Network Working Group
Internet-Draft
Updates: 4568,4585 (if approved)
Intended status: Standards Track
Expires: July 27, 2017

A. Hutton
Unify
R. Jesske
Deutsche Telekom
A. Johnston
Unaffiliated
G. Salgueiro
Cisco
B. Aboba
Microsoft

January 23, 2017

Negotiating SRTP and RTCP Feedback using the RTP/AVP Profile draft-hutton-mmusic-opportunistic-negotiation-00

Abstract

This document describes how the use of the Secure Real-time transport protocol (SRTP) [RFC3711]. can be negotiated using the AVP (Audio Video Profile) defined in [RFC3551]. Such a mechanism is used to provide a means for encrypted media to be used in environments where support for encryption is not known in advance, and not required. The same mechanism is also applied to negotiation of the Extended RTP Profile for Real-time Transport Control Protocol Based Feedback (RTP/AVPF) [RFC4585].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 27, 2017.

Table of Contents

1. Introduction

Opportunistic Security [RFC7435] is an approach to security that defines a third mode for security between "cleartext" and "comprehensive protection" that allows encryption and authentication to be used if supported but will not result in failures if it is not supported. In terms of secure media, cleartext is RTP [RFC3550] media which is negotiated with the AVP (Audio Video Profile) profile defined [RFC3551]. Comprehensive protection is Secure RTP [RFC3711], negotiated with a secure profile, such as SAVP or SAVPF [RFC5124].

[I-D.ietf-sipbrandy-osrtp] describes how Secure Real-time transport protocol (SRTP) can be negotiated opportunistically.

[RFC4568] however requires that SRTP is only negotiated using the RTP/SAVP profile [RFC3711] or the RTP/SAVPF profile [RFC5124]. This document relaxes this rule by allowing SRTP to be used with the RTP/AVP profile when negotiated opportunistically.

Similarly [RFC4585] requires that the RTCP extended reports are only used in media sessions for which the "AVPF" profile is specified. This document therefore also relaxes this rule allowing RTCP based feedback to be used with the RTP/AVP profile.

2. Normative Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

3. Motivation

In theory SDP [RFC4566] allows different RTP profiles such as SAVP, AVPF, and AVP to be offered as separate m-lines, and allows the answerer to reject profiles it does not support or does not wish to use. However the use of multiple m-lines for such a negotiation is not well defined and implementations receiving such an offer are likely to reject the SDP Offer rather than use the profile they support. This negotiation failure has been observed when negotiating the secure profile (SAVP) and also when negotiating RTCP based feedback messages [RFC4585] (RTP/AVPF) or both (RTP/SAVPF).

To avoid using multiple m-lines to negotiate RTP profiles this draft recognized that existing implementation of SRTP, and RTCP feedback, make use of the relevant SDP attributes to indicate such capabilities. The approach therefore taken in this draft uses the "a=" lines in SDP to negotiate these capabilities in a single offer/answer exchange, by offering the AVP profile but indicating the supported functionality in a=lines.

4. Use of RTP/AVP profile with SRTP

To negotiate SRTP in an opportunistic way such as that described in [I-D.ietf-sipbrandy-osrtp] requires a fallback to unencrypted media to occur if the remote endpoint does not support SRTP.

Therefore when negotiating SRTP opportunistically the SDP offerer MUST use the AVP profile [RFC3551]. This is independent of the key exchange mechanism used.

The SDP answerer will use the AVP profile if it does not encrypt the media and may use the AVP if it encrypts the media. The exact negotiation mechanism is however outside the scope of this document, an example mechanism can be found in [I-D.ietf-sipbrandy-osrtp].

Therefore when negotiating SRTP opportunistically the SDP offerer MUST use the AVP profile [RFC3551]. This is independent of the key exchange mechanism used.

5. Use of RTP/AVP profile with RTCP Feedback

Negotiating the use of the Extended RTP Profile for RTCP Based Feedback (RTP/AVPF) [RFC4585] opportunistically also requires the offerer to use the AVP profile otherwise the offer is likely to be rejected by an answerer who does not support AVPF.

Therefore when negotiating RTCP Based Feedback opportunistically the SDP offerer MUST use the AVP profile [RFC3551] and include the "a=rtcp-fb" SDP attribute as described in [RFC4585]. This is an update to [RFC4585] which requires that the "a=rtcp-fb" attribute is only used with the AVPF profile. All other [RFC4585] procedures remain unchanged.

6. IANA Considerations

None

7. Security Considerations

The security considerations of [RFC7435] apply to any opportunistic approach to SRTP.

It is important to note that negotiating SRTP in an opportunistic way makes no changes, and has no effect on media sessions in which the offer contains a secure profile of RTP, such as SAVP or SAVPF. As discussed in [RFC7435] this is the "comprehensive protection" for media mode.

8. Acknowledgements

This document is dedicated to our friend and colleague Francois Audet who is greatly missed in our community. His work on improving security in SIP and RTP provided the foundation for this work.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[I-D.ietf-sipbrandy-osrtp]
Johnston, A., Aboba, B., Hutton, A., Jesske, R., and T. Stach, "An Opportunistic Approach for Secure Real-time Transport Protocol (OSRTP)", draft-ietf-sipbrandy-osrtp-01 (work in progress), October 2016.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <http://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <http://www.rfc-editor.org/info/rfc4568>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <http://www.rfc-editor.org/info/rfc4585>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <http://www.rfc-editor.org/info/rfc5124>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

Authors' Addresses

Andrew Hutton
Unify
Brickhill Street
Milton Keynes MK15 0DJ
UK

Email: andrew.hutton@unify.com

Roland Jesske
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt 64295
Germany

Email: R.Jesske@telekom.de

Alan Johnston
Unaffiliated
Bellevue, WA
USA

Email: alan.b.johnston@gmail.com

Gonzalo Salgueiro
Cisco
7200-12 Kit Creek Road
RTP, NC 27709
USA

Email: gsalguei@cisco.com

Bernard Aboba
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

Email: bernard.aboba@gmail.com

Network Working Group
Internet-Draft
Obsoletes: 4572 (if approved)
Intended status: Standards Track
Expires: August 6, 2017

J. Lennox
Vidyo
C. Holmberg
Ericsson

February 2, 2017

Connection-Oriented Media Transport over TLS in SDP
draft-ietf-mmusic-4572-update-13

Abstract

This document specifies how to establish secure connection-oriented media transport sessions over the Transport Layer Security (TLS) protocol using the Session Description Protocol (SDP). It defines the SDP protocol identifier, 'TCP/TLS'. It also defines the syntax and semantics for an SDP 'fingerprint' attribute that identifies the certificate that will be presented for the TLS session. This mechanism allows media transport over TLS connections to be established securely, so long as the integrity of session descriptions is assured.

This document obsoletes RFC 4572, by clarifying the usage of multiple fingerprints.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 6, 2017.

Copyright Notice

Table of Contents

1. Introduction

The Session Description Protocol (SDP) [8] provides a general-purpose format for describing multimedia sessions in announcements or invitations. For many applications, it is desirable to establish, as part of a multimedia session, a media stream that uses a connection-oriented transport. RFC 4145, Connection-Oriented Media Transport in the Session Description Protocol (SDP) [7], specifies a general mechanism for describing and establishing such connection-oriented streams; however, the only transport protocol it directly supports is TCP. In many cases, session participants wish to provide confidentiality, data integrity, and authentication for their media sessions. This document therefore extends the Connection-Oriented

Media specification to allow session descriptions to describe media sessions that use the Transport Layer Security (TLS) protocol [10].

The TLS protocol allows applications to communicate over a channel that provides confidentiality and data integrity. The TLS specification, however, does not specify how specific protocols establish and use this secure channel; particularly, TLS leaves the question of how to interpret and validate authentication certificates as an issue for the protocols that run over TLS. This document specifies such usage for the case of connection-oriented media transport.

Complicating this issue, endpoints exchanging media will often be unable to obtain authentication certificates signed by a well-known root certification authority (CA). Most certificate authorities charge for signed certificates, particularly host-based certificates; additionally, there is a substantial administrative overhead to obtaining signed certificates, as certification authorities must be able to confirm that they are issuing the signed certificates to the correct party. Furthermore, in many cases endpoints' IP addresses and host names are dynamic: they may be obtained from DHCP, for example. It is impractical to obtain a CA-signed certificate valid for the duration of a DHCP lease. For such hosts, self-signed certificates are usually the only option. This specification defines a mechanism that allows self-signed certificates can be used securely, provided that the integrity of the SDP description is assured. It provides for endpoints to include a secure hash of their certificate, known as the "certificate fingerprint", within the session description. Provided that the fingerprint of the offered certificate matches the one in the session description, end hosts can trust even self-signed certificates.

The rest of this document is laid out as follows. An overview of the problem and threat model is given in Section 3. Section 4 gives the basic mechanism for establishing TLS-based connected-oriented media in SDP. Section 5 describes the SDP fingerprint attribute, which, assuming that the integrity of SDP content is assured, allows the secure use of self-signed certificates. Section 6 describes which X.509 certificates are presented, and how they are used in TLS. Section 7 discusses additional security considerations.

1.1. Changes From RFC 4572

This document obsoletes RFC 4572 [20] but remains backwards compatible with older implementations. The changes from [20] are that it clarifies that multiple 'fingerprint' attributes can be used to carry fingerprints, calculated using different hash functions, associated with a given certificate, and to carry fingerprints

associated with multiple certificates. The fingerprint matching procedure, when multiple fingerprints are provided, are also clarified. The document also updates the preferred hash function with a stronger cipher suite, and removes the requirement to use the same hash function for calculating a certificate fingerprint and certificate signature.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

3. Overview

This section discusses the threat model that motivates TLS transport for connection-oriented media streams. It also discusses in more detail the need for end systems to use self-signed certificates.

3.1. SDP Operational Modes

There are two principal operational modes for multimedia sessions: advertised and offer-answer. Advertised sessions are the simpler mode. In this mode, a server publishes, in some manner, an SDP session description of a multimedia session it is making available. The classic example of this mode of operation is the Session Announcement Protocol (SAP) [15], in which SDP session descriptions are periodically transmitted to a well-known multicast group. Traditionally, these descriptions involve multicast conferences, but unicast sessions are also possible. (Connection-oriented media, obviously, cannot use multicast.) Recipients of a session description connect to the addresses published in the session description. These recipients may not previously have been known to the advertiser of the session description.

Alternatively, SDP conferences can operate in offer-answer mode [4]. This mode allows two participants in a multimedia session to negotiate the multimedia session between them. In this model, one participant offers the other a description of the desired session from its perspective, and the other participant answers with the desired session from its own perspective. In this mode, each of the participants in the session has knowledge of the other one. This is the mode of operation used by the Session Initiation Protocol (SIP) [17].

3.2. Threat Model

Participants in multimedia conferences often wish to guarantee confidentiality, data integrity, and authentication for their media sessions. This section describes various types of attackers and the ways they attempt to violate these guarantees. It then describes how the TLS protocol can be used to thwart the attackers.

The simplest type of attacker is one who listens passively to the traffic associated with a multimedia session. This attacker might, for example, be on the same local-area or wireless network as one of the participants in a conference. This sort of attacker does not threaten a connection's data integrity or authentication, and almost any operational mode of TLS can provide media stream confidentiality.

More sophisticated is an attacker who can send his own data traffic over the network, but who cannot modify or redirect valid traffic. In SDP's 'advertised' operational mode, this can barely be considered an attack; media sessions are expected to be initiated from anywhere on the network. In SDP's offer-answer mode, however, this type of attack is more serious. An attacker could initiate a connection to one or both of the endpoints of a session, thus impersonating an endpoint, or acting as a man in the middle to listen in on their communications. To thwart these attacks, TLS uses endpoint certificates. So long as the certificates' private keys have not been compromised, the endpoints have an external trusted mechanism (most commonly, a mutually-trusted certification authority) to validate certificates, and the endpoints know what certificate identity to expect, endpoints can be certain that such an attack has not taken place.

Finally, the most serious type of attacker is one who can modify or redirect session descriptions: for example, a compromised or malicious SIP proxy server. Neither TLS itself nor any mechanisms that use it can protect an SDP session against such an attacker. Instead, the SDP description itself must be secured through some mechanism; SIP, for example, defines how S/MIME [22] can be used to secure session descriptions.

3.3. The Need for Self-Signed Certificates

SDP session descriptions are created by any endpoint that needs to participate in a multimedia session. In many cases, such as SIP phones, such endpoints have dynamically-configured IP addresses and host names and must be deployed with nearly zero configuration. For such an endpoint, it is for practical purposes impossible to obtain a certificate signed by a well-known certification authority.

If two endpoints have no prior relationship, self-signed certificates cannot generally be trusted, as there is no guarantee that an attacker is not launching a man-in-the-middle attack. Fortunately, however, if the integrity of SDP session descriptions can be assured, it is possible to consider those SDP descriptions themselves as a prior relationship: certificates can be securely described in the session description itself. This is done by providing a secure hash of a certificate, or "certificate fingerprint", as an SDP attribute; this mechanism is described in Section 5.

3.4. Example SDP Description for TLS Connection

Figure 1 illustrates an SDP offer that signals the availability of a T.38 fax session over TLS. For the purpose of brevity, the main portion of the session description is omitted in the example, showing only the 'm' line and its attributes. (This example is the same as the first one in RFC 4145 [7], except for the proto parameter and the fingerprint attribute.) See the subsequent sections for explanations of the example's TLS-specific attributes.

(Note: due to RFC formatting conventions, this document splits SDP across lines whose content would exceed 72 characters. A backslash character marks where this line folding has taken place. This backslash and its trailing CRLF and whitespace would not appear in actual SDP content.)

```
m=image 54111 TCP/TLS t38
c=IN IP4 192.0.2.2
a=setup:passive
a=connection:new
a=fingerprint:SHA-256 \
   12:DF:3E:5D:49:6B:19:E5:7C:AB:4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF: \
   3E:5D:49:6B:19:E5:7C:AB:4A:AD
a=fingerprint:SHA-1 \
   4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

Figure 1: Example SDP Description Offering a TLS Media Stream

4. Protocol Identifiers

The 'm' line in SDP specifies, among other items, the transport protocol to be used for the media in the session. See the "Media Descriptions" section of SDP [8] for a discussion on transport protocol identifiers.

This specification defines the protocol identifier, 'TCP/TLS', which indicates that the media described will use the Transport Layer Security protocol [10] over TCP. (Using TLS over other transport

protocols is not discussed in this document.) The 'TCP/TLS' protocol identifier describes only the transport protocol, not the upper-layer protocol. An 'm' line that specifies 'TCP/TLS' MUST further qualify the protocol using a fmt identifier to indicate the application being run over TLS.

Media sessions described with this identifier follow the procedures defined in RFC 4145 [7]. They also use the SDP attributes defined in that specification, 'setup' and 'connection'.

5. Fingerprint Attribute

Parties to a TLS session indicate their identities by presenting authentication certificates as part of the TLS handshake procedure. Authentication certificates are X.509 [2] certificates, as profiled by RFC 3279 [5], RFC 5280 [11], and RFC 4055 [6].

In order to associate media streams with connections and to prevent unauthorized barge-in attacks on the media streams, endpoints MUST provide a certificate fingerprint. If the X.509 certificate presented for the TLS connection matches the fingerprint presented in the SDP, the endpoint can be confident that the author of the SDP is indeed the initiator of the connection.

A certificate fingerprint is a secure one-way hash of the DER (distinguished encoding rules) form of the certificate. (Certificate fingerprints are widely supported by tools that manipulate X.509 certificates; for instance, the command "openssl x509 -fingerprint" causes the command-line tool of the openssl package to print a certificate fingerprint, and the certificate managers for Mozilla and Internet Explorer display them when viewing the details of a certificate.)

A fingerprint is represented in SDP as an attribute (an 'a' line). It consists of the name of the hash function used, followed by the hash value itself. The hash value is represented as a sequence of uppercase hexadecimal bytes, separated by colons. The number of bytes is defined by the hash function. (This is the syntax used by openssl and by the browsers' certificate managers. It is different from the syntax used to represent hash values in, e.g., HTTP digest authentication [24], which uses unseparated lowercase hexadecimal bytes. It was felt that consistency with other applications of fingerprints was more important.)

The formal syntax of the fingerprint attribute is given in Augmented Backus-Naur Form [9] in Figure 2. This syntax extends the BNF syntax of SDP [8].

```
attribute              =/ fingerprint-attribute

fingerprint-attribute  =  "fingerprint" ":" hash-func SP fingerprint

hash-func              =  "sha-1" / "sha-224" / "sha-256" /
                          "sha-384" / "sha-512" /
                          "md5" / "md2" / token
                          ; Additional hash functions can only come
                          ; from updates to RFC 3279

fingerprint            =  2UHEX *(":" 2UHEX)
                          ; Each byte in upper-case hex, separated
                          ; by colons.

