[MS-SDPEXT]: Session Description Protocol (SDP) Version 2.0 Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial version
04/25/2008	0.2		Revised and edited technical content
06/27/2008	1.0		Revised and edited technical content
08/15/2008	1.01		Revised and edited technical content
12/12/2008	2.0		Revised and edited technical content
02/13/2009	2.01		Revised and edited technical content
03/13/2009	2.02		Revised and edited technical content
07/13/2009	2.03	Major	Revised and edited the technical content
08/28/2009	2.04	Editorial	Revised and edited the technical content
11/06/2009	2.05	Minor	Revised and edited the technical content
02/19/2010	2.06	Editorial	Revised and edited the technical content
03/31/2010	2.07	Major	Updated and revised the technical content
04/30/2010	2.08	Editorial	Revised and edited the technical content
06/07/2010	2.09	Editorial	Revised and edited the technical content
06/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.10	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies a proprietary extension to the **Session Description Protocol (SDP)** and the SDP extensions described in [MS-SDP] to support audio/video and application sharing calls.

SDP is used to negotiate and establish **session (3)** characteristics during call setup.

Unless explicitly specified, this protocol follows the offer/answer model to represent session characteristics using an SDP to establish a session.

This protocol is used to negotiate audio/video and application sharing call setup and adding video (or audio) to an existing audio (or video) only call.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Augmented Backus-Naur Form (ABNF)
network address translation (NAT)
server

The following terms are defined in [MS-OFCGLOS]:

200 OK
audio video profile (AVP)
Client Scale Secure Real-Time Transport Protocol (Client Scale-SRTP)
Common Intermediate Format (CIF)
Content-Type header
dual-tone multi-frequency (DTMF)
Interactive Connectivity Establishment (ICE)
INVITE
Real-Time Transport Protocol (RTP)
Scale Secure Real-Time Transport Protocol (SSRTP)
Secure Real-Time Transport Protocol (SRTP)
Server Scale Secure Real-Time Transport Protocol (Server SSRTP)
session
Session Description Protocol (SDP)
Session Initiation Protocol (SIP)

The following terms are specific to this document:

multiple points of presence (MPOP): A condition in which a single user signs in from multiple devices. A user who has multiple points of presence can be contacted through any of these devices.

secure audio video profile (SAVP): A protocol that extends the audio-video profile specification to include the Secure Real-Time Transport Protocol, as described in [RFC3711].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[IETFDRAFT-ICENAT-06] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-06, October 2005, http://tools.ietf.org/html/draft-ietf-mmusic-ice-06

[IETFDRAFT-ICENAT-19] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19, October 2007, http://tools.ietf.org/html/draft-ietf-mmusic-ice-19

[IETFDRAFT-ICETCP-07] Rosenberg, J., "TCP Candidates with Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-tcp-07, July 2008, http://tools.ietf.org/html/draft-ietf-mmusic-ice-tcp-07

[IETFDRAFT-OFFANS-08] Sawada, T., Kyzivat, P., "SIP (Session Initiation Protocol) Usage of the Offer/Answer Model", April 2008, http://tools.ietf.org/html/draft-ietf-sipping-sip-offeranswer-08

[IETFDRAFT-RCITD-199-01] Holmberg, C., "Response Code for Indication of Terminated Dialog", draft-ietf-sip-199-01.txt, August 2008, http://tools.ietf.org/id/draft-ietf-sip-199-01.txt

[MS-DTMF] Microsoft Corporation, "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals Extensions", June 2008.

[MS-ICE2] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions 2.0", March 2009.

[MS-OCER] Microsoft Corporation, "Client Error Reporting Protocol Specification", June 2008.

[MS-OCPSTN] Microsoft Corporation, "Session Initiation Protocol (SIP) Extensions for PSTN Calls", June 2008.

[MS-RTASPF] Microsoft Corporation, "RTP Payload Format for Application Sharing Extensions", March 2009.

[MS-SDP] Microsoft Corporation, "Session Description Protocol (SDP) Extensions", August 2007.

[MS-SIPRE] Microsoft Corporation, "Session Initiation Protocol (SIP) Routing Extensions", June 2008.

[MS-SSRTP] Microsoft Corporation, "Scale Secure Real-time Transport Protocol (SSRTP) Extensions", June 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002, http://www.ietf.org/rfc/rfc3261.txt

[RFC3262] Rosenberg, J., and Schulzrinne, H., "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)", RFC 3262, June 2002, http://www.ietf.org/rfc/rfc3262.txt

[RFC3264] Rosenberg, J., Schulzrinne, H., "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002, http://www.ietf.org/rfc/rfc3264.txt

[RFC3389] Zopf, R., "Real-Time Transport Protocol (RTP) Payload for Comfort Noise (CN)", September 2002, http://www.ietf.org/rfc/rfc3389.txt

[RFC3551] Schulzrinne, H., and Casner, S., "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003, http://www.ietf.org/rfc/rfc3551.txt

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) Attribute in Session Description Protocol (SDP)", RFC 3605, October 2003, http://www.ietf.org/rfc/rfc3605.txt

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K., "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004, http://www.ietf.org/rfc/rfc3711.txt

[RFC4145] Yon, D., Camarillo, G., "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005, http://www.ietf.org/rfc/rfc4145.txt

[RFC4235] Rosenberg, J. and Schulzrinne, H., "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", November 2005, ftp://ftp.rfc-editor.org/in-notes/rfc4235.txt

[RFC4566] Handley, M., Jacobson, V., and Perkins, C., "SDP: Session Description Protocol", RFC 4566, July 2006, http://www.ietf.org/rfc/rfc4566.txt

[RFC4568] Andreasen, F., Baugher, M., Wing, D., "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006, http://www.ietf.org/rfc/rfc4568.txt

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, July 2006, http://www.ietf.org/rfc/rfc4571.txt

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-ICE] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions", June 2008.

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary", June 2008.

[MS-RTP] Microsoft Corporation, "Real-time Transport Protocol (RTP) Extensions", June 2008.

[MS-RTPRADEX] Microsoft Corporation, "RTP Payload for Redundant Audio Data Extensions", June 2008.

1.3 Protocol Overview (Synopsis)

Initiation of a multimedia session (3) or conference requires the exchange of the media details, transport addresses, and other metadata between the parties involved. This exchange facilitates the negotiation of media characteristics for establishing the session. The media characteristics associated with a session are specified in Session Description Protocol (SDP). The exchange and negotiation of the session properties follows the specification of the Offer/Answer Model with the SDP. In applications, these protocols are used to negotiate and establish a multimedia session.

It is a common requirement that the media being exchanged in a multimedia session be protected using some form of encryption. When the **Real-Time Transport Protocol (RTP)** is used to exchange media, the media can be protected using the **Secure Real-Time Transport Protocol (SRTP)**, which requires exchange of attributes related to SRTP, such as cryptographic parameters. These characteristics can also be negotiated using SDP when the cryptographic characteristics of the media stream are described by a new SDP attribute name 'crypto' that is defined in the Security Description for Media Streams.

After the session is established, media can flow between the participating parties. Often, other networking components, such as **network address translation (NAT)**, are present between two parties and prevent media from traversing between the two parties. In such cases, **Interactive Connectivity Establishment (ICE)** can be used to facilitate media traversal through these network components. For information about the Microsoft® proprietary extension to the ICE protocol , see [MS-ICE]. ICE specifies a protocol for setting up the audio/video Real-Time Transport Protocol (RTP) streams in a way that allows the streams to perform network address translation (NAT).

The Microsoft Unified Communications system uses and extends these protocols to support multimedia sessions. The protocol extension consists of the following additions, enhancements and restrictions:

- **Supported values and parameters of the crypto attribute:** Specifies the parameter and values that are supported for the 'a=crypto,' as specified in Security Description for Media Streams.
- **SRTP and Scale Secure Real-Time Transport Protocol (SSRTP) encryption parameters are not renegotiated once the session is established.**
- A new SDP attribute, 'a=cryptoscale', is used for the negotiation of all the cryptographic parameters associated with SSRTP.
- RTAudio and RTVideo codecs in addition to the new RTData are supported.
- **SRTP encryption can be used optionally:** This option allows for support of remote peers that do not support Secure-RTP.
- The 'm=' line is used for preference specification, but the formats are not listed in the order of preference: This is a deviation from the specification of An Offer/Answer Model with the Session Description Protocol (SDP).
- TCP Media addresses in SDP are not used when ICE is not used. This is a deviation from the specification of TCP-Based Media Transport in the SDP.
- TCP Media addresses in SDP are not used in the first SIP Invite method when ICE is used.
- Addresses are not used for the 'rtcp' attribute**.**
- Limited support for the 'setup' attribute**:** This is a deviation from the TCP-Based Media Transport in the SDP.
- Limited support for the 'connection' attribute**:** This is a deviation from the TCP-Based Media Transport in the SDP.
- Early media support: This protocol handles early media only in very specific scenarios in a constrained manner and does not support early media in any other scenarios. Section 3 has more details on these scenarios and constraints.

- Usage of 'a=encryption' SDP attribute: This protocol has added limitations to the 'a=encryption' attribute.
- Ignore 'a=fmtp' attribute for video media.
- Only IPv4 addresses are used for addresses in Session Description Protocol.
- **Deviations from ICE:** This item captures all the deviations from the Interactive Connectivity Establishment (ICE) specifications.
- Session version on the 'o=' line is not incremented.
- Format for Dual-tone Multi-Frequency(DTMF) in SDP.
- Restriction on the name of the RTP Payload for Redundant Audio Data.
- Restriction on the name and sampling rate for comfort noise.
- **A media type 'm=applicationsharing'**: Identifies an RDP based application sharing media stream (session) over RTP. In the context of the application sharing media stream (m=applicationsharing), four new attributes are defined:

  ▪**a=x-applicationsharing-**session**-id:** Identifies an RDP session.

  ▪**a=x-applicationsharing-role:** Determines the party sharing role.

  ▪**(sharer or viewer)a=x-applicationsharing-media-type:** Negotiates the RDP media type.

  ▪**a=mid:** An identifier of the media described by the containing m= line.
- **A media level attribute 'a=tty':** Indicates that a UA has been configured to optimize the transfer of tones used in text telephony.
- **A video media level attribute 'a=x-caps: '**: Indicates the video capabilities supported by a video receiver.
- A MIME type '**application/GW-SDP**' is defined. This MIME type holds the gateway SDP or SIP trunk SDP.
- A media level attribute 'a=x-bypassid'. It is a declarative attribute used to indicate the location of the media endpoint/media processor associated with this SDP. It is used to optimize the media path with a gateway in the same location.
- A media level attribute 'a=x-bypass'. It is a declarative attribute that signifies that the media line with which it is associated involves bypass. It is a media level attribute sent in an answer when the answerer has chosen the bypass path.
- A media level attribute "a=x-mediasettings". It contains the stream capabilities of the sender.

  A media level attribute "a=x-ms-SDP-diagnostics:" is used to notify recipient of additional diagnostic information for that media line.

For details about these extensions and deviations, see Section 3.

1.4 Relationship to Other Protocols

This protocol depends on:

- Session Description Protocol (SDP), as described in [RFC4566], for media negotiation.
- **Session Initiation Protocol (SIP)**, as described in [RFC3261], for establishing and initializing a session.
- SDP for Media Streams, as described in [RFC4568], for media encryption.
- An Offer/Answer Model for SDP, as described in [RFC3264], to represent session characteristics used with SDP.
- A methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, as described in [IETFDRAFT-ICENAT-06] and [IETFDRAFT-ICENAT-19], for media to traverse NAT and firewalls.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable to:

- **Negotiation of SSRTP parameters between the two communicating peers:** The SSRTP encryption can be used by an application in conference scenarios when communicating with an MCU.
- **Ability to negotiate whether the media is encrypted using SRTP/SSRTP:** Ability to negotiate SRTP-encryption or SSRTP-encryption optionally enables the application to communicate using these encryption mechanisms when the remote peer cannot support either of these encryptions.
- **Ability to support 3 new codecs.**
- **Ability to do connection-oriented (TCP) media in selected scenarios.**
- **Ability to do early media in selected scenarios.**
- **Negotiation of video receive capabilities between two video peers:** Ability to negotiate video receive capabilities enables a video source to know what a video receiver is capable of receiving (frame rate, resolution, bit rate, number of video streams).

