Internet Engineering Task Force (IETF)
Request for Comments: 6366
Category: Informational
ISSN: 2070-1721
J. Valin
Mozilla
K. Vos
Skype Technologies, S.A.
August 2011

Requirements for an Internet Audio Codec

Abstract

This document provides specific requirements for an Internet audio codec. These requirements address quality, sampling rate, bit-rate, and packet-loss robustness, as well as other desirable properties.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6366.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

This document provides requirements for an audio codec designed specifically for use over the Internet. The requirements attempt to address the needs of the most common Internet interactive audio transmission applications and ensure good quality when operating in conditions that are typical for the Internet. These requirements also address the quality, sampling rate, delay, bit-rate, and packet-loss robustness. Other desirable codec properties are considered as well.

2. Definitions

Throughout this document, the following conventions refer to the sampling rate of a signal:

Narrowband: 8 kilohertz (kHz)

Wideband: 16 kHz

Super-wideband: 24/32 kHz

Full-band: 44.1/48 kHz

Codec bit-rates in bits per second (bit/s) will be considered without counting any overhead ((IP/UDP/RTP) headers, padding, etc.). The codec delay is the total algorithmic delay when one adds the codec frame size to the "look-ahead". Thus, it is the minimum theoretically achievable end-to-end delay of a transmission system that uses the codec.

3. Applications

The following applications should be considered for Internet audio codecs, along with their requirements:

- Point-to-point calls
- Conferencing
- Telepresence
- Teleoperation
- In-game voice chat
- Live distributed music performances / Internet music lessons
- Delay-tolerant networking or push-to-talk services
- Other applications

3.1. Point-to-Point Calls

Point-to-point calls are voice over IP (VoIP) calls from two "standard" (fixed or mobile) phones, and implemented in hardware or software. For these applications, a wideband codec is required, along with narrowband support for compatibility with a public switched telephone network (PSTN). It is expected for the range of useful bit-rates to be 12 - 32 kilobits per second (kbit/s) for wideband speech and 8 - 16 kbit/s for narrowband speech. The codec delay must be less than 40 milliseconds (ms), but no more than 25 ms is desirable. Support for encoding music is not required, but it is desirable for the codec not to make background (on-hold) music excessively unpleasant to hear. Also, the codec should be robust to noise (produce intelligible speech and no annoying artifacts) even at lower bit-rates.

3.2. Conferencing

Conferencing applications (that support multi-party calls) have additional requirements on top of the requirements for point-to-point calls. Conferencing systems often have higher-fidelity audio equipment and have greater network bandwidth available -- especially when video transmission is involved. Therefore, support for super-wideband audio becomes important, with useful bit-rates in the 32 - 64 kbit/s range. The ability to vary the bit-rate, according to the "difficulty" of the audio signal, is a desirable feature for the codec. This not only saves bandwidth "on average", but it can also help conference servers make more efficient use of the available bandwidth, by using more bandwidth for important audio streams and less bandwidth for less important ones (e.g., background noise).

Conferencing end-points often operate in hands-free conditions, which creates acoustic echo problems. Therefore, lower delay is important, as it reduces the quality degradation due to any residual echo after acoustic echo cancellation (AEC). Consequently, the codec delay must be less than 30 ms for this application. An optional low-delay mode with less than 10 ms delay is desirable, but not required.

Most conferencing systems operate with a bridge that mixes some (or all) of the audio streams and sends them back to all the participants. In that case, it is important that the codec not produce annoying artifacts when two voices are present at the same time. Also, this mixing operation should be as easy as possible to perform. To make it easier to determine which streams have to be mixed (and which are noise/silence), it must be possible to measure (or estimate) the voice activity in a packet without having to fully decode the packet (saving most of the complexity when the packet need not be decoded). Also, the ability to save on the computational complexity when mixing is also desirable, but not required. For example, a transform codec may make it possible to mix the streams in the transform domain, without having to go back to time-domain. Low-complexity up-sampling and down-sampling within the codec is also a desirable feature when mixing streams with different sampling rates.