UHEX                   =  DIGIT / %x41-46 ; A-F uppercase
```

Figure 2: Augmented Backus-Naur Syntax for the Fingerprint Attribute

Following RFC 3279 [5] as updated by RFC 4055 [6], therefore, the defined hash functions are 'SHA-1' [1] [16], 'SHA-224' [1], 'SHA-256' [1], 'SHA-384'[1], 'SHA-512' [1], 'MD5' [13] and 'MD2' [23], with 'SHA-256' preferred. A new IANA registry of Hash Function Textual Names, specified in Section 8, allows for addition of future tokens, but they may only be added if they are included in RFCs that update or obsolete RFC 3279 [5].

Implementations compliant to this specification MUST NOT use the MD2 and MD5 hash functions to calculate fingerprints, or to verify received fingerprints that have been calculated using them.

NOTE: The MD2 and MD5 hash functions are listed in this specification so that implementations can recognize them. Implementations that log unused hash functions might log occurrences of these algorithms differently to unknown hash algorithms.

The fingerprint attribute may be either a session-level or a media-level SDP attribute. If it is a session-level attribute, it applies to all TLS sessions for which no media-level fingerprint attribute is defined.

5.1. Multiple Fingerprints

Multiple SDP fingerprint attributes can be associated with an 'm' line. This can occur if multiple fingerprints have been calculated for a certificate using different hash functions. It can also occur if one or more fingerprints associated with multiple certificates have been calculated. This might be needed if multiple certificates will be used for media associated with an 'm' line (e.g., if separate

certificates are used for RTP and RTCP), or where it is not known which certificate will be used when the fingerprints are exchanged. In such cases, one or more fingerprints MUST be calculated for each possible certificate.

An endpoint MUST, as a minimum, calculate a fingerprint using both the 'SHA-256' hash function algorithm and the hash function used to generate the signature on the certificate for each possible certificate. Including the hash from the signature algorithm ensures interoperability with strict implementations of RFC 4572 [20]. Either of these fingerprints MAY be omitted if the endpoint includes a hash with a stronger hash algorithm that it knows that the peer supports, if it is known that the peer does not support the hash algorithm, or if local policy mandates use of stronger algorithms.

If fingerprints associated with multiple certificates are calculated, the same set of hash functions MUST be used to calculate fingerprints for each certificate associated with the 'm' line.

An endpoint MUST select the set of fingerprints which use its most preferred hash function (out of those offered by the peer) and verify that each certificate used matches one fingerprint out of that set. If a certificate does not match any such fingerprint, the endpoint MUST NOT establish the TLS connection.

NOTE: The SDP fingerprint attribute does not contain a reference to a specific certificate. Endpoints need to compare the fingerprint with a certificate hash in order to look for a match.

6. Endpoint Identification

6.1. Certificate Choice

An X.509 certificate binds an identity and a public key. If SDP describing a TLS session is transmitted over a mechanism that provides integrity protection, a certificate asserting any syntactically valid identity MAY be used. For example, an SDP description sent over HTTP/TLS [14] or secured by S/MIME [22] MAY assert any identity in the certificate securing the media connection.

Security protocols that provide only hop-by-hop integrity protection (e.g., the sips protocol [17], SIP over TLS) are considered sufficiently secure to allow the mode in which any valid identity is accepted. However, see Section 7 for a discussion of some security implications of this fact.

In situations where the SDP is not integrity-protected, however, the certificate provided for a TLS connection MUST certify an appropriate

identity for the connection. In these scenarios, the certificate presented by an endpoint MUST certify either the SDP connection address, or the identity of the creator of the SDP message, as follows:

- If the connection address for the media description is specified as an IP address, the endpoint MAY use a certificate with an iPAddress subjectAltName that exactly matches the IP in the connection-address in the session description's 'c' line. Similarly, if the connection address for the media description is specified as a fully-qualified domain name, the endpoint MAY use a certificate with a dNSName subjectAltName matching the specified 'c' line connection-address exactly. (Wildcard patterns MUST NOT be used.)

- Alternately, if the SDP session description of the session was transmitted over a protocol (such as SIP [17]) for which the identities of session participants are defined by uniform resource identifiers (URIs), the endpoint MAY use a certificate with a uniformResourceIdentifier subjectAltName corresponding to the identity of the endpoint that generated the SDP. The details of what URIs are valid are dependent on the transmitting protocol. (For more details on the validity of URIs, see Section 7.)

Identity matching is performed using the matching rules specified by RFC 5280 [11]. If more than one identity of a given type is present in the certificate (e.g., more than one dNSName name), a match in any one of the set is considered acceptable. To support the use of certificate caches, as described in Section 7, endpoints SHOULD consistently provide the same certificate for each identity they support.

6.2. Certificate Presentation

In all cases, an endpoint acting as the TLS server (i.e., one taking the 'setup:passive' role, in the terminology of connection-oriented media) MUST present a certificate during TLS initiation, following the rules presented in Section 6.1. If the certificate does not match the original fingerprint, the client endpoint MUST terminate the media connection with a bad_certificate error.

If the SDP offer/answer model [4] is being used, the client (the endpoint with the 'setup:active' role) MUST also present a certificate following the rules of Section 6.1. The server MUST request a certificate, and if the client does not provide one, or if the certificate does not match a provided fingerprint, the server endpoint MUST terminate the media connection with a bad_certificate error.

Note that when the offer/answer model is being used, it is possible for a media connection to outrace the answer back to the offerer. Thus, if the offerer has offered a 'setup:passive' or 'setup:actpass' role, it MUST (as specified in RFC 4145 [7]) begin listening for an incoming connection as soon as it sends its offer. However, it MUST NOT assume that the data transmitted over the TLS connection is valid until it has received a matching fingerprint in an SDP answer. If the fingerprint, once it arrives, does not match the client's certificate, the server endpoint MUST terminate the media connection with a bad_certificate error, as stated in the previous paragraph.

If offer/answer is not being used (e.g., if the SDP was sent over the Session Announcement Protocol [15]), there is no secure channel available for clients to communicate certificate fingerprints to servers. In this case, servers MAY request client certificates, which SHOULD be signed by a well-known certification authority, or MAY allow clients to connect without a certificate.

7. Security Considerations

This entire document concerns itself with security. The problem to be solved is addressed in Section 1, and a high-level overview is presented in Section 3. See the SDP specification [8] for security considerations applicable to SDP in general.

Offering a TCP/TLS connection in SDP (or agreeing to one in SDP offer/answer mode) does not create an obligation for an endpoint to accept any TLS connection with the given fingerprint. Instead, the endpoint must engage in the standard TLS negotiation procedure to ensure that the TLS stream cipher and MAC algorithm chosen meet the security needs of the higher-level application. (For example, an offered stream cipher of TLS_NULL_WITH_NULL_NULL SHOULD be rejected in almost every application scenario.)

Like all SDP messages, SDP messages describing TLS streams are conveyed in an encapsulating application protocol (e.g., SIP, Media Gateway Control Protocol (MGCP), etc.). It is the responsibility of the encapsulating protocol to ensure the integrity of the SDP security descriptions. Therefore, the application protocol SHOULD either invoke its own security mechanisms (e.g., secure multiparts) or, alternatively, utilize a lower-layer security service (e.g., TLS or IPsec). This security service SHOULD provide strong message authentication as well as effective replay protection.

However, such integrity protection is not always possible. For these cases, end systems SHOULD maintain a cache of certificates that other parties have previously presented using this mechanism. If possible, users SHOULD be notified when an unsecured certificate associated

with a previously unknown end system is presented and SHOULD be strongly warned if a different unsecured certificate is presented by a party with which they have communicated in the past. In this way, even in the absence of integrity protection for SDP, the security of this document's mechanism is equivalent to that of the Secure Shell (ssh) protocol [18], which is vulnerable to man-in-the-middle attacks when two parties first communicate, but can detect ones that occur subsequently. (Note that a precise definition of the "other party" depends on the application protocol carrying the SDP message.) Users SHOULD NOT, however, in any circumstances be notified about certificates described in SDP descriptions sent over an integrity-protected channel.

To aid interoperability and deployment, security protocols that provide only hop-by-hop integrity protection (e.g., the sips protocol [17], SIP over TLS) are considered sufficiently secure to allow the mode in which any syntactically valid identity is accepted in a certificate. This decision was made because sips is currently the integrity mechanism most likely to be used in deployed networks in the short to medium term. However, in this mode, SDP integrity is vulnerable to attacks by compromised or malicious middleboxes, e.g., SIP proxy servers. End systems MAY warn users about SDP sessions that are secured in only a hop-by-hop manner, and definitions of media formats running over TCP/TLS MAY specify that only end-to-end integrity mechanisms be used.

Depending on how SDP messages are transmitted, it is not always possible to determine whether or not a subjectAltName presented in a remote certificate is expected for the remote party. In particular, given call forwarding, third-party call control, or session descriptions generated by endpoints controlled by the Gateway Control Protocol [21], it is not always possible in SIP to determine what entity ought to have generated a remote SDP response. In general, when not using authenticity and integrity protection of SDP descriptions, a certificate transmitted over SIP SHOULD assert the endpoint's SIP Address of Record as a uniformResourceIndicator subjectAltName. When an endpoint receives a certificate over SIP asserting an identity (including an iPAddress or dNSName identity) other than the one to which it placed or received the call, it SHOULD alert the user and ask for confirmation. This applies whether certificates are self-signed, or signed by certification authorities; a certificate for "sip:bob@example.com" may be legitimately signed by a certification authority, but may still not be acceptable for a call to "sip:alice@example.com". (This issue is not one specific to this specification; the same consideration applies for S/MIME-signed SDP carried over SIP.)

This document does not define a mechanism for securely transporting RTP and RTP Control Protocol (RTCP) packets over a connection-oriented channel. Please see RFC 7850 [19] for more details.

TLS is not always the most appropriate choice for secure connection-oriented media; in some cases, a higher- or lower-level security protocol may be appropriate.

This document improves security from the RFC 4572 [20]. It updates the preferred hash function from SHA-1 to SHA-256, and deprecates the usage of the MD2 and MD5 hash functions.
By clarifying the usage and handling of multiple fingerprints, the document also enables hash agility, and incremental deployment of newer, and more secure, hash functions.

8. IANA Considerations

Note to IANA. No IANA considerations are changed from RFC4572 [20] so the only actions required are to update the registries to point at this specification.

This document defines an SDP proto value: 'TCP/TLS'. Its format is defined in Section 4. This proto value has been registered by IANA under "Session Description Protocol (SDP) Parameters" under "proto".

This document defines an SDP session and media-level attribute: 'fingerprint'. Its format is defined in Section 5. This attribute has been registered by IANA under "Session Description Protocol (SDP) Parameters" under "att-field (both session and media level)".

The SDP specification [8] states that specifications defining new proto values, like the 'TCP/TLS' proto value defined in this one, must define the rules by which their media format (fmt) namespace is managed. For the TCP/TLS protocol, new formats SHOULD have an associated MIME registration. Use of an existing MIME subtype for the format is encouraged. If no MIME subtype exists, it is RECOMMENDED that a suitable one be registered through the IETF process [12] by production of, or reference to, a standards-track RFC that defines the transport protocol for the format.

This specification takes over the IANA registry named "Hash Function Textual Names", that was created in [20]. It will not be part of the SDP Parameters.

The names of hash functions used for certificate fingerprints are registered by the IANA. Hash functions MUST be defined by standards-track RFCs that update or obsolete RFC 3279 [5].

When registering a new hash function textual name, the following information MUST be provided:

- The textual name of the hash function.
- The Object Identifier (OID) of the hash function as used in X.509 certificates.
- A reference to the standards-track RFC, updating or obsoleting RFC 3279 [5], defining the use of the hash function in X.509 certificates.

Table 1 contains the initial values of this registry.

Hash Function Name	OID	Reference
"md2"	1.2.840.113549.2.2	RFC 3279
"md5"	1.2.840.113549.2.5	RFC 3279
"sha-1"	1.3.14.3.2.26	RFC 3279
"sha-224"	2.16.840.1.101.3.4.2.4	RFC 4055
"sha-256"	2.16.840.1.101.3.4.2.1	RFC 4055
"sha-384"	2.16.840.1.101.3.4.2.2	RFC 4055
"sha-512"	2.16.840.1.101.3.4.2.3	RFC 4055

Table 1: IANA Hash Function Textual Name Registry

9. References

9.1. Normative References

[1] National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-2, August 2002, <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>.

[2] International Telecommunications Union, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, ISO Standard 9594-8, March 2000.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[4] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[5] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, DOI 10.17487/RFC3279, April 2002, <http://www.rfc-editor.org/info/rfc3279>.

[6] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, DOI 10.17487/RFC4055, June 2005, <http://www.rfc-editor.org/info/rfc4055>.

[7] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <http://www.rfc-editor.org/info/rfc4145>.

[8] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[9] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[10] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[11] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[12] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

9.2. Informative References

[13] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, DOI 10.17487/RFC1321, April 1992, <http://www.rfc-editor.org/info/rfc1321>.

[14] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[15] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <http://www.rfc-editor.org/info/rfc2974>.

[16] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, DOI 10.17487/RFC3174, September 2001, <http://www.rfc-editor.org/info/rfc3174>.

[17] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[18] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <http://www.rfc-editor.org/info/rfc4251>.

[19] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, DOI 10.17487/RFC4571, July 2006, <http://www.rfc-editor.org/info/rfc4571>.

[20] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <http://www.rfc-editor.org/info/rfc4572>.

[21] Taylor, T., "Reclassification of RFC 3525 to Historic", RFC 5125, DOI 10.17487/RFC5125, February 2008, <http://www.rfc-editor.org/info/rfc5125>.

[22] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <http://www.rfc-editor.org/info/rfc5751>.

[23] Turner, S. and L. Chen, "MD2 to Historic Status", RFC 6149, DOI 10.17487/RFC6149, March 2011, <http://www.rfc-editor.org/info/rfc6149>.

[24] Shekh-Yusef, R., Ed., Ahrens, D., and S. Bremer, "HTTP Digest Access Authentication", RFC 7616, DOI 10.17487/RFC7616, September 2015, <http://www.rfc-editor.org/info/rfc7616>.

Appendix A. Acknowledgments

This version of the document included significant contributions by Cullen Jennings, Paul Kyzivat, Roman Shpount, and Martin Thomson. Elwyn Davies performed the Gen-ART review of the document.

Authors' Addresses

Jonathan Lennox
Vidyo

Email: jonathan@vidyo.com

Christer Holmberg
Ericsson

Email: christer.holmberg@ericsson.com

MMUSIC
Internet-Draft
Obsoletes: 5245 (if approved)
Intended status: Standards Track
Expires: September 14, 2017

M. Petit-Huguenin
Impedance Mismatch
A. Keranen
Ericsson
S. Nandakumar
Cisco Systems

March 13, 2017

Using Interactive Connectivity Establishment (ICE) with Session Description Protocol (SDP) offer/answer and Session Initiation Protocol (SIP)

draft-ietf-mmusic-ice-sip-sdp-12

Abstract

This document describes how Interactive Connectivity Establishment (ICE) is used with Session Description Protocol (SDP) offer/answer and Session Initiation Protocol (SIP).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document describes how Interactive Connectivity Establishment (ICE) is used with Session Description Protocol (SDP) offer/answer [RFC3264] and Session Initiation Protocol (SIP). The ICE specification [ICE-BIS] describes procedures that are common to all usages of ICE and this document gives the additional details needed to use ICE with SDP offer/answer and SIP.

Note that ICE is not intended for NAT traversal for SIP, which is assumed to be provided via another mechanism [RFC5626].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Readers should be familiar with the terminology defined in [RFC3264], in [RFC7656], in [ICE-BIS] and the following:

Default Destination/Candidate: The default destination for a component of a media stream is the transport address that would be used by an agent that is not ICE aware. A default candidate for a component is one whose transport address matches the default destination for that component. For the RTP component, the default IP address is in the "c=" line of the SDP, and the port is in the "m=" line. For the RTCP component, the address and port are indicated using the "a=rtcp" attribute defined in [RFC3605], if present; otherwise, the RTCP component address is same as the address of the RTP component, and its port is one greater than the port of the RTP component.

3. ICE Candidate Exchange and Offer/Answer Mapping

[ICE-BIS] defines ICE candidate exchange as the process for ICE agents (Initiator and Responder) to exchange their candidate information required for ICE processing at the agents. For the purposes of this specification, the candidate exchange process corresponds to the [RFC3264] Offer/Answer protocol and the terminologies offerer and answerer correspond to the initiator and responder terminologies from [ICE-BIS] respectively.

4. SDP Offer/Answer Procedures

4.1. Initial Offer/Answer Exchange

4.1.1. Sending the Initial Offer

The offerer shall follow the procedures defined in section 4 of [ICE-BIS] to gather, prioritize and eliminate the redundant candidates. It then chooses the default candidates and encodes them in the SDP to be sent to its peer, the answerer.

4.1.1.1. Choosing Default Candidates

A candidate is said to be default if it would be the target of media from a non-ICE peer; that target is called the DEFAULT DESTINATION. If the default candidates are not selected by the ICE algorithm when

communicating with an ICE-aware peer, an updated offer/answer will be required after ICE processing completes in order to "fix up" the SDP so that the default destination for media matches the candidates selected by ICE. If ICE happens to select the default candidates, no updated offer/answer is required.

An agent MUST choose a set of candidates, one for each component of each in-use media stream, to be default. A media stream is in-use if it does not have a port of zero (which is used in RFC 3264 to reject a media stream). Consequently, a media stream is in-use even if it is marked as a=inactive [RFC4566] or has a bandwidth value of zero.

An ageny may choose any type of the candidate as the default, if the chosen candidates increases the likelihood of success with the peer that is being contacted if ICE is not being used.

It is RECOMMENDED that default candidates be chosen based on the likelihood of those candidates to work with the peer that is being contacted if ICE is not being used. Many factors may influence such a decision in a given agent. In scenarios where the agent is fully aware of its peer’s location and can reach the peer directly, choosing the host candidates as the default may well be sufficient. If the network configuration under which the agents operates is static and known beforehand, either the host or the server reflexives candidates can serve as the default candidates (depending on if a given agent is behind NAT and their reachability). If the agent is completely unaware of the peer’s location or no assumptions can be made of network characteristics and the connectivity, the relayed candidates might be the only option as the default candidate. Having the decision of choosing the default candidate as a configurable option in the implementations might provide agents the flexibility to take into account the aforementioned criteria. Barring such configuration flexibility, it is RECOMMENDED that the default candidates be the relayed candidates (if relayed candidates are available), server reflexive candidates (if server reflexive candidates are available), and finally host candidates.

4.1.1.2. Encoding the SDP

The process of encoding the SDP is identical between full and lite implementations.

The agent will include an "m=" line for each Source Stream [RFC7656] it wishes to use. The ordering of source streams in the SDP is relevant for ICE. ICE will perform its connectivity checks for the first "m=" line first, and consequently media will be able to flow for that stream first. Agents SHOULD place their most important source stream, if there is one, first in the SDP.

There will be a candidate attribute for each candidate for a particular source stream. Section 5 provides detailed rules for constructing this attribute.

STUN connectivity checks between agents are authenticated using the short-term credential mechanism defined for STUN [RFC5389]. This mechanism relies on a username and password that are exchanged through protocol machinery between the client and server. The username fragment and password are exchanged in the ice-ufrag and ice-pwd attributes, respectively.

If an agent is a lite implementation, it MUST include an "a=ice-lite" session-level attribute in its SDP to indicate this. If an agent is a full implementation, it MUST NOT include this attribute.

Section 7 of [ICE-BIS] defines a new ICE option, 'ice2'. This option is used by ICE Agents to indicate their compliancy with [ICE-BIS] specification as compared to the [RFC5245]. If the Offering agent is a [ICE-BIS] compliant implementation, a session level ICE option to indicate the same (via the "a=ice-options:ice2" SDP line) MUST be included.

The default candidates are added to the SDP as the default destination for media. For source streams based on RTP, this is done by placing the IP address and port of the RTP candidate into the "c=" and "m=" lines, respectively. If the agent is utilizing RTCP and if RTCP candidate is present and is not equal to the same address and the next higher port number of the RTP candidate, the agent MUST encode the RTCP candidate using the a=rtcp attribute as defined in [RFC3605]. If RTCP is not in use, the agent MUST signal that using b=RS:0 and b=RR:0 as defined in [RFC3556]

The transport addresses that will be the default destination for media when communicating with non-ICE peers MUST also be present as candidates in one or more a=candidate lines.

ICE provides for extensibility by allowing an offer or answer to contain a series of tokens that identify the ICE extensions used by that agent. If an agent supports an ICE extension, it MUST include the token defined for that extension in the ice-options attribute.

The following is an example SDP message that includes ICE attributes (lines folded for readability):

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:ice2
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
 10.0.1.1 rport 8998
```

Once an agent has sent its offer or its answer, that agent MUST be prepared to receive both STUN and media packets on each candidate. As discussed in section 9.1 of [ICE-BIS], media packets can be sent to a candidate prior to its appearance as the default destination for media in an offer or answer.

4.1.2. Receiving the Initial Offer

On receiving the offer, the answerer verifies the support for ICE (section 5.1.1 of [ICE-BIS]), determines its role (section 5.1.2 of [ICE-BIS]), gathers candidates (section 4 of [ICE-BIS]), encodes the candidates in an SDP answer and sends it to its peer, the offerer. The answerer shall then follow the steps defined in sections 5.1.3 and 5.1.4 of [ICE-BIS] to schedule the ICE connectivity checks.

The below sub-sections provide additional requirements associated with the processing of the offerer’s SDP pertaining to this specification.

4.1.2.1. ICE Option "ice2" considerations

If the SDP offer contains a session level ICE option, "ice2" , and if the answering ICE Agent is also an [ICE-BIS] compliant implementation, then the generated SDP answer MUST include the session level "a=ice-options:ice2" SDP line.

4.1.2.2. Choosing Default Candidates

The process for selecting default candidates at the answerer is identical to the process followed by the offerer, as described in Section 4.1.1.1 for full implementations in this specification and section 4.2 of [ICE-BIS] for lite implementations.

4.1.2.3. Verifying ICE Support

The agent will proceed with the ICE procedures defined in [ICE-BIS] and this specification if, for each media stream in the SDP it received, the default destination for each component of that media stream appears in a candidate attribute. For example, in the case of RTP, the IP address and port in the "c=" and "m=" lines, respectively, appear in a candidate attribute and the value in the rtcp attribute appears in a candidate attribute.

If this condition is not met, the agent MUST process the SDP based on normal RFC 3264 procedures, without using any of the ICE mechanisms described in the remainder of this specification with the following exceptions:

1. The agent MUST follow the rules of section 8 of [ICE-BIS], which describe keepalive procedures for all agents.

2. If the agent is not proceeding with ICE because there were a=candidate attributes, but none that matched the default destination of the media stream, the agent MUST include an a=ice-mismatch attribute in its answer.

3. If the default candidates were relayed candidates learned through a TURN server, the agent MUST create permissions in the TURN server for the IP addresses learned from its peer in the SDP it just received. If this is not done, initial packets in the media stream from the peer may be lost.

4.1.2.4. Determining Role

In unusual cases, described in Appendix C, it is possible for both agents to mistakenly believe they are controlled or controlling. To resolve this, each agent MUST select a random number, called the tie-breaker, uniformly distributed between 0 and (2**64) - 1 (that is, a 64-bit positive integer). This number is used in connectivity checks to detect and repair this case, as described in section 6.1.2.3 of [ICE-BIS].

4.1.3. Receipt of the Initial Answer

When ICE is used with SIP, forking may result in a single offer generating a multiplicity of answers. In that case, ICE proceeds completely in parallel and independently for each answer, treating the combination of its offer and each answer as an independent offer/answer exchange, with its own set of local candidates, pairs, check lists, states, and so on. The only case in which processing of one

pair impacts another is freeing of candidates, discussed below in Section 4.1.5.

On receiving the SDP answer,the offerer performs steps similar to answerer's processing of the offer. The offerer verifies the answerer's ICE support determines, its role, and processes the answerer's candidates to schedule the connectivity checks (section 6 of [ICE-BIS]).

If the offerer had included the "ice2" ICE Option in the offer and the SDP answer also includes a similar session level ICE option, then the peers are [ICE-BIS] compliant implementations. On the other hand, if the SDP Answer lacks such a ICE option, the offerer defaults to the procedures that are backward compatible with the [RFC5245] specification.

4.1.3.1. Verifying ICE Support

The logic at the offerer is identical to that of the answerer as described in section 5.1.1 of [ICE-BIS], with the exception that an offerer would not ever generate a=ice-mismatch attributes in an SDP.

In some cases, the answer may omit a=candidate attributes for the media streams, and instead include an a=ice-mismatch attribute for one or more of the media streams in the SDP. This signals to the offerer that the answerer supports ICE, but that ICE processing was not used for the session because a signaling intermediary modified the default destination for media components without modifying the corresponding candidate attributes. See Section 11.2.2 for a discussion of cases where this can happen. This specification provides no guidance on how an agent should proceed in such a failure case.

4.1.4. Performing Connectivity Checks

The possibility for role conflicts described in section 6.1.3.1.1 of [ICE-BIS] applies to this usage and hence all full agents MUST implement the role conflict repairing mechanism. Also both full and lite agents MUST utilize the ICE-CONTROLLED and ICE-CONTROLLING attributes as described in section 6.1.2.3 of [ICE-BIS].

4.1.5. Concluding ICE

Once the state of each check list is Completed, If an agent is controlling, it examines the highest-priority nominated candidate pair for each component of each media stream. If any of those candidate pairs differ from the default candidate pairs in the most

recent offer/answer exchange, the controlling agent MUST generate an updated offer as described in Section 4.2.

When ICE is used with SIP, and an offer is forked to multiple recipients, ICE proceeds in parallel and independently with each answerer, all using the same local candidates. Once ICE processing has reached the Completed state for all peers for media streams using those candidates, the agent SHOULD wait an additional three seconds, and then it MAY cease responding to checks or generating triggered checks on that candidate. It MAY free the candidate at that time.

Freeing of server reflexive candidates is never explicit; it happens by lack of a keepalive. The three-second delay handles cases when aggressive nomination is used, and the selected pairs can quickly change after ICE has completed.

4.2. Subsequent Offer/Answer Exchanges

Either agent MAY generate a subsequent offer at any time allowed by [RFC3264]. The rules in Section 4.1.5 will cause the controlling agent to send an updated offer at the conclusion of ICE processing when ICE has selected different candidate pairs from the default pairs. This section defines rules for construction of subsequent offers and answers.

Should a subsequent offer fail, ICE processing continues as if the subsequent offer had never been made.

4.2.1. Generating the Offer

4.2.1.1. Procedures for All Implementations

4.2.1.1.1. ICE Restarts

An agent MAY restart ICE processing for an existing media stream as defined in section 6.3 of [ICE-BIS].

The rules governing the ICE restart imply that setting the IP address in the "c=" line to 0.0.0.0 will cause an ICE restart. Consequently, ICE implementations MUST NOT utilize this mechanism for call hold, and instead MUST use a=inactive and a=sendonly as described in [RFC3264].

To restart ICE, an agent MUST change both the ice-pwd and the ice-ufrag for the media stream in an offer. Note that it is permissible to use a session-level attribute in one offer, but to provide the same ice-pwd or ice-ufrag as a media-level attribute in a subsequent

offer. This is not a change in password, just a change in its representation, and does not cause an ICE restart.

An agent sets the rest of the fields in the SDP for this media stream as it would in an initial offer of this media stream (see Section 4.1.1.2). Consequently, the set of candidates MAY include some, none, or all of the previous candidates for that stream and MAY include a totally new set of candidates.

4.2.1.1.2. Removing a Media Stream

If an agent removes a media stream by setting its port to zero, it MUST NOT include any candidate attributes for that media stream and SHOULD NOT include any other ICE-related attributes defined in Section 5 for that media stream.

4.2.1.1.3. Adding a Media Stream

If an agent wishes to add a new media stream, it sets the fields in the SDP for this media stream as if this was an initial offer for that media stream (see Section 4.1.1.2). This will cause ICE processing to begin for this media stream.

4.2.1.2. Procedures for Full Implementations

This section describes additional procedures for full implementations, covering existing media streams.

4.2.1.2.1. Existing Media Streams with ICE Running

If an agent generates an updated offer including a media stream that was previously established, and for which ICE checks are in the Running state, the agent follows the procedures defined here.

An agent MUST include candidate attributes for all local candidates it had signaled previously for that media stream. The properties of that candidate as signaled in SDP -- the priority, foundation, type, and related transport address -- SHOULD remain the same. The IP address, port, and transport protocol, which fundamentally identify that candidate, MUST remain the same (if they change, it would be a new candidate). The component ID MUST remain the same. The agent MAY include additional candidates it did not offer previously (see section 4.2.4.1.1), but which it has gathered since the last offer/ answer exchange, including peer reflexive candidates.

The agent MAY change the default destination for media. As with initial offers, there MUST be a set of candidate attributes in the offer matching this default destination.

4.2.1.2.2. Existing Media Streams with ICE Completed

If an agent generates an updated offer including a media stream that was previously established, and for which ICE checks are in the Completed state, the agent follows the procedures defined here.

The default destination for media (i.e., the values of the IP addresses and ports in the "m=" and "c=" lines used for that media stream) MUST be the local candidate from the highest-priority nominated pair in the valid list for each component. This "fixes" the default destination for media to equal the destination ICE has selected for media.

The agent MUST include candidate attributes for candidates matching the default destination for each component of the media stream, and MUST NOT include any other candidates.

In addition, if the agent is controlling, it MUST include the a=remote-candidates attribute for each media stream whose check list is in the Completed state. The attribute contains the remote candidates from the highest-priority nominated pair in the valid list for each component of that media stream. It is needed to avoid a race condition whereby the controlling agent chooses its pairs, but the updated offer beats the connectivity checks to the controlled agent, which doesn't even know these pairs are valid, let alone selected. See Appendix B for elaboration on this race condition.

4.2.1.3. Procedures for Lite Implementations

4.2.1.3.1. Existing Media Streams with ICE Running

This section describes procedures for lite implementations for existing streams for which ICE is running.

A lite implementation MUST include all of its candidates for each component of each media stream in an a=candidate attribute in any subsequent offer. These candidates are formed identically to the procedures for initial offers, as described in section 4.2 of [ICE-BIS].

A lite implementation MUST NOT add additional host candidates in a subsequent offer. If an agent needs to offer additional candidates, it MUST restart ICE.

The username fragments, password, and implementation level MUST remain the same as used previously. If an agent needs to change one of these, it MUST restart ICE for that media stream.