1.7 Versioning and Capability Negotiation

No version number is defined in this protocol. Session characteristics are negotiated using Session Description Protocol (SDP) and the Offer/Answer Model for SDP.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

As an extension to SDP, this protocol prescribes the format of session descriptions intended to support audio video and application sharing calls and can use any transport protocol used to carry SDP messages.

Session Initiation Protocol (SIP) is a commonly used transport for SDP messages. In this case, session descriptions, represented as SDP messages, MUST be included in the body of SIP messages. The **Content-Type header** of such SIP message MUST contain type/sub-type of 'application/sdp' or 'application/gw-sdp' (see **[RFC1521]** for examples and details of the Content-Type header). The 'application/gw-sdp' MIME type holds the gateway SDP or SIP trunk SDP and contains x-bypassid as a parameter. A client can use this parameter to decide if further parsing of the SDP is needed or not, thereby optimizing its processing.

SIP messages can be transported over TCP or TLS. TLS SHOULD be used to protect the encryption key, as the key is passed in the SIP/SDP signaling.

2.2 Message Syntax

The messages for this protocol are SDP messages. An SDP message contains the description of a media session. The session and media characteristics are described by a set of <type>=<value> lines, as specified in [RFC4566]. The extensions are defined as custom SDP attributes.

3 Protocol Details

3.1 Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

All the message processing events and sequencing rules for media negotiation conform to the SDP specifications [RFC4566] and [RFC3264], with some exceptions or modifications for the custom attributes introduced in this document.

Also note that the behavior described in [RFC4566] and [RFC3264] does not follow a simple client/server behavior. The two parties involved in an SDP exchange are peers. Which peer creates or modifies a session changes which peer is the offerer or answerer as described in [RFC3264]. In addition, some SDP attributes follow an offer/answer behavior and some SDP attributes supply information to the other peer with no answer expected.

3.1.5.1 Supported Values and Parameters for the 'a=crypto' Attribute

3.1.5.1.1 Parameter and Values Specification

The ‘a=crypto’ tag WSP crypto-suite WSP key-params *(WSP session-param) attribute is as specified in [RFC4568], with the exception that a single white space MUST be used. The attribute has the following format expressed using the **Augmented Backus-Naur Form (ABNF)** notation.

tag field: The < tag > field is used to specify a decimal number to identify a particular cryptographic attribute in the SDP Security Description for Media Streams [RFC4568]. In the current extension, the semantics of the tag field is more restricted, in that the decimal value MUST be unique across the 'a=crypto' and 'a=cryptoscale' attributes. 'a=cryptoscale' is a new attribute defined by this protocol and is specified in more detail in section 3.1.5.2.

crypto-suite field: The crypto-suite field is used to specify cryptographic methods or algorithms for media encryption. The only <crypto-suite> option supported is AES_CM_128_HMAC_SHA1_80. In other words, <crypto-suite> MUST be AES_CM_128_HMAC_SHA1_80. In [RFC4568], this is defined in the context of 'RTP/SAVP' as the transport. In the current extensions, use of this field is extended to the case when the transport is 'RTP/AVP' in an SDP offer. This deviation from [RFC4568] is required to support negotiation of SRTP optionally, as specified in section 3.1.5.8.

key-params field: The key-params field is used to specify the keying information. The key-params are further defined in [RFC4568], as follows:

```
key-params = <key-method> ":" <key-info>
```

where the key-method subfield is used to specify the provisional method of the keying information. As specified in [RFC4568], the only method that MUST be used is 'inline', indicating that the keying material is provided in the key-info field.

The definition of **key-info** is specified in [RFC4568]. The specification of key-info in [RFC4568] is specifically targeted to the 'RTP/SAVP' transport. In the current extension the key-info field can be used for both 'RTP/SAVP' and 'RTP/AVP'. This extension is required to support negotiation of SRTP optionally, as specified in section 3.1.5.8.

More than one key-params instance per line of a=crypto MUST NOT be used.

Details of key-info field: Here is the format specified in [RFC4568] for the key-info field.

```
"inline:" <key||salt> ["|" lifetime] ["|" MKI ":" length]
```

Here is a list of constraints and values accepted for the key-info field

- MKI MUST be used and the MKI length MUST be 1 byte.
- Value for lifetime MUST be 2^31 on sending.
- Value of lifetime MUST be ignored on the receive and 2^31 used instead.

Session-params field: Session-params field MUST NOT be used.

The following example is a 'a=crypto' attribute from Session Description Protocol.

```
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:t20I47Tyj1NDG6H+gWNpIzAzRPfYeQg8pP+ukwoy|2^31|1:1
```

Horizontal tab between tokens MUST NOT be used by the application.

3.1.5.2 Specifying and Negotiating SSRTP

The new ‘a=cryptoscale’ attribute is introduced to support SSRTP encryption of the media in an audio/video session. This attribute has the following format expressed in the ABNF notation:

```
"a=cryptosclae:" tag WSP scale-srtp-flavor WSP crypto-suite WSP key-params *(session-param)
```

The definition of tag, crypto-suite, and key-params are the same as specified for the a=crypto attribute. The new field, scale-srtp-flavor, is used to specify whether the encryption is done by the **server** or the protocol client. The field value can be either 'client' or 'server', specified as follows:

```
scale-srtp-flavor="client" | "server"
```

All the extensions to or deviations from [RFC4568] related to the a=crypto attribute, as specified in section 3.1.5.1 of this document, also apply to the 'a=cryptoscale' attribute.

An application supporting media encryption using **Client Scale Secure Real-Time Transport Protocol (Client Scale-SRTP)** chooses a value of 'client' for the scale-srtp-flavor field. An application using **Server Scale Secure Real-Time Transport Protocol (Server SSRTP)** chooses a value of 'server' for the scale-srtp-flavor' field. The choice depends upon the application characteristics. Typically, an application sending media to and receiving media from multiple peers SHOULD use the Server SSRTP encryption. An application MUST use either the Client Scale SRTP encryption or the Server SSRTP encryption. It MUST NOT use both at the same time.

Note that the <tag> field MUST be a unique decimal value across all the 'a=crypto' and 'a=cryptoscale' attributes.

The fields of the attribute 'a=crypto' and 'a=cryptoscale' are themselves not encrypted. Protection of the fields and encryption information is provided by the TLS transport over which the Session Initiation Protocol (SIP) signaling is carried.

The details of Server SSRTP/Client Scale SRTP can be found in [MS-SSRTP] section 2.2.

The following example is the 'a=cryptoscale' attribute used with SDP.

```
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:85Sm2QWogZ9N256qxTRhfIRxjUp9q1ISMxwbiloc|2^31|1:1
```

3.1.5.2.1 Processing and Negotiating SSRTP

The 'a=cryptoscale' attribute is used to negotiate SSRTP encryption of media.

The following table specifies how an application can communicate its SRTP and SSRTP encryption preferences.

Protocol element	Description
a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]	Supports SRTP encryption.
a=cryptoscale:<tag> client <crypto-suite> <key-params> [<session-params>]	Supports the client flavor of SSRTP encryption.
a=cryptoscale:<tag> server <crypto-suite> <key-params> [<session-params>]	Supports the server flavor of SSRTP encryption.

With the current extensions, an application expresses its ability to support SRTP and SSRTP by specifying the 'a=crypto' and 'a=cryptoscale' attributes, respectively, in an SDP message as the body of a SIP request.

An application MUST propose to support only one type of the SSRTP encryption in the SDP. The application MUST NOT add both (client and server) types of SSRTP to the SDP message.

An application SHOULD respond to the SIP request with only one preferred encryption in the SDP message in the SIP response, out of all the proposed encryptions specified in the SDP message of the SIP request.

Media is encrypted using SSRTP only when one peer proposes the Client Scale SRTP [MS-SSRTP] and the other peer proposes the Server SSRTP [MS-SSRTP]. If both peers propose the same type of SSRTP, media is not encrypted using SSRTP.

The following table summarizes the application behavior based on the negotiations. The behavior applies to both an initial SIP **INVITE** (as specified in [RFC3261]) and a re-INVITE to add new modality.

SDP offer contains	SDP answer contains	Result encryption from offerer to answerer	Result encryption from answerer to offerer
SRTP	SRTP	SRTP encrypted	SRTP encrypted
Client Scale SRTP	Server SSRTP	SRTP encrypted	SSRTP encrypted
Client Scale SRTP	Client Scale SRTP	No encryption	No encryption
Server SSRTP	Server SSRTP	No encryption	No encryption
Server SSRTP	Client Scale SRTP	SRTP encrypted	SSRTP encrypted
SRTP or Client Scale SRTP	SRTP	SRTP encrypted	SRTP encrypted
SRTP or Client Scale SRTP	Client Scale SRTP	No encryption	No encryption
SRTP or Client Scale SRTP	Server SSRTP	SRTP encrypted	SSRTP encrypted
SRTP or Server SSRTP	SRTP	SRTP encrypted	SRTP encrypted
SRTP or Server SSRTP	Client Scale SRTP	SRTP encrypted	SSRTP encrypted
SRTP or Server SSRTP	Server SSRTP	No encryption	No encryption

An application can specify multiple a=crypto and a=cryptoscale attributes in an SDP message. But there MUST NOT be more than one such attribute with the same SRTP type (SRTP or SSRTP) and the crypto-suite field.

3.1.5.2.2 Renegotiation of Encryption

An application MUST NOT use SIP re-INVITE to re-negotiate the encryption type (SRTP or SSRTP) or any other parameter in the a=crypto or a=cryptoscale lines.

3.1.5.3 Representing new Payload Types

This protocol adds support for three new payload types: RTAudio, RTVideo and RTData for audio, video and application sharing streams. The media formats of these payload types are described by the parameters in the following table that SHOULD<1> be specified using the 'a=rtpmap:' and 'a=fmtp:' attributes for dynamic payload types, as specified in [RFC4566].

Payload	Encoding name	Clock rate	Bit rate
RTAudio	x-msrta	16000	29000
RTAudio	x-msrta	8000	11800

Payload	Encoding name	Clock rate	Bit rate
RTVideo	x-rtvc1	90000	NA
RTData	x-data	90000	NA

As an example, the following SDP message fragment specifies an RTAudio payload type of an audio stream:

```
m=audio 37632 RTP/AVP 114 ...
…
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
…
```

Negotiation of these payload types are similar to the negotiation of other payload types, as specified in [RFC3264]. Any dynamic payload type can be chosen for these payloads following the RTP Profile for Audio and Video Conferences with Minimum Control specification, as specified in [RFC3551]. Specifying these parameters in the 'a=rtpmap' attribute in a media description section of an SDP message indicates the preference of these codecs for that payload type.

Applications that do not support these codecs MUST NOT advertise these codecs in the SDP. In the case of RTP, if a particular codec was referenced with a specific payload type number specified in the 'a=rtpmap:' attribute in the offer, that same payload type number MUST be used for that codec in the answer.

For more information about the payload type x-data (a=rtpmap:127 x-data/90000), see [MS-RTASPF].

For complete list of codecs supported, see [MS-RTP] section 2.2.1.

3.1.5.4 Interpreting the Preference of Formats in the Format List

In this protocol, the list of formats specified in an 'm=' for a particular media stream indicates the supported media formats, and does not represent any order of preference. This is different from what is specified in the Offer/Answer specification [RFC3264], which stipulates that the listed formats in the 'm=' line are listed according to the order of preference.

3.1.5.5 Format for Dual-Tone Multi-Frequency(DTMF) in SDP

The RTP Payload type number used to specify **dual-tone multi-frequency (DTMF)** in the 'm' line of the SDP MUST be 101<2> as specified in [MS-DTMF].

3.1.5.6 Restriction on the Name of the RTP Payload for Redundant Audio Data

The name of the payload used for Redundant Audio Data [MS-RTPRADEX] MUST be 'RED' and is case-sensitive.

3.1.5.7 Restriction on the Name and sampling rate for comfort noise

The name of the payload used for comfort noise SHOULD<3> be "CN" and the sampling rate SHOULD<4> be 8,000 or 16,000. For more information, see [RFC3389].

3.1.5.8 Negotiating SRTP Optionally

To require SRTP encryption for a media stream, an application can use the Secure Real-Time Transport Protocol (SRTP), as specified in [RFC3711], to specify the **secure audio video profile (SAVP)** in an 'm=' line of an SDP message, as part of the SRTP negotiation. This is shown in the following example.

```
m=audio 50004 RTP/SAVP 8 97 101
```

This description, however, does not allow for the possibility to negotiate SRTP encryption optionally, in that the support of the SRTP encryption is desired but not required.