3.3. Telepresence

Most telepresence applications can be considered to be essentially very high-quality video-conferencing environments, so all of the conferencing requirements also apply to telepresence. In addition, telepresence applications require super-wideband and full-band audio capability with useful bit-rates in the 32 - 80 kbit/s range. While voice is still the most important signal to be encoded, it must be possible to obtain good quality (even if not transparent) music.

Most telepresence applications require more than one audio channel, so support for stereo and multi-channel is important. While this can always be accomplished by encoding multiple single-channel streams, it is preferable to take advantage of the redundancy that exists between channels.

3.4. Teleoperation and Remote Software Services

Teleoperation applications are similar to telepresence, with the exception that they involve remote physical interactions. For example, the user may be controlling a robot while receiving real-time audio feedback from that robot. For these applications, the delay has to be less than 10 ms. The other requirements of telepresence (quality, bit-rate, multi-channel) apply to teleoperation as well. The only exception is that mixing is not an important issue for teleoperation.

The requirements for remote software services are similar to those of teleoperation. These applications include remote desktop applications, remote virtualization, and interactive media application being rendered remotely (e.g., video games rendered on central servers). For all these applications, full-band audio with an algorithmic delay below 10 ms are important.

3.5. In-Game Voice Chat

An increasing number of computer/console games make use of VoIP to allow players to communicate in real time. The requirements for gaming are similar to those of conferencing, with the main difference being that narrowband compatibility is not necessary. While for most applications a codec delay up to 30 ms is acceptable, a low-delay (< 10 ms) option is highly desirable, especially for games with rapid interactions. The ability to use variable bit-rate (VBR) (with a maximum allowed bit-rate) is also highly desirable because it can significantly reduce the bandwidth requirement for a game server.

3.6. Live Distributed Music Performances / Internet Music Lessons

Live music over the Internet requires extremely low end-to-end delay and is one of the most demanding applications for interactive audio transmission. It has been observed that for most scenarios, total end-to-end delays up to 25 ms could be tolerated by musicians, with the absolute limit (where none of the scenarios are possible) being around 50 ms [carot09]. In order to achieve this low delay on the Internet -- either in the same city or in a nearby city -- the network propagation time must be taken into account. When also subtracting the delay of the audio buffer, jitter buffer, and acoustic path, that leaves around 2 ms to 10 ms for the total delay of the codec. Considering the speed of light in fiber, every 1 ms reduction in the codec delay increases the range over which synchronization is possible by approximately 200 km.

Acoustic echo is expected to be an even more important issue for network music than it is in conferencing, especially considering that the music quality requirements essentially forbid the use of a "non-linear processor" (NLP) with AEC. This is another reason why very low delay is essential.

Considering that the application is music, the full audio bandwidth (44.1 or 48 kHz sampling rate) must be transmitted with a bit-rate that is sufficient to provide near-transparent to transparent quality. With the current audio coding technology, this corresponds to approximately 64 kbit/s to 128 kbit/s per channel. As for telepresence, support for two or more channels is often desired, so it would be useful for a codec to be able to take advantage of the redundancy that is often present between audio channels.

3.7. Delay-Tolerant Networking or Push-to-Talk Services

Internet transmissions are subjected to interruptions of connectivity that severely disturb a phone call. This may happen in cases of route changes, handovers, slow fading, or device failures. To overcome this distortion, the phone call can be halted and resumed after the connectivity has been reestablished again.

Also, if transmission capacity is lower than the minimal coding rate, switching to a push-to-talk mode still allows for effective communication. In this situation, voice is transmitted at slower-than-real-time bit-rate and conversations are interrupted until the speech has been transmitted.

These modes require interrupting the audio playout and continuing after a pause of arbitrary duration.

3.8. Other Applications

The above list is by no means a complete list of all applications involving interactive audio transmission on the Internet. However, it is believed that meeting the needs of all these different applications should be sufficient to ensure that the needs of other applications not listed will also be met.