4.2.1.3.2. Existing Media Streams with ICE Completed

If ICE has completed for a media stream, the default destination for that media stream MUST be set to the remote candidate of the candidate pair for that component in the valid list. For a lite implementation, there is always just a single candidate pair in the valid list for each component of a media stream. Additionally, the agent MUST include a candidate attribute for each default destination.

Additionally, if the agent is controlling (which only happens when both agents are lite), the agent MUST include the a=remote-candidates attribute for each media stream. The attribute contains the remote candidates from the candidate pairs in the valid list (one pair for each component of each media stream).

4.2.2. Receiving the Offer and Generating an Answer

4.2.2.1. Procedures for All Implementations

When receiving a subsequent offer within an existing session, an agent MUST reapply the verification procedures in Section 4.1.2.3 without regard to the results of verification from any previous offer/answer exchanges. Indeed, it is possible that a previous offer/answer exchange resulted in ICE not being used, but it is used as a consequence of a subsequent exchange.

4.2.2.1.1. Detecting ICE Restart

If the offer contained a change in the a=ice-ufrag or a=ice-pwd attributes compared to the previous SDP from the peer, it indicates that ICE is restarting for this media stream. If all media streams are restarting, then ICE is restarting overall.

If ICE is restarting for a media stream:

- The agent MUST change the a=ice-ufrag and a=ice-pwd attributes in the answer.

- The agent MAY change its implementation level in the answer.

An agent sets the rest of the fields in the SDP for this media stream as it would in an initial answer to this media stream (see Section 4.1.1.2). Consequently, the set of candidates MAY include some, none, or all of the previous candidates for that stream and MAY include a totally new set of candidates.

4.2.2.1.2. New Media Stream

If the offer contains a new media stream, the agent sets the fields in the answer as if it had received an initial offer containing that media stream (see Section 4.1.1.2). This will cause ICE processing to begin for this media stream.

4.2.2.1.3. Removed Media Stream

If an offer contains a media stream whose port is zero, the agent MUST NOT include any candidate attributes for that media stream in its answer and SHOULD NOT include any other ICE-related attributes defined in Section 5 for that media stream.

4.2.2.2. Procedures for Full Implementations

Unless the agent has detected an ICE restart from the offer, the username fragments, password, and implementation level MUST remain the same as used previously. If an agent needs to change one of these it MUST restart ICE for that media stream by generating an offer; ICE cannot be restarted in an answer.

Additional behaviors depend on the state of ICE processing for that media stream.

4.2.2.2.1. Existing Media Streams with ICE Running and no remote-candidates

If ICE is running for a media stream, and the offer for that media stream lacked the remote-candidates attribute, the rules for construction of the answer are identical to those for the offerer as described in Section 4.2.1.2.1.

4.2.2.2.2. Existing Media Streams with ICE Completed and no remote-candidates

If ICE is Completed for a media stream, and the offer for that media stream lacked the remote-candidates attribute, the rules for construction of the answer are identical to those for the offerer as described in Section 4.2.1.2.2, except that the answerer MUST NOT include the a=remote-candidates attribute in the answer.

4.2.2.2.3. Existing Media Streams and remote-candidates

A controlled agent will receive an offer with the a=remote-candidates attribute for a media stream when its peer has concluded ICE processing for that media stream. This attribute is present in the offer to deal with a race condition between the receipt of the offer,

and the receipt of the Binding Response that tells the answerer the candidate that will be selected by ICE. See Appendix B for an explanation of this race condition. Consequently, processing of an offer with this attribute depends on the winner of the race.

The agent forms a candidate pair for each component of the media stream by:

- Setting the remote candidate equal to the offerer's default destination for that component (e.g., the contents of the "m=" and "c=" lines for RTP, and the a=rtcp attribute for RTCP)

- Setting the local candidate equal to the transport address for that same component in the a=remote-candidates attribute in the offer.

The agent then sees if each of these candidate pairs is present in the valid list. If a particular pair is not in the valid list, the check has "lost" the race. Call such a pair a "losing pair".

The agent finds all the pairs in the check list whose remote candidates equal the remote candidate in the losing pair:

- If none of the pairs are In-Progress, and at least one is Failed, it is most likely that a network failure, such as a network partition or serious packet loss, has occurred. The agent SHOULD generate an answer for this media stream as if the remote-candidates attribute had not been present, and then restart ICE for this stream.

- If at least one of the pairs is In-Progress, the agent SHOULD wait for those checks to complete, and as each completes, redo the processing in this section until there are no losing pairs.

Once there are no losing pairs, the agent can generate the answer. It MUST set the default destination for media to the candidates in the remote-candidates attribute from the offer (each of which will now be the local candidate of a candidate pair in the valid list). It MUST include a candidate attribute in the answer for each candidate in the remote-candidates attribute in the offer.

4.2.2.3. Procedures for Lite Implementations

If the received offer contains the remote-candidates attribute for a media stream, the agent forms a candidate pair for each component of the media stream by:

- Setting the remote candidate equal to the offerer's default destination for that component (e.g., the contents of the "m=" and "c=" lines for RTP, and the a=rtcp attribute for RTCP).

- Setting the local candidate equal to the transport address for that same component in the a=remote-candidates attribute in the offer.

It then places those candidates into the Valid list for the media stream. The state of ICE processing for that media stream is set to Completed.

Furthermore, if the agent believed it was controlling, but the offer contained the remote-candidates attribute, both agents believe they are controlling. In this case, both would have sent updated offers around the same time. However, the signaling protocol carrying the offer/answer exchanges will have resolved this glare condition, so that one agent is always the 'winner' by having its offer received before its peer has sent an offer. The winner takes the role of controlling, so that the loser (the answerer under consideration in this section) MUST change its role to controlled. Consequently, if the agent was going to send an updated offer since, based on the rules in section 6.2 of [ICE-BIS], it was controlling, it no longer needs to.

Besides the potential role change, change in the Valid list, and state changes, the construction of the answer is performed identically to the construction of an offer as described in Section 4.2.1.3.

4.2.3. Receiving the Answer for a Subsequent Offer

Some deployments of ICE include e.g. SDP-Modifying Signaling-only Back-to-Back User Agents (B2BUAs) [RFC7092] that modify the SDP body during the subsequent offer/answer exchange. With the B2BUA being ICE-unaware, a subsequent answer might be manipulated and might not include ICE candidates although the initial answer did.

An example of a situation where such an "unexpected" answer might be experienced appears when such a B2BUA introduces a media server during call hold using 3rd party call-control procedures. Omitting further details how this is done this could result in an answer being received at the holding UA that was constructed by the B2BUA. With the B2BUA being ICE-unaware, that answer would not include ICE candidates.

Receiving an answer without ICE attributes in this situation might be unexpected, but would not necessarily impair the user experience.

In addition to procedures for the expected answer, the following section advices on how to recover from the unexpected situation.

4.2.3.1. Procedures for All Implementations

When receiving an answer within an existing session for a subsequent offer as specified in Section 4.2.1.2.2, an agent MUST verify ICE support as specified in Section 4.1.3.1.

If ICE support is indicated in the SDP answer and the offer was a restart, the agent MUST perform ICE restart procedures as specified in Section 4.2.4. If ICE support is no longer indicated in the SDP answer, the agent MUST fall-back to [RFC3264] procedures and SHOULD NOT drop the dialog just because of missing ICE support. If the agent sends a new offer later on, it SHOULD perform an ICE restart as specified in Section 4.2.1.1.1.

If ICE support is indicated in the SDP answer and ICE is running, the agent MUST continue ICE procedures as specified in Section 4.2.4.1.4. If ICE support is no longer indicated in the SDP answer, the agent MUST abort the ongoing ICE processing and fall-back to [RFC3264] procedures. The agent SHOULD NOT drop the dialog just because of missing ICE support. If the agent sends a new offer later on, it SHOULD perform an ICE restart as specified in Section 4.2.1.1.1.

If ICE support is indicated in the SDP answer and if ICE is completed and the answer conforms to Section 4.2.2.2.3, the agent MUST remain in the ICE Completed state. If ICE support is no longer indicated in the SDP answer, the agent MUST fall-back to [RFC3264] procedures and SHOULD NOT drop the dialog just because of this unexpected answer. Once the agent sends a new offer later on it MUST perform an ICE restart.

4.2.4. Updating the Check and Valid Lists

4.2.4.1. Procedures for Full Implementations

4.2.4.1.1. ICE Restarts

The agent MUST remember the highest-priority nominated pairs in the Valid list for each component of the media stream, called the previous selected pairs, prior to the restart. The agent will continue to send media using these pairs, as described in Section 7.1. Once these destinations are noted, the agent MUST flush the valid and check lists, and then recompute the check list and its states as described in section 5.1.3 of [ICE-BIS].

4.2.4.1.2. New Media Stream

If the offer/answer exchange added a new media stream, the agent MUST create a new check list for it (and an empty Valid list to start of course), as described in section 5.1.3 of [ICE-BIS].

4.2.4.1.3. Removed Media Stream

If the offer/answer exchange removed a media stream, or an answer rejected an offered media stream, an agent MUST flush the Valid list for that media stream. It MUST terminate any STUN transactions in progress for that media stream. An agent MUST remove the check list for that media stream and cancel any pending ordinary checks for it.

4.2.4.1.4. ICE Continuing for Existing Media Stream

The valid list is not affected by an updated offer/answer exchange unless ICE is restarting.

If an agent is in the Running state for that media stream, the check list is updated (the check list is irrelevant if the state is completed). To do that, the agent recomputes the check list using the procedures described in section 5.1.3 of [ICE-BIS]. If a pair on the new check list was also on the previous check list, and its state was Waiting, In-Progress, Succeeded, or Failed, its state is copied over. Otherwise, its state is set to Frozen.

If none of the check lists are active (meaning that the pairs in each check list are Frozen), the full-mode agent follows steps in Section 5.1.3.6 of [ICE-BIS] to place appropriate candidates in the Waiting state to further continue ICE processing.

4.2.4.2. Procedures for Lite Implementations

If ICE is restarting for a media stream, the agent MUST start a new Valid list for that media stream. It MUST remember the pairs in the previous Valid list for each component of the media stream, called the previous selected pairs, and continue to send media there as described in Section 7.1. The state of ICE processing for each media stream MUST change to Running, and the state of ICE processing MUST change to Running.

5. Grammar

This specification defines eight new SDP attributes -- the "candidate", "remote-candidates", "ice-lite", "ice-mismatch", "ice-ufrag", "ice-pwd", "ice-pacing", and "ice-options" attributes. This

section also provides non-normative examples of the attributes defined.

The syntax for the attributes follow Augmented BNF as defined in [RFC5234].

5.1. "candidate" Attribute

The candidate attribute is a media-level attribute only. It contains a transport address for a candidate that can be used for connectivity checks.

```
candidate-attribute   = "candidate" ":" foundation SP component-id SP
                        transport SP
                        priority SP
                        connection-address SP     ;from RFC 4566
                        port         ;port from RFC 4566
                        SP cand-type
                        [SP rel-addr]
                        [SP rel-port]
                        *(SP extension-att-name SP
                             extension-att-value)

foundation            = 1*32ice-char
component-id          = 1*5DIGIT
transport             = "UDP" / transport-extension
transport-extension   = token              ; from RFC 3261
priority              = 1*10DIGIT
cand-type             = "typ" SP candidate-types
candidate-types       = "host" / "srflx" / "prflx" / "relay" / token
rel-addr              = "raddr" SP connection-address
rel-port              = "rport" SP port
extension-att-name    = token
extension-att-value   = *VCHAR
ice-char              = ALPHA / DIGIT / "+" / "/"
```

This grammar encodes the primary information about a candidate: its IP address, port and transport protocol, and its properties: the foundation, component ID, priority, type, and related transport address:

<connection-address>: is taken from RFC 4566 [RFC4566]. It is the IP address of the candidate. When parsing this field, an agent can differentiate an IPv4 address and an IPv6 address by presence of a colon in its value -- the presence of a colon indicates IPv6. An agent MUST ignore candidate lines that include candidates with IP address versions that are not supported or recognized. An IP address SHOULD be used, but an FQDN MAY be used in place of an IP

address. In that case, when receiving an offer or answer containing an FQDN in an a=candidate attribute, the FQDN is looked up in the DNS first using an AAAA record (assuming the agent supports IPv6), and if no result is found or the agent only supports IPv4, using an A record. The rules from section 6 of [RFC6724] is followed by fixing the source address to be one from the candidate pair to be matched against destination addresses reported by FQDN, in cases where the DNS query returns more than one IP address.

<port>: is also taken from RFC 4566 [RFC4566]. It is the port of the candidate.

<transport>: indicates the transport protocol for the candidate. This specification only defines UDP. However, extensibility is provided to allow for future transport protocols to be used with ICE, such as the Datagram Congestion Control Protocol (DCCP) [RFC4340].

<foundation>: is composed of 1 to 32 <ice-char>s. It is an identifier that is equivalent for two candidates that are of the same type, share the same base, and come from the same STUN server. The foundation is used to optimize ICE performance in the Frozen algorithm as described in section 5.1.3 of [ICE-BIS]

<component-id>: is a positive integer between 1 and 256 that identifies the specific component of the media stream for which this is a candidate. It MUST start at 1 and MUST increment by 1 for each component of a particular candidate. For media streams based on RTP, candidates for the actual RTP media MUST have a component ID of 1, and candidates for RTCP MUST have a component ID of 2. See section 10 in [ICE-BIS] for additional discussion on extending ICE to new media streams.

<priority>: is a positive integer between 1 and (2**31 - 1). The procedures for computing candidate’s priority is described in section 4.1.2 of [ICE-BIS].

<cand-type>: encodes the type of candidate. This specification defines the values "host", "srflx", "prflx", and "relay" for host, server reflexive, peer reflexive, and relayed candidates, respectively. The set of candidate types is extensible for the future.

<rel-addr> and <rel-port>: convey transport addresses related to the candidate, useful for diagnostics and other purposes. <rel-addr> and <rel-port> MUST be present for server reflexive, peer reflexive, and relayed candidates. If a candidate is server or

peer reflexive, <rel-addr> and <rel-port> are equal to the base for that server or peer reflexive candidate. If the candidate is relayed, <rel-addr> and <rel-port> are equal to the mapped address in the Allocate response that provided the client with that relayed candidate (see section Appendix B.3 of [ICE-BIS] for a discussion of its purpose). If the candidate is a host candidate, <rel-addr> and <rel-port> MUST be omitted.

In some cases, e.g., for privacy reasons, an agent may not want to reveal the related address and port. In this case the address MUST be set to "0.0.0.0" (for IPv4 candidates) or "::" (for IPv6 candidates) and the port to zero.

The candidate attribute can itself be extended. The grammar allows for new name/value pairs to be added at the end of the attribute. An implementation MUST ignore any name/value pairs it doesn't understand.

Example: SDP line for UDP server reflexive candidate attribute for the RTP compo
nent

a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ
srflx raddr 10.0.1.1 rport 8998

5.2. "remote-candidates" Attribute

The syntax of the "remote-candidates" attribute is defined using Augmented BNF as defined in [RFC5234]. The remote-candidates attribute is a media-level attribute only.

```
remote-candidate-att = "remote-candidates:" remote-candidate
                           0*(SP remote-candidate)
remote-candidate = component-ID SP connection-address SP port
```

The attribute contains a connection-address and port for each component. The ordering of components is irrelevant. However, a value MUST be present for each component of a media stream. This attribute MUST be included in an offer by a controlling agent for a media stream that is Completed, and MUST NOT be included in any other case.

Example: Remote candidates SDP lines for the RTP and RTCP components:

a=remote-candidates:1 192.0.2.3 45664
a=remote-candidates:2 192.0.2.3 45665

5.3. "ice-lite" and "ice-mismatch" Attributes

The syntax of the "ice-lite" and "ice-mismatch" attributes, both of which are flags, is:

```
ice-lite              = "ice-lite"
ice-mismatch          = "ice-mismatch"
```

"ice-lite" is a session-level attribute only, and indicates that an agent is a lite implementation. "ice-mismatch" is a media-level attribute only, and when present in an answer, indicates that the offer arrived with a default destination for a media component that didn't have a corresponding candidate attribute.

5.4. "ice-ufrag" and "ice-pwd" Attributes

The "ice-ufrag" and "ice-pwd" attributes convey the username fragment and password used by ICE for message integrity. Their syntax is:

```
ice-pwd-att           = "ice-pwd:" password
ice-ufrag-att         = "ice-ufrag:" ufrag
password              = 22*256ice-char
ufrag                 = 4*256ice-char
```

The "ice-pwd" and "ice-ufrag" attributes can appear at either the session-level or media-level. When present in both, the value in the media-level takes precedence. Thus, the value at the session-level is effectively a default that applies to all media streams, unless overridden by a media-level value. Whether present at the session or media-level, there MUST be an ice-pwd and ice-ufrag attribute for each media stream. If two media streams have identical ice-ufrag's, they MUST have identical ice-pwd's.

The ice-ufrag and ice-pwd attributes MUST be chosen randomly at the beginning of a session. The ice-ufrag attribute MUST contain at least 24 bits of randomness, and the ice-pwd attribute MUST contain at least 128 bits of randomness. This means that the ice-ufrag attribute will be at least 4 characters long, and the ice-pwd at least 22 characters long, since the grammar for these attributes allows for 6 bits of information per character. The attributes MAY be longer than 4 and 22 characters, respectively, of course, up to 256 characters. The upper limit allows for buffer sizing in implementations. Its large upper limit allows for increased amounts of randomness to be added over time. For compatibility with the 512 character limitation for the STUN username attribute value and for bandwidth conservation considerations, the ice-ufrag attribute MUST NOT be longer than 32 characters when sending, but an implementation MUST accept up to 256 characters when receiving.

Example shows sample ice-ufrag and ice-pwd SDP lines:

```
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
```

5.5. "ice-pacing" Attribute

The "ice-pacing" attribute indicates the desired connectivity check pacing, in milliseconds, for this agent (see section 11 of [ICE-BIS]). The syntax is:

```
ice-pacing-att            = "ice-pacing:" pacing-value
pacing-value              = 1*10DIGIT
```

Example shows ice-pacing value of 5 ms:

```
a=ice-pacing:5
```

5.6. "ice-options" Attribute

The "ice-options" attribute is a session- and media-level attribute. It contains a series of tokens that identify the options supported by the agent. Its grammar is:

```
ice-options           = "ice-options:" ice-option-tag
                          0*(SP ice-option-tag)
ice-option-tag        = 1*ice-char
```

The existence of an ice-option can indicate that a certain extension is supported by the agent and will be used or that the extension is used only if the other agent is willing to use it too. In order to avoid ambiguity, documents defining new options must indicate which case applies to the defined extensions.

Example shows ’rtp+ecn’ ice-option SDP line from <<RFC6679>>:

```
a=ice-options:rtp+ecn
```

6. Keepalives

All the ICE agents MUST follow the procedures defined in section 9 of [ICE-BIS] for sending keepalives. The keepalives MUST be sent regardless of whether the media stream is currently inactive, sendonly, recvonly, or sendrecv, and regardless of the presence or value of the bandwidth attribute. An agent can determine that its peer supports ICE by the presence of a=candidate attributes for each media session.

7. Media Handling

7.1. Sending Media

The selected pair for a component of a media stream might not equal the default pair for that same component from the most recent offer/answer exchange. When this happens, the selected pair is used for media, not the default pair. When ICE first completes, if the selected pairs aren't a match for the default pairs, the controlling agent sends an updated offer/answer exchange to remedy this disparity. However, until that updated offer arrives, there will not be a match. Furthermore, in very unusual cases, the default candidates in the updated offer/answer will not be a match.

7.1.1. Procedures for All Implementations

section 9.1.3 of [ICE-BIS] defines procedures for sending media common across Full and Lite implementations.

7.2. Receiving Media

See section 9.2 of [ICE-BIS] for procedures on receiving media.

8. Usage with SIP

8.1. Latency Guidelines

ICE requires a series of STUN-based connectivity checks to take place between endpoints. These checks start from the answerer on generation of its answer, and start from the offerer when it receives the answer. These checks can take time to complete, and as such, the selection of messages to use with offers and answers can affect perceived user latency. Two latency figures are of particular interest. These are the post-pickup delay and the post-dial delay. The post-pickup delay refers to the time between when a user "answers the phone" and when any speech they utter can be delivered to the caller. The post-dial delay refers to the time between when a user enters the destination address for the user and ringback begins as a consequence of having successfully started alerting the called user agent.

Two cases can be considered -- one where the offer is present in the initial INVITE and one where it is in a response.

8.1.1. Offer in INVITE

To reduce post-dial delays, it is RECOMMENDED that the caller begin gathering candidates prior to actually sending its initial INVITE. This can be started upon user interface cues that a call is pending, such as activity on a keypad or the phone going off-hook.

On the receipt of the offer, the answerer SHOULD generate an answer in a provisional response once it has compelted candidate gathering. ICE requires that a provisional response with an SDP be transmitted reliably. This can be done through the existing Provisional Response Acknowledgment (PRACK) mechanism [RFC3262] or through an ICE specific optimization, wherein, the agent retransmits the provisional response with the exponential backoff timers described in [RFC3262]. Such retransmissions MUST cease on receipt of a STUN Binding request for one of the media streams signaled in that SDP or on transmission of the answer in a 2xx response. If no Binding request is received prior to the last retransmit, the agent does not consider the session terminated. For the ICE lite peers, the agent MUST cease retransmitting the 18x after sending it four times (ICE will actually work even if the peer never receives the 18x; however, experience has shown that sending it is important for middleboxes and firewall traversal).

It should be noted that the ICE specific optimization is very specific to provisional response carrying answers that start ICE processing and it is not a general technique for 1xx reliability. Also such an optimization SHOULD NOT be used if both agents support PRACK.

Despite the fact that the provisional response will be delivered reliably, the rules for when an agent can send an updated offer or answer do not change from those specified in [RFC3262]. Specifically, if the INVITE contained an offer, the same answer appears in all of the 1xx and in the 2xx response to the INVITE. Only after that 2xx has been sent can an updated offer/answer exchange occur.

Alternatively, an agent MAY delay sending an answer until the 200 OK; however, this results in a poor user experience and is NOT RECOMMENDED.

Once the answer has been sent, the agent SHOULD begin its connectivity checks. Once candidate pairs for each component of a media stream enter the valid list, the answerer can begin sending media on that media stream.

However, prior to this point, any media that needs to be sent towards the caller (such as SIP early media [RFC3960]) MUST NOT be transmitted. For this reason, implementations SHOULD delay alerting the called party until candidates for each component of each media stream have entered the valid list. In the case of a PSTN gateway, this would mean that the setup message into the PSTN is delayed until this point. Doing this increases the post-dial delay, but has the effect of eliminating 'ghost rings'. Ghost rings are cases where the called party hears the phone ring, picks up, but hears nothing and cannot be heard. This technique works without requiring support for, or usage of, preconditions [RFC3312]. It also has the benefit of guaranteeing that not a single packet of media will get clipped, so that post-pickup delay is zero. If an agent chooses to delay local alerting in this way, it SHOULD generate a 180 response once alerting begins.

8.1.2. Offer in Response

In addition to uses where the offer is in an INVITE, and the answer is in the provisional and/or 200 OK response, ICE works with cases where the offer appears in the response. In such cases, which are common in third party call control [RFC3725], ICE agents SHOULD generate their offers in a reliable provisional response (which MUST utilize [RFC3262]), and not alert the user on receipt of the INVITE. The answer will arrive in a PRACK. This allows for ICE processing to take place prior to alerting, so that there is no post-pickup delay, at the expense of increased call setup delays. Once ICE completes, the callee can alert the user and then generate a 200 OK when they answer. The 200 OK would contain no SDP, since the offer/answer exchange has completed.