To support SRTP encryption optionally, this protocol deviates from the specification [RFC3711]; in a SIP INVITE request, an application MUST use **audio video profile (AVP)** in the 'm=' line of SDP, together with the 'a=crypto' or 'a=cryptoscale' attribute to negotiate media encryption using SRTP or SSRTP. The application SHOULD bypass the negotiation of SRTP encryption by not specifying any 'a=crypto' and 'a=cryptoscale' attributes. To acknowledge the ability to support the SRTP encryption, the remote peer MUST respond to the SIP request in a SIP (**200 OK**) response with an SDP message specifying SAVP in the 'm=' line and the 'a=crypto' or 'a=cryptoscale' attribute as part of the media description. All subsequent SIP re-INVITE requests MUST continue to have SAVP. If the remote peer cannot support SRTP encryption, the remote peer MUST specify AVP in the 'm=' line of SDP and MUST NOT specify any 'a=crypto' and 'a=cryptoscale' attributes in the SIP response.

The following are examples of negotiating encryption.

If a peer sends an SDP in a SIP request to specify that it can support SRTP encryption, but the support is not mandatory, an example SDP is:

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

If the Peer responding to the request is capable of supporting, and does support, SRTP encryption, an example SDP message is:

```
m=audio 50014 RTP/SAVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:v0ncVM8eKP2bkOINeRaqcFeqjXwGMXo0sRalidZc|2^31|1:1
```

If the Peer responding to the request is not capable of supporting or does not support SRTP encryption, an example SDP message is:

```
m=audio 50104 RTP/AVP 8 97 101
```

If a peer sends an SDP in a SIP request to mandate SRTP encryption support, an example SDP message is:

```
m=audio 50004 RTP/SAVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

3.1.5.9 Connection-Oriented Media Address Support

In the SDP, as specified in [RFC4566], the 'm=' line is used to specify the transport for the given media type. The supported transport can be either UDP or TCP. TCP is a connection-oriented transport by which the media is exchanged. UDP is not connection-oriented. The following is an SDP message fragment showing an 'm=' line specifying that TCP be used as the media transport, as defined in [RFC4571]:

```
m=audio 50004 TCP/RTP/AVP 8 97 101
```

However, the connection-oriented transport SHOULD NOT<5> be used when Interactive Connectivity Establishment (ICE), as defined in [IETFDRAFT-ICENAT-06] or [IETFDRAFT-ICENAT-19], is not enabled on the offering application and the answering application. This applies to any offer or answer received from an application that does not support ICE.

If the offer or answer received supports ICE, according to the ICE specifications in [IETFDRAFT-ICENAT-06] and [IETFDRAFT-ICENAT-19], the port and the transport specified in the 'm=' line are referred to as the active address or the address that will be tried first in the ICE methodology to establish connection.

Application sharing SHOULD<6> use [MS-ICE2] over TCP.

We are in the ICE scenario if both the peers can support the ICE, which can be established by examining the offer SDP and answer SDP.

Connection-oriented (TCP) transport for the active addresses in the first offer or answer SDP MUST NOT be used. Subsequent SDP (SDPs in the reinvite) SHOULD<7> have connection-oriented (TCP) transport for the active address.

Any specification of a connection-oriented transport specified in the 'm=' line of SDP in the first offer or answer MUST be rejected, as specified in [RFC3264] section 6 or section 7.

3.1.5.10 Limited support for 'setup' and 'connection' Attributes

TCP-based Media Transport in the Session Description Protocol, as specified in [RFC4145], adds two new attributes to the Session Description Protocol. These are the 'a=setup' and 'a=connection' attributes. These attributes are used to establish and maintain TCP connections for the media exchange. However, the support for these attributes is limited in this protocol. These limitations are discussed in this section.

3.1.5.10.1 Limited support for the 'a=setup' Attribute

TCP-based Media Transport in the Session Description Protocol specification, as defined in [RFC4145], specifies that the 'a=setup' attribute can have the following roles for the purpose of establishing a TCP connection:

- active
- passive
- actpass
- holdconn

When used in the context of this protocol, the 'a=setup' attribute MUST have one of the following two values:

- active
- passive

The behavior of the roles "active" and "passive" are the same as specified in [RFC4145] with the following exceptions. For example, the peer with the 'active' role initiates an outgoing TCP connection. The peer with the 'passive' role is ready to accept an incoming TCP connection.

If the initial offer has a value of 'a=setup:active', the answer also has a value of 'a=setup:active', but the offerer role is still considered to be 'active' because it is the endpoint that is initiating the outgoing connection. Subsequent offers/answers contain the correct values of 'active' and 'passive'.

3.1.5.10.2 Limited support for the 'a=connection' Attribute

The TCP-based Media Transport in the Session Description Protocol specification, as defined in [RFC4145], specifies that the 'a=connection' attribute can have the following values to indicate the status of a TCP connection.

- new
- existing

When used in the context of this protocol, the 'a=connection' attribute SHOULD<8> have the following value only:

- existing

The semantic of the 'existing' value is specified in [RFC4145].

3.1.5.11 Text Telephony Support

A new media level attribute 'a=tty' is defined. It is included by a UA to indicate that it has been configured to optimize the transfer of tones used in text telephony. Such tones could, for example, originate from a telecommunications device for the deaf (TDD) – also referred to as a TTY (teletypewriter) – that interacts with the UC device via an audio coupler. The presence of this attribute can be used by the peer UA as an indication to enable detection for such tones, and optimize for their transfer once detected.

Note that this attribute has no offerer/answer behavior. It is used to inform the peer UA that a TTY device can be used.

3.1.5.12 Early Media Support

Early media refers to media exchange taking place before a session INVITE is accepted. This could be the initial greeting received by the user while the SIP handshake is under way. Early media support amounts to getting an SDP answer in a provisional SIP response of the 18x levels and starting media exchange after processing the SDP answer. Provisional responses are specified in detail in [RFC3261] section 13. Early media support discussed in this document is not based on any specific RFC. It is the subject of the following sections.

3.1.5.12.1 Restriction to Receiving a SDP Answer in Provisional Response

To support early media, all of the following conditions MUST be met when a UA receives an SDP answer in the provisional response.

- Interactive Connectivity Establishment (ICE) MUST be supported for early media.

- All SDP answers in the provisional responses MUST be the same.
- When the offer is forked, SDP answers not in reliable provisional responses SHOULD<9> be sent by a maximum of 1 device. For information about how to determine if an offer is forked, see [MS-SIPRE].

3.1.5.12.2 Receiving a SDP Answer in Provisional Response and Starting Media Streams

Media streams will be started after receiving a SDP answer in the provisional response, depending upon whether the SIP INVITE request was forked to **multiple points of presence (MPOP)**.

A SIP INVITE request was forked if an 'ms-forking' SIP header exists in any of the provision response.The 'ms-forking' header is added when the call is forked to multiple points of presence (MPOP) by the SIP proxy or server. More detailed specification of this header can be found in [MS-SIPRE].

Depending on whether the SIP INVITE request was forked, media streams will be started as follows:

- If a SDP answer is received in a provisional response and the SIP INVITE request was forked, the following are applicable

  - the received streams SHOULD<10> be started, if they are not already started.
  - the send stream SHOULD<11> be started for sending DTMF only after receiving one or more RTP media packets (via the corresponding receive stream).

- If a SDP answer is received in a provisional response and the SIP INVITE request was not forked, both the receive and send streams (for sending DTMF only) SHOULD<12> be started with the consideration that the send stream is started only after receiving one or more RTP media packets (via the corresponding receive stream).

Additionally, speech for the media streams in the forward direction SHOULD NOT<13> be started until the 200 OK is received for the INVITE.

For more details about starting media streams, see section 3.1.5.5.

3.1.5.12.3 SDP Answer in Provisional and Final Responses

SDP answers received in the provisional response (of the 18x-level) and the final response (of the 200-level) can be different depending on whether the call is forked. Specifically, if the 18x arrived from one fork and the 200-level from another fork, the SDP answers received can be different.

If an answer was contained in an 18x-level, it SHOULD<14> be repeated (without any changes) in the 200 for the same fork.

If the call is not forked, the SDP answer received in the final response (200) SHOULD<15> be the same as the one received in the provisional response (18x).

3.1.5.12.4 ICE Processing When a SDP Answer is Received in the Provisional Response

When a SIP INVITE request is NOT forked and an SDP answer is received in the provisional response, ICE processing SHOULD<16> proceed as if the SDP was received in the final response.

3.1.5.13 Extensions for reliable provisional response processing and related offer/answer models

[RFC3262] specifies a means for SIP entities to send reliable provisional response within an early or established dialog. The following sections define client behavior and considerations specific to reliable provisional response and early media.

When negotiating early offer/answer prior to the call being answered, SIP UA's SHOULD<17> use the procedures described in [RFC3262], with the following exceptions:

- A SIP UA MUST NOT send any SIP request containing a Require header with the option tag of 100rel.
- A SIP UA SHOULD include Require:100rel in 183 responses. SIP UAs SHOULD use a reliable provisional 183 response with SDP to perform early connectivity checks or to negotiate early media.

Furthermore, SIP UAs SHOULD<18> use the procedures described in [IETFDRAFT-OFFANS-08] when sending reliable provisional response with SDP, with the following exceptions:

- SIP UAs MUST NOT negotiate more than one offer/answer before the call is answered.
- SIP UAs MUST NOT include SDP in PRACKs or 200 OK responses to PRACKs.
- SIP UA's MUST use a 1XX reliable response when responding to an INVITE with early media.
- SIP UA's MUST use a 2XX response when responding to an INVITE of an established dialog.

When dealing with forked endpoints and early media, SIP UAs SHOULD<19> also process 199 response code specified in [IETFDRAFT-RCITD-199-01] to clean up early media state, if any. Information regarding when a 199 is sent is specified in [MS-SIPRE].

3.1.5.14 No Support for Renegotiation of SRTP or SSRTP Encryption Parameters

SRTP encryption parameters MUST NOT be renegotiated after the encryption is negotiated and the session is established.

SSRTP encryption parameters MUST NOT be renegotiated after the encryption is negotiated and the session is established.

3.1.5.15 Ignore 'a=fmtp' Attribute for Video and Panoramic Video Media

"a=fmtp" attributes of SDP, as specified in [RFC4566], are not supported for video modality by this protocol and SHOULD be ignored.

3.1.5.16 Usage of 'a=encryption' SDP Attribute

The "a=encryption"<20> attribute, as specified in [MS-SDP], is meant for negotiating DES encryption and MUST be used in conjunction with the 'k=' attribute of SDP. The 'a=encryption' attribute is applied only to the negotiation of the DES encryption and does not affect the negotiation of SRTP or SSRTP negotiation.

3.1.5.17 Restricted Address Types in 'c=' and 'a=candidate' Lines

Session Description Protocol (SDP), as specified in [RFC4566], states that the IP addresses assigned to the "c=", "o=" and "a=candidate" lines in an SDP message MUST be IPv4 addresses.

3.1.5.18 No Support for Optional Parameters in the 'a=rtcp' Attribute

As specified in [RFC3605], the "a=rtcp" attribute has the following format in the ABNF notation:

```
rtcp-attribute =  "a=rtcp:" port  [nettype space addrtype space
                        connection-address] CRLF
```

Optional parameters are allowed in addition to the "port" parameter. However, this protocol only supports the use of the "port" parameter in the "a=rtcp" attribute and stipulates that the optional parameters (after the "port" parameter) MUST NOT be used.

3.1.5.19 Application sharing media stream/type 'm=applicationsharing'

This protocol defines a new media type – 'applicationsharing'- which represents an RDP-based media stream/session. An application sharing m= line identifies exactly one RDP session.

Application sharing media requires a lossless transport and therefore the only candidates supported are TCP-based, as specified in [RFC4145] and [IETFDRAFT-ICETCP-07].

Application sharing does not support early media.

In the context of this media type four new SDP attributes are defined.

3.1.5.19.1 'a=x-applicationsharing-session-id' attribute

The session-id attribute is used to uniquely identify an RDP session on one end.

This attribute is optional; if missing a viewer is going to be connected to the first available session.

Session-id has the following format in the ABNF notation:

```
session-id-attribute = "a=x-applicationsharing-session-id:" *(alphanum) CRLF
```

3.1.5.19.2 'a=x-applicationsharing-role' attribute

This attribute determines the RDP sharing role of the party.

This attribute SHOULD<21> be present. The party that is sharing SHOULD<22> set the role to "sharer" and the party that is viewing SHOULD<23> set the role to "viewer." The following table lists the appropriate role for the answer based on the role in the offer.

Offer	Answer
a=x-applicationsharing-role:sharer	a=x-applicationsharing-role:viewer
a=x-applicationsharing-role:viewer	a=x-applicationsharing-role:sharer

If the SDP session contains multiple application sharing m=lines, the (session-id, role) pair SHOULD<24> be unique for each active m=line.

The RDP role attribute has the following format in the ABNF notation:

```
role-attribute = "a=x-applicationsharing-role:" ( "sharer" | "viewer" ) CRLF
```

3.1.5.19.3 'a=x-applicationsharing-media-type' attribute

This attribute is used to negotiate the RDP media type to be used. This attribute is optional; default is "" indicating that this is a signaling only session with no associated media stream.Media-type attribute has the following format in the ABNF notation:

```
rdp-media-type-attribute = "a=x-applicationsharing-media-type:" <list-of-supported-medias>
CRLF
<list-of-supported-medias>: <rdp-flavor> *( SPACE <rdp-flavor> )
<rdp-flavor>: "rdp" | "webrdp" | ""
SPACE: %d32
```

3.1.5.19.4 'a=mid' attribute

This attribute is used as an identifier of the media described by the m=line. This attribute SHOULD<25> be included.

Every time a new m=line media is added to the SDP the value of a=mid is incremented by 1.

The media-identifier attribute has the following format in the ABNF notation:

```
media-identifier-attribute="a=mid:" 1*DIGIT CRLF
```

3.1.5.20 Interpretation of 'o=' line in the SDP

The 'o=' line of an SDP message, as specified in [RFC4566], specifies the session originator and session identifiers that include the session ID, session version, network type, address type, and unicast address. The ABNF is as follows:

```
O=<username> <sess-id> <sess-version> <nettype> <addrtype> <unicast-address>
```

The parameter <sess-id> MUST be ignored on receive. The parameter <sess-version> MUST be a numeric value but the value MUST ignored on receive. The parameter <nettype> MUST be "IN", the <addrtype> MUST be IPv4, and the <unicast-address> MUST be the dotted-decimal representation of the IP version 4 address. An application MUST NOT increment the session version value (<sess-version>) in the 'o=' line in any subsequent SDP offers.

3.1.5.21 Deviations from ICE-06

ICE, as specified in [IETFDRAFT-ICENAT-06], is a methodology to let media traverse NAT and firewalls to reach the remote peer. The following subsections describe the deviations from the standard ICE specification.

3.1.5.21.1 General Outline of the ICE Methodology

In general, ICE works as follows. First a peer (offerer) gets all its reachable addresses and provides them in an SDP offer. The SDP offer is then sent to the remote peer. The remote peer gets all its reachable addresses and provides them in an SDP answer. On receiving the SDP offer, both the offerer and the answerer begin to exchange packets to determine the optimal path for media traversal. This process of determining the optimal path is referred to as connectivity-checks in the subsequent discussions. After this optimal path is determined, the offerer sends a SIP re-INVITE to the remote peer, communicating the optimal address in the SDP offer. This SIP re-INVITE is referred to as an ICE re-INVITE in the subsequent sections of this document. An indicator of the ICE re-INVITE is the existence of an 'a=remote-candidate' attribute for a modality. This attribute is absent in the previous SIP INVITE or SIP re-INVITE. For more details, see [IETFDRAFT-ICENAT-06].

3.1.5.21.2 ICE RE-INVITE Initiator

According to [IETFDRAFT-ICENAT-06], an ICE re-INVITE is always sent by the offerer of that media. This protocol deviates from that specification, and stipulates that the ICE re-INVITE MUST always be sent by the offerer of the call and not the offerer of the modality. This means that the caller MUST send the ICE re-INVITE.

This also means that if the local peer starts an audio call with a remote peer and then, after some time, the remote peer adds video to this call, the ICE re-INVITE for the video stream MUST be sent by the local peer, instead of by the remote peer. In contrast, the [IETFDRAFT-ICENAT-06] specification requires that the ICE re-INVITE for video is sent by the remote-peer in a similar case.

3.1.5.21.3 No Update of Candidates Between INVITE and ICE RE-INVITE

According to [IETFDRAFT-ICENAT-06], the list of addresses exchanged in the original SIP INVITE can be updated anytime between the first INVITE and the ICE re-INVITE by sending a SIP UPDATE or SIP re-INVITE request. However, this protocol stipulates that an application MUST NOT add or remove addresses using SIP UPDATE or SIP re-INVITE until the connectivity-checks are done or until an ICE re-INVITE is exchanged successfully.

3.1.5.21.4 Extending the Transport to Connection-Oriented (TCP)

ICE, as specified in [IETFDRAFT-ICENAT-06], specifies that UDP be used as the transport and allows extensions to add other transport. This protocol adds TCP to the supported transport type in ICE. Thus, the following examples are both permitted:

```
a=candidate:ir84fUlcDqYH50bs2M/Xn/pDNE+fVfxRTbXBWG34PM8 2 1vvq9h3j8xixI3npD0X9VA UDP 0.830
10.56.65.184 63616  a=candidate:Mbmhbdy6gJ1nwkKtoJWa8h9LHlpQ90uT/EiBDOvBPB4 1
76CTu2GXyKtnYlu2ZydjXA TCP 0.190 172.29.105.45 50563
```

3.1.5.22 Deviation from ICE V19

ICE, as specified in [IETFDRAFT-ICENAT-19] , is a methodology to let media traverse NAT and firewalls to reach the remote peer. Support of ICE in this protocol differs from that specified in that document. The following subsections describe deviations from the standard ICE specification.

3.1.5.22.1 LITE implementation

According to [IETFDRAFT-ICENAT-19], there are two implementations of ICE – FULL implementation and LITE implementation. This protocol does not support the LITE implementation. This means that this protocol SHOULD<26> gather the RELAYED, SERVER REFLEXIVE candidates and perform connectivity checks as described in [MS-ICE2].

3.1.5.22.2 Ice-options attributes

According to [IETFDRAFT-ICENAT-19], an offer or an answer is allowed to use the ice-options attribute to identity the ice extensions supported by that agent. If an agent supports an extension, it MUST include the token that represents that extension in the ice-options attributes.

This protocol does not support the ice-options attribute. It SHOULD NOT<27> generate an SDP with this attribute and SHOULD ignore this attribute if it is present in the SDP received.

3.1.5.22.3 Ice-mismatch attributes

According to [IETFDRAFT-ICENAT-19], this attribute, when present in an answer, indicates that the agent that sends the offer contains a default destination for a media component that did not have a corresponding candidate attribute.

This protocol does not support this attribute. It SHOULD NOT<28> generate an answer with this attribute. If received as an answer to an offer, this protocol SHOULD ignore this attribute.

For offers that are generated by this protocol, the default destination for a media component SHOULD<29> have a corresponding candidate attribute.

3.1.5.22.4 ice-ufrag and ice-pwd attributes

According to [IETFDRAFT-ICENAT-19], the ice-ufrag attribute can be 4 to 256 bytes long and the ice-pwd attribute can be 22 to 256 bytes long, and they SHOULD be in clear text. This protocol determines if base 64 encoding is used in the offer by checking their lengths. Therefore, in order to not have the answering agent treat clear text as encoded string, this protocol SHOULD NOT<30> use an ice-ufrag attribute of 6 bytes and an ice-pwd attribute of 32 bytes long in an offer.

3.1.5.23 Deviation from ICE-TCP-07

ICE, as described in [IETFDRAFT-ICENAT-06] and [IETFDRAFT-ICENAT-19], defines ways for media traffic to traverse NAT and a firewall. These specifications provide a general framework for describing candidates, which only use the UDP transport protocol.

[IETFDRAFT-ICETCP-07] extends the ICE protocol to include TCP transport protocol.

The deviations from these specifications are described in the subsections that follow.

3.1.5.23.1 Default Candidate

For audio/video calls, the default candidate SHOULD NOT<31> be TCP. For application sharing calls, the default candidate SHOULD<32> be TCP.

3.1.5.23.2 Local Candidate

For audio and video media type, this protocol does not gather passive local host candidates for the TCP protocol. Therefore the SDP SHOULD NOT have any passive TCP local host candidates.

For application sharing media type, the local candidates SHOULD<33> be TCP.

3.1.5.24 Extensions for call hold and retrieve

The following specifies client behavior for the offer and answer negotiated for hold and un-hold operations when in an audio and/or video call.

3.1.5.24.1 Invoking hold

A protocol client invoking hold is required to do the following for all audio and video media streams in the resulting offer:

- Client SHOULD<34> change the direction of all streams to inactive
- Client SHOULD<35> include sip.rendering, as specified in [MS-SIPRE], with a value of "no".

3.1.5.24.2 Clearing hold (retrieve)

In-order to clear the hold (retrieve the call) the protocol client is required to do the following for all audio and video media streams in the resulting offer:

- Client SHOULD<36> change the direction of all streams to sendrecv.
- Client SHOULD<37> exclude sip.rendering, as specified in [MS-SIPRE].

3.1.5.25 Extension for video receive capabilities `a=x-caps'

This protocol defines a video media level attribute `a=x-caps' which represents what a video receiver is capable of receiving. A video capability a= line defines video capabilities for each of the video codec the video receiver is capable of receiving.

This attribute is optional; if missing, a video sender SHOULD<38> set the video receive capabilities of the remote peer as **Common Intermediate Format (CIF)** at 15 fps and VGA at 15 fps for main video and 15fps for panoramic video. QCIF, CIF and VGA MUST be advertised in the list of Video capabilities for the main video. A video capability of VGA 13 fps MUST be treated as VGA 15 fps. This media attribute has the following format in ABNF notation:

Video-Capabilities-media-type-attribute: "a=x-caps:" <video-payload-type > SPACE <list-of-video-capabilities> CRLF

video-payload-type: The RTP payload type number for video (such as 121 for x-rtvc1 and 34 for H.263)

list-of-video-capabilities: <video-capability>";"<video-capability>

video-capability: <Capability-ID>":"<Width-of-video-frame>":"<Height-of-video-frame>":"<frames-per-second>":"<maximum-bitrate-bits-per-second>":"<additional-attributes> SPACE: %d32**capability-id(integer):** A unique random integer among the listed capability ID for that m= line and is between 1 and 2147483647 in the entire video-capabilities-media-type-attribute.

width-of-video-frame (integer): The width is one of these values:

- 176 for QCIF
- 352 for CIF
- 640 for VGA
- 1280 for HD

Height-of-the-video-frame (integer): The height is one of these values:

- 144 for QCIF
- (288 for CIF
- 480 for VGA
- 720 for HD)

frames-per-second (float): Value is always less than 30.0 fps. Any values beyond 30.0 SHOULD be treated as 30.0. It specifies the maximum frame rate the receiver is capable of receiving.

maximum-bitrate-in bits-per-second (integer): It is ignored and is reserved for future use.

Any additional attribute SHOULD be ignored and is reserved for future use.

A protocol peer, upon receiving the video capabilities SHOULD<39> do the following:

- If there is a "," in a <video-capability> attribute, anything from "," till the end of <video-capability> (";" or CRLF) SHOULD be ignored.
- If there is a syntax error in a=x-caps the whole a=x-caps SHOULD be ignored and video receive capabilities of remote peer SHOULD be set as CIF at 15 fps or VGA at 15 fps

The followng example is the 'a=x-caps' attribute used with Session Description Protocol.