4. Constraints Imposed by the Internet on the Codec

Packet losses are inevitable on the Internet, and dealing with them is one of the most fundamental requirements for an Internet audio codec. While any audio codec can be combined with a good packet-loss concealment (PLC) algorithm, the important aspect is what happens on the first packets received _after_ the loss. More specifically, this means that:

- it should be possible to interpret the contents of any received packet, irrespective of previous losses as specified in BCP 36 [PAYLOADS]; and

- the decoder should re-synchronize as quickly as possible (i.e., the output should quickly converge to the output that would have been obtained if no loss had occurred).

The constraint of being able to decode any packet implies the following considerations for an audio codec:

- The size of a compressed frame must be kept smaller than the MTU to avoid fragmentation;

- The interpretation of any parameter encoded in the bit-stream must not depend on information contained in other packets. For example, it is not acceptable for a codec to allow signaling a mode change in one packet and assume that subsequent frames will be decoded according to that mode.

Although the interpretation of parameters cannot depend on other packets, it is still reasonable to use some amount of prediction across frames, provided that the predictors can resynchronize quickly in case of a lost packet. In this case, it is important to use the best compromise between the gain in coding efficiency and the loss in packet loss robustness due to the use of inter-frame prediction. It is a desirable property for the codec to allow some real-time control of that trade-off, so that it can take advantage of more prediction when the loss rate is small, while being more robust to losses when the loss rate is high.

To improve the robustness to packet loss, it would be desirable for the codec to allow an adaptive (data- and network-dependent) amount of side information to help improve audio quality when losses occur. For example, side information may include the retransmission of certain parameters encoded in the previous frame(s).

To ensure freedom of implementation, decoder-side-only error concealment does not need to be specified, although a functional PLC algorithm is desirable as part of the codec reference implementation. Obviously, any information signaled in the bit-stream intended to aid PLC needs to be specified.

Another important property of the Internet is that it is mostly a best-effort network, with no guaranteed bandwidth. This means that the codec has to be able to vary its output bit-rate dynamically (in real time), without requiring an out-of-band signaling mechanism, and without causing audible artifacts at the bit-rate change boundaries. Additional desirable features are:

- o Having the possibility to use smooth bit-rate changes with one byte/frame resolution;

- o Making it possible for a codec to adapt its bit-rate based on the source signal being encoded (source-controlled VBR) to maximize the quality for a certain _average_ bit-rate.

Because the Internet transmits data in bytes, a codec should produce compressed data in integer numbers of bytes. In general, the codec design should take into consideration explicit congestion notification (ECN) and may include features that would improve the quality of an ECN implementation.

The IETF has defined a set of application-layer protocols to be used for transmitting real-time transport of multimedia data, including voice. Thus, it is important for the resulting codec to be easy to use with these protocols. For example, it must be possible to create an [RTP] payload format that conforms to BCP 36 [PAYLOADS]. If any codec parameters need to be negotiated between end-points, the negotiation should be as easy as possible to carry over session initiation protocol (SIP) [RFC3261]/ session description protocol (SDP) [RFC4566] or alternatively over extensible messaging and presence protocol (XMPP) [RFC6120] / Jingle [XEP-0167].

5. Detailed Basic Requirements

This section summarizes all the constraints imposed by the target applications and by the Internet into a set of actual requirements for codec development.

5.1. Operating Space

The operating space for the target applications can be divided in terms of delay: most applications require a "medium delay" (20-30 ms), while a few require a "very low delay" (< 10 ms). It makes sense to divide the space based on delay because lowering the delay has a cost in terms of quality versus bit-rate.

For medium delay, the resulting codec must be able to efficiently operate within the following range of bit-rates (per channel):

- Narrowband: 8 kbit/s to 16 kbit/s
- Wideband: 12 to 32 kbit/s
- Super-wideband: 24 to 64 kbit/s
- Full-band: 32 to 80 kbit/s

Obviously, a lower-delay codec that can operate in the above range is also acceptable.