Alternatively, agents MAY place the offer in a 2xx instead (in which case the answer comes in the ACK). When this happens, the callee will alert the user on receipt of the INVITE, and the ICE exchanges will take place only after the user answers. This has the effect of reducing call setup delay, but can cause substantial post-pickup delays and media clipping.

8.2. SIP Option Tags and Media Feature Tags

[RFC5768] specifies a SIP option tag and media feature tag for usage with ICE. ICE implementations using SIP SHOULD support this specification, which uses a feature tag in registrations to facilitate interoperability through signaling intermediaries.

8.3. Interactions with Forking

ICE interacts very well with forking. Indeed, ICE fixes some of the problems associated with forking. Without ICE, when a call forks and the caller receives multiple incoming media streams, it cannot determine which media stream corresponds to which callee.

With ICE, this problem is resolved. The connectivity checks which occur prior to transmission of media carry username fragments, which in turn are correlated to a specific callee. Subsequent media packets that arrive on the same candidate pair as the connectivity check will be associated with that same callee. Thus, the caller can perform this correlation as long as it has received an answer.

8.4. Interactions with Preconditions

Quality of Service (QoS) preconditions, which are defined in [RFC3312] and [RFC4032], apply only to the transport addresses listed as the default targets for media in an offer/answer. If ICE changes the transport address where media is received, this change is reflected in an updated offer that changes the default destination for media to match ICE's selection. As such, it appears like any other re-INVITE would, and is fully treated in RFCs 3312 and 4032, which apply without regard to the fact that the destination for media is changing due to ICE negotiations occurring "in the background".

Indeed, an agent SHOULD NOT indicate that QoS preconditions have been met until the checks have completed and selected the candidate pairs to be used for media.

ICE also has (purposeful) interactions with connectivity preconditions [RFC5898]. Those interactions are described there. Note that the procedures described in Section 8.1 describe their own type of "preconditions", albeit with less functionality than those provided by the explicit preconditions in [RFC5898].

8.5. Interactions with Third Party Call Control

ICE works with Flows I, III, and IV as described in [RFC3725]. Flow I works without the controller supporting or being aware of ICE. Flow IV will work as long as the controller passes along the ICE attributes without alteration. Flow II is fundamentally incompatible with ICE; each agent will believe itself to be the answerer and thus never generate a re-INVITE.

The flows for continued operation, as described in Section 7 of [RFC3725], require additional behavior of ICE implementations to support. In particular, if an agent receives a mid-dialog re-INVITE

that contains no offer, it MUST restart ICE for each media stream and go through the process of gathering new candidates. Furthermore, that list of candidates SHOULD include the ones currently being used for media.

9. Relationship with ANAT

[RFC4091], the Alternative Network Address Types (ANAT) Semantics for the SDP grouping framework, and [RFC4092], its usage with SIP, define a mechanism for indicating that an agent can support both IPv4 and IPv6 for a media stream, and it does so by including two "m=" lines, one for v4 and one for v6. This is similar to ICE, which allows for an agent to indicate multiple transport addresses using the candidate attribute. However, ANAT relies on static selection to pick between choices, rather than a dynamic connectivity check used by ICE.

It is RECOMMENDED that ICE be used in realizing the dual-stack use-cases in agents that support ICE.

10. Setting Ta and RTO for RTP Media Streams

During the gathering phase of ICE (section 4.1.1 [ICE-BIS]) and while ICE is performing connectivity checks (section 6 [ICE-BIS]), an agent sends STUN and TURN transactions. These transactions are paced at a rate of one every Ta milliseconds, and utilize a specific RTO. See Section 12 of [ICE-BIS] for details on how the values of Ta and RTO are computed with a real-time media stream of known maximum bandwidth to rate-control the ICE exchanges.

11. Security Considerations

11.1. Attacks on the Offer/Answer Exchanges

An attacker that can modify or disrupt the offer/answer exchanges themselves can readily launch a variety of attacks with ICE. They could direct media to a target of a DoS attack, they could insert themselves into the media stream, and so on. These are similar to the general security considerations for offer/answer exchanges, and the security considerations in [RFC3264] apply. These require techniques for message integrity and encryption for offers and answers, which are satisfied by the TLS mechanism [RFC3261] when SIP is used. As such, the usage of TLS with ICE is RECOMMENDED.

11.2. Insider Attacks

In addition to attacks where the attacker is a third party trying to insert fake offers, answers, or STUN messages, there are several

attacks possible with ICE when the attacker is an authenticated and valid participant in the ICE exchange.

11.2.1. The Voice Hammer Attack

The voice hammer attack is an amplification attack. In this attack, the attacker initiates sessions to other agents, and maliciously includes the IP address and port of a DoS target as the destination for media traffic signaled in the SDP. This causes substantial amplification; a single offer/answer exchange can create a continuing flood of media packets, possibly at high rates (consider video sources). This attack is not specific to ICE, but ICE can help provide remediation.

Specifically, if ICE is used, the agent receiving the malicious SDP will first perform connectivity checks to the target of media before sending media there. If this target is a third-party host, the checks will not succeed, and media is never sent.

Unfortunately, ICE doesn’t help if it’s not used, in which case an attacker could simply send the offer without the ICE parameters. However, in environments where the set of clients is known, and is limited to ones that support ICE, the server can reject any offers or answers that don’t indicate ICE support.

User Agents that are not willing to receive non-ICE answers MUST include an "ice" Option Tag in the Require Header Field in their offer. Clients that rejects non-ICE offers SHOULD use a 421 response code, together with an Option Tag "ice" in the Require Header Field in the response.

11.2.2. Interactions with Application Layer Gateways and SIP

Application Layer Gateways (ALGs) are functions present in a Network Address Translation (NAT) device that inspect the contents of packets and modify them, in order to facilitate NAT traversal for application protocols. Session Border Controllers (SBCs) are close cousins of ALGs, but are less transparent since they actually exist as application-layer SIP intermediaries. ICE has interactions with SBCs and ALGs.

If an ALG is SIP aware but not ICE aware, ICE will work through it as long as the ALG correctly modifies the SDP. A correct ALG implementation behaves as follows:

- The ALG does not modify the "m=" and "c=" lines or the rtcp attribute if they contain external addresses.

- If the "m=" and "c=" lines contain internal addresses, the modification depends on the state of the ALG:

  If the ALG already has a binding established that maps an external port to an internal IP address and port matching the values in the "m=" and "c=" lines or rtcp attribute, the ALG uses that binding instead of creating a new one.

  If the ALG does not already have a binding, it creates a new one and modifies the SDP, rewriting the "m=" and "c=" lines and rtcp attribute.

Unfortunately, many ALGs are known to work poorly in these corner cases. ICE does not try to work around broken ALGs, as this is outside the scope of its functionality. ICE can help diagnose these conditions, which often show up as a mismatch between the set of candidates and the "m=" and "c=" lines and rtcp attributes. The ice-mismatch attribute is used for this purpose.

ICE works best through ALGs when the signaling is run over TLS. This prevents the ALG from manipulating the SDP messages and interfering with ICE operation. Implementations that are expected to be deployed behind ALGs SHOULD provide for TLS transport of the SDP.

If an SBC is SIP aware but not ICE aware, the result depends on the behavior of the SBC. If it is acting as a proper Back-to-Back User Agent (B2BUA), the SBC will remove any SDP attributes it doesn't understand, including the ICE attributes. Consequently, the call will appear to both endpoints as if the other side doesn't support ICE. This will result in ICE being disabled, and media flowing through the SBC, if the SBC has requested it. If, however, the SBC passes the ICE attributes without modification, yet modifies the default destination for media (contained in the "m=" and "c=" lines and rtcp attribute), this will be detected as an ICE mismatch, and ICE processing is aborted for the call. It is outside of the scope of ICE for it to act as a tool for "working around" SBCs. If one is present, ICE will not be used and the SBC techniques take precedence.

12. IANA Considerations

12.1. SDP Attributes

The original ICE specification defined seven new SDP attributes per the procedures of Section 8.2.4 of [RFC4566]. The registration information is reproduced here.

12.1.1. candidate Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: candidate

Long Form: candidate

Type of Attribute: media-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and provides one of many possible candidate addresses for communication. These addresses are validated with an end-to-end connectivity check using Session Traversal Utilities for NAT (STUN).

Appropriate Values: See Section 5 of RFC XXXX.

12.1.2. remote-candidates Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: remote-candidates

Long Form: remote-candidates

Type of Attribute: media-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and provides the identity of the remote candidates that the offerer wishes the answerer to use in its answer.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.3. ice-lite Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-lite

Long Form: ice-lite

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and indicates that an agent has the minimum functionality required to support ICE inter-operation with a peer that has a full implementation.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.4. ice-mismatch Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-mismatch

Long Form: ice-mismatch

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and indicates that an agent is ICE capable, but did not proceed with ICE due to a mismatch of candidates with the default destination for media signaled in the SDP.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.5. ice-pwd Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-pwd

Long Form: ice-pwd

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and provides the password used to protect STUN connectivity checks.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.6. ice-ufrag Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-ufrag

Long Form: ice-ufrag

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and provides the fragments used to construct the username in STUN connectivity checks.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.7. ice-pacing Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-pacing

Long Form: ice-pacing

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE) to indicate desired connectivity check pacing values.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.8. ice-options Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-options

Long Form: ice-options

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset attribute.

Purpose: This attribute is used with Interactive Connectivity Establishment (ICE), and indicates the ICE options or extensions used by the agent.

Appropriate Values: See Section 5 of RFC XXXX.

12.2. Interactive Connectivity Establishment (ICE) Options Registry

IANA maintains a registry for ice-options identifiers under the Specification Required policy as defined in "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226].

ICE options are of unlimited length according to the syntax in Section 5.6; however, they are RECOMMENDED to be no longer than 20 characters. This is to reduce message sizes and allow for efficient parsing.

In [RFC5245] ICE options could only be defined at the session level. ICE options can now also be defined at the media level. This can be used when aggregating between different ICE agents in the same endpoint, but future options may require to be defined at the media-level. To ensure compatibility with legacy implementation, the media-level ICE options MUST be aggregated into a session-level ICE option. Because aggregation rules depend on the specifics of each option, all new ICE options MUST also define in their specification how the media-level ICE option values are aggregated to generate the value of the session-level ICE option.

[RFC6679] defines the "rtp+ecn" ICE option. The aggregation rule for this ICE option is that if all aggregated media using ICE contain a media-level "rtp+ecn" ICE option then an "rtp+ecn" ICE option MUST be inserted at the session-level. If one of the media does not contain the option, then it MUST NOT be inserted at the session-level.

Section 7 of [ICE-BIS] defines "ice2" ICE option. Since "ice2" is a session level ICE option, no aggregation rules apply.

A registration request MUST include the following information:

- The ICE option identifier to be registered
- Name, Email, and Address of a contact person for the registration

- Organization or individuals having the change control
- Short description of the ICE extension to which the option relates
- Reference(s) to the specification defining the ICE option and the related extensions

13. Acknowledgments

A large part of the text in this document was taken from [RFC5245], authored by Jonathan Rosenberg.

Some of the text in this document was taken from [RFC6336], authored by Magnus Westerlund and Colin Perkins.

Thanks to Thomas Stach for the text in Section 4.2.3, Roman Shpount for suggesting RTCP candidate handling in Section 4.1.1.2 and Simon Perreault for advising on IPV6 address selection when candidate-address includes FQDN.

Thanks to following experts for their reviews and constructive feedback: Christer Holmberg, Adam Roach.

14. References

14.1. Normative References

[ICE-BIS] Keranen, A. and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-ice-rfc5245bis-00 (work in progress), March 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, DOI 10.17487/RFC3262, June 2002, <http://www.rfc-editor.org/info/rfc3262>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3312] Camarillo, G., Ed., Marshall, W., Ed., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, DOI 10.17487/RFC3312, October 2002, <http://www.rfc-editor.org/info/rfc3312>.

[RFC3556] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, DOI 10.17487/RFC3556, July 2003, <http://www.rfc-editor.org/info/rfc3556>.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <http://www.rfc-editor.org/info/rfc3605>.

[RFC4032] Camarillo, G. and P. Kyzivat, "Update to the Session Initiation Protocol (SIP) Preconditions Framework", RFC 4032, DOI 10.17487/RFC4032, March 2005, <http://www.rfc-editor.org/info/rfc4032>.

[RFC4091] Camarillo, G. and J. Rosenberg, "The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework", RFC 4091, June 2005, <http://www.rfc-editor.org/info/rfc4091>.

[RFC4092] Camarillo, G. and J. Rosenberg, "Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)", RFC 4092, June 2005, <http://www.rfc-editor.org/info/rfc4092>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

[RFC5768] Rosenberg, J., "Indicating Support for Interactive Connectivity Establishment (ICE) in the Session Initiation Protocol (SIP)", RFC 5768, DOI 10.17487/RFC5768, April 2010, <http://www.rfc-editor.org/info/rfc5768>.

[RFC6336] Westerlund, M. and C. Perkins, "IANA Registry for Interactive Connectivity Establishment (ICE) Options", RFC 6336, April 2010, <http://www.rfc-editor.org/info/rfc6336>.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <http://www.rfc-editor.org/info/rfc6679>.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", RFC 7092, DOI 10.17487/RFC7092, December 2013, <http://www.rfc-editor.org/info/rfc7092>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <http://www.rfc-editor.org/info/rfc7656>.

14.2. Informative References

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, DOI 10.17487/RFC3725, April 2004, <http://www.rfc-editor.org/info/rfc3725>.

[RFC3960] Camarillo, G. and H. Schulzrinne, "Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)", RFC 3960, DOI 10.17487/RFC3960, December 2004, <http://www.rfc-editor.org/info/rfc3960>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <http://www.rfc-editor.org/info/rfc4340>.

[RFC5626] Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed., "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, DOI 10.17487/RFC5626, October 2009, <http://www.rfc-editor.org/info/rfc5626>.

[RFC5898] Andreasen, F., Camarillo, G., Oran, D., and D. Wing, "Connectivity Preconditions for Session Description Protocol (SDP) Media Streams", RFC 5898, DOI 10.17487/RFC5898, July 2010, <http://www.rfc-editor.org/info/rfc5898>.

Appendix A. Examples

For the example shown in section 12 of [ICE-BIS] the resulting offer (message 5) encoded in SDP looks like:

```
   v=0
   o=jdoe 2890844526 2890842807 IN IP6 $L-PRIV-1.IP
   s=
   c=IN IP6 $NAT-PUB-1.IP
   t=0 0
   a=ice-pwd:asd88fgpdd777uzjYhagZg
   a=ice-ufrag:8hhY
   m=audio $NAT-PUB-1.PORT RTP/AVP 0
   b=RS:0
   b=RR:0
   a=rtpmap:0 PCMU/8000
   a=candidate:1 1 UDP 2130706431 $L-PRIV-1.IP $L-PRIV-1.PORT typ host
   a=candidate:2 1 UDP 1694498815 $NAT-PUB-1.IP $NAT-PUB-1.PORT typ
    srflx raddr $L-PRIV-1.IP rport $L-PRIV-1.PORT
```

The offer, with the variables replaced with their values, will look like (lines folded for clarity):

```
v=0
o=jdoe 2890844526 2890842807 IN IP6 fe80::6676:baff:fe9c:ee4a
s=
c=IN IP6 2001:420:c0e0:1005::61
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 fe80::6676:baff:fe9c:ee4a 8998 typ host
a=candidate:2 1 UDP 1694498815 2001:420:c0e0:1005::61 45664 typ srflx raddr
 fe80::6676:baff:fe9c:ee4a rport 8998
```

The resulting answer looks like:

```
   v=0
   o=bob 2808844564 2808844564 IN IP4 $R-PUB-1.IP
   s=
   c=IN IP4 $R-PUB-1.IP
   t=0 0
   a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
   a=ice-ufrag:9uB6
   m=audio $R-PUB-1.PORT RTP/AVP 0
   b=RS:0
   b=RR:0
   a=rtpmap:0 PCMU/8000
   a=candidate:1 1 UDP 2130706431 $R-PUB-1.IP $R-PUB-1.PORT typ host
```

With the variables filled in:

```
v=0
o=bob 2808844564 2808844564 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
m=audio 3478 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 192.0.2.1 3478 typ host
```

Appendix B. The remote-candidates Attribute

The a=remote-candidates attribute exists to eliminate a race condition between the updated offer and the response to the STUN Binding request that moved a candidate into the Valid list. This race condition is shown in Figure 1. On receipt of message 4, agent L adds a candidate pair to the valid list. If there was only a single media stream with a single component, agent L could now send an updated offer. However, the check from agent R has not yet generated a response, and agent R receives the updated offer (message 7) before getting the response (message 9). Thus, it does not yet know that this particular pair is valid. To eliminate this condition, the actual candidates at R that were selected by the offerer (the remote candidates) are included in the offer itself, and the answerer delays its answer until those pairs validate.

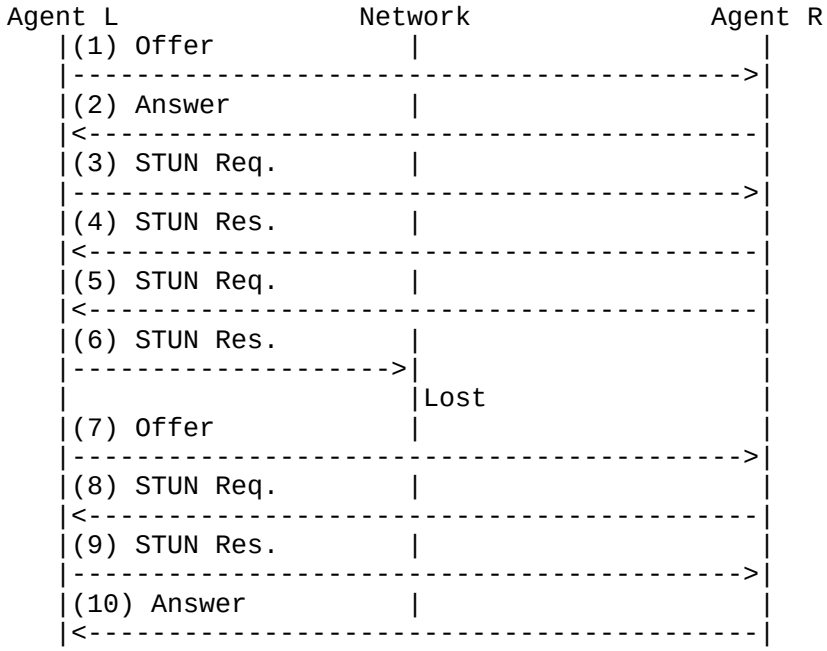


Figure 1: Race Condition Flow

Appendix C. Why Is the Conflict Resolution Mechanism Needed?

When ICE runs between two peers, one agent acts as controlled, and the other as controlling. Rules are defined as a function of implementation type and offerer/answerer to determine who is controlling and who is controlled. However, the specification mentions that, in some cases, both sides might believe they are controlling, or both sides might believe they are controlled. How can this happen?

The condition when both agents believe they are controlled shows up in third party call control cases. Consider the following flow:

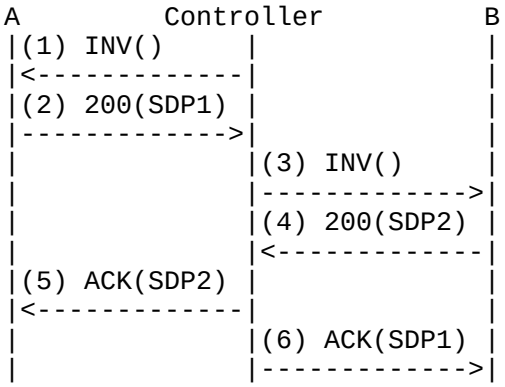


Figure 2: Role Conflict Flow

This flow is a variation on flow III of RFC 3725 [RFC3725]. In fact, it works better than flow III since it produces fewer messages. In this flow, the controller sends an offerless INVITE to agent A, which responds with its offer, SDP1. The agent then sends an offerless INVITE to agent B, which it responds to with its offer, SDP2. The controller then uses the offer from each agent to generate the answers. When this flow is used, ICE will run between agents A and B, but both will believe they are in the controlling role. With the role conflict resolution procedures, this flow will function properly when ICE is used.

At this time, there are no documented flows that can result in the case where both agents believe they are controlled. However, the conflict resolution procedures allow for this case, should a flow arise that would fit into this category.

Appendix D. Why Send an Updated Offer?

Section 11.1 describes rules for sending media. Both agents can send media once ICE checks complete, without waiting for an updated offer. Indeed, the only purpose of the updated offer is to "correct" the SDP so that the default destination for media matches where media is being sent based on ICE procedures (which will be the highest-priority nominated candidate pair).

This begs the question -- why is the updated offer/answer exchange needed at all? Indeed, in a pure offer/answer environment, it would not be. The offerer and answerer will agree on the candidates to use through ICE, and then can begin using them. As far as the agents themselves are concerned, the updated offer/answer provides no new information. However, in practice, numerous components along the signaling path look at the SDP information. These include entities

performing off-path QoS reservations, NAT traversal components such as ALGs and Session Border Controllers (SBCs), and diagnostic tools that passively monitor the network. For these tools to continue to function without change, the core property of SDP -- that the existing, pre-ICE definitions of the addresses used for media -- the "m=" and "c=" lines and the rtcp attribute -- must be retained. For this reason, an updated offer must be sent.

Authors' Addresses

Marc Petit-Huguenin
Impedance Mismatch

Email: marc@petit-huguenin.org

Ari Keranen
Ericsson
Jorvas 02420
Finland

Email: ari.keranen@ericsson.com

Suhas Nandakumar
Cisco Systems
707 Tasman Dr
Milpitas, CA 95035
USA

Email: snandaku@cisco.com

Network Working Group
Internet-Draft
Updates: 4855 (if approved)
Intended status: Standards Track
Expires: September 14, 2017

P. Thatcher
Google
M. Zanaty
S. Nandakumar
Cisco Systems
B. Burman
Ericsson
A. Roach
B. Campen
Mozilla

March 13, 2017

RTP Payload Format Restrictions
draft-ietf-mmusic-rid-10

Abstract

In this specification, we define a framework for specifying restrictions on RTP streams in the Session Description Protocol. This framework defines a new "rid" SDP attribute to unambiguously identify the RTP Streams within a RTP Session and restrict the streams' payload format parameters in a codec-agnostic way beyond what is provided with the regular Payload Types.

This specification updates RFC4855 to give additional guidance on choice of Format Parameter (fmtp) names, and on their relation to the restrictions defined by this document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Table of Contents

1. Terminology

The terms "Source RTP Stream", "Endpoint", "RTP Session", and "RTP Stream" are used as defined in [RFC7656].

[RFC4566] and [RFC3264] terminology is also used where appropriate.

2. Introduction

The Payload Type (PT) field in RTP provides a mapping between the RTP payload format and the associated SDP media description. The SDP rtpmap and/or fmtp attributes are used, for a given PT, to describe the properties of the media that is carried in the RTP payload.

Recent advances in standards have given rise to rich multimedia applications requiring support for multiple RTP Streams within a RTP session [I-D.ietf-mmusic-sdp-bundle-negotiation], [I-D.ietf-mmusic-sdp-simulcast] or having to support a large number of codecs. These demands have unearthed challenges inherent with:

- The restricted RTP PT space in specifying the various payload configurations,
- The codec-specific constructs for the payload formats in SDP,
- Missing or underspecified payload format parameters,
- Overloading of PTs to indicate not just codec configurations, but individual streams within an RTP session.

To expand on these points: [RFC3550] assigns 7 bits for the PT in the RTP header. However, the assignment of static mapping of RTP payload type numbers to payload formats and multiplexing of RTP with other protocols (such as RTCP) could result in a limited number of payload type numbers available for application usage. In scenarios where the number of possible RTP payload configurations exceed the available PT space within a RTP Session, there is a need for a way to represent

the additional restrictions on payload configurations and to effectively map an RTP Stream to its corresponding restrictions. This issue is exacerbated by the increase in techniques - such as simulcast and layered codecs - which introduce additional streams into RTP Sessions.