```
a=x-caps:121
263:1280:720:30.0:1500000:1;4359:640:480:30.0:600000:1;8455:352:288:15.0:250000:1;12551:176:1
44:15.0:180000:1
```

3.1.5.26 Extensions to optimize the media path to a gateway

This section describes the extensions used by the client to optimize the media path to a gateway in the same location. These extensions SHOULD be used by SIP clients that support ms-bypass [MS-OCPSTN]. <40>

3.1.5.26.1 'a=x-bypassid' attribute

This is a declarative attribute used to indicate the location of the media endpoint associated with this SDP. It is a media level attribute, which MUST be sent in an offer by the client if it wants to establish a baseline for the possibility of doing media bypass.

When the SIP client receives a multipart/alternative MIME body in the offer it first looks for a part of type of application/GW-SDP and if one is found and the x-bypassid values match then that is the part chosen.

3.1.5.26.2 'a=x-bypass' attribute

This is a declarative attribute that signifies that the media line with which it is associated involves bypass. It is a media level attribute which MUST be sent when the answerer has chosen the bypass path.

3.1.5.26.3 'a=x-mediasettings' attribute

This attribute SHOULD be added by a UA to signify its following stream capabilities:

- holdrtcpunsupported: This value SHOULD be added by a UA to signify that RTCP is not supported when the call is on hold. If present in the negotiated SDP, the client MUST NOT expect RTCP when the call is on hold.
- rtcpunsupported: This value SHOULD be added by a UA to signify that RTCP is not supported. If present in the negotiated SDP, the client MUST NOT expect RTCP for the call.

- signalboostunsupported: This value SHOULD be added by a UA to signify that its media stream is not amplified. If present in the negotiated SDP, the client SHOULD apply amplification on the incoming media stream.

The grammar for this attribute is defined below

a=x-mediasettings:(holdrtcpunsupported/rtcpunsupported/signalboostunsupported)*(SPACE holdrtcpunsupported/rtcpunsupported/signalboostunsupported)

3.1.5.27 Extensions for diagnostic info in SDP

This protocol defines a new media level attribute **a=x-ms-SDP-diagnostics**<41>. An SDP endpoint SHOULD add this attribute if it wants the receiving endpoint to display a notification regarding the status of the SDP session.

The format for the **a=x-ms-SDP-diagnostics** in the Backus–Naur form (BNF) is specified as follows. It is similar to that of the **ms-public-diagnostics** header defined in [MS-OCER].

The parameters **EQUAL**, **HCOLON**, **SEMI**, **generic-param**, and **quoted-string** are as defined in **[RFC3261]** Section 25.1.

```
a EQUAL x-ms-SDP-diagnostics HCOLON  ErrorId  *(.SubErrorId) SEMI reason-param * (SEMI
generic-param)
ErrorId = unsigned-integer

SubErrorId = unsigned-integer

reason-param = "reason=" reason-value

reason-value = quoted-string
```

ErrorId (unsigned-integer): Required. Value MUST be within unsigned 32-bit integer range. Represents a specific error condition, and SHOULD be used by the SIP client to determine appropriate error handling behavior.

SubErrorId (unsigned-integer): Optional. If present, its value MUST be within unsigned 32-bit integer range. **SubErrorId** can be used to differentiate related scenarios that result in the same **ErrorId**, and can be used by the SIP client to determine appropriate error handling behavior.

reason-value: Optional. Reason SHOULD indicate a specific reason for an explanation of the error. A SIP client SHOULD NOT use this parameter value to determine error handling behavior. This parameter value can be used for SIP server troubleshooting purposes.

***(SEMI generic-param):** Optional. Can be used to define custom attribute-value pairs, to convey additional troubleshooting information to the SIP client.

The following table lists the allowed values. More values might be added in the future. If an ErrorId not listed here is received by an SDP endpoint, it SHOULD be ignored.

ErrorId	Reason string	Explanation
53000	Insufficient Bandwidth Available	Bandwidth policy checks on server failed for this particular m= line

ErrorId	Reason string	Explanation
53001	Candidates Restricted	Bandwidth policy checks required some of the original a=candidate lines to be removed for bandwidth limitation reasons

Following is an example SDP.

```
v=0
o=- 168 0 IN IP4 172.18.0.106
s=session
c=IN IP4 172.18.0.106
b=CT:1000
t=0 0
m=audio 51038 RTP/SAVP 9 111 0 8 97 101 13 118
c=IN IP4 172.18.0.106
a=rtcp:51039
a=ice-ufrag:VUa7
a=ice-pwd:uSvOqE8rrlf2065N/AymKLpL
a=candidate:1 1 UDP 2130706431 172.18.0.106 51038 typ host
a=candidate:1 2 UDP 2130705918 172.18.0.106 51039 typ host
a=candidate:3 1 tcp-act 1684798719 172.18.0.106 50112 typ srflx raddr 172.18.0.106 rport
50112
a=candidate:3 2 tcp-act 1684798206 172.18.0.106 50112 typ srflx raddr 172.18.0.106 rport
50112
a=label:main-audio
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:8q4vdHtbV3uIGM7z+jgLTxltWIhd9vedIMXiO4MB|2^31|1:1
a=rtpmap:9 g722/8000
a=fmtp:9 bitrate=64000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 RED/8000
a=fmtp:97 red/8000
a=rtpmap:101 telephone-event/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=ptime:20
a=encryption:rejected
m=video 0 RTP/AVP 121
a=x-ms-sdp-diagnostics:53000; reason="Insufficient Bandwidth Available"
a=label:main-video
a=rtpmap:121 x-rtvc1/90000
a=fmtp:121 CIF=15;VGA=15;PANO=15
```

3.1.5.28 Extensions for Music-on-Hold

This section specifies SDP extensions that a UA can use in SDP offers to indicate that music-on-hold is being streamed in a given media session.

3.1.5.28.1 'a=feature' attribute

This is a declarative media-level attribute that specifies additional features for its associated media line. Its syntax is as follows:

```
a EQUAL feature HCOLON 1*ALPHANUM
```

The parameters **EQUAL**, **HCOLON**, and **ALPHANUM** are as defined in [RFC3261] section 25.1. The alphanumeric string to the right of the colon indicates the particular feature being attributed to the associated media. For music-on-hold, it MUST be "MoH". Additional values MAY be defined in the future to signal other features besides music-on-hold.

3.1.5.28.2 UA behavior for 'a=feature' attribute

If a UA wishes to use these extensions to signal that it is streaming music-on-hold, its offer MUST contain a=sendonly and a=feature:MoH attribute lines for those media. The a=feature:MoH line MUST NOT appear under any other media line than m=audio.

If all media lines in the SDP contain an a=feature:MoH attribute line, the SIP Contact header SHOULD include include sip.rendering, specified in [RFC4235] section 5.2, with a value of "no".

When a UA receives an SDP offer with a=feature:MoH, it MAY chose to render an appropriate user interface for hold or music-on-hold. When a UA receives an SDP offer with features that it does not understand, it SHOULD ignore them.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Generic Examples

4.1.1 Client Makes an Offer using ICE as described in IETFDRAFT-ICENAT-06

Following are some SDP examples that demonstrate the offer with most of the extensions specified in this protocol.

The following example is an offer sent by a client.

```
v=0
o=- 0 0 IN IP4 10.56.65.184
s=session
c=IN IP4 10.56.65.184
b=CT:99980
t=0 0
m=audio 37632 RTP/AVP 114 9 111 112 115 116 4 8 0 97101
k=base64:4/oLIAyteGDaKsIPrsoEYnf2FNxKS8H4RxQetMXAq+phbKbECwC7nXwvmk8V
a=candidate:ir84fUlcDqYH50bs2M/Xn/pDNE+fVfxRTbXBWG34PM8 1 1vvq9h3j8xixI3npD0X9VA UDP 0.830
10.56.65.184 37632 a=candidate:ir84fUlcDqYH50bs2M/Xn/pDNE+fVfxRTbXBWG34PM8 2
1vvq9h3j8xixI3npD0X9VA UDP 0.830 10.56.65.184 63616
a=candidate:Mbmhbdy6gJ1nwkKtoJWa8h9LHlpQ90uT/EiBDOvBPB4 1 76CTu2GXyKtnYlu2ZydjXA TCP 0.190
172.29.105.45 50563 a=candidate:Mbmhbdy6gJ1nwkKtoJWa8h9LHlpQ90uT/EiBDOvBPB4 2
76CTu2GXyKtnYlu2ZydjXA TCP 0.190 172.29.105.45 50563
a=candidate:L6SFpc1rY2GenmqDg0N7eqYMWN0/jI3nH6vttRoU0VE 1 L4J04UBiONZgYNUCyOlT9Q UDP 0.490
172.29.105.45 50403 a=candidate:L6SFpc1rY2GenmqDg0N7eqYMWN0/jI3nH6vttRoU0VE 2
L4J04UBiONZgYNUCyOlT9Q UDP 0.490 172.29.105.45 57283
a=candidate:sct7Qs0hpryFGR/K94UBURz0NOWuThCD7a1iTJyLF8Q 1 ozhWUy01WJw83GTHGukOiw TCP 0.250
10.56.65.184 16512 a=candidate:sct7Qs0hpryFGR/K94UBURz0NOWuThCD7a1iTJyLF8Q 2
ozhWUy01WJw83GTHGukOiw TCP 0.250 10.56.65.184 16512
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:85Sm2QWogZ9N256qxTRhfIRxjUp9q1ISMxwbiloc|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:t20I47Tyj1NDG6H+gWNpIzAzRPfYeQg8pP+ukwoy|2^31|1:1
a=maxptime:200
a=rtcp:63616
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:4 G723/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:97 RED/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:optional
m=video 24832 RTP/AVP 12134
k=base64:OBwIjVz1CCOJA9t2QUIhoLSd6Zk5ZbHOqaxUoqgpQggTmSrV7m5OBNMUI1rt
a=candidate:kR94HVUEeM0GCz7TfUzEoBojVMo/V+fSSbYUv2MFCxg 1 VzH+zfgjCGjhGEF9aa6ujg UDP 0.840
10.56.65.184 24832 a=candidate:kR94HVUEeM0GCz7TfUzEoBojVMo/V+fSSbYUv2MFCxg 2
VzH+zfgjCGjhGEF9aa6ujg UDP 0.840 10.56.65.184 39552
a=candidate:Sluz8sKaw20lFkZ8/m6UjK9HU/hYudqY3Xv4yJ1QcQI 1 HX1SFTd1yDyb0gmg5Fl6wQ TCP 0.190
```

```
172.29.105.45 55585 a=candidate:Sluz8sKaw20lFkZ8/m6UjK9HU/hYudqY3Xv4yJ1QcQI 2
HX1SFTd1yDyb0gmg5Fl6wQ TCP 0.190 172.29.105.45 55585
a=candidate:J8ubfJUv8xZqKbnKzkH0MvqpRcQE+6jf4/22WG0qzPI 1 r14RJIjw2dTtunLCxLxNGw UDP 0.490
172.29.105.45 56913 a=candidate:J8ubfJUv8xZqKbnKzkH0MvqpRcQE+6jf4/22WG0qzPI 2
r14RJIjw2dTtunLCxLxNGw UDP 0.490 172.29.105.45 57169
a=candidate:Ya8xTTDo0z9kK5Ty6W++HLmVzc95OM1rFnaJ8TT9/hc 1 pt8XROAfQJ9Q0k9nFSaHGg TCP 0.250
10.56.65.184 7680 a=candidate:Ya8xTTDo0z9kK5Ty6W++HLmVzc95OM1rFnaJ8TT9/hc 2
pt8XROAfQJ9Q0k9nFSaHGg TCP 0.250 10.56.65.184 7680
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:BPTL7aWOS9oqHOexSUMoWRCBwGT00ATCrWDI8Pkl|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:N4XsS82yDHiZdPuG2xXvXplKbbPXjeuvup7B9M4H|2^31|1:1
a=maxptime:200
a=rtcp:39552
a=rtpmap:121 x-rtvc1/90000
a=rtpmap:34 H263/90000
a=encryption:optional
```

In the preceding example, the offerer is proposing audio and video as modalities. The offerer supports both SRTP and SSRTP as the mode for encryption and proposes that in the SDP, using the 'a=crypto' and 'a=cryptoscale' attributes. The offerer also wants to only encrypt the media optionally. This is described by specifying 'RTP/AVP' as the transport, even though there are 'a=crypto' and 'a=cryptoscale' attributes present in the SDP message.

The 'a=encryption'<42> attribute is used to negotiation Data Encryption Standard (DES) encryption, and used along with a 'k=' attribute. The values of the 'a=encryption' attribute only apply to the DES encryption.

Also note that RTAudio and RTVideo codecs are represented in the codec using dynamic payloads of 114, 115, and 121 and are identified using their encoding names of 'x-msrta' and 'x-rtvc1' in their corresponding 'a=rtpmap' attributes.

4.1.2 Client Receives Response with SSRTP to ICENAT-06 Offer

The following example is a response (SDP answer) received for the preceding offer.

```
v=0
o=- 0 0 IN IP4 172.29.106.5
s=session
c=IN IP4 172.29.106.5
b=CT:1000
t=0 0
m=audio 57472 RTP/SAVP 9 111 8 0 97 101
c=IN IP4172.29.106.5
a=rtcp:59648
a=candidate:vu6VFdaIZf91YO6DePy/FBzJ0pHopn1lRD/vlUSSJU0 1 bhmEv8fu4QTnweUlMXuiiA UDP 0.900
172.29.106.5 57472
a=candidate:vu6VFdaIZf91YO6DePy/FBzJ0pHopn1lRD/vlUSSJU0 2 bhmEv8fu4QTnweUlMXuiiA UDP 0.900
172.29.106.5 59648
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:L1gAdIcRtzb7OdDbZJhf1PTH2Pjlkq7gxJWva7zX|2^31|1:1
a=label:main-audio
a=encryption:rejected
a=rtpmap:9 G722/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:97 RED/8000
```

```
a=fmtp:97 red/8000
a=rtpmap:101 telephone-event/8000
a=ptime:60
m=video 58496 RTP/SAVP 121
c=IN IP4 172.29.106.5
a=rtcp:54656
a=candidate:HfCkQziV8VGEey2/VVPSm3m8b0otY/xZilAoWGRo6BM 1 SzMsl46X7YwBpVsbapBY/g UDP 0.900
172.29.106.5 58496
a=candidate:HfCkQziV8VGEey2/VVPSm3m8b0otY/xZilAoWGRo6BM 2 SzMsl46X7YwBpVsbapBY/g UDP 0.900
172.29.106.5 54656
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:sCkL4JFpu5JbaworoJYsXuPvDbpJLavl15JLOJE6|2^31|1:1
a=label:main-video
a=encryption:rejected
a=rtpmap:121 x-rtvc1/90000
a=fmtp:121 CIF=15;VGA=15;PANO=15
a=x-sourceid:MainCamera
```

The answerer (remote peer) also wants to encrypt the media using SRTP, and so it replies with an SDP answer that has 'RTP/SAVP' in the transport in the 'm=' line. The answerer does not want to support Data Encryption Standard (DES) encryption, and so it returns an 'a=encryption' attribute with a value of 'rejected'. Note that the media is still encrypted using SRTP because the 'a=encryption' attribute value only applies to DES encryption.

Also note that the remote peer really wants to do SSRTP, and thus returns only the 'a=cryptoscale' attribute with the 'server' value for the **scale-srtp-flavor** parameter. After this exchange of offer and answer, the call is set up and the media is encrypted using SSRTP.

4.1.3 Client Makes an Offer using ICE as described in IETFDRAFT-ICENAT-19

Following are some Session Description Protocol (SDP) examples that demonstrate the offer with most of the extensions specified in this document.

The following example is an offer sent by a client.

```
v=0
o=- 0 0 IN IP4 172.24.32.152
s=session
c=IN IP4 172.24.32.152
b=CT:99980
t=0 0
m=audio 50005 RTP/AVP 114 9 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag: 6nxO
a=ice-pwd: G6rUJNNaobz8IdDZrAbyFDoO
a=candidate:1 1 UDP 2130706431 172.24.32.152 50005 typ host
a=candidate:1 2 UDP 2130705918 172.24.32.152 50009 typ host
a=candidate:2 1 TCP-PASS 6556159 172.29.105.171 53127 typ relay raddr 172.29.105.171 rport
53127
a=candidate:2 2 TCP-PASS 6556158 172.29.105.171 53127 typ relay raddr 172.29.105.171 rport
53127
a=candidate:3 1 UDP 16648703 172.29.105.171 59353 typ relay raddr 172.29.105.171 rport 59353
a=candidate:3 2 UDP 16648702 172.29.105.171 59627 typ relay raddr 172.29.105.171 rport 59627
a=candidate:4 1 TCP-ACT 7076863 172.29.105.171 53127 typ relay raddr 172.29.105.171 rport
53127
a=candidate:4 2 TCP-ACT 7076350 172.29.105.171 53127 typ relay raddr 172.29.105.171 rport
53127
```

```
a=candidate:5 1 TCP-ACT 1684797951 172.24.32.152 50004 typ srflx raddr 172.24.32.152 rport
50004
a=candidate:5 2 TCP-ACT 1684797438 172.24.32.152 50004 typ srflx raddr 172.24.32.152 rport
50004
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:15PHFDbUI8l9/bOHUYM9geb2IakQY3tMe31TgoPC|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:C62B/j2xrqnk18t4bxXthuGv/Lxc9DmYDG4mnAOK|2^31|1:1
a=maxptime:200
a=rtcp:50009
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:4 G723/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:optional
m=video 50012 RTP/AVP 121 34
a=ice-ufrag: m7A0
a=ice-pwd: yfKPbeepmE8/PvGoIDFq4OId
a=candidate:1 1 UDP 2130706431 172.24.32.152 50012 typ host
a=candidate:1 2 UDP 2130705918 172.24.32.152 50011 typ host
a=candidate:2 1 TCP-PASS 6556159 172.29.105.171 59400 typ relay raddr 172.29.105.171 rport
59400
a=candidate:2 2 TCP-PASS 6556158 172.29.105.171 59400 typ relay raddr 172.29.105.171 rport
59400
a=candidate:3 1 UDP 16648703 172.29.105.171 54004 typ relay raddr 172.29.105.171 rport 54004
a=candidate:3 2 UDP 16648702 172.29.105.171 58581 typ relay raddr 172.29.105.171 rport 58581
a=candidate:4 1 TCP-ACT 7076863 172.29.105.171 59400 typ relay raddr 172.29.105.171 rport
59400
a=candidate:4 2 TCP-ACT 7076350 172.29.105.171 59400 typ relay raddr 172.29.105.171 rport
59400
a=candidate:5 1 TCP-ACT 1684797951 172.24.32.152 50003 typ srflx raddr 172.24.32.152 rport
50003
a=candidate:5 2 TCP-ACT 1684797438 172.24.32.152 50003 typ srflx raddr 172.24.32.152 rport
50003
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:KaSgBMqbVyQDtY12ihKmnNslPtpYnq1X7xko32nY|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:1smNZ23vqTBP4oQmBHJ5NsGbSJZG/BWgS6onqlV8|2^31|1:1
a=rtcp:50011
a=rtpmap:121 x-rtvc1/90000
a=rtpmap:34 H263/90000
a=encryption:optional
```

In the example above, the offerer is proposing audio and video as modalities. The offerer supports both SRTP and SSRTP as the mode for encryption, and proposes that in the SDP using the 'a=crypto' and 'a=cryptoscale' attributes. The offerer also wants to only encrypt the media optionally. This is described by specifying 'RTP/AVP' as the transport, even though there are 'a=crypto' and 'a=cryptoscale' attributes present in the SDP message.

The 'a=encryption' attribute is used to negotiate DES encryption, and used along with a 'k=' attribute. The values of the 'a=encryption' attribute only apply to the DES encryption.

Also note that RTAudio and RTVideo codecs are represented in the codec using dynamic payloads of 114, 115, and 121, and are identified using their encoding names of 'x-msrta' and 'x-rtvc1' in their corresponding 'a=rtpmap' attributes.

4.1.4 Client Receives Response with SSRTP to ICENAT-19 Offer

The following example is a response (SDP answer) received for the preceding offer.

```
v=0
o=- 0 0 IN IP4 172.24.32.125
s=session
c=IN IP4 172.24.32.125
b=CT:99980
t=0 0
m=audio 50018 RTP/SAVP 114 9 111 112 115 116 4 8 0 97 13 118 101
a=ice-ufrag:yYmQ
a=ice-pwd:T8P5yKtikiFpupO0pOqGatje
a=candidate:1 1 UDP 2130706431 172.24.32.125 50018 typ host
a=candidate:1 2 UDP 2130705918 172.24.32.125 50007 typ host
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:E8zKbdtM9sJdQenqGWVb3sYBp52rxFgS4uwMWy/k|2^31|1:1
a=remote-candidates:1 172.24.32.152 50005 2 172.24.32.152 50009
a=maxptime:200
a=rtcp:50007
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:4 G723/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:required
m=video 50002 RTP/SAVP 121 34
a=ice-ufrag: tTaJ
a=ice-pwd: 4jUT5Tp48gTR3iEvJiWVVDpG
a=x-caps:121
196611:640:480:25.0:600000:1;262148:352:288:15.0:256000:1;327685:176:144:15.0:180000:1
a=x-caps:34 65537:352:288:15.0:256000:1;131074:176:144:15.0:180000:1
a=candidate:1 1 UDP 2130706431 172.24.32.125 50002 typ host
a=candidate:1 2 UDP 2130705918 172.24.32.125 50008 typ host
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:o6ZolyOppaJqBxlYQ9/R4ykPCjgKDJMisiVvXSMb|2^31|1:1
a=remote-candidates:1 172.24.32.152 50012 2 172.24.32.152 50011
```

```
a=rtcp:50008
a=rtpmap:121 x-rtvc1/90000
a=rtpmap:34 H263/90000
a=encryption:required
```

The answerer (remote peer) also wants to encrypt the media using SRTP, so it replies with an SDP answer that has 'RTP/SAVP' in the transport in the 'm=' line.

The answerer wants to mandate the use of DES encryption, so it returns an 'a=encryption' attribute with a value of 'required'. Note that the media is still encrypted using SRTP because the 'a=encryption' attribute value only applies to DES encryption.

4.2 Encryption Using SRTP Examples that Demonstrate Extensions

Following are some examples. For brevity, only the pertinent portions of the SDP are displayed.

- Application optionally wanting to encrypt the media using either SRTP or Client Scale SRTP:

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

- Application optionally wanting to encrypt the media using either SRTP or Server SSRTP:

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

- Application optionally wanting to encrypt the media using only SRTP:

```
m=audio 50004 RTP/AVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

- Application compulsorily wanting to encrypt the media using either SRTP or Client Scale SRTP:

```
m=audio 50004 RTP/SAVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

- Application compulsorily wanting to encrypt the media using either SRTP or Server SSRTP:

```
m=audio 50004 RTP/SAVP 8 97 101
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
```

```
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

- Application compulsorily wanting to encrypt the media using only SRTP:

```
m=audio 50004 RTP/SAVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3 Offer/Answer Exchange for Various SRTP Encryption Scenarios

The following subsections contain examples. Only the relevant portion of the SDP message is included.

4.3.1 Offerer Wanting SRTP or Client Scale-SRTP Encryption Optionally and Answerer Wanting SRTP or Client Scale-SRTP Encryption Optionally

4.3.1.1 Offer

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3.1.2 Answer

```
m=audio 50004 RTP/SAVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:z8aIuyfeJZ2bkLVNPadciqjXwGMXo0s0IomrZr|2^31|1:1
```

4.3.1.3 Noteworthy points

- The answerer supported only SRTP or Client Scale SRTP. Thus, it responds only to the 'a=crypto' line of the offer. In this case, the offerer and answerer can only support the same flavor of the SSRTP, and SSRTP cannot be used.
- The answerer uses the same tag value for his 'a=crypto' attribute to signify that it is in response to the 'a=crypto' attribute with the same tag value in the offer.
- The answerer changes the transport profile from '**AVP**' to '**SAVP**' because both the offerer and answerer have negotiated SRTP for doing encryption.

4.3.2 Offerer Wanting SRTP or Client Scale-SRTP Optionally and Answerer Wanting SRTP or Server SSRTP Encryption Optionally

4.3.2.1 Offer

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3.2.2 Answer

```
m=audio 50004 RTP/SAVP 8 97 101
a=cryptoscale:1 server AES_CM_128_HMAC_SHA1_80
inline:Qr98aafIklbkPOReAKItaeUjXwZrOadI893ilaD|2^31|1:1
```

4.3.2.3 Noteworthy points

- The answerer supported only SRTP or server SSRTP, and thus responds only to the 'a=cryptoscale' line of the offer. In this case, the offerer and answerer can support the different types of SSRTP, and SSRTP can be used.
- The answerer uses the same tag value for his 'a=cryptoscale' attribute to signify that it is in response to the 'a=cryptoscale' attribute with the same tag value in the offer.
- The answerer changes the transport profile from '**AVP**' to '**SAVP**' because both the offerer and answerer have negotiated SSRTP for doing encryption.

4.3.3 Offerer Wanting SRTP or Client Scale-SRTP Encryption Optionally and Answerer Wanting SRTP Encryption Optionally

4.3.3.1 Offer

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3.3.2 Answer

```
m=audio 50004 RTP/SAVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:z8aIuyfeJZ2bkLVNPadciqjXwGMXo0s0IomrZr|2^31|1:1
```

4.3.3.3 Noteworthy points

- The answerer supported only SRTP, and thus responds only to the 'a=crypto' line of the offer.
- The answerer uses the same tag value for his 'a=crypto' attribute to signify that it is in response to the 'a=crypto' attribute with the same tag value in the offer.
- The answerer changes the transport profile from '**AVP**' to '**SAVP**' because both the offerer and answerer have negotiated SRTP for doing encryption.

4.3.4 Offerer Wanting SRTP or Client Scale-SRTP Encryption Optionally and Answerer Cannot Support SRTP or SSRTP Encryption

4.3.4.1 Offer

```
m=audio 50004 RTP/AVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3.4.2 Answer

```
m=audio 50004 RTP/AVP 8 97 101
```

4.3.4.3 Noteworthy points:

- The answerer cannot support SRTP or SSRTP and does not respond with any crypto or crypto scale attributes.

4.3.5 Offerer Wanting SRTP or Client Scale-SRTP Encryption Compulsorily and Answerer Wanting SRTP Encryption Optionally

4.3.5.1 Offer

```
m=audio 50004 RTP/SAVP 8 97 101
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:vV5wrmv9u07pd0QvyHw7rf6yL8e3xXt07AI74T3J|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:Oi0nVM8eJZ2bkLVNeRaqtUeqjXwGMXo0s0IrmoKh|2^31|1:1
```

4.3.5.2 Answer

```
m=audio 50004 RTP/SAVP 8 97 101
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:z8aIuyfeJZ2bkLVNPadciqjXwGMXo0s0IomrZr|2^31|1:1
```

4.3.5.3 Noteworthy points

- The Offerer wanted to encrypt compulsorily using SRTP or SSRTP, and thus set the transport profile to '**SAVP**'.
- The answerer supported only SRTP, and thus responds only to the 'a=crypto' line of the offer.
- The answerer uses the same tag value for the 'a=crypto' attribute to signify that it is in response to the 'a=crypto' attribute with the same tag value in the offer.

4.4 Restriction to the name and sampling rate for wide band comfort noise

Following is an example of an offer with support for comfort noise.

```
m=audio 57472 RTP/AVP 118 8 0 97 101
c=IN IP4172.29.106.5
a=rtpmap:118 CN/16000
```

4.5 Offer/Answer Exchange for application sharing

4.5.1 Offer

In the following example, the offerer is proposing application sharing as a modality in the role of a viewer.

```
m=applicationsharing 25865 TCP/RTP/SAVP 127
a=ice-ufrag:YVBHg
```

```
a=ice-pwd:ttsbflut41Em7/nM7qBatyZKEV
a=candidate:1 1 TCP-PASS 2120613887 157.56.65.134 7967 typ host
a=candidate:1 2 TCP-PASS 2120613374 157.56.65.134 7967 typ host
a=candidate:2 1 TCP-ACT 2121006591 157.56.65.134 25865 typ host
a=candidate:2 2 TCP-ACT 2121006078 157.56.65.134 25865 typ host
a=candidate:3 1 TCP-PASS 6556159 172.29.105.171 57506 typ relay raddr 172.29.105.171 rport 
57506
a=candidate:3 2 TCP-PASS 6556158 172.29.105.171 57506 typ relay raddr 172.29.105.171 rport 
57506
a=candidate:4 1 TCP-ACT 7076607 172.29.105.171 57506 typ relay raddr 172.29.105.171 rport 
57506
a=candidate:4 2 TCP-ACT 7076094 172.29.105.171 57506 typ relay raddr 172.29.105.171 rport 
57506
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80 
inline:/qIJxtX8+/VEpKGlTEgcQf84Hzq77umuaFL3y+fA|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:hhVTXYObDO1a5joyG5v0mnm+Djx7E6Hd01Y0Avkt|2^31|1:1
a=setup:passive
a=connection:new
a=rtcp:25865
a=mid:1
a=rtpmap:127 x-data/90000
a=x-applicationsharing-session-id:1
a=x-applicationsharing-role:sharer
a=x-applicationsharing-media-type:rdp
```

4.5.2 Answer

The answerer accepts the offer in the role of a sharer.

```
m=applicationsharing 53076 TCP/RTP/SAVP 127
c=IN IP4 172.29.105.171
a=rtpmap:127 x-data/90000
a=mid:1
a=connection:new
a=setup:active
a=rtcp:53076
a=ice-ufrag:A0nvw
a=ice-pwd:dp7UG//SD5FPVC7kD4San8b1YsHaL
a=candidate:1 1 tcp-pass 2120613887 172.29.105.158 57857 typ host raddr 172.29.105.158 rport 
57857
a=candidate:1 2 tcp-pass 2120613374 172.29.105.158 57857 typ host raddr 172.29.105.158 rport 
57857
a=candidate:2 1 tcp-act 2121006591 172.29.105.158 55959 typ host raddr 172.29.105.158 rport 
55959
a=candidate:2 2 tcp-act 2121006078 172.29.105.158 55959 typ host raddr 172.29.105.158 rport 
55959
a=candidate:3 1 tcp-pass 6555135 172.29.105.171 53076 typ relay raddr 172.29.105.171 rport 
53076
a=candidate:3 2 tcp-pass 6555134 172.29.105.171 53076 typ relay raddr 172.29.105.171 rport 
53076
a=candidate:4 1 tcp-act 7076607 172.29.105.171 53076 typ relay raddr 172.29.105.171 rport 
53076
a=candidate:4 2 tcp-act 7076094 172.29.105.171 53076 typ relay raddr 172.29.105.171 rport 
53076
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:WgJ76m2+jmIICUHA4wWyrpVJJBoMlgGDuY+1Jz5R|2^31|1:1
a=label:applicationsharing
a=x-applicationsharing-session-id:1
a=x-applicationsharing-role:viewer
```

```
a=x-applicationsharing-media-type:rdp
```

4.5.3 Noteworthy points

- The offerer has a role of a viewer while the answerer has a role of a sharer.
- The offerer wanted to encrypt compulsorily using SRTP, and thus set the transport profile as '**SAVP**'.(SSRTP is not used).
- The answerer supported only SRTP, and thus responds only to the 'a=crypto' line of the offer.
- The answerer uses the same tag value for the 'a=crypto' attribute to signify that it is in response to the 'a=crypto' attribute with the same tag value in the offer.
- RTP is used over [MS-ICE2] using TCP.

4.6 Offer/Answer Exchange with optimized media path to a gateway

This section describes examples of inbound and outbound calls between the client and a gateway with the media path bypassing OCS.

4.6.1 Incoming call from gateway to client

Note: There is a CONTENT-ID MIME header associated with each application/sdp and application/gw-sdp that are part of the multipart/alternative offer. In the following example, the application/GW-SDP offered to the client indicates that the gateway doesn't amplify media and its bypass id is 9CD08A01-E998-456a-AC8A-D028930E5933.

```
Content-Type: application/sdp
Content-ID: <f5806c1e-a58b-492f-a274-27e84ea28920>
Content-Disposition: Session;handling=optional;ms-proxy-2007fallback
v=0
o=- 5 0 IN IP4 192.168.104.102
s=session
c=IN IP4 192.168.104.102
b=CT:1000000
t=0 0
m=audio 56868 RTP/AVP 0 8 115 13 118 97 101
c=IN IP4 192.168.104.102
a=rtcp:56869
a=candidate:E3q9M8OJWFaigFVFtD0+u6FqPp0nkHYGAePLOMBTJRc 1 HYiiMeZUh7p4AUdo6XSncw UDP 0.830
192.168.104.102 56868
a=candidate:E3q9M8OJWFaigFVFtD0+u6FqPp0nkHYGAePLOMBTJRc 2 HYiiMeZUh7p4AUdo6XSncw UDP 0.830
192.168.104.102 56869
a=candidate:UzFFBI7awxelfHqPVFlhESQbd1jrYZ5PTn5+6tyH3aU 1 LF4n5rfHFi1/rLoHFWHUPw TCP 0.150
10.9.66.105 56821
a=candidate:UzFFBI7awxelfHqPVFlhESQbd1jrYZ5PTn5+6tyH3aU 2 LF4n5rfHFi1/rLoHFWHUPw TCP 0.150
10.9.66.105 56821
a=candidate:25u0MNHHZjaAh9RPPkpVe7Ba7EdCaxjUYRRqvoIfRkY 1 1sK0tfrBJVJiw820Lcvj3w UDP 0.450
10.9.66.105 59709
a=candidate:25u0MNHHZjaAh9RPPkpVe7Ba7EdCaxjUYRRqvoIfRkY 2 1sK0tfrBJVJiw820Lcvj3w UDP 0.450
10.9.66.105 52813
a=candidate:2DxliVgEarpkaYkb05bFTo8qq9e7BH3eW8ijJ2E3k4M 1 jJ5nCZyij2lvPO66RpXZpA TCP 0.250
192.168.104.102 55429
a=candidate:2DxliVgEarpkaYkb05bFTo8qq9e7BH3eW8ijJ2E3k4M 2 jJ5nCZyij2lvPO66RpXZpA TCP 0.250
192.168.104.102 55429
```

```
a=label:main-audio
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:2eyQLFO8vaoOX2GBLg9Qx9mMIJhsuGlL3Vfy65YG|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:QwzL7xoJ9BOMU50/FI72zI9Uh9jolO+LvBKmw+Q0|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:wBfC6An1JRyv1gwfQEkUnPekR6eGRVUyobeGbJHp|2^31
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:97 RED/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16,36
--aE6c0vI9iMfFYM08fnyeWG1occh4Naqt
Content-Type: application/sdp
Content-ID: <713e032e-fde6-48e5-83be-738f1bfdfe36>
v=0
o=- 6 0 IN IP4 192.168.104.102
s=session
c=IN IP4 192.168.104.102
b=CT:1000000
t=0 0
m=audio 51390 RTP/AVP 0 8 115 13 118 97 101
c=IN IP4 192.168.104.102
a=rtcp:51391
a=ice-ufrag:fAgr
a=ice-pwd:fUzyxypL9YjgIpFilSuHWHjW
a=candidate:1 1 UDP 2130706431 192.168.104.102 51390 typ host
a=candidate:1 2 UDP 2130705918 192.168.104.102 51391 typ host
a=candidate:2 1 tcp-pass 6555135 10.9.66.105 57678 typ relay raddr 192.168.104.102 rport
53641
a=candidate:2 2 tcp-pass 6555134 10.9.66.105 57678 typ relay raddr 192.168.104.102 rport
53641
a=candidate:3 1 UDP 16647679 10.9.66.105 53655 typ relay raddr 192.168.104.102 rport 55932
a=candidate:3 2 UDP 16647678 10.9.66.105 54870 typ relay raddr 192.168.104.102 rport 55933
a=candidate:4 1 tcp-act 7076863 10.9.66.105 57678 typ relay raddr 192.168.104.102 rport 53641
a=candidate:4 2 tcp-act 7076350 10.9.66.105 57678 typ relay raddr 192.168.104.102 rport 53641
a=candidate:5 1 tcp-act 1684797951 192.168.104.102 53641 typ srflx raddr 192.168.104.102
rport 53641
a=candidate:5 2 tcp-act 1684797438 192.168.104.102 53641 typ srflx raddr 192.168.104.102
rport 53641
a=label:main-audio
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:2eyQLFO8vaoOX2GBLg9Qx9mMIJhsuGlL3Vfy65YG|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:QwzL7xoJ9BOMU50/FI72zI9Uh9jolO+LvBKmw+Q0|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:wBfC6An1JRyv1gwfQEkUnPekR6eGRVUyobeGbJHp|2^31
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:97 RED/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16,36
--aE6c0vI9iMfFYM08fnyeWG1occh4Naqt
Content-Type: application/gw-sdp; x-bypassid=9CD08A01-E998-456a-AC8A-D028930E5933
Content-ID: <bcb3fcca-1dc7-4632-aefc-3d4e9947c64f>
```

```
Content-Disposition: Session;handling=optional
v=0
o=PSTNgateway1 1344430046 1344429731 IN IP4 192.168.107.12
s=session
c=IN IP4 192.168.107.12
t=0 0
m=audio 6390 RTP/SAVP 0 8 4 2 3 101
c=IN IP4 192.168.107.12
a=rtcp:6391
a=x-bypassid:9CD08A01-E998-456a-AC8A-D028930E5933
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:EtVylZp2HonR5Vwd7PFV8kKILnC4P3sKILMY3mAy|2^31|203:1
a=sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=fmtp:4 annexa=no
a=rtpmap:2 G726-32/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=x-mediasettings:signalboostunsupported
--aE6c0vI9iMfFYM08fnyeWG1occh4Naqt—
```

The following code is the answer from the OC, assuming it is in the same location as the gateway and has selected the gateway SDP.

```
ms-accepted-content-id: <bcb3fcca-1dc7-4632-aefc-3d4e9947c64f>
v=0
o=- 0 0 IN IP4 192.168.40.165
s=session
c=IN IP4 192.168.40.165
b=CT:99980
t=0 0
m=audio 28636 RTP/SAVP 0 8 4 101
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:2y9Pl2hgW0bgz/t8CRurDcRQjjOmEpqbztRk2O/L|2^31|1:1
a=maxptime:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=x-bypass
```

4.6.2 Outbound call from client to gateway

In the following example, the offer from the client indicates that its bypass id is 9CD08A01-E998-456a-AC8A-D028930E5933.

```
------=_NextPart_000_0754_01CAA68D.A421F990
Content-Type: application/sdp
Content-Transfer-Encoding: 7bit
Content-ID: <b36a0b797c2d448684b4cd88213e687b>
```

```
Content-Disposition: session; handling=optional; ms-proxy-2007fallback
v=0
o=- 0 0 IN IP4 192.168.40.165
s=session
c=IN IP4 192.168.40.165
b=CT:99980
t=0 0
m=audio 31984 RTP/AVP 114 9 112 111 0 8 116 115 4 97 13 118 101
a=candidate:Eo6N4ZUF5f08I+5P0uqR1tY20IRoszjUqaAq/X2kIts 1 U3pPAm1UlyRGOdhpy2femA UDP 0.830
192.168.40.165 31984
a=candidate:Eo6N4ZUF5f08I+5P0uqR1tY20IRoszjUqaAq/X2kIts 2 U3pPAm1UlyRGOdhpy2femA UDP 0.830
192.168.40.165 31985
a=candidate:3DOp61IDKKyJMLBEbV3e1xQLe4NJD1SlXVzafFyiqgk 1 Obw5WAGIyt1kU/zo7ons/Q TCP 0.190
10.9.66.105 59349
a=candidate:3DOp61IDKKyJMLBEbV3e1xQLe4NJD1SlXVzafFyiqgk 2 Obw5WAGIyt1kU/zo7ons/Q TCP 0.190
10.9.66.105 59349
a=candidate:PzI3B9tYBN+qhYwJcb0j0C42c5ZTR5TyoDWRfb7yXXk 1 eiWWwDXKkSxP58wBK+R/hQ UDP 0.490
10.9.66.105 51744
a=candidate:PzI3B9tYBN+qhYwJcb0j0C42c5ZTR5TyoDWRfb7yXXk 2 eiWWwDXKkSxP58wBK+R/hQ UDP 0.490
10.9.66.105 52795
a=candidate:2kUGrKpDjD4YDF2AS9k1NvGLoCeIEYHSaUfgLxEfdcQ 1 ZFTOm8nfx79vTVzbFxMaKQ TCP 0.250
192.168.40.165 16567
a=candidate:2kUGrKpDjD4YDF2AS9k1NvGLoCeIEYHSaUfgLxEfdcQ 2 ZFTOm8nfx79vTVzbFxMaKQ TCP 0.250
192.168.40.165 16567
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:8XdWiybvpW9FAj7ItDedcqhWjHCKr7gCVq0q56ek|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:s3BkRJUaog532qlFBRdFTpcSCYhoa/hwzr8wV39v|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:nYToZWyhCoDcl/CQLFTE4bTJiCv8YqDlnfF9CVv/|2^31
a=maxptime:200
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=fmtp:9 bitrate=64000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:4 G723/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:optional
a=x-bypassid:9CD08A01-E998-456a-AC8A-D028930E5933
------=_NextPart_000_0754_01CAA68D.A421F990
Content-Type: ap
TL_INFO(TF_PROTOCOL) [1]0BDC.0BC8::02/06/2010-01:57:30.724.00073d14
(S4,SipMessage.DataLoggingHelper:sipmessage.cs(581))$$$SplitMessageSegmentBegin$$$Transfer-
Encoding: 7bit
Content-ID: <6d62e6a07ddd4ddbab5d9f6474f4175c>
Content-Disposition: session; handling=optional
v=0
o=- 0 0 IN IP4 192.168.40.165
s=session
```

```
c=IN IP4 192.168.40.165
b=CT:99980
t=0 0
m=audio 13510 RTP/AVP 114 9 112 111 0 8 116 115 4 97 13 118 101
a=ice-ufrag:oTw+
a=ice-pwd:J7jponEfEPTn6YX8lbeaImJh
a=candidate:1 1 UDP 2130706431 192.168.40.165 13510 typ host
a=candidate:1 2 UDP 2130705918 192.168.40.165 13511 typ host
a=candidate:2 1 TCP-PASS 6556159 10.9.66.105 56378 typ relay raddr 192.168.40.165 rport 12134
a=candidate:2 2 TCP-PASS 6556158 10.9.66.105 56378 typ relay raddr 192.168.40.165 rport 12134
a=candidate:3 1 UDP 16648703 10.9.66.105 58427 typ relay raddr 192.168.40.165 rport 17214
a=candidate:3 2 UDP 16648702 10.9.66.105 52415 typ relay raddr 192.168.40.165 rport 17215
a=candidate:4 1 TCP-ACT 7076863 10.9.66.105 56378 typ relay raddr 192.168.40.165 rport 12134
a=candidate:4 2 TCP-ACT 7076350 10.9.66.105 56378 typ relay raddr 192.168.40.165 rport 12134
a=candidate:5 1 TCP-ACT 1684797951 192.168.40.165 12134 typ srflx raddr 192.168.40.165 rport
12134
a=candidate:5 2 TCP-ACT 1684797438 192.168.40.165 12134 typ srflx raddr 192.168.40.165 rport
12134
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:8XdWiybvpW9FAj7ItDedcqhWjHCKr7gCVq0q56ek|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:s3BkRJUaog532qlFBRdFTpcSCYhoa/hwzr8wV39v|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:nYToZWyhCoDcl/CQLFTE4bTJiCv8YqDlnfF9CVv/|2^31
a=maxptime:200
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=fmtp:9 bitrate=64000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:4 G723/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:optional
a=x-bypassid:9CD08A01-E998-456a-AC8A-D028930E5933
------=_NextPart_000_0754_01CAA68D.A421F990—
```

The following example is the answer received by the OC, assuming the gateway is in the same location as OC and has opted to bypass.

```
Ms-Accepted-Content-ID: <6d62e6a07ddd4ddbab5d9f6474f4175c>
Rseq: 1
v=0
o=PSTNgateway1 696126319 696126004 IN IP4 192.168.107.12
s=session
c=IN IP4 192.168.107.12
t=0 0
```

```
m=audio 6470 RTP/SAVP 0 101
c=IN IP4 192.168.107.12
a=rtcp:6471
a=x-bypass
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:SnMuCMywfjsGUGMiv2Q7aky9OFeHzcZ35VgIllsv|2^31|129:1
a=sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=x-mediasettings:signalboostunsupported
```

4.7 Extensions for music-on-hold

Note that the following examples illustrate SDP only. They do not show sip.rendering in the SIP Contact header.

4.7.1 Offer specifying music-on-hold

```
m=audio 52033 RTP/SAVP 114 111 112 115 116 4 8 0 97 13 118 101
a=sendonly
a=feature:MoH
```

This offer includes a=sendonly and a=feature:MoH under the m=audio line, indicating that that audio channel is streaming music-on-hold.

4.7.2 Offer removing music-on-hold

```
m=audio 52033 RTP/SAVP 114 111 112 115 116 4 8 0 97 13 118 101
a=sendrecv
```

This offer has a=sendrecv and no a=feature:MoH, indicating that the audio session is no longer on hold (and is no longer streaming music-on-hold).

5 Security

5.1 Security Considerations for Implementers

Although media encryption is supported, the exchange of encryption information to encrypt the media is not encrypted. To protect the encryption information during the exchange, the application can use TLS to carry the SIP traffic. Any other security considerations are covered by SIP and SDP.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Office Communications Server 2007
- Microsoft® Office Communications Server 2007 R2
- Microsoft® Office Communicator 2007
- Microsoft® Office Communicator 2007 R2
- Microsoft® Lync™ Server 2010
- Microsoft® Lync™ 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 3.1.5.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, these parameters are required.

<2> Section 3.1.5.5: This restriction only applies to Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2.

<3> Section 3.1.5.7: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the name of the payload used for comfort noise is required to be "CN".

<4> Section 3.1.5.7: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the sampling rate is required to be 8,000 or 16,000.

<5> Section 3.1.5.9: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the connection-oriented transport cannot be used when ICE is not enabled on the offering application and the answering application. This applies to any offer or answer received from an application that does not support ICE.

<6> Section 3.1.5.9: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other product, application sharing is required to use ICE over TCP.

<7> Section 3.1.5.9: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<8> Section 3.1.5.10.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported. For all other products, when used in the context of this protocol, the 'a=connection' attribute is required to have the value 'existing.'

<9> Section 3.1.5.12.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported. For all other products, when the offer is forked, SDP answers not in reliable provisional responses are required to be sent only from a 0 or 1 device.

<10> Section 3.1.5.12.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<11> Section 3.1.5.12.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<12> Section 3.1.5.12.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<13> Section 3.1.5.12.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<14> Section 3.1.5.12.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, if an answer was contained in an 18x-level, it is required to be repeated (without any changes) in the 200 for the same fork.

<15> Section 3.1.5.12.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, if the call is not forked, the SDP answer received in the final response (200) is required to be the same as the one received in the provisional response (18x).

<16> Section 3.1.5.12.4: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported. For all other products, when a SIP INVITE request is NOT forked and an SDP answer is received in the provisional response, ICE processing is required to proceed as if the SDP was received in the final response.

<17> Section 3.1.5.13: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products are required to follow the exceptions as specified.

<18> Section 3.1.5.13: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products are required to follow the exceptions as specified.

<19> Section 3.1.5.13: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<20> Section 3.1.5.16: Lync Server 2010, Lync 2010: This behavior is not supported.

<21> Section 3.1.5.19.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products MUST include this attribute.

<22> Section 3.1.5.19.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products, the sharing party MUST set the role to "sharer."

<23> Section 3.1.5.19.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products, the viewing party MUST set the role to "viewer."

<24> Section 3.1.5.19.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the (session-id, role) pair is required to be unique for each active m=line.

<25> Section 3.1.5.19.4: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the 'a=mid' attribute is required.

<26> Section 3.1.5.22.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other product are required to gather the RELAYED, SERVER REFLEXIVE candidates and perform connectivity checks.

<27> Section 3.1.5.22.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products are not allowed to generate an SDP with the ice-options attribute.

<28> Section 3.1.5.22.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. All other products are not allowed to generate an SDP with the ice-mismatch attribute.

<29> Section 3.1.5.22.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the default destination for a media component is required to have a corresponding candidate attribute.

<30> Section 3.1.5.22.4: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<31> Section 3.1.5.23.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, for audio/video calls, the default candidate is not allowed to be TCP.

<32> Section 3.1.5.23.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, for application sharing calls, the default candidate is required to be TCP.

<33> Section 3.1.5.23.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, for application sharing media type, the local candidates are required to be TCP.

<34> Section 3.1.5.24.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the client is required to change the direction of all streams to inactive.

<35> Section 3.1.5.24.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<36> Section 3.1.5.24.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported. For all other products, the client is required to change the direction of all streams to sendrecv.

<37> Section 3.1.5.24.2: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<38> Section 3.1.5.25: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<39> Section 3.1.5.25: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2 This behavior is not supported. For all other products, the protocol peer is required to follow the requirements listed.

<40> Section 3.1.5.26: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2 This behavior is not supported.

<41> Section 3.1.5.27: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<42> Section 4.1.1: Lync Server 2010, Lync 2010: This behavior is not supported.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

N

O

P

R

S

T

V

Release: Sunday, December 19, 2010