For very low delay, the resulting codec will need to operate within the following range of bit-rates (per channel):

- Super-wideband: 32 to 80 kbit/s
- Full-band: 48 to 128 kbit/s
- (Narrowband and wideband not required)

5.2. Quality and Bit-Rate

The quality of a codec is directly linked to the bit-rate, so these two must be considered jointly. When comparing the bit-rate of codecs, the overhead of IP/UDP/RTP headers should not be considered, but any additional bits required in the RTP payload format, after the header (e.g., required signaling), should be considered. In terms of quality versus bit-rate, the codec to be developed must be better than the following codecs, that are generally considered royalty-free:

- For narrowband: Speex (NB) [Speex], and internet low bit-rate codec (iLBC)(*) [RFC3951]
- For wideband: Speex (WB) [Speex], G.722.1(*) [ITU.G722.1]
- For super-wideband/fullband: G.722.1C(*) [ITU.G722.1]

The codecs marked with (*) have additional licensing restrictions, but the codec to be developed should still not perform significantly worse. In addition to the quality targets listed above, a desirable objective is for the codec quality to be no worse than Adaptive Multi-Rate (AMR-NB) and Adaptive Multi-Rate Wideband (AMR-WB). Quality should be measured for multiple languages, including tonal languages. The case of multiple simultaneous voices (as sometimes happens in conferencing) should be evaluated as well.

The comparison with the above codecs assumes that the codecs being compared have similar delay characteristics. The bit-rate required, for a certain level of quality, may be higher than the referenced codecs in cases where a much lower delay is required. In that case, the increase in bit-rate must be less than the ratio between the delays.

It is desirable for the codecs to support source-controlled variable bit-rate (VBR) to take advantage of different inputs, that require a different bit-rate, to achieve the same quality. However, it should still be possible to use the codec at a truly constant bit-rate to ensure that no information leak is possible when using an encrypted channel.

5.3. Packet-Loss Robustness

Robustness to packet loss is a very important aspect of any codec to be used on the Internet. Codecs must maintain acceptable quality at loss rates up to 5% and maintain good intelligibility up to 15% loss rate. At any sampling rate, bit-rate, and packet-loss rate, the quality must be no less than the quality obtained with the Speex codec or the Global System for Mobile Communications - Full Rate (GSM-FR) codec in the same conditions. The actual packet-loss "patterns" to be used in testing must be obtained from real packet-loss traces collected on the Internet, rather than from loss models. These traces should be representative of the typical environments in which the applications of Section 3 operate. For example, traces related to VoIP calls should consider the loss patterns observed for typical home broadband and corporate connections.

5.4. Computational Resources

The resulting codec should be implementable on a wide range of devices, so there should be a fixed-point implementation or at least assurance that a reasonable fixed-point is possible. The computational resources figures listed below are meant to be upper bounds. Even below these bounds, resources should still be minimized. Any proposed increase in computational resources consumption (e.g., to increase quality) should be carefully evaluated even if the resulting resource consumption is below the upper bound. Having variable complexity would be useful (but not required) in achieving that goal as it would allow trading quality/bit-rate for lower complexity.

The computational requirements for real-time encoding and decoding of a mono signal on one core of a recent x86 CPU (as measured with the Unix "time" utility or equivalent) are as follows:

- Narrowband: 40 megahertz (MHz) (2% of a 2 gigahertz (GHz) CPU core)

- Wideband: 80 MHz (4% of a 2 GHz CPU core)

- Super-wideband/fullband: 200 MHz (10% of a 2 GHz CPU core)

It is desirable that the MHz values listed above also be achievable on fixed-point digital signal processors that are capable of single-cycle multiply-accumulate operations (16x16 multiplication accumulated into 32 bits).

For applications that require mixing (e.g., conferencing), it should be possible to estimate the energy and/or the voice activity status of the decoded signal with less than 10% of the complexity figures listed above.

It is the intent to maximize the range of devices on which a codec can be implemented. Therefore, the reference implementation must not depend on special hardware features or instructions to be present in order to meet the complexity requirement. However, it may be desirable to take advantage of such hardware when available, (e.g., hardware accelerators for operations like Fast Fourier Transforms (FFT) and convolutions). A codec should also minimize the use of saturating arithmetic so as to be implementable on architectures that do not provide hardware saturation (e.g., ARMv4).