This specification defines a new SDP framework for restricting Source RTP Streams (Section 2.1.10 [RFC7656]), along with the SDP attributes to restrict payload formats in a codec-agnostic way. This framework can be thought of as a complementary extension to the way the media format parameters are specified in SDP today, via the "a=fmtp" attribute.

The additional restrictions on individual streams are indicated with a new "a=rid" SDP attribute. Note that the restrictions communicated via this attribute only serve to further restrict the parameters that are established on a PT format. They do not relax any existing restrictions.

This specification makes use of the RTP Stream Identifier SDES RTCP item defined in [I-D.ietf-avtext-rid] to provide correlation between the RTP Packets and their format specification in the SDP.

As described in Section 6.2.1, this mechanism achieves backwards compatibility via the normal SDP processing rules, which require unknown a= lines to be ignored. This means that implementations need to be prepared to handle successful offers and answers from other implementations that neither indicate nor honor the restrictions requested by this mechanism.

Further, as described in Section 6 and its subsections, this mechanism achieves extensibility by: (a) having offerers include all supported restrictions in their offer, and (b) having answerers ignore "a=rid" lines that specify unknown restrictions.

3. Key Words for Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]

4. SDP "a=rid" Media Level Attribute

This section defines new SDP media-level attribute [RFC4566], "a=rid", ("restriction identifier") used to communicate a set of restrictions to be applied to an identified RTP Stream. Roughly speaking, this attribute takes the following form (see Section 10 for a formal definition).

a=rid:<rid-id> <direction> [pt=<fmt-list>;]<restriction>=<value>...

An "a=rid" SDP media attribute specifies restrictions defining a unique RTP payload configuration identified via the "rid-id" field. This value binds the restriction to the RTP Stream identified by its RTP Stream Identifier SDES item [I-D.ietf-avtext-rid]. To be clear, implementations that use the "a=rid" parameter in SDP MUST support the RtpStreamId SDES item described in [I-D.ietf-avtext-rid]. Such implementations MUST send it for all streams in an SDP media description ("m=") that have "a=rid" lines remaining after applying the rules in Section 6 and its subsections.

The "direction" field identifies the direction of the RTP Stream packets to which the indicated restrictions are applied. It may be either "send" or "recv". Note that these restriction directions are expressed independently of any "inactive", "sendonly", "recvonly", or "sendrecv" attributes associated with the media section. It is, for example, valid to indicate "recv" restrictions on a "sendonly" stream; those restrictions would apply if, at a future point in time, the stream were changed to "sendrecv" or "recvonly".

The optional "pt=<fmt-list>" lists one or more PT values that can be used in the associated RTP Stream. If the "a=rid" attribute contains no "pt", then any of the PT values specified in the corresponding "m=" line may be used.

The list of zero or more codec-agnostic restrictions (Section 5) describe the restrictions that the corresponding RTP Stream will conform to.

This framework MAY be used in combination with the "a=fmtp" SDP attribute for describing the media format parameters for a given RTP Payload Type. In such scenarios, the "a=rid" restrictions (Section 5) further restrict the equivalent "a=fmtp" attributes.

A given SDP media description MAY have zero or more "a=rid" lines describing various possible RTP payload configurations. A given "rid-id" MUST NOT be repeated in a given media description ("m=" section).

The "a=rid" media attribute MAY be used for any RTP-based media transport. It is not defined for other transports, although other documents may extend its semantics for such transports.

Though the restrictions specified by the "rid" restrictions follow a syntax similar to session-level and media-level parameters, they are

defined independently. All "rid" restrictions MUST be registered with IANA, using the registry defined in Section 12.

Section 10 gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for the "rid" attribute. The "a=rid" media attribute is not dependent on charset.

5. "a=rid" restrictions

This section defines the "a=rid" restrictions that can be used to restrict the RTP payload encoding format in a codec-agnostic way.

The following restrictions are intended to apply to video codecs in a codec-independent fashion.

- max-width, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the width of the stream when a rotation of zero degrees is encoded.

- max-height, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the height of the stream when a rotation of zero degrees is encoded.

- max-fps, for frame rate in frames per second. For encoders that do not use a fixed framerate for encoding, this value should restrict the minimum amount of time between frames: the time between any two consecutive frames SHOULD NOT be less than 1/max-fps seconds.

- max-fs, for frame size in pixels per frame. This is the product of frame width and frame height, in pixels, for rectangular frames.

- max-br, for bit rate in bits per second. The restriction applies to the media payload only, and does not include overhead introduced by other layers (e.g., RTP, UDP, IP, or Ethernet). The exact means of keeping within this limit are left up to the implementation, and instantaneous excursions outside the limit are permissible. For any given one-second sliding window, however, the total number of bits in the payload portion of RTP SHOULD NOT exceed the value specified in "max-br."

- max-pps, for pixel rate in pixels per second. This value SHOULD be handled identically to max-fps, after performing the following conversion: max-fps = max-pps / (width * height). If the stream resolution changes, this value is recalculated. Due to this

recalculation, excursions outside the specified maximum are possible near resolution change boundaries.

- o max-bpp, for maximum number of bits per pixel, calculated as an average of all samples of any given coded picture. This is expressed as a floating point value, with an allowed range of 0.0001 to 48.0. These values MUST be encoded with at most four digits to the right of the decimal point.

- o depend, to identify other streams that the stream depends on. The value is a comma-separated list of rid-ids. These rid-ids identify RTP streams that this stream depends on in order to allow for proper interpretation. The mechanism defined in this document allows for such dependencies to be expressed only when the streams are in the same media section.

All the restrictions are optional and are subject to negotiation based on the SDP Offer/Answer rules described in Section 6.

This list is intended to be an initial set of restrictions. Future documents may define additional restrictions; see Section 12.2. While this document does not define restrictions for audio codecs or any media types other than video, there is no reason such restrictions should be precluded from definition and registration by other documents.

Section 10 provides formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for each of the "a=rid" restrictions defined in this section.

6. SDP Offer/Answer Procedures

This section describes the SDP Offer/Answer [RFC3264] procedures when using this framework.

Note that "rid-id" values are only required to be unique within a media section ("m-line"); they do not necessarily need to be unique within an entire RTP session. In traditional usage, each media section is sent on its own unique 5-tuple, which provides an unambiguous scope. Similarly, when using BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], MID values associate RTP streams uniquely to a single media description.

6.1. Generating the Initial SDP Offer

For each RTP media description in the offer, the offerer MAY choose to include one or more "a=rid" lines to specify a configuration profile for the given set of RTP Payload Types.

In order to construct a given "a=rid" line, the offerer must follow these steps:

1. It MUST generate a "rid-id" that is unique within a media description

2. It MUST set the direction for the "rid-id" to one of "send" or "recv"

3. It MAY include a listing of SDP media formats (usually corresponding to RTP payload types) allowed to appear in the RTP Stream. Any Payload Types chosen MUST be a valid payload type for the media section (that is, it must be listed on the "m=" line). The order of the listed formats is significant; the alternatives are listed from (left) most preferred to (right) least preferred. When using RID, this preference overrides the normal codec preference as expressed by format type ordering on the "m="-line, using regular SDP rules.

4. The Offerer then chooses zero or more "a=rid" restrictions (Section 5) to be applied to the RTP Stream, and adds them to the "a=rid" line.

5. If the offerer wishes the answerer to have the ability to specify a restriction, but does not wish to set a value itself, it includes the name of the restriction in the "a=rid" line, but without any indicated value.

Note: If an "a=fmtp" attribute is also used to provide media-format-specific parameters, then the "a=rid" restrictions will further restrict the equivalent "a=fmtp" parameters for the given Payload Type for the specified RTP Stream.

If a given codec would require an "a=fmtp" line when used without "a=rid" then the offer MUST include a valid corresponding "a=fmtp" line even when using "a=rid".

6.2. Answerer processing the SDP Offer

6.2.1. "a=rid"-unaware Answerer

If the receiver doesn’t support the framework defined in this specification, the entire "a=rid" line is ignored following the standard [RFC3264] Offer/Answer rules.

Section 6.1 requires the offer to include a valid "a=fmtp" line for any media formats that otherwise require it (in other words, the "a=rid" line cannot be used to replace "a=fmtp" configuration). As a

result, ignoring the "a=rid" line is always guaranteed to result in a valid session description.

6.2.2. "a=rid"-aware Answerer

If the answerer supports the "a=rid" attribute, the following verification steps are executed, in order, for each "a=rid" line in a received offer:

1. The answerer ensures that the "a=rid" line is syntactically well formed. In the case of a syntax error, the "a=rid" line is discarded.

2. Extract the rid-id from the "a=rid" line and verify its uniqueness within a media section. In the case of a duplicate, the entire "a=rid" line, and all "a=rid" lines with rid-ids that duplicate this line, are discarded and MUST NOT be included in the SDP Answer.

3. If the "a=rid" line contains a "pt=", the list of payload types is verified against the list of valid payload types for the media section (that is, those listed on the "m=" line). Any PT missing from the "m=" line is discarded from the set of values in the "pt=". If no values are left in the "pt=" parameter after this processing, then the "a=rid" line is discarded.

4. If the "direction" field is "recv", The answerer ensures that "a=rid" restrictions are supported. In the case of an unsupported restriction, the "a=rid" line is discarded.

5. If the "depend" restriction is included, the answerer MUST make sure that the listed rid-ids unambiguously match the rid-ids in the media description. Any "depend" "a=rid" lines that do not are discarded.

6. The answerer verifies that the restrictions are consistent with at least one of the codecs to be used with the RTP Stream. If the "a=rid" line contains a "pt=", it contains the list of such codecs; otherwise, the list of such codecs is taken from the associated "m=" line. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec properties for all codecs, then the "a=rid" line is discarded.

Note that the answerer does not need to understand every restriction present in a "send" line: if a stream sender restricts the stream in a way that the receiver does not understand, this causes no issues with interoperability.

6.3. Generating the SDP Answer

Having performed verification of the SDP offer as described in Section 6.2.2, the answerer shall perform the following steps to generate the SDP answer.

For each "a=rid" line:

1. The value of the "direction" field is reversed: "send" is changed to "recv", and "recv" is changed to "send".

2. The answerer MAY choose to modify specific "a=rid" restriction values in the answer SDP. In such a case, the modified value MUST be more restricted than the ones specified in the offer. The answer MUST NOT include any restrictions that were not present in the offer.

3. The answerer MUST NOT modify the "rid-id" present in the offer.

4. If the "a=rid" line contains a "pt=", the answerer is allowed to discard one or more media formats from a given "a=rid" line. If the answerer chooses to discard all the media formats from an "a=rid" line, the answerer MUST discard the entire "a=rid" line. If the offer did not contain a "pt=" for a given "a=rid" line, then the answer MUST NOT contain a "pt=" in the corresponding line.

5. In cases where the answerer is unable to support the payload configuration specified in a given "a=rid" line with a direction of "recv" in the offer, the answerer MUST discard the corresponding "a=rid" line. This includes situations in which the answerer does not understand one or more of the restrictions in an "a=rid" line with a direction of "recv".

Note: in the case that the answerer uses different PT values to represent a codec than the offerer did, the "a=rid" values in the answer use the PT values that are present in its answer.

6.4. Offerer Processing of the SDP Answer

The offerer SHALL follow these steps when processing the answer:

1. The offerer matches the "a=rid" line in the answer to the "a=rid" line in the offer using the "rid-id". If no matching line can be located in the offer, the "a=rid" line is ignored.

2. If the answer contains any restrictions that were not present in the offer, then the offerer SHALL discard the "a=rid" line.

3. If the restrictions have been changed between the offer and the answer, the offerer MUST ensure that the modifications can be supported; if they cannot, the offerer SHALL discard the "a=rid" line.

4. If the "a=rid" line in the answer contains a "pt=" but the offer did not, the offerer SHALL discard the "a=rid" line.

5. If the "a=rid" line in the answer contains a "pt=" and the offer did as well, the offerer verifies that the list of payload types is a subset of those sent in the corresponding "a=rid" line in the offer. Note that this matching must be performed semantically rather than on literal PT values, as the remote end may not be using symmetric PTs. For the purpose of this comparison: for each PT listed on the "a=rid" line in the answer, the offerer looks up the corresponding "a=rtpmap" and "a=fmtp" lines in the answer. It then searches the list of "pt=" values indicated in the offer, and attempts to find one with an equivalent set of "a=rtpmap" and "a=fmtp" lines in the offer. If all PTs in the answer can be matched, then the "pt=" values pass validation; otherwise, it fails. If this validation fails, the offerer SHALL discard the "a=rid" line. Note that this semantic comparison necessarily requires an understanding of the meaning of codec parameters, rather than a rote byte-wise comparison of their values.

6. If the "a=rid" line contains a "pt=", the offerer verifies that the attribute values provided in the "a=rid" attributes are consistent with the corresponding codecs and their other parameters. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec properties, then the offerer SHALL discard the "a=rid" line.

7. The offerer verifies that the restrictions are consistent with at least one of the codecs to be used with the RTP Stream. If the "a=rid" line contains a "pt=", it contains the list of such codecs; otherwise, the list of such codecs is taken from the associated "m=" line. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec properties for all codecs, then the offerer SHALL discard the "a=rid" line.

Any "a=rid" line present in the offer that was not matched by step 1 above has been discarded by the answerer, and does not form part of the negotiated restrictions on an RTP Stream. The offerer MAY still apply any restrictions it indicated in an "a=rid" line with a direction field of "send", but it is not required to do so.

It is important to note that there are several ways in which an offer can contain a media section with "a=rid" lines, but the corresponding media section in the response does not. This includes situations in which the answerer does not support "a=rid" at all, or does not support the indicated restrictions. Under such circumstances, the offerer MUST be prepared to receive a media stream to which no restrictions have been applied.

6.5. Modifying the Session

Offers and answers inside an existing session follow the rules for initial session negotiation. Such an offer MAY propose a change in the number of RIDs in use. To avoid race conditions with media, any RIDs with proposed changes SHOULD use a new ID, rather than re-using one from the previous offer/answer exchange. RIDs without proposed changes SHOULD re-use the ID from the previous exchange.

7. Use with Declarative SDP

This document does not define the use of RID in declarative SDP. If concrete use cases for RID in declarative SDP use are identified in the future, we expect that additional specifications will address such use.

8. Interaction with Other Techniques

Historically, a number of other approaches have been defined that allow restricting media streams via SDP. These include:

- Codec-specific configuration set via format parameters ("a=fmtp"); for example, the H.264 "max-fs" format parameter [RFC6184]

- Size restrictions imposed by image attribute attributes ("a=imageattr") [RFC6236]

When the mechanism described in this document is used in conjunction with these other restricting mechanisms, it is intended to impose additional restrictions beyond those communicated in other techniques.

In an offer, this means that "a=rid" lines, when combined with other restrictions on the media stream, are expected to result in a non-empty union. For example, if image attributes are used to indicate that a PT has a minimum width of 640, then specification of "max-width=320" in an "a=rid" line that is then applied to that PT is nonsensical. According to the rules of Section 6.2.2, this will result in the corresponding "a=rid" line being ignored by the recipient.

In an answer, the "a=rid" lines, when combined with the other restrictions on the media stream, are also expected to result in a non-empty union. If the implementation generating an answer wishes to restrict a property of the stream below that which would be allowed by other parameters (e.g., those specified in "a=fmtp" or "a=imageattr"), its only recourse is to discard the "a=rid" line altogether, as described in Section 6.3. If it instead attempts to restrict the stream beyond what is allowed by other mechanisms, then the offerer will ignore the corresponding "a=rid" line, as described in Section 6.4.

The following subsections demonstrate these interactions using commonly-used video codecs. These descriptions are illustrative of the interaction principles outlined above, and are not normative.

8.1. Interaction with VP8 Format Parameters

[RFC7741] defines two format parameters for the VP8 codec. Both correspond to restrictions on receiver capabilities, and never indicate sending restrictions.

8.1.1. max-fr - Maximum Framerate

The VP8 "max-fr" format parameter corresponds to the "max-fps" restriction defined in this specification. If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fps" parameter, then the sent stream will conform to the smaller of the two values.

8.1.2. max-fs - Maximum Framesize, in VP8 Macroblocks

The VP8 "max-fs" format parameter corresponds to the "max-fs" restriction defined in this document, by way of a conversion factor of the number of pixels per macroblock (typically 256). If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fs" parameter, then the sent stream will conform to the smaller of the two values; that is, the number of pixels per frame will not exceed:

```
min(rid_max_fs, fmtp_max_fs * macroblock_size)
```

This fmtp parameter also has bearing on the max-height and max-width parameters. Section 6.1 of [RFC7741] requires that the width and height of the frame in macroblocks are also required to be less than int(sqrt(fmtp_max_fs * 8)). Accordingly, the maximum width of a transmitted stream will be limited to:

```
min(rid_max_width, int(sqrt(fmtp_max_fs * 8)) * macroblock_width)
```

Similarly, the stream's height will be limited to:

```
min(rid_max_height, int(sqrt(fmtp_max_fs * 8)) * macroblock_height)
```

8.2. Interaction with H.264 Format Parameters

[RFC6184] defines format parameters for the H.264 video codec. The majority of these parameters do not correspond to codec-independent restrictions:

- deint-buf-cap
- in-band-parameter-sets
- level-asymmetry-allowed
- max-rcmd-nalu-size
- max-cpb
- max-dpb
- packetization-mode
- redundant-pic-cap
- sar-supported
- sar-understood
- sprop-deint-buf-req
- sprop-init-buf-time
- sprop-interleaving-depth
- sprop-level-parameter-sets
- sprop-max-don-diff
- sprop-parameter-sets
- use-level-src-parameter-sets

Note that the max-cpb and max-dpb format parameters for H.264 correspond to restrictions on the stream, but they are specific to

the way the H.264 codec operates, and do not have codec-independent equivalents.

The following codec format parameters correspond to restrictions on receiver capabilities, and never indicate sending restrictions.

8.2.1. profile-level-id and max-recv-level - Negotiated Sub-Profile

These parameters include a "level" indicator, which acts as an index into Table A-1 of [H264]. This table contains a number of parameters, several of which correspond to the restrictions defined in this document. [RFC6184] also defines format parameters for the H.264 codec that may increase the maximum values indicated by the negotiated level. The following sections describe the interaction between these parameters and the restrictions defined by this document. In all cases, the H.264 parameters being discussed are the maximum of those indicated by [H264] Table A-1 and those indicated in the corresponding "a=fmtp" line.

8.2.2. max-br / MaxBR - Maximum Video Bitrate

The H.264 "MaxBR" parameter (and its equivalent "max-br" format parameter) corresponds to the "max-bps" restriction defined in this specification, by way of a conversion factor of 1000 or 1200; see [RFC6184] for details regarding which factor gets used under differing circumstances.

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fps" parameter, then the sent stream will conform to the smaller of the two values - that is:

```
min(rid_max_br, h264_MaxBR * conversion_factor)
```

8.2.3. max-fs / MaxFS - Maximum Framesize, in H.264 Macroblocks

The H.264 "MaxFs" parameter (and its equivalent "max-fs" format parameter) corresponds roughly to the "max-fs" restriction defined in this document, by way of a conversion factor of 256 (the number of pixels per macroblock).

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fs" parameter, then the sent stream will conform to the smaller of the two values - that is:

```
min(rid_max_fs, h264_MaxFs * 256)
```

8.2.4. max-mbps / MaxMBPS - Maximum Macroblock Processing Rate

The H.264 "MaxMBPS" parameter (and its equivalent "max-mbps" format parameter) corresponds roughly to the "max-pps" restriction defined in this document, by way of a conversion factor of 256 (the number of pixels per macroblock).

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-pps" parameter, then the sent stream will conform to the smaller of the two values - that is:

```
min(rid_max_pps, h264_MaxMBPS * 256)
```

8.2.5. max-smbps - Maximum Decoded Picture Buffer

The H.264 "max-smbps" format parameter operates the same way as the "max-mpbs" format parameter, under the hypothetical assumption that all macroblocks are static macroblocks. It is handled by applying the conversion factor described in Section 8.1 of [RFC6184], and the result of this conversion is applied as described in Section 8.2.4.

9. Format Parameters for Future Payloads

Registrations of future RTP payload format specifications that define media types that have parameters matching the RID restrictions specified in this memo SHOULD name those parameters in a manner that matches the names of those RID restrictions, and SHOULD explicitly state what media type parameters are restricted by what RID restrictions.

10. Formal Grammar

This section gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for each of the new media and "a=rid" attributes defined in this document.

```
rid-syntax        = "a=rid:" rid-id SP rid-dir
                    [ rid-pt-param-list / rid-param-list ]

rid-id            = 1*(alpha-numeric / "-" / "_")

alpha-numeric     = < as defined in {{RFC4566}} >

rid-dir           = "send" / "recv"

rid-pt-param-list = SP rid-fmt-list *(";" rid-param)
```

```
   rid-param-list    = SP rid-param *(";" rid-param)

   rid-fmt-list      = "pt=" fmt *( "," fmt )

   fmt               = < as defined in {{RFC4566}} >

   rid-param         = rid-width-param
                     / rid-height-param
                     / rid-fps-param
                     / rid-fs-param
                     / rid-br-param
                     / rid-pps-param
                     / rid-bpp-param
                     / rid-depend-param
                     / rid-param-other

   rid-width-param   = "max-width" [ "=" int-param-val ]

   rid-height-param  = "max-height" [ "=" int-param-val ]

   rid-fps-param     = "max-fps" [ "=" int-param-val ]

   rid-fs-param      = "max-fs" [ "=" int-param-val ]

   rid-br-param      = "max-br" [ "=" int-param-val ]

   rid-pps-param     = "max-pps" [ "=" int-param-val ]

   rid-bpp-param     = "max-bpp" [ "=" float-param-val ]

   rid-depend-param  = "depend=" rid-list

   rid-param-other   = 1*(alpha-numeric / "-") [ "=" param-val ]

   rid-list          = rid-id *( "," rid-id )

   int-param-val     = 1*DIGIT

   float-param-val   = 1*DIGIT "." 1*DIGIT

   param-val         = *( %x20-58 / %x60-7E )
                     ; Any printable character except semicolon
```

11. SDP Examples

Note: see [I-D.ietf-mmusic-sdp-simulcast] for examples of RID used in simulcast scenarios.

11.1. Many Bundled Streams using Many Codecs

In this scenario, the offerer supports the Opus, G.722, G.711 and DTMF audio codecs, and VP8, VP9, H.264 (CBP/CHP, mode 0/1), H.264-SVC (SCBP/SCHP) and H.265 (MP/M10P) for video. An 8-way video call (to a mixer) is supported (send 1 and receive 7 video streams) by offering 7 video media sections (1 sendrecv at max resolution and 6 recvonly at smaller resolutions), all bundled on the same port, using 3 different resolutions. The resolutions include:

- 1 receive stream of 720p resolution is offered for the active speaker.
- 2 receive streams of 360p resolution are offered for the prior 2 active speakers.
- 4 receive streams of 180p resolution are offered for others in the call.

NOTE: The SDP given below skips a few lines to keep the example short and focused, as indicated by either the "..." or the comments inserted.

The offer for this scenario is shown below.

```
...
m=audio 10000 RTP/SAVPF 96 9 8 0 123
a=rtpmap:96 OPUS/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:123 telephone-event/8000
a=mid:a1
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=rtpmap:98 VP8/90000
a=fmtp:98 max-fs=3600; max-fr=30
a=rtpmap:99 VP9/90000
a=fmtp:99 max-fs=3600; max-fr=30
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42401f; packetization-mode=0
a=rtpmap:101 H264/90000
```

```
   a=fmtp:101 profile-level-id=42401f; packetization-mode=1
   a=rtpmap:102 H264/90000
   a=fmtp:102 profile-level-id=640c1f; packetization-mode=0
   a=rtpmap:103 H264/90000
   a=fmtp:103 profile-level-id=640c1f; packetization-mode=1
   a=rtpmap:104 H264-SVC/90000
   a=fmtp:104 profile-level-id=530c1f
   a=rtpmap:105 H264-SVC/90000
   a=fmtp:105 profile-level-id=560c1f
   a=rtpmap:106 H265/90000
   a=fmtp:106 profile-id=1; level-id=93
   a=rtpmap:107 H265/90000
   a=fmtp:107 profile-id=2; level-id=93
   a=sendrecv
   a=mid:v1 (max resolution)
   a=rid:1 send max-width=1280;max-height=720;max-fps=30
   a=rid:2 recv max-width=1280;max-height=720;max-fps=30
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v2 (medium resolution)
   a=rid:3 recv max-width=640;max-height=360;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v3 (medium resolution)
   a=rid:3 recv max-width=640;max-height=360;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v4 (small resolution)
   a=rid:4 recv max-width=320;max-height=180;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
   ...same rtpmap/fmtp as above...
   ...same rid:4 as above for mid:v5,v6,v7 (small resolution)...
   ...
```

11.2. Scalable Layers

Adding scalable layers to a session within a multiparty conference gives a selective forwarding unit (SFU) further flexibility to selectively forward packets from a source that best match the bandwidth and capabilities of diverse receivers. Scalable encodings have dependencies between layers, unlike independent simulcast streams. RIDs can be used to express these dependencies using the "depend" restriction. In the example below, the highest resolution is offered to be sent as 2 scalable temporal layers (using MRST). See [I-D.ietf-mmusic-sdp-simulcast] for additional detail about simulcast usage.

```
Offer:
...
m=audio ...same as previous example ...
...
m=video ...same as previous example ...
...same rtpmap/fmtp as previous example ...
a=sendrecv
a=mid:v1 (max resolution)
a=rid:0 send max-width=1280;max-height=720;max-fps=15
a=rid:1 send max-width=1280;max-height=720;max-fps=30;depend=0
a=rid:2 recv max-width=1280;max-height=720;max-fps=30
a=rid:5 send max-width=640;max-height=360;max-fps=15
a=rid:6 send max-width=320;max-height=180;max-fps=15
a=simulcast: send rid=0;1;5;6 recv rid=2
...
...same m=video sections as previous example for mid:v2-v7...
...
```

12. IANA Considerations

This specification updates [RFC4855] to give additional guidance on choice of Format Parameter (fmtp) names, and on their relation to RID restrictions.

12.1. New SDP Media-Level attribute

This document defines "rid" as SDP media-level attribute. This attribute must be registered by IANA under "Session Description Protocol (SDP) Parameters" under "att-field (media level only)".

The "rid" attribute is used to identify properties of RTP stream with in a RTP Session. Its format is defined in Section 10.

The formal registration information for this attribute follows.

Contact name, email address, and telephone number

IETF MMUSIC Working Group
mmusic@ietf.org
+1 510 492 4080

Attribute name (as it will appear in SDP)

rid

Long-form attribute name in English

Restriction Identifier

Type of attribute (session level, media level, or both)

Media Level

Whether the attribute value is subject to the charset attribute

The attribute is not dependent on charset.

A one-paragraph explanation of the purpose of the attribute

The "rid" SDP attribute is used to to unambiguously identify the RTP Streams within a RTP Session and restrict the streams' payload format parameters in a codec-agnostic way beyond what is provided with the regular Payload Types.

A specification of appropriate attribute values for this attribute

Valid values are defined by the ABNF in [RFCXXXXX]

Multiplexing (Mux) Category

SPECIAL

12.2. Registry for RID-Level Parameters

This specification creates a new IANA registry named "att-field (rid level)" within the SDP parameters registry. The "a=rid" restrictions MUST be registered with IANA and documented under the same rules as for SDP session-level and media-level attributes as specified in [RFC4566].

Parameters for "a=rid" lines that modify the nature of encoded media MUST be of the form that the result of applying the modification to the stream results in a stream that still complies with the other

parameters that affect the media. In other words, restrictions always have to restrict the definition to be a subset of what is otherwise allowable, and never expand it.

New restriction registrations are accepted according to the "Specification Required" policy of [RFC5226], provided that the specification includes the following information:

- contact name, email address, and telephone number
- restriction name (as it will appear in SDP)
- long-form restriction name in English
- whether the restriction value is subject to the charset attribute
- an explanation of the purpose of the restriction
- a specification of appropriate attribute values for this restriction
- an ABNF definition of the restriction

The initial set of "a=rid" restriction names, with definitions in Section 5 of this document, is given below:

Type	SDP Name	Reference
att-field	(rid level)	
	max-width	[RFCXXXX]
	max-height	[RFCXXXX]
	max-fps	[RFCXXXX]
	max-fs	[RFCXXXX]
	max-br	[RFCXXXX]
	max-pps	[RFCXXXX]
	max-bpp	[RFCXXXX]
	depend	[RFCXXXX]

It is conceivable that a future document wants to define a RID-level restrictions that contain string values. These extensions need to take care to conform to the ABNF defined for rid-param-other. In particular, this means that such extensions will need to define escaping mechanisms if they want to allow semicolons, unprintable characters, or byte values greater than 127 in the string.

13. Security Considerations

As with most SDP parameters, a failure to provide integrity protection over the "a=rid" attributes provides attackers a way to modify the session in potentially unwanted ways. This could result in an implementation sending greater amounts of data than a recipient wishes to receive. In general, however, since the "a=rid" attribute can only restrict a stream to be a subset of what is otherwise allowable, modification of the value cannot result in a stream that is of higher bandwidth than would be sent to an implementation that does not support this mechanism.

The actual identifiers used for RIDs are expected to be opaque. As such, they are not expected to contain information that would be sensitive, were it observed by third-parties.

14. Acknowledgements

Many thanks to review from Cullen Jennings, Magnus Westerlund, and Paul Kyzivat. Thanks to Colin Perkins for input on future payload type handing.

15. References

15.1. Normative References

[I-D.ietf-avtext-rid]
Roach, A., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDES)", draft-ietf-avtext-rid-09 (work in progress), October 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <http://www.rfc-editor.org/info/rfc4855>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

15.2. Informative References

[H264] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services (V9)", February 2014, <http://www.itu.int/rec/T-REC-H.264-201304-I>.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-36 (work in progress), October 2016.

[I-D.ietf-mmusic-sdp-simulcast] Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-07 (work in progress), January 2017.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <http://www.rfc-editor.org/info/rfc6184>.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, DOI 10.17487/RFC6236, May 2011, <http://www.rfc-editor.org/info/rfc6236>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <http://www.rfc-editor.org/info/rfc7656>.

[RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <http://www.rfc-editor.org/info/rfc7741>.

Authors' Addresses

Peter Thatcher
Google

Email: pthatcher@google.com

Mo Zanaty
Cisco Systems

Email: mzanaty@cisco.com

Suhas Nandakumar
Cisco Systems

Email: snandaku@cisco.com

Bo Burman
Ericsson

Email: bo.burman@ericsson.com

Adam Roach
Mozilla

Email: adam@nostrum.com

Byron Campen
Mozilla

Email: bcampen@mozilla.com

MMUSIC Working Group
Internet-Draft
Updates: 3264 (if approved)
Intended status: Standards Track
Expires: April 30, 2017

C. Holmberg
Ericsson
H. Alvestrand
Google
C. Jennings
Cisco

October 27, 2016

Negotiating Media Multiplexing Using the Session Description Protocol (SDP)

draft-ietf-mmusic-sdp-bundle-negotiation-36.txt

Abstract

This specification defines a new Session Description Protocol (SDP) Grouping Framework extension, 'BUNDLE'. The extension can be used with the SDP Offer/Answer mechanism to negotiate the usage of a single address:port combination (BUNDLE address) for receiving media, referred to as bundled media, specified by multiple SDP media descriptions ("m=" lines).

To assist endpoints in negotiating the use of bundle this specification defines a new SDP attribute, 'bundle-only', which can be used to request that specific media is only used if bundled. The specification also updates RFC 3264, to allow usage of zero port values without meaning that media is rejected.

There are multiple ways to correlate the bundled RTP packets with the appropriate media descriptions. This specification defines a new Real-time Transport Protocol (RTP) source description (SDES) item and a new RTP header extension that provides an additional way to do this correlation by using them to carry a value that associates the RTP/RTCP packets with a specific media description.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 30, 2017.

Copyright Notice

Table of Contents

1. Introduction

When multimedia communications are established, each 5-tuple reserved for an individual media stream consume additional resources (especially when Interactive Connectivity Establishment (ICE) [RFC5245] is used). For this reason, it is attractive to use a 5-tuple for multiple media streams.

This specification defines a way to use a single address:port combination (BUNDLE address) for receiving media specified by multiple SDP media descriptions ("m=" lines).

This specification defines a new SDP Grouping Framework [RFC5888] extension called 'BUNDLE'. The extension can be used with the Session Description Protocol (SDP) Offer/Answer mechanism [RFC3264] to negotiate the usage of a BUNDLE group. Within the BUNDLE group, a BUNDLE address is used for receiving media specified by multiple "m=" lines. This is referred to as bundled media.

The offerer and answerer [RFC3264] use the BUNDLE extension to negotiate the BUNDLE addresses, one for the offerer (offerer BUNDLE address) and one for the answerer (answerer BUNDLE address), to be used for receiving the bundled media specified by a BUNDLE group. Once the offerer and the answerer have negotiated a BUNDLE group, they associate their respective BUNDLE address with each "m=" line in

the BUNDLE group. The BUNDLE addresses are used to receive all media specified by the BUNDLE group.

The use of a BUNDLE group and a BUNDLE address also allows the usage of a single set of Interactive Connectivity Establishment (ICE) [RFC5245] candidates for multiple "m=" lines.

This specification also defines a new SDP attribute, 'bundle-only', which can be used to request that specific media is only used if kept within a BUNDLE group. The specification also updates RFC 3264, to allow usage of zero port values without meaning that media is rejected.

As defined in RFC 4566 [RFC4566], the semantics of assigning the same transport address (IP address and port) to multiple "m=" lines are undefined, and there is no grouping defined by such means. Instead, an explicit grouping mechanism needs to be used to express the intended semantics. This specification provides such an extension.

This specification also updates sections 5.1, 8.1 and 8.2 of RFC 3264 [RFC3264]. The update allows an answerer to assign a non-zero port value to an "m=" line in an SDP answer, even if the "m=" line in the associated SDP offer contained a zero port value.

This specification also defines a new Real-time Transport Protocol (RTP) [RFC3550] source description (SDES) item, 'MID', and a new RTP SDES header extension that can be used to associate RTP streams with media descriptions.

SDP bodies can contain multiple BUNDLE groups. A given BUNDLE address MUST only be associated with a single BUNDLE group. The procedures in this specification apply independently to a given BUNDLE group. All RTP based media flows described by a single BUNDLE group belong to a single RTP session [RFC3550].

The BUNDLE extension is backward compatible. Endpoints that do not support the extension are expected to generate offers and answers without an SDP 'group:BUNDLE' attribute, and are expected to associate a unique address with each "m=" line within an offer and answer, according to the procedures in [RFC4566] and [RFC3264]

2. Terminology

"m=" line: SDP bodies contain one or more media descriptions. Each media description is identified by an SDP "m=" line.

5-tuple: A collection of the following values: source address, source port, destination address, destination port, and transport-layer protocol.

Unique address: An IP address and port combination that is associated with only one "m=" line in an offer or answer.

Shared address: An IP address and port combination that is associated with multiple "m=" lines within an offer or answer.

Offerer BUNDLE-tag: The first identification-tag in a given SDP ’group:BUNDLE’ attribute identification-tag list in an offer.

Answerer BUNDLE-tag: The first identification-tag in a given SDP ’group:BUNDLE’ attribute identification-tag list in an answer.

Offerer BUNDLE address: Within a given BUNDLE group, an IP address and port combination used by an offerer to receive all media specified by each "m=" line within the BUNDLE group.

Answerer BUNDLE address: Within a given BUNDLE group, an IP address and port combination used by an answerer to receive all media specified by each "m=" line within the BUNDLE group.

BUNDLE group: A set of "m=" lines, created using an SDP Offer/Answer exchange, which uses the same BUNDLE address for receiving media.

Bundled "m=" line: An "m=" line, whose identification-tag is placed in an SDP ’group:BUNDLE’ attribute identification-tag list in an offer or answer.

Bundle-only "m=" line: A bundled "m=" line with an associated SDP ’bundle-only’ attribute.

Bundled media: All media specified by a given BUNDLE group.

Initial offer: The first offer, within an SDP session (e.g. a SIP dialog when the Session Initiation Protocol (SIP) [RFC3261] is used to carry SDP), in which the offerer indicates that it wants to create a given BUNDLE group.

Subsequent offer: An offer which contains a BUNDLE group that has been created as part of a previous offer/answer exchange.

Identification-tag: A unique token value that is used to identify an "m=" line. The SDP ’mid’ attribute [RFC5888], associated with an "m=" line, carries an unique identification-tag. The session-level SDP ’group’ attribute [RFC5888] carries a list of identification-

tags, identifying the "m=" lines associated with that particular 'group' attribute.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP offer/answer mechanism [RFC3264]. Declarative usage of SDP is out of scope of this document, and is thus undefined.

5. SDP Grouping Framework BUNDLE Extension

This section defines a new SDP Grouping Framework extension [RFC5888], 'BUNDLE'. The BUNDLE extension can be used with the SDP Offer/Answer mechanism to negotiate the usage of a single address:port combination (BUNDLE address) for receiving bundled media.

A single address:port combination is also used for sending bundled media. The address:port combination used for sending bundled media MAY be the same as the BUNDLE address, used to receive bundled media, depending on whether symmetric RTP [RFC4961] is used.

All media associated with a BUNDLE group MUST be transport using the same transport-layer protocol (e.g., UDP or TCP).

The BUNDLE extension is indicated using an SDP 'group' attribute with a "BUNDLE" semantics value [RFC5888]. An identification-tag is associated with each bundled "m=" line, and each identification-tag is listed in the SDP 'group:BUNDLE' attribute identification-tag list. Each "m=" line whose identification-tag is listed in the identification-tag list is associated with a given BUNDLE group.

SDP bodies can contain multiple BUNDLE groups. Any given bundled "m=" line MUST NOT be associated with more than one BUNDLE group.

NOTE: The order of the "m=" lines listed in the SDP 'group:BUNDLE' attribute identification-tag list does not have to be the same as the order in which the "m=" lines occur in the SDP.

Section 8 defines the detailed SDP Offer/Answer procedures for the BUNDLE extension.

6. SDP 'bundle-only' Attribute

This section defines a new SDP media-level attribute [RFC4566], 'bundle-only'. 'bundle-only' is a property attribute [RFC4566], and hence has no value.

Name: bundle-only

Value: N/A

Usage Level: media

Charset Dependent: no

Example:

```
a=bundle-only
```

In order to ensure that an answerer that does not support the BUNDLE extension always rejects a bundled "m=" line, the offerer can assign a zero port value to the "m=" line. According to [RFC3264] an answerer will reject such "m=" line. By associating an SDP 'bundle-only' attribute with such "m=" line, the offerer can request that the answerer accepts the "m=" line if the answerer supports the Bundle extension, and if the answerer keeps the "m=" line within the associated BUNDLE group.

NOTE: Once the offerer BUNDLE address has been selected, the offerer does not need to include the 'bundle-only' attribute in subsequent offers. By associating the offerer BUNDLE address with an "m=" line of a subsequent offer, the offerer will ensure that the answerer will either keep the "m=" line within the BUNDLE group, or the answerer will have to reject the "m=" line.

The usage of the 'bundle-only' attribute is only defined for a bundled "m=" line with a zero port value, within an offer. Other usage is unspecified.

Section 8 defines the detailed SDP Offer/Answer procedures for the 'bundle-only' attribute.

7. SDP Information Considerations

This section describes restrictions associated with the usage of SDP parameters within a BUNDLE group. It also describes, when parameter and attribute values have been associated with each bundled "m=" line, how to calculate a value for the whole BUNDLE group.

7.1. Connection Data (c=)

The "c=" line nettype value [RFC4566] associated with a bundled "m=" line MUST be 'IN'.

The "c=" line addrtype value [RFC4566] associated with a bundled "m=" line MUST be 'IP4' or 'IP6'. The same value MUST be associated with each "m=" line.

NOTE: Extensions to this specification can specify usage of the BUNDLE mechanism for other nettype and addrtype values than the ones listed above.

7.2. Bandwidth (b=)

An offerer and answerer MUST use the rules and restrictions defined in [I-D.ietf-mmusic-sdp-mux-attributes] for associating the SDP bandwidth (b=) line with bundled "m=" lines.

7.3. Attributes (a=)

An offerer and answerer MUST use the rules and restrictions defined in [I-D.ietf-mmusic-sdp-mux-attributes] for associating SDP attributes with bundled "m=" lines.

8. SDP Offer/Answer Procedures

This section describes the SDP Offer/Answer [RFC3264] procedures for:

- Negotiating and creating a BUNDLE group; and
- Selecting the BUNDLE addresses (offerer BUNDLE address and answerer BUNDLE address); and
- Adding an "m=" line to a BUNDLE group; and
- Moving an "m=" line out of a BUNDLE group; and
- Disabling an "m=" line within a BUNDLE group.

The generic rules and procedures defined in [RFC3264] and [RFC5888] also apply to the BUNDLE extension. For example, if an offer is rejected by the answerer, the previously negotiated SDP parameters and characteristics (including those associated with a BUNDLE group) apply. Hence, if an offerer generates an offer in which the offerer wants to create a BUNDLE group, and the answerer rejects the offer, the BUNDLE group is not created.

The procedures in this section are independent of the media type or "m=" line proto value represented by a bundled "m=" line. Section 10 defines additional considerations for RTP based media. Section 6 defines additional considerations for the usage of the SDP 'bundle-only' attribute. Section 11 defines additional considerations for the usage of Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] mechanism.

SDP offers and answers can contain multiple BUNDLE groups. The procedures in this section apply independently to a given BUNDLE group.

8.1. Mux Category Considerations

When an offerer associates SDP attributes with a bundled "m=" line associated with a shared address, IDENTICAL and TRANSPORT mux category SDP attributes [I-D.ietf-mmusic-sdp-mux-attributes] are associated with the "m=" line only if the "m=" line is also associated with the offerer BUNDLE-tag. Otherwise the offerer MUST NOT associate such SDP attributes with the "m=" line.

When an answerer associates SDP attributes with a bundled "m=" line, IDENTICAL and TRANSPORT mux category SDP attributes are associated with the "m=" line only if the "m=" line is also associated with the answerer BUNDLE-tag. Otherwise the answerer MUST NOT associated such SDP attributes with the "m=" line.

NOTE: As bundled "m=" lines associated with a shared address will share the same IDENTICAL and TRANSPORT mux category SDP attributes, and attribute values, there is no need to associate such SDP attributes with each "m=" line. The attributes and attribute values are implicitly applied to each "m=" line associated with the shared address.

8.2. Generating the Initial SDP Offer

When an offerer generates an initial offer, in order to create a BUNDLE group, it MUST:

- Assign a unique address to each "m=" line within the offer, following the procedures in [RFC3264], unless the media line is a 'bundle-only' "m=" line (see below); and

- Add an SDP 'group:BUNDLE' attribute to the offer; and

- Place the identification-tag of each bundled "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list; and

- Indicate which unique address the offerer suggests as the offerer BUNDLE address [Section 8.2.1].

If the offerer wants to request that the answerer accepts a given bundled "m=" line only if the answerer keeps the "m=" line within the BUNDLE group, the offerer MUST:

- Associate an SDP 'bundle-only' attribute [Section 8.2.1] with the "m=" line; and

- Assign a zero port value to the "m=" line.

NOTE: If the offerer assigns a zero port value to an "m=" line, but does not also associate an SDP 'bundle-only' attribute with the "m=" line, it is an indication that the offerer wants to disable the "m=" line [Section 8.5.4].

[Section 17.1] shows an example of an initial offer.

8.2.1. Suggesting the offerer BUNDLE address

In the offer, the address associated with the "m=" line associated with the offerer BUNDLE-tag indicates the address that the offerer suggests as the offerer BUNDLE address.

The "m=" line associated with the offerer BUNDLE-tag MUST NOT contain a zero port value or an SDP 'bundle-only' attribute.

8.2.2. Example: Initial SDP Offer

The example shows an initial SDP offer. The offer includes two "m=" lines in the SDP, and suggests that both are included in a BUNDLE group. The audio "m=" line is associated with the offerer BUNDLE-tag (placed first in the SDP group:BUNDLE attribute identificatoin-id list).

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

8.3. Generating the SDP Answer

When an answerer generates an answer that contains a BUNDLE group, the following general SDP grouping framework restrictions, defined in [RFC5888], also apply to the BUNDLE group:

- The answerer MUST NOT include a BUNDLE group in the answer, unless the offerer requested the BUNDLE group to be created in the corresponding offer; and

- The answerer MUST NOT include an "m=" line within a BUNDLE group, unless the offerer requested the "m=" line to be within that BUNDLE group in the corresponding offer.

If the answer contains a BUNDLE group, the answerer MUST:

- Select an Offerer BUNDLE Address [Section 8.3.1]; and

- Select an Answerer BUNDLE Address [Section 8.3.2];

The answerer is allowed to select a new Answerer BUNDLE address each time it generates an answer to an offer.

If the answerer does not want to keep an "m=" line within a BUNDLE group, it MUST:

- Move the "m=" line out of the BUNDLE group [Section 8.3.3]; or
- Reject the "m=" line [Section 8.3.4];

If the answerer keeps a bundle-only "m=" line within the BUNDLE group, it follows the procedures (associates the answerer BUNDLE address with the "m=" line etc) for any other "m=" line kept within the BUNDLE group.

If the answerer does not want to keep a bundle-only "m=" line within the BUNDLE group, it MUST reject the "m=" line [Section 8.3.4].

The answerer MUST NOT associate an SDP 'bundle-only' attribute with any "m=" line in an answer.

NOTE: If a bundled "m=" line in an offer contains a zero port value, but the "m=" line does not contain an SDP 'bundle-only' attribute, it is an indication that the offerer wants to disable the "m=" line [Section 8.5.4].

8.3.1. Answerer Selection of Offerer Bundle Address

In an offer, the address (unique or shared) associated with the bundled "m=" line associated with the offerer BUNDLE-tag indicates the address that the offerer suggests as the offerer BUNDLE address [Section 8.2.1]. The answerer MUST check whether that "m=" line fulfils the following criteria:

- The answerer will not move the "m=" line out of the BUNDLE group [Section 8.3.3]; and
- The answerer will not reject the "m=" line [Section 8.3.4]; and
- The "m=" line does not contain a zero port value.

If all of the criteria above are fulfilled, the answerer MUST select the address associated with the "m=" line as the offerer BUNDLE address. In the answer, the answerer BUNDLE-tag represents the "m=" line, and the address associated with the "m=" line in the offer becomes the offerer BUNDLE address.

If one or more of the criteria are not fulfilled, the answerer MUST select the next identification-tag in the identification-tag list, and perform the same criteria check for the "m=" line associated with that identification-tag. If there are no more identification-tags in

the identification-tag list, the answerer MUST NOT create the BUNDLE group. In addition, unless the answerer rejects the whole offer, the answerer MUST apply the answerer procedures for moving an "m=" line out of a BUNDLE group [Section 8.3.3] to each bundled "m=" line in the offer when creating the answer.

[Section 17.1] shows an example of an offerer BUNDLE address selection.

8.3.2. Answerer Selection of Answerer BUNDLE Address

When the answerer selects a BUNDLE address for itself, referred to as the answerer BUNDLE address, it MUST associate that address with each bundled "m=" line within the created BUNDLE group in the answer.

The answerer MUST NOT associate the answerer BUNDLE address with an "m=" line that is not within the BUNDLE group, or to an "m=" line that is within another BUNDLE group.

[Section 17.1] shows an example of an answerer BUNDLE address selection.

8.3.3. Moving A Media Description Out Of A BUNDLE Group

When an answerer wants to move an "m=" line out of a BUNDLE group, it MUST first check the following criteria:

- In the corresponding offer, the "m=" line is associated with a shared address (e.g. a previously selected offerer BUNDLE address); or

- In the corresponding offer, an SDP ’bundle-only’ attribute is associated with the "m=" line, and the "m=" line contains a zero port value.

If either criteria above is fulfilled, the answerer MUST reject the "m=" line [Section 8.3.4].

Otherwise, if in the corresponding offer the "m=" line is associated with a unique address, the answerer MUST associate a unique address with the "m=" line in the answer (the answerer does not reject the "m=" line).

In addition, in either case above, the answerer MUST NOT place the identification-tag, associated with the moved "m=" line, in the SDP ’group’ attribute identification-tag list associated with the BUNDLE group.

8.3.4. Rejecting A Media Description In A BUNDLE Group

When an answerer rejects an "m=" line, it MUST associate an address with a zero port value with the "m=" line in the answer, according to the procedures in [RFC3264].

In addition, the answerer MUST NOT place the identification-tag, associated with the rejected "m=" line, in the SDP 'group' attribute identification-tag list associated with the BUNDLE group.

8.3.5. Example: SDP Answer

The example shows an SDP answer, based on the SDP offer in [Section 8.2.2]. The answers acceppts both "m=" lines in the BUNDLE group.

SDP Answer

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

8.4. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answer contains a BUNDLE group, the offerer MUST check that any bundled "m=" line in the answer was indicated as bundled in the corresponding offer. If there is no mismatch, the offerer MUST use the offerer BUNDLE address, selected by the answerer [Section 8.3.1], as the address for each bundled "m=" line.

NOTE: As the answerer might reject one or more bundled "m=" lines, or move a bundled "m=" line out of a BUNDLE group, each bundled "m=" line in the offer might not be indicated as bundled in the answer.

If the answer does not contain a BUNDLE group, the offerer MUST process the answer as a normal answer.

8.5. Modifying the Session

When an offerer generates a subsequent offer, it MUST associate the previously selected offerer BUNDLE address [Section 8.3.1] with each bundled "m=" line (including any bundle-only "m=" line), except if:

- The offerer suggests a new offerer BUNDLE address [Section 8.5.1]; or
- The offerer wants to add a bundled "m=" line to the BUNDLE group [Section 8.5.2]; or
- The offerer wants to move a bundled "m=" line out of the BUNDLE group [Section 8.5.3]; or
- The offerer wants to disable the bundled "m=" line [Section 8.5.4].

In addition, the offerer MUST select an offerer BUNDLE-tag [Section 8.2.1] associated with the previously selected offerer BUNDLE address, unless the offerer suggests a new offerer BUNDLE address.

8.5.1. Suggesting a new offerer BUNDLE address

When an offerer generates an offer, in which it suggests a new offerer BUNDLE address [Section 8.2.1], the offerer MUST:

- Assign the address (shared address) to each "m=" line within the BUNDLE group; or
- Assign the address (unique address) to one bundled "m=" line.

In addition, the offerer MUST indicate that the address is the new suggested offerer BUNDLE address [Section 8.2.1].

NOTE: Unless the offerer associates the new suggested offerer BUNDLE address with each bundled "m=" line, it can associate unique addresses with any number of bundled "m=" lines (and the previously selected offerer BUNDLE address to any remaining bundled "m=" line)

if it wants to suggest multiple alternatives for the new offerer BUNDLE address.

8.5.2. Adding a media description to a BUNDLE group

When an offerer generates an offer, in which it wants to add a bundled "m=" line to a BUNDLE group, the offerer MUST:

- Assign a unique address to the added "m=" line; or
- Assign the previously selected offerer BUNDLE address to the added "m=" line; or
- If the offerer associates a new (shared address) suggested offerer BUNDLE address with each bundled "m=" line [Section 8.5.1], also associate that address with the added "m=" line.

In addition, the offerer MUST add the identification-tag associated with the added "m=" line to the SDP 'group:BUNDLE' attribute identification-tag list with the BUNDLE group [Section 8.2.1].

NOTE: Assigning a unique address to the "m=" line allows the answerer to move the "m=" line out of the BUNDLE group [Section 8.3.3], without having to reject the "m=" line.

If the offerer associates a unique address with the added "m=" line, and if the offerer suggests that address as the new offerer BUNDLE address [Section 8.5.1], the offerer BUNDLE-tag MUST represent the added "m=" line [Section 8.2.1].

If the offerer associates a new suggested offerer BUNDLE address with each bundled "m=" line [Section 8.5.1], including the added "m=" line, the offerer BUNDLE-tag MAY represent the added "m=" line [Section 8.2.1].

[Section 17.3] shows an example where an offerer sends an offer in order to add a bundled "m=" line to a BUNDLE group.

8.5.3. Moving A Media Description Out Of A BUNDLE Group

When an offerer generates an offer, in which it wants to move a bundled "m=" line out of a BUNDLE group it was added to in a previous offer/answer transaction, the offerer:

- MUST associate a unique address with the "m=" line; and

- MUST NOT place the identification-tag associated with the "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group.

NOTE: If the removed "m=" line is associated with the previously selected BUNDLE-tag, the offerer needs to suggest a new BUNDLE-tag [Section 8.2.1].

NOTE: If an "m=" line, when being moved out of a BUNDLE group, is added to another BUNDLE group, the offerer applies the procedures in [Section 8.5.2] to the "m=" line.

[Section 17.4] shows an example of an offer for moving an "m=" line out of a BUNDLE group.

8.5.4. Disabling A Media Description In A BUNDLE Group

When an offerer generates an offer, in which it wants to disable a bundled "m=" line (added to the BUNDLE group in a previous offer/answer transaction), the offerer:

- MUST associate an address with a zero port value with the "m=" line, following the procedures in [RFC4566]; and
- MUST NOT place the identification-tag associated with the "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group.

[Section 17.5] shows an example of an offer for disabling an "m=" line within a BUNDLE group.

9. Protocol Identification

Each "m=" line within a BUNDLE group MUST use the same transport-layer protocol. If bundled "m=" lines use different protocols on top of the transport-layer protocol, there MUST exist a publicly available specification which describes a mechanism, for this particular protocol combination, how to associate received data with the correct protocol.

In addition, if received data can be associated with more than one bundled "m=" line, there MUST exist a publicly available specification which describes a mechanism for associating the received data with the correct "m=" line.

This document describes a mechanism to identify the protocol of received data among the STUN, DTLS and SRTP protocols (in any combination), when UDP is used as transport-layer protocol, but does

not describe how to identify different protocols transported on DTLS. While the mechanism is generally applicable to other protocols and transport-layer protocols, any such use requires further specification around how to multiplex multiple protocols on a given transport-layer protocol, and how to associate received data with the correct protocols.

9.1. STUN, DTLS, SRTP

Section 5.1.2 of [RFC5764] describes a mechanism to identify the protocol of a received packet among the STUN, Datagram Transport Layer Security (DTLS) and SRTP protocols (in any combination). If an offer or answer includes bundled "m=" lines that represent these protocols, the offerer or answerer MUST support the mechanism described in [RFC5764], and no explicit negotiation is required in order to indicate support and usage of the mechanism.

[RFC5764] does not describe how to identify different protocols transported on DTLS, only how to identify the DTLS protocol itself. If multiple protocols are transported on DTLS, there MUST exist a specification describing a mechanism for identifying each individual protocol. In addition, if a received DTLS packet can be associated with more than one "m=" line, there MUST exist a specification which describes a mechanism for associating the received DTLS packet with the correct "m=" line.

[Section 10.2] describes how to associate the packets in a received SRTP stream with the correct "m=" line.

10. RTP Considerations

10.1. Single RTP Session

All RTP-based media within a single BUNDLE group belong to a single RTP session [RFC3550].

Since a single RTP session is used for each bundle group, all "m=" lines representing RTP-based media in a bundle group will share a single SSRC numbering space [RFC3550].

The following rules and restrictions apply for a single RTP session:

- A specific payload type value can be used in multiple bundled "m=" lines only if each codec associated with the payload type number shares an identical codec configuration [Section 10.1.1].

- The proto value in each bundled RTP-based "m=" line MUST be identical (e.g. RTP/AVPF).

- The RTP MID header extension MUST be enabled, by associating an SDP 'extmap' attribute [RFC5285], with a 'urn:ietf:params:rtp-hdrext:sdes:mid' URI value, with each bundled RTP-based "m=" line in every offer and answer.

- A given SSRC MUST NOT transmit RTP packets using payload types that originate from different bundled "m=" lines.

NOTE: The last bullet above is to avoid sending multiple media types from the same SSRC. If transmission of multiple media types are done with time overlap, RTP and RTCP fail to function. Even if done in proper sequence this causes RTP Timestamp rate switching issues [RFC7160]. However, once an SSRC has left the RTP session (by sending an RTCP BYE packet), that SSRC can be reused by another source (possibly associated with a different bundled "m=" line) after a delay of 5 RTCP reporting intervals (the delay is to ensure the SSRC has timed out, in case the RTCP BYE packet was lost [RFC3550]).

10.1.1. Payload Type (PT) Value Reuse

Multiple bundled "m=" lines might represent RTP based media. As all RTP based media specified by a BUNDLE group belong to the same RTP session, in order for a given payload type value to be used inside more than one bundled "m=" line, all codecs associated with the payload type number MUST share an identical codec configuration. This means that the codecs MUST share the same media type, encoding name, clock rate and any parameter that can affect the codec configuration and packetization.
[I-D.ietf-mmusic-sdp-mux-attributes] lists SDP attributes, whose attribute values must be identical for all codecs that use the same payload type value.

10.2. Associating RTP/RTCP Streams With Correct SDP Media Description

As described in [RFC3550], RTP/RTCP packets are associated with RTP streams [RFC7656]. Each RTP stream is identified by an SSRC value, and each RTP/RTCP packet carries an SSRC value that is used to associate the packet with the correct RTP stream (an RTCP packet can carry multiple SSRC values, and might therefore be associated with multiple RTP streams).

In order to be able to process received RTP/RTCP packets correctly it must be possible to associate an RTP stream with the correct "m=" line, as the "m=" line and SDP attributes associated with the "m=" line contain information needed to process the packets.

As all RTP streams associated with a BUNDLE group are using the same address:port combination for sending and receiving RTP/RTCP packets,

the local address:port combination cannot be used to associate an RTP stream with the correct "m=" line. In addition, multiple RTP streams might be associated with the same "m=" line.

Also, as described in [Section 10.1.1], the same payload type value might be used by multiple RTP streams, in which case the payload type value cannot be used to associate an RTP stream with the correct "m=" line.

An offerer and answerer can inform each other which SSRC values they will use for an RTP stream by using the SDP 'ssrc' attribute [RFC5576]. However, an offerer will not know which SSRC values the answerer will use until the offerer has received the answer providing that information. Due to this, before the offerer has received the answer, the offerer will not be able to associate an RTP stream with the correct "m=" line using the SSRC value associated with the RTP stream. In addition, the offerer and answerer may start using new SSRC values mid-session, without informing each other using the SDP 'ssrc' attribute.

In order for an offerer and answerer to always be able to associate an RTP stream with the correct "m=" line, the offerer and answerer using the BUNDLE extension MUST support the mechanism defined in Section 14, where the offerer and answerer insert the identification-tag (provided by the remote peer) associated with an "m=" line in RTP and RTCP packets associated with a BUNDLE group.

The mapping from an SSRC to an identification-tag is carried in RTCP SDES packets or in RTP header extensions (Section 14). Since a compound RTCP packet can contain multiple RTCP SDES packets, and each RTCP SDES packet can contain multiple chunks, an RTCP packet can contain several SSRC to identification-tag mappings. The offerer and answerer maintain tables mapping RTP streams identified by SSRC, to a€œm=a€œ lines identified by the identification-tag. These tables are updated each time an RTP/RTCP packet containing one or more mappings from SSRC to identification-tag is received. Note that the mapping from SSRC to identification-tag can change at any time during an RTP session. Once an offerer or answerer recieve an RTP/RTCP packet carrying an identification-tag and an SSRC value (an RTCP packet might carry multiple identification-tags and SSRC values), it creates a mapping between the SSRC value and the identification-tag, in order to associate the RTP stream with the "m=" line associated with the identification-tag. Note that the mapping might change mid-session if, for a given SSRC value, a different identification-tag is provided in an RTP/RTCP packet.

If an offerer and answerer is not able to associate an RTP stream with an "m=" line (using the mechanisms described in this section, or using other appropriate mechanism, e.g, based on the payload type value if it is unique to a single "m=" line), it MUST either drop the RTP/RTCP packets associated with the RTP stream, or process them in an application specific manner, once non-stream specific processing (e.g., related to congestion control) of the packets have occurred. Note that RTCP packets can report on multiple RTP streams.

10.3. RTP/RTCP Multiplexing

Within a BUNDLE group, the offerer and answerer MUST enable RTP/RTCP multiplexing [RFC5761] for the RTP-based media specified by the BUNDLE group.

When RTP/RTCP multiplexing is enabled, the same address:port combination will be used for sending all RTP packets and the RTCP packets associated with the BUNDLE group. Each endpoint will send the packets towards the BUNDLE address of the other endpoint. The same address:port combination MAY be used for receiving RTP packets and RTCP packets.

10.3.1. SDP Offer/Answer Procedures

This section describes how an offerer and answerer use the SDP 'rtcp-mux' attribute [RFC5761] and the SDP 'rtcp-mux-only' attribute [I-D.ietf-mmusic-mux-exclusive] to negotiate usage of RTP/RTCP multiplexing for RTP-based media specified by a BUNDLE group.

The procedures in this section only apply to RTP-based "m=" lines.

10.3.1.1. Generating the Initial SDP Offer

When an offerer generates an initial offer, the offerer MUST associate an SDP 'rtcp-mux' attribute [RFC5761] with each bundled RTP-based "m=" line in the offer, including a bundle-only "m=" line. In addition, the offerer MUST associate an SDP 'rtcp-mux-only' attribute [I-D.ietf-mmusic-mux-exclusive] with each RTP-based bundle-only "m=" line, and MAY associated an SDP 'rtcp-mux-only' attribute with other bundled RTP-based "m=" lines.

NOTE: Whether the offerer associates an SDP 'rtcp-mux-only' attribute with a bundled "m=" line or not depends on whether the offerer supports fallback to usage of a separate port for RTCP in case the answerer does not include the "m=" line in the BUNDLE group.

NOTE: If the offerer associates an SDP 'rtcp-mux' attribute with a bundled "m=" line, but does not associate an SDP 'rtcp-mux-only'

attribute with the "m=" line, the offerer can also associate an SDP 'rtcp' attribute [RFC3605] with the "m=" line in order to provide a fallback port for RTCP, as described in [RFC5761]. However, the fallback port will only be used in case the answerer does not include the "m=" line in the BUNDLE group.

In the initial offer, the address:port combination for RTCP MUST be unique in each bundled RTP-based "m=" line (excluding a bundle-only "m=" line), similar to RTP.

10.3.1.2. Generating the SDP Answer

When an answerer generates an answer, if the answerer accepts one or more RTP-based "m=" lines within a BUNDLE group, the answerer MUST enable usage of RTP/RTCP multiplexing. The answerer MUST associate an SDP "rtcp-mux" attribute with each RTP-based "m=" line in the answer. In addition, if an "m=" line in the corresponding offer contained an SDP "rtcp-mux-only" attribute, the answerer MUST associate an SDP "rtcp-mux-only" attribute with the "m=" line in the answer.

If an RTP-based "m=" line in the corresponding offer did not contain an SDP 'rtcp-mux' attribute, the answerer MUST NOT include the "m=" line within a BUNDLE group in the answer.

If an RTP-based "m=" line in the corresponding offer contained an SDP "rtcp-mux-only" attribute, and if the answerer moves the "m=" line out of the BUNDLE group in the answer Section 8.3.3, the answerer MUST still either enable RTP/RTCP multiplexing for the media associated with the "m=" line, or reject the "m=" line Section 8.3.4.

The answerer MUST NOT associate an SDP 'rtcp' attribute with any bundled "m=" line in the answer. The answerer will use the port value of the selected offerer BUNDLE address for sending RTP and RTCP packets associated with each RTP-based bundled "m=" line towards the offerer.

If the usage of RTP/RTCP multiplexing within a BUNDLE group has been negotiated in a previous offer/answer transaction, the answerer MUST associate an SDP 'rtcp-mux' attribute with each bundled RTP-based "m=" line in the answer.

10.3.1.3. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answerer has accepted the usage of RTP/RTCP multiplexing (see Section 10.3.1.2), the answerer follows the procedures for RTP/RTCP multiplexing defined in [RFC5761]. The offerer will use the port value associated with the

answerer BUNDLE address for sending RTP and RTCP packets associated with each RTP-based bundled "m=" line towards the answerer.

NOTE: It is considered a protocol error if the answerer has not accepted the usage of RTP/RTCP multiplexing for RTP-based "m=" lines that the answerer included in the BUNDLE group.

10.3.1.4. Modifying the Session

When an offerer generates a subsequent offer, for each RTP-based "m=" line that was previously added to the BUNDLE group the offerer MUST associate an SDP 'rtcp-mux' attribute and an SDP 'rtcp-mux-only' attribute with the "m=" line in the same way it was previously done, unless the offerer wants to disable or remove the "m=" line from the BUNDLE group.

If the offerer wants to add a bundled RTP-based "m=" line to the BUNDLE group, it associates an SDP 'rtcp-mux' attribute and an SDP 'rtcp-mux-only' attribute with the "m=" line using the procedures in [Section 10.3.1.1].

11. ICE Considerations

This section describes how to use the BUNDLE grouping extension together with the Interactive Connectivity Establishment (ICE) mechanism [I-D.ietf-ice-rfc5245bis].

The generic procedures for negotiating usage of ICE using SDP, defined in [I-D.ietf-mmusic-ice-sip-sdp], also apply to usage of ICE with BUNDLE, with the following exceptions:

- When BUNDLE addresses for a BUNDLE group have been selected for both endpoints, ICE connectivity checks and keep-alives only need to be performed for the whole BUNDLE group, instead of per bundled "m=" line.

- Among bundled "m=" lines with which the offerer has associated a shared address, the offerer only associates ICE-related media-level SDP attributes with the "m=" line associated with the offerer BUNDLE-tag.

- Among bundled "m=" lines with which the answerer has associated a shared address, the answerer only associates ICE-related media-level SDP attributes with the "m=" line associated with the answerer BUNDLE-tag.

Support and usage of ICE mechanism together with the BUNDLE extension is OPTIONAL.

11.1. SDP Offer/Answer Procedures

When an offerer associates a unique address with a bundled "m=" line (excluding any bundle-only "m=" line), the offerer MUST associate SDP 'candidate' attributes (and other applicable ICE-related media-level SDP attributes), containing unique ICE properties (candidates etc), with the "m=" line, according to the procedures in [I-D.ietf-mmusic-ice-sip-sdp].

When an offerer associates a shared address with a bundled "m=" line, if the "m=" line is associated with the offerer BUNDLE-tag, the offerer MUST associate SDP 'candidate' attributes (and other applicable ICE-related media-level SDP attributes), containing shared ICE properties, with the "m=" line. If the "m=" line is not associated with the offerer BUNDLE-tag, the offerer MUST NOT associate ICE-related SDP attributes with the "m=" line.

When an answerer associates a shared address with a bundled "m=" line, if the "m=" line is associated with the answerer BUNDLE-tag, the answerer MUST associate SDP 'candidate' attributes (and other applicable ICE-related media-level SDP attributes), containing shared ICE properties, with the "m=" line. If the "m=" line is not associated with the answerer BUNDLE-tag, the answerer MUST NOT associate ICE-related SDP attributes with the "m=" line.

NOTE: As most ICE-related media-level SDP attributes belong to the TRANSPORT mux category [I-D.ietf-mmusic-sdp-mux-attributes], the offerer and answerer follow the rules in Section 8.1. However, in the case of ICE-related media-level attributes, the rules apply to all attributes (see note below), even if they belong to a different mux category.

NOTE: The following ICE-related media-level SDP attributes are defined in [I-D.ietf-mmusic-ice-sip-sdp]: 'candidiate', 'remote-candidates', 'ice-mismatch', 'ice-ufrag', 'ice-pwd', and 'ice-pacing'.

11.1.1. Generating the Initial SDP Offer

When an offerer generates an initial offer, the offerer MUST associate ICE-related media-level SDP attributes with each bundled "m=" line, according to [Section 11.1].

11.1.2. Generating the SDP Answer

When an answerer generates an answer that contains a BUNDLE group, the answerer MUST associate ICE-related SDP attributes with the "m="

line associated with the answerer BUNDLE-tag, according to [Section 11.1].

11.1.3. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answerer supports and uses the ICE mechanism and the BUNDLE extension, the offerer MUST associate the ICE properties associated with the offerer BUNDLE address, selected by the answerer [Section 8.3.1], with each bundled "m=" line.

11.1.4. Modifying the Session

When an offerer generates a subsequent offer, it MUST associate unique or shared ICE properties to one or more bundled "m=" lines, according to [Section 11.1].

12. DTLS Considerations

One or more media streams within a BUNDLE group might use the Datagram Transport Layer Security (DTLS) protocol [RFC6347] in order to encrypt the data, or to negotiate encryption keys if another encryption mechanism is used to encrypt media.

When DTLS is used within a BUNDLE group, the following rules apply:

- There can only be one DTLS association [RFC6347] associated with the BUNDLE group; and
- Each usage of the DTLS association within the BUNDLE group MUST use the same mechanism for determining which endpoints (the offerer or answerer) become DTLS client and DTLS server; and
- Each usage of the DTLS association within the Bundle group MUST use the same mechanism for determining whether an offer or answer will trigger the establishment of a new DTLS association, or whether an existing DTLS association will be used; and
- If the DTLS client supports DTLS-SRTP [RFC5764] it MUST include the ’use_srtp’ extension [RFC5764] in the DTLS ClientHello message [RFC5764], The client MUST include the extension even if the usage of DTLS-SRTP is not negotiated as part of the multimedia session (e.g., SIP session [RFC3261].

NOTE: The inclusion of the ’use_srtp’ extension during the initial DTLS handshake ensures that a DTLS renegotiation will not be required in order to include the extension, in case DTLS-SRTP encrypted media is added to the BUNDLE group later during the multimedia session.

13. Update to RFC 3264

This section replaces the text of the following sections of RFC 3264:

- Section 5.1 (Unicast Streams).
- Section 8.2 (Removing a Media Stream).
- Section 8.4 (Putting a Unicast Media Stream on Hold).

13.1. Original text of section 5.1 (2nd paragraph) of RFC 3264

For recvonly and sendrecv streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For sendonly RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer indicates that the stream is offered but MUST NOT be used. This has no useful semantics in an initial offer, but is allowed for reasons of completeness, since the answer can contain a zero port indicating a rejected stream (Section 6). Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero indicates that the media stream is not wanted.

13.2. New text replacing section 5.1 (2nd paragraph) of RFC 3264

For recvonly and sendrecv streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For sendonly RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer by default indicates that the stream is offered but MUST NOT be used, but an extension mechanism might specify different semantics for the usage of a zero port value. Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero by default indicates that the media stream is not wanted.

13.3. Original text of section 8.2 (2nd paragraph) of RFC 3264

A stream that is offered with a port of zero MUST be marked with port zero in the answer. Like the offer, the answer MAY omit all attributes present previously, and MAY list just a single media format from amongst those in the offer.

13.4. New text replacing section 8.2 (2nd paragraph) of RFC 3264

A stream that is offered with a port of zero MUST by default be marked with port zero in the answer, unless an extension mechanism, which specifies semantics for the usage of a non-zero port value, is used. If the stream is marked with port zero in the answer, the answer MAY omit all attributes present previously, and MAY list just a single media format from amongst those in the offer."

13.5. Original text of section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn’t allow for RTCP to be used with held streams, doesn’t work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn’t know the addresses and ports at the time of the offer. Of course, when used, the port number MUST NOT be zero, which would specify that the stream has been disabled. An agent MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

13.6. New text replacing section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn’t allow for RTCP to be used with held streams, doesn’t work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn’t know the addresses and ports at the time of the offer. Of course, when used, the port number MUST NOT be zero, if it would specify that the stream has been disabled. However, an extension mechanism might specify different semantics of the zero port number usage. An agent MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

14. RTP/RTCP extensions for identification-tag transport

SDP Offerers and Answerers [RFC3264] can associate identification-tags with "m=" lines within SDP Offers and Answers, using the procedures in [RFC5888]. Each identification-tag uniquely represents an "m=" line.

This section defines a new RTCP SDES item [RFC3550], 'MID', which is used to carry identification-tags within RTCP SDES packets. This section also defines a new RTP SDES header extension [RFC7941], which is used to carry the 'MID' RTCP SDES item in RTP packets.

The SDES item and RTP SDES header extension make it possible for a receiver to associate each RTP stream with with a specific "m=" line, with which the receiver has associated an identification-tag, even if those "m=" lines are part of the same RTP session. The RTP SDES header extension also ensures that the media recipient gets the identification-tag upon receipt of the first decodable media and is able to associate the media with the correct application.

A media recipient informs the media sender about the identification-tag associated with an "m=" line through the use of an 'mid' attribute [RFC5888]. The media sender then inserts the identification-tag in RTCP and RTP packets sent to the media recipient.

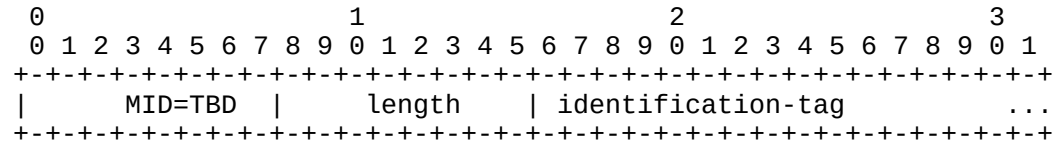
NOTE: This text above defines how identification-tags are carried in SDP Offers and Answers. The usage of other signalling protocols for carrying identification-tags is not prevented, but the usage of such protocols is outside the scope of this document.

[RFC3550] defines general procedures regarding the RTCP transmission interval. The RTCP MID SDES item SHOULD be sent in the first few RTCP packets sent after joining the session, and SHOULD be sent regularly thereafter. The exact number of RTCP packets in which this SDES item is sent is intentionally not specified here, as it will depend on the expected packet loss rate, the RTCP reporting interval, and the allowable overhead.

The RTP SDES header extension for carrying the 'MID' RTCP SDES SHOULD be included in some RTP packets at the start of the session and whenever the SSRC changes. It might also be useful to include the header extension in RTP packets that comprise access points in the media (e.g., with video I-frames). The exact number of RTP packets in which this header extension is sent is intentionally not specified here, as it will depend on expected packet loss rate and loss patterns, the overhead the application can tolerate, and the importance of immediate receipt of the identification-tag.

For robustness purpose, endpoints need to be prepared for situations where the reception of the identification-tag is delayed, and SHOULD NOT terminate sessions in such cases, as the identification-tag is likely to arrive soon.

14.1. RTCP MID SDES Item



The identification-tag payload is UTF-8 encoded, as in SDP.

The identification-tag is not zero terminated.

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

14.2. RTP SDES Header Extension For MID

The payload, containing the identification-tag, of the RTP SDES header extension element can be encoded using either the one-byte or two-byte header [RFC7941]. The identification-tag payload is UTF-8 encoded, as in SDP.

The identification-tag is not zero terminated. Note, that the set of header extensions included in the packet needs to be padded to the next 32-bit boundary using zero bytes [RFC5285].

As the identification-tag is included in either an RTCP SDES item or an RTP SDES header extension, or both, there should be some consideration about the packet expansion caused by the identification-tag. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension’s size needs to be taken into account when encoding the media.

It is recommended that the identification-tag is kept short. Due to the properties of the RTP header extension mechanism, when using the one-byte header, a tag that is 1-3 bytes will result in a minimal number of 32-bit words used for the RTP SDES header extension, in case no other header extensions are included at the same time. Note, do take into account that some single characters when UTF-8 encoded will result in multiple octets. The identification-tag MUST NOT contain any user information, and applications SHALL avoid generating

the identification-tag using a pattern that enables application identification.

15. IANA Considerations

15.1. New SDES item

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

This document adds the MID SDES item to the IANA "RTP SDES item types" registry as follows:

```
Value:     TBD
Abbrev.:   MID
Name:      Media Identification
Reference: RFCXXXX
```

15.2. New RTP SDES Header Extension URI

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new extension URI in the RTP SDES Compact Header Extensions sub-registry of the RTP Compact Header Extensions registry sub-registry, according to the following data:

```
Extension URI: urn:ietf:params:rtp-hdrext:sdes:mid
Description:   Media identification
Contact:       christer.holmberg@ericsson.com
Reference:     RFCXXXX
```

The SDES item does not reveal privacy information about the users. It is simply used to associate RTP-based media with the correct SDP media description (m- line) in the SDP used to negotiate the media.

The purpose of the extension is for the offerer to be able to associate received multiplexed RTP-based media before the offerer receives the associated SDP answer.

15.3. New SDP Attribute

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new SDP media-level attribute, 'bundle-only', according to the following data:

Attribute name:	bundle-only
Type of attribute:	media
Subject to charset:	No
Purpose:	Request a media description to be accepted in the answer only if kept within a BUNDLE group by the answerer.
Appropriate values:	N/A
Contact name:	Christer Holmberg
Contact e-mail:	christer.holmberg@ericsson.com
Reference:	RFCXXXX
Mux category:	NORMAL

15.4. New SDP Group Semantics

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document registers the following semantics with IANA in the "Semantics for the "group" SDP Attribute" subregistry (under the "Session Description Protocol (SDP) Parameters" registry:

Semantics	Token	Reference
Media bundling	BUNDLE	[RFCXXXX]

16. Security Considerations

The security considerations defined in [RFC3264] and [RFC5888] apply to the BUNDLE extension. Bundle does not change which information flows over the network but only changes which addresses and ports that information is flowing on and thus has very little impact on the security of the RTP sessions.

When the BUNDLE extension is used, a single set of security credentials might be used for all media streams specified by a BUNDLE group.

When the BUNDLE extension is used, the number of SSRC values within a single RTP session increases, which increases the risk of SSRC collision. [RFC4568] describes how SSRC collision may weaken SRTP and SRTCP encryption in certain situations.

17. Examples

17.1. Example: Bundle Address Selection

The example below shows:

- An offer, in which the offerer associates a unique address with each bundled "m=" line within the BUNDLE group.
- An answer, in which the answerer selects the offerer BUNDLE address, and then selects its own BUNDLE address (the answerer BUNDLE address) and associates it with each bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

17.2. Example: BUNDLE Extension Rejected

The example below shows:

- An offer, in which the offerer associates a unique address with each bundled "m=" line within the BUNDLE group.
- An answer, in which the answerer rejects the offered BUNDLE group, and associates a unique address with each "m=" line (following normal RFC 3264 procedures).

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
m=audio 20000 RTP/AVP 0
b=AS:200
a=rtcp-mux
a=rtpmap:0 PCMU/8000
m=video 30000 RTP/AVP 32
b=AS:1000
a=rtcp-mux
a=rtpmap:32 MPV/90000
```

17.3. Example: Offerer Adds A Media Description To A BUNDLE Group

The example below shows:

- A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer exchange), in which the offerer adds a new "m=" line, represented by the "zen" identification-tag, to a previously negotiated BUNDLE group, associates a unique address with the added "m=" line, and associates the previously selected offerer BUNDLE address with each of the other bundled "m=" lines within the BUNDLE group.

- An answer, in which the answerer associates the answerer BUNDLE address with each bundled "m=" line (including the newly added "m=" line) within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar zen
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
```

```
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar zen
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

17.4. Example: Offerer Moves A Media Description Out Of A BUNDLE Group

The example below shows:

- A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer transaction), in which the offerer moves a bundled "m=" line out of a BUNDLE group, associates a unique address with the moved "m=" line, and associates the offerer BUNDLE address with each other bundled "m=" line within the BUNDLE group.

- An answer, in which the answerer moves the "m=" line out of the BUNDLE group, associates a unique address with the moved "m=" line, and associates the answerer BUNDLE address with each of the remaining bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
```

```
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 50000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 60000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
```

```
a=rtpmap:66 H261/90000
```

17.5. Example: Offerer Disables A Media Description Within A BUNDLE Group

The example below shows:

- A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer transaction), in which the offerer disables a bundled "m=" line within a BUNDLE group, assigns a zero port number to the disabled "m=" line, and associates the offerer BUNDLE address with each of the other bundled "m=" lines within the BUNDLE group.

- An answer, in which the answerer moves the disabled "m=" line out of the BUNDLE group, assigns a zero port value to the disabled "m=" line, and associates the answerer BUNDLE address with each of the remaining bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

18. Acknowledgements

The usage of the SDP grouping extension for negotiating bundled media is based on a similar alternatives proposed by Harald Alvestrand and Cullen Jennings. The BUNDLE extension described in this document is based on the different alternative proposals, and text (e.g., SDP examples) have been borrowed (and, in some cases, modified) from those alternative proposals.

The SDP examples are also modified versions from the ones in the Alvestrand proposal.

Thanks to Paul Kyzivat, Martin Thomson, Flemming Andreasen, Thomas Stach, Ari Keranen, Adam Roach, Christian Groves, Roman Shpount, Suhas Nandakumar, Nils Ohlmeier, Jens Guballa, Raju Makaraju and Justin Uberti for reading the text, and providing useful feedback.

Thanks to Magnus Westerlund, Colin Perkins and Jonathan Lennox for providing help and text on the RTP/RTCP procedures.

Thanks to Spotify for providing music for the countless hours of document editing.

19. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-35

- Editorial changes on RTP streaming mapping section based on comments from Colin Perkins.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-34

- RTP streams, instead of RTP packets, are associated with m- lines.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-33

- Editorial changes based on comments from Eric Rescorla and Cullen Jennings:

- - Changes regarding usage of RTP/RTCP multiplexing attributes.

- - Additional text regarding associating RTP/RTCP packets with SDP m- lines.

- - Reference correction.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-32

- Editorial changes based on comments from Eric Rescorla and Cullen Jennings:

- - Justification for mechanism added to Introduction.

- - Clarify that the order of m- lines in the group:BUNDLE attribute does not have to be the same as the order in which the m- lines are listed in the SDP.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-31

- Editorial changes based on GitHub Pull requests by Martin Thomson:

- - https://github.com/cdh4u/draft-sdp-bundle/pull/2

- - https://github.com/cdh4u/draft-sdp-bundle/pull/1

- Editorial change based on comment from Diederick Huijbers (9th July 2016).

- Changes based on comments from Flemming Andreasen (21st June 2016):
- - Mux category for SDP bundle-only attribute added.
- - Mux category considerations editorial clarification.
- - Editorial changes.
- RTP SDES extension according to draft-ietf-avtext-sdes-hdr-ext.
- Note whether Design Considerations appendix is to be kept removed:
- - Appendix is kept within document.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-30

- Indicating in the Abstract and Introduction that the document updates RFC 3264.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-29

- Change based on WGLC comment from Colin Perkins.
- - Clarify that SSRC can be reused by another source after a delay of 5 RTCP reporting intervals.
- Change based on WGLC comment from Alissa Cooper.
- - IANA registry name fix.
- - Additional IANA registration information added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-28

- - Alignment with exclusive mux procedures.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-27

- - Yet another terminology change.
- - Mux category considerations added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-26

- - ICE considerations modified: ICE-related SDP attributes only added to the bundled m- line representing the selected BUNDLE address.

- o - Reference to draft-ietf-mmusic-ice-sip-sdp added.
- o - Reference to RFC 5245 replaced with reference to draft-ietf-ice-rfc5245bis.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-25

- o - RTP/RTCP mux procedures updated with exclusive RTP/RTCP mux considerations.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-24

- o - Reference and procedures associated with exclusive RTP/RTCP mux added

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-23

- o - RTCP-MUX mandatory for bundled RTP m- lines
- o - Editorial fixes based on comments from Flemming Andreasen

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-22

- o - Correction of Ari’s family name
- o - Editorial fixes based on comments from Thomas Stach
- o - RTP/RTCP correction based on comment from Magnus Westerlund
- o -- http://www.ietf.org/mail-archive/web/mmusic/current/msg14861.html

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-21

- o - Correct based on comment from Paul Kyzivat
- o -- ’received packets’ replaced with ’received data’

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-20

- o - Clarification based on comment from James Guballa
- o - Clarification based on comment from Flemming Andreasen

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-19

- o - DTLS Considerations section added.

- o - BUNDLE semantics added to the IANA Considerations
- o - Changes based on WGLC comments from Adam Roach
- o -- http://www.ietf.org/mail-archive/web/mmusic/current/msg14673.html

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-18

- o - Changes based on agreements at IETF#92
- o -- BAS Offer removed, based on agreement at IETF#92.
- o -- Procedures regarding usage of SDP "b=" line is replaced with a reference to to draft-ietf-mmusic-sdp-mux-attributes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-17

- o - Editorial changes based on comments from Magnus Westerlund.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-16

- o - Modification of RTP/RTCP multiplexing section, based on comments from Magnus Westerlund.
- o - Reference updates.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-15

- o - Editorial fix.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-14

- o - Editorial changes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-13

- o Changes to allow a new suggested offerer BUNDLE address to be assigned to each bundled m- line.
- o Changes based on WGLC comments from Paul Kyzivat
- o - Editorial fixes

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-12

- o Usage of SDP ’extmap’ attribute added

- SDP ’bundle-only’ attribute scoped with "m=" lines with a zero port value
- Changes based on WGLC comments from Thomas Stach
- - ICE candidates not assigned to bundle-only m- lines with a zero port value
- - Editorial changes
- Changes based on WGLC comments from Colin Perkins
- - Editorial changes:
- -- "RTP SDES item" -> "RTCP SDES item"
- -- "RTP MID SDES item" -> "RTCP MID SDES item"
- - Changes in section 10.1.1:
- -- "SHOULD NOT" -> "MUST NOT"
- -- Additional text added to the Note
- - Change to section 13.2:
- -- Clarify that mid value is not zero terminated
- - Change to section 13.3:
- -- Clarify that mid value is not zero terminated
- -- Clarify padding
- Changes based on WGLC comments from Paul Kyzivat
- - Editorial changes:
- Changes based on WGLC comments from Jonathan Lennox
- - Editorial changes:
- - Defintion of SDP bundle-only attribute alligned with structure in 4566bis draft

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-11

- Editorial corrections based on comments from Harald Alvestrand.

- Editorial corrections based on comments from Cullen Jennings.
- Reference update (RFC 7160).
- Clarification about RTCP packet sending when RTP/RTCP multiplexing is not used (http://www.ietf.org/mail-archive/web/mmusic/current/msg13765.html).
- Additional text added to the Security Considerations.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-10

- SDP bundle-only attribute added to IANA Considerations.
- SDES item and RTP header extension added to Abstract and Introduction.
- Modification to text updating section 8.2 of RFC 3264.
- Reference corrections.
- Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-09

- Terminology change: "bundle-only attribute assigned to m= line" to "bundle-only attribute associated with m= line".
- Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-08

- Editorial corrections.
- - "of"->"if" (8.3.2.5).
- - "optional"->"OPTIONAL" (9.1).
- - Syntax/ABNF for ’bundle-only’ attribute added.
- - SDP Offer/Answer sections merged.
- - ’Request new offerer BUNDLE address’ section added

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-07

- OPEN ISSUE regarding Receiver-ID closed.

- o - RTP MID SDES Item.
- o - RTP MID Header Extension.
- o OPEN ISSUE regarding insertion of SDP 'rtcp' attribute in answers closed.
- o - Indicating that, when rtcp-mux is used, the answerer MUST NOT include an 'rtcp' attribute in the answer, based on the procedures in section 5.1.3 of RFC 5761.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-06

- o Draft title changed.
- o Added "SDP" to section names containing "Offer" or "Answer".
- o Editorial fixes based on comments from Paul Kyzivat (http://www.ietf.org/mail-archive/web/mmusic/current/msg13314.html).
- o Editorial fixed based on comments from Colin Perkins (http://www.ietf.org/mail-archive/web/mmusic/current/msg13318.html).
- o - Removed text about extending BUNDLE to allow multiple RTP sessions within a BUNDLE group.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-05

- o Major re-structure of SDP Offer/Answer sections, to align with RFC 3264 structure.
- o Additional definitions added.
- o - Shared address.
- o - Bundled "m=" line.
- o - Bundle-only "m=" line.
- o - Offerer suggested BUNDLE mid.
- o - Answerer selected BUNDLE mid.
- o Q6 Closed (IETF#88): An Offerer MUST NOT assign a shared address to multiple "m=" lines until it has received an SDP Answer indicating support of the BUNDLE extension.

- Q8 Closed (IETF#88): An Offerer can, before it knows whether the Answerer supports the BUNDLE extension, assign a zero port value to a 'bundle-only' "m=" line.
- SDP 'bundle-only' attribute section added.
- Connection data nettype/addrtype restrictions added.
- RFC 3264 update section added.
- Indicating that a specific payload type value can be used in multiple "m=" lines, if the value represents the same codec configuration in each "m=" line.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-04

- Updated Offerer procedures (http://www.ietf.org/mail-archive/web/mmusic/current/msg12293.html).
- Updated Answerer procedures (http://www.ietf.org/mail-archive/web/mmusic/current/msg12333.html).
- Usage of SDP 'bundle-only' attribute added.
- Reference to Trickle ICE document added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-02

- Mechanism modified, to be based on usage of SDP Offers with both different and identical port number values, depending on whether it is known if the remote endpoint supports the extension.
- Cullen Jennings added as co-author.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-01

- No changes. New version due to expiration.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-00

- No changes. New version due to expiration.

Changes from draft-holmberg-mmusic-sdp-multiplex-negotiation-00

- Draft name changed.
- Harald Alvestrand added as co-author.

- "Multiplex" terminology changed to "bundle".
- Added text about single versus multiple RTP Sessions.
- Added reference to RFC 3550.

20. References

20.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <http://www.rfc-editor.org/info/rfc3605>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, DOI 10.17487/RFC4961, July 2007, <http://www.rfc-editor.org/info/rfc4961>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <http://www.rfc-editor.org/info/rfc5761>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <http://www.rfc-editor.org/info/rfc5764>.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <http://www.rfc-editor.org/info/rfc5888>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <http://www.rfc-editor.org/info/rfc7656>.

[RFC7941] Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items", RFC 7941, DOI 10.17487/RFC7941, August 2016, <http://www.rfc-editor.org/info/rfc7941>.

[I-D.ietf-ice-rfc5245bis] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-04 (work in progress), June 2016.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-14 (work in progress), September 2016.

[I-D.ietf-mmusic-mux-exclusive] Holmberg, C., "Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP", draft-ietf-mmusic-mux-exclusive-10 (work in progress), August 2016.

[I-D.ietf-mmusic-ice-sip-sdp]
Petit-Huguenin, M., Keranen, A., and S. Nandakumar, "Using Interactive Connectivity Establishment (ICE) with Session Description Protocol (SDP) offer/answer and Session Initiation Protocol (SIP)", draft-ietf-mmusic-ice-sip-sdp-10 (work in progress), July 2016.

20.2. Informative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <http://www.rfc-editor.org/info/rfc4568>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <http://www.rfc-editor.org/info/rfc5576>.

[RFC7160] Petit-Huguenin, M. and G. Zorn, Ed., "Support for Multiple Clock Rates in an RTP Session", RFC 7160, DOI 10.17487/RFC7160, April 2014, <http://www.rfc-editor.org/info/rfc7160>.

[I-D.ietf-mmusic-trickle-ice]
Ivov, E., Rescorla, E., and J. Uberti, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", draft-ietf-mmusic-trickle-ice-02 (work in progress), January 2015.

Appendix A. Design Considerations

One of the main issues regarding the BUNDLE grouping extensions has been whether, in SDP Offers and SDP Answers, the same port value should be inserted in "m=" lines associated with a BUNDLE group, as the purpose of the extension is to negotiate the usage of a single address:port combination for media specified by the "m=" lines. Issues with both approaches, discussed in the Appendix have been raised. The outcome was to specify a mechanism which uses SDP Offers with both different and identical port values.

Below are the primary issues that have been considered when defining the "BUNDLE" grouping extension:

- 1) Interoperability with existing UAs.
- 2) Interoperability with intermediary B2BUA- and proxy entities.
- 3) Time to gather, and the number of, ICE candidates.
- 4) Different error scenarios, and when they occur.
- 5) SDP Offer/Answer impacts, including usage of port number value zero.

A.1. UA Interoperability

Consider the following SDP Offer/Answer exchange, where Alice sends an SDP Offer to Bob:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

SDP Answer

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
m=audio 20000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 20002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

RFC 4961 specifies a way of doing symmetric RTP but that is an a later invention to RTP and Bob can not assume that Alice supports RFC 4961. This means that Alice may be sending RTP from a different port than 10000 or 10002 - some implementation simply send the RTP from an ephemeral port. When Bob's endpoint receives an RTP packet, the only way that Bob knows if it should be passed to the video or audio codec is by looking at the port it was received on. This lead some SDP implementations to use the fact that each "m=" line had a different port number to use that port number as an index to find the correct m line in the SDP. As a result, some implementations that do support symmetric RTP and ICE still use a SDP data structure where SDP with "m=" lines with the same port such as:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10000 RTP/AVP 98
a=rtpmap:98 H261/90000
```

will result in the second "m=" line being considered an SDP error because it has the same port as the first line.

A.2. Usage of port number value zero

In an SDP Offer or SDP Answer, the media specified by an "m=" line can be disabled/rejected by setting the port number value to zero. This is different from e.g., using the SDP direction attributes, where RTCP traffic will continue even if the SDP "inactive" attribute is indicated for the associated "m=" line.

If each "m=" line associated with a BUNDLE group would contain different port values, and one of those port values would be used for a BUNDLE address associated with the BUNDLE group, problems would occur if an endpoint wants to disable/reject the "m=" line associated with that port, by setting the port value to zero. After that, no "m=" line would contain the port value which is used for the BUNDLE address. In addition, it is unclear what would happen to the ICE candidates associated with the "m=" line, as they are also used for the BUNDLE address.

A.3. B2BUA And Proxy Interoperability

Some back to back user agents may be configured in a mode where if the incoming call leg contains an SDP attribute the B2BUA does not understand, the B2BUA still generates that SDP attribute in the Offer for the outgoing call leg. Consider a B2BUA that did not understand the SDP "rtcp" attribute, defined in RFC 3605, yet acted this way. Further assume that the B2BUA was configured to tear down any call where it did not see any RTCP for 5 minutes. