Communication in Networks - accepted for publication in Journal of Universal Computer Science (J. UCS), ISSN: 0948-695X, Graz Univ. Technology, Inst. Information Systems Computer Media-IICM (accepted for publication)
11. Fridrich J, Goljan M, Du R (2001) Invertible Authentication Watermark for JPEG Images, ITCC 2001, Las Vegas, Nevada, pp 223–227
12. Garofolo JS, et al (1993) TIMIT Acoustic-Phonetic Continuous Speech Corpus Linguistic Data Consortium, Philadelphia
13. GTK + (GIMP Toolkit) documentation, URL: http://www.gtk.org
14. Guha S, Daswani N, Jain R (2006) An experimental study of the Skype peer-to-peer VoIP system. Sixth International Workshop on Peer-to-Peer Systems (IPTPS)
15. Hamdaqa M, Tahvildari L, ReLACK: A Reliable VoIP Steganography Approach, In Proc. of Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2011), Korea, 2011, pp 189–197
16. Huang Y, Tang S, Yuan J (2011) Steganography in Inactive Frames of VoIP Streams Encoded by Source Codec. IEEE Trans on Inf Forensics and Secur 6(2):296–306
17. I)ruid, Real-time Steganography with RTP, Technical Report, September, 2007 URL: http://www.uninformed.org/?v=8&a=3&t=pdf
18. ITU-T G.114, One-way Transmission Time, SERIES G: Transmission Systems and Media, Digital System and Networks, ITU, Geneva, Switzerland, 2003
19. ITU-T, Recommendation G. 107, The E-Model, a computational model for use in transmission planning, 2002
20. ITU-T, Recommendation. P.800, Methods for subjective determination of transmission quality, 1996
21. ITU-T, Recommendation. P.862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, 2001

22. ITU-T, Recommendation. P.862.3, Application guide for objective quality measurement based on Recommendations P.862, P.862.1 and P.862.2, November 2007
23. Krätzer C, Dittmann J, Vogel T, Hillert R (2006) Design and Evaluation of Steganography for Voice-over-IP, In Proc. of IEEE International Symposium on Circuits and Systems, (ISCAS) 2006, Kos, Greece
24. Lee W, Lee M, McGowan J (2009) Enhancing objective evaluation of speech quality algorithm: current efforts, limitations and future directions. Eur Trans Telecommun 20(6):594–603
25. Linphone soft-phone, URL: http://www.linphone.org/
26. Lubacz J, Mazurczyk W, Szczypiorski K (2010) Vice over IP (invited paper), In: IEEE Spectrum, ISSN: 0018–9235, pp 40–45
27. Mazurczyk W, Lost Audio Packets Steganography: A First Practical Evaluation, International Journal of Security and Communication Networks, John Wiley & Sons, ISSN: 1939–0114 (in press), doi: 10.1002/sec.502
28. Mazurczyk W, Kotulski Z (2006) New security and control protocol for VoIP based on steganography and digital watermarking, In Proc. of 5th International Conference on Computer Science - Research and Applications (IBIZA 2006), Poland, Kazimierz Dolny 9–11
29. Mazurczyk W, Lubacz J (2010) LACK – a VoIP Steganographic Method, In: Telecommunication Systems: Modelling, Analysis, Design and Management, Vol. 45, Numbers 2–3, ISSN: 1018–4864 (print version), ISSN: 1572–9451 (electronic version), Springer US, Journal no. 1123
30. Mazurczyk W, Szczypiorski S (2008) Covert Channels in SIP for VoIP signalling, In: Hamid Jahankhani, Kenneth Revett, and Dominic Palmer-Brown (Eds.): ICGeS 2008 - Communications in Computer and Information Science (CCIS) 12, Springer Verlag Berlin Heidelberg, Proc. of 4th International Conference on Global E-security 2008, London, United Kingdom, 23–25, pp 65–70
31. Mazurczyk W, Szczypiorski S (2008) Steganography of VoIP Streams, In: Robert Meersman and Zahir Tari (Eds.): OTM 2008, Part II - Lecture Notes in Computer Science (LNCS) 5332, Springer-Verlag Berlin Heidelberg, Proc. of OnTheMove Federated Conferences and Workshops: The 3rd International Symposium on Information Security (IS'08), Monterrey, Mexico, November 9–14, pp 1001–1018
32. Miao R, Huang Y (2011) An Approach of Covert Communication Based on the Adaptive Steganography Scheme on Voice over IP, Communications, IEEE International Conference on (ICC 2011)
33. Mills D, Delaware U, Martin J, Burbank J, Kasch W (2010) Network Time Protocol Version 4: Protocol and Algorithms Specification, IETF RFC 5905
34. Murdoch, S., Lewis, S.: Embedding Covert Channels into TCP/IP. Information Hiding, 247–266 (2005)
35. Netfilter framework documentation, URL: http://www.netfilter.org
36. NTP: The Network Time Protocol documentation, URL: http://ntp.org
37. Pevny T, Fridrich J (2008) Detection of Double-Compression for Applications in Steganography. IEEE Trans on Inf Secur and Forensic 3(2):247–258
38. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A (2002) SIP: Session Initiation Protocol, IETF, RFC 3261
39. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RTP: A Transport Protocol for Real-Time Applications, IETF, RFC 3550
40. Simmons GJ (1994) Subliminal channels; past and present. Eur Trans Telecommun 5(4):459–474
41. Takahashi T, Lee W (2007) An Assessment of VoIP Covert Channel Threats. In: Proc. of 3rd International Conference on Security and Privacy in Communication Networks (SecureComm 2007), Nice, France
42. The Multi-Generator MGEN documentation, URL: http://cs.itd.nrl.navy.mil/work/mgen/
43. The TRace Plot Real-time (TRPR) documentation, URL: http://pf.itd.nrl.navy.mil/proteantools/trpr.html
44. Tian H, Zhou K, Jiang H, Liu J, Huang Y, Feng D (2009) An adaptive steganography scheme for voice over IP, IEEE International Symposium on Circuits and Systems (ISCAS 2009), Taipei, Taiwan, 24–27
45. Tian H, Zhou K, Jiang H, Liu J, Huang Y, Feng D (2009) An M-Sequence Based Steganography Model for Voice over IP, IEEE International Conference on Communications (ICC 2009), pp. 1–5, Dresden, Germany, 14–18
46. Wang C, Wu W (2007) Information Hiding in Real-Time VoIP Streams, Ninth IEEE International Symposium on Multimedia (ISM 2007), pp. 255 – 262, Taichung, Taiwan, 10–12
47. Wright C, Ballard L, Coulls S, Monrose F, Masson G (2008) Spot me if you can: recovering spoken phrases in encrypted VoIP conversations. In Proceedings of IEEE Symposium on Security and Privacy
48. Zander S, Armitage G, Branch P (2007) A Survey of Covert Channels and Countermeasures in Computer Network Protocols, IEEE Communications Surveys & Tutorials, 3rd Quarter 2007, Volume: 9, Issue: 3, pp. 44–57, ISSN: 1553-877X
49. Zhou Y, Chan W (2007) E-model based comparison of multiple description coding and layered coding in packet networks. Eur Trans Telecommun 18(7):661–668

Wojciech Mazurczyk holds an M.Sc. (2004) and a Ph.D. (2009, with honours) in telecommunication both from Faculty of Electronics and Information Technology, WUT; assistant professor at WUT; author of over 80 scientific papers, 2 patent application and 30 invited talks on information security and telecommunications; main research interests: information hiding techniques, network anomalies detection, digital forensics, network security and multimedia services. Research co-leader of Network Security Group (secgroup.pl). A TPC member of refereed conferences, including IEEE MASCOTS, IEEE ICCT, IEEE ICC and ACSAC. He also serves as the reviewer of refereed international magazines and journals among others (from master journal list): IEEE Transactions on Information Forensics and Security, Computer Communications, IET Information Security and Multimedia Tools and Applications. Personal website: http://mazurczyk.com.

Paweł Szaga has been studying telecommunications at Warsaw University of Technology (WUT, Poland) since 2007. His main areas of interest are network security and information hiding. Member of the Network Security Group at WUT (secgroup.pl).

Krzysztof Szczypiorski holds M.Sc. (1997, with honours), Ph.D. (2007, with honours) and D.Sc. (habilitation, 2012) in telecommunications from Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT). Associate Professor at WUT. Founder and head of International Telecommunication Union Internet Training Centre (ITUITC) established in 2003. Research leader of Network Security Group at WUT (secgroup.pl). His research interests include: network security, steganography and wireless networks. He is the author or the co-author of 160+ publications including 110 papers and 50+ invited talks. He is the inventor of 3 patents and pending applications. For almost 20 years he also serves as the independent consultant in fields of network security and telecommunications. Personal website: http://ksz.tele.pw.edu.pl.