The combined codec size and data read-only memory (ROM) should be small enough not to cause significant implementation problems on typical embedded devices. The codec context/state size required should be no more than 2*R*C bytes in floating-point, where R is the sampling rate and C is the number of channels. For fixed-point, that size should be less than R*C. The scratch space required should also be less than 2*R*C bytes for floating point or less than R*C bytes for fixed-point.

6. Additional Considerations

There are additional features or characteristics that may be desirable under some circumstances, but should not be part of the strict requirements. The benefit of meeting these considerations should be weighted against the associated cost.

6.1. Low-Complexity Audio Mixing

In many applications that require a mixing server (e.g., conferencing, games), it is important to minimize the computational cost of the mixing. As much as possible, it should be possible to perform the mixing with fewer computations than it would take to decode all the streams, mix them, and re-encode the result. Properties that reduce the complexity of the mixing process include:

- o The ability to derive sufficient parameters, such as loudness and/or spectral envelope, for estimating voice activity of a compressed frame without fully decoding that frame;

- o The ability to mix the streams in an intermediate representation (e.g., transform domain), rather than having to fully decode the signals before the mixing;

- o The use of bit-stream layers (Section 6.3) by aggregating a small number of active streams at lower quality.

For conferencing applications, the total complexity of the decoding, voice activity detection (VAD), and mixing should be considered when evaluating proposals.

6.2. Encoder Side Potential for Improvement

In many codecs, it is possible to improve the quality by improving the encoder without breaking compatibility (i.e., without changing the decoder). Potential for improvement varies from one codec to another. It is generally low for pulse code modulation (PCM) or adaptive differential pulse code modulation (ADPCM) codecs and higher for perceptual transform codecs. All things being equal, being able to improve a codec after the bit-stream is a desirable property. However, this should not be done at the expense of quality in the reference encoder. Other potential improvements include signal-adaptive frame size selection and improved discontinuous transmission (DTX) algorithms that take advantage of predicting the decoder sides packet loss concealment (PLC) algorithms.

6.3. Layered Bit-Stream

A layered codec makes it possible to transmit only a certain subset of the bits and still obtain a valid bit-stream with a quality that is equivalent to the quality that would be obtained from encoding at the corresponding rate. While this is not a necessary feature for most applications, it can be desirable for cases where a "mixing server" needs to handle a large number of streams with limited computational resources.

6.4. Partial Redundancy

One possible way of increasing robustness to packet loss is to include partial redundancy within packets. This can be achieved either by including the base layer of the previous frame (for a layered codec) or by transmitting other parameters from the previous frame(s) to assist the PLC algorithm in case of loss. The ability to include partial redundancy for high-loss scenarios is desirable, provided that the feature can be dynamically turned on or off (so that no bandwidth is wasted in case of loss-free transmission).

6.5. Stereo Support

It is highly desirable for the codec to have stereo support. At a minimum, the codec should be able to encode two channels independently without causing significant stereo image artifacts. It is also desirable for the codec to take advantage of the inter-channel redundancy in stereo audio to reduce the bit-rate (for an equivalent quality) of stereo audio compared to coding channels independently.

6.6. Bit Error Robustness

The vast majority of Internet-based applications do not need to be robust to bit errors because packets either arrive unaltered or do not arrive at all. Therefore, the emphasis should be on packet-loss robustness and packet-loss concealment. That being said, often, the extra robustness to bit errors can be achieved at no cost at all (i.e., no increase in size, complexity, or bit-rate; no decrease in quality, or packet-loss robustness, etc.). In those cases, it is useful to make a change that increases the robustness to bit errors. This can be useful for applications that use UDP Lite transmission (e.g., over a wireless LAN). Robustness to packet loss should *never* be sacrificed to achieve higher bit error robustness.

6.7. Time Stretching and Shortening

When adaptive jitter buffers are used, it is often necessary to stretch or shorten the audio signal to allow changes in buffering. While this operation can be performed directly on the decoder's output, it is often more computationally efficient to stretch or shorten the signal directly within the decoder. It is desirable for the reference implementation to provide a time stretching/shortening implementation, although it should not be normative.

6.8. Input Robustness

The systems providing input to the encoder and receiving output from the decoder may be far from ideal in actual use. Input and output audio streams may be corrupted by compounding non-linear artifacts from analog hardware and digital processing. The codecs to be developed should be tested to ensure that they degrade gracefully under adverse audio input conditions. Types of digital corruption that may be tested include tandeming, transcoding, low-quality resampling, and digital clipping. Types of analog corruption that may be tested include microphones with substantial background noise, analog clipping, and loudspeaker distortion. No specific end-to-end quality requirements are mandated for use with the proposed codec. It is advisable, however, that several typical in situ environments/ processing chains be specified for the purpose of benchmarking end-to-end quality with the proposed codec.

6.9. Support of Audio Forensics

Emergency calls can be analyzed using audio forensics if the context and situation of the caller has to be identified. Thus, it is important to transmit not only the voice of the callers well, but also to transmit background noise at high quality. In these situations, sounds or noises of low volume should also not be compressed or dropped. Therefore, the encoder must allow DTX to be disabled when required (e.g., for emergency calls).

6.10. Legacy Compatibility

In order to create the best possible codec for the Internet, there is no requirement for compatibility with legacy Internet codecs.

7. Security Considerations

Although this document itself does not have security considerations, this section describes the security requirements for the codec.

As for any protocol to be used over the Internet, security is a very important aspect to consider. This goes beyond the obvious considerations of preventing buffer overflows and similar attacks that can lead to denial-of-service (DoS) or remote code execution. One very important security aspect is to make sure that the decoders have a bounded and reasonable worst-case complexity. This prevents an attacker from causing a DoS by sending packets that are specially crafted to take a very long (or infinite) time to decode.

A more subtle aspect is the information leak that can occur when the codec is used over an encrypted channel (e.g., [SRTP]). For example, it was suggested [wright08] [white11] that use of source-controlled VBR may reveal some information about a conversation through the size of the compressed packets. Therefore, it should be possible to use the codec at a truly constant bit-rate, if needed.

8. Acknowledgments

We would like to thank all the people who contributed directly or indirectly to this document, including Slava Borilin, Christopher Montgomery, Raymond (Juin-Hwey) Chen, Jason Fischl, Gregory Maxwell, Alan Duric, Jonathan Christensen, Julian Spittka, Michael Knappe, Christian Hoene, and Henry Sinnreich. We would also like to thank Cullen Jennings, Jonathan Rosenberg, and Gregory Lebovitz for their advice.

9. Informative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[XEP-0167] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, December 2009.

[RFC3951] Andersen, S., Duric, A., Astrom, H., Hagen, R., Kleijn, W., and J. Linden, "Internet Low Bit Rate Codec (iLBC)", RFC 3951, December 2004.

[ITU.G722.1] International Telecommunications Union, "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss", ITU-T Recommendation G.722.1, May 2005.

[Speex] Xiph.Org Foundation, "Speex: http://www.speex.org/", 2003.

[carot09] Carot, A., Werner, C., and T. Fischinger, "Towards a Comprehensive Cognitive Analysis of Delay-Influenced Rhythmical Interaction: http://www.carot.de/icmc2009.pdf", 2009.

[PAYLOADS] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, December 1999.

[RTP] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[SRTP] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[wright08] Wright, C., Ballard, L., Coull, S., Monrose, F., and G. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations: http://www.cs.jhu.edu/~cwright/oakland08.pdf", 2008.

[white11] White, A., Matthews, A., Snow, K., and F. Monrose, "Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on fon-iks http://www.cs.unc.edu/~fabian/papers/foniks-oak11.pdf", 2011.

Authors' Addresses

Jean-Marc Valin
Mozilla
650 Castro Street
Mountain View, CA 94041
USA

EMail: jmvalin@jmvalin.ca

Koen Vos
Skype Technologies, S.A.
Stadsgarden 6
Stockholm, 11645
Sweden

EMail: koen.vos@skype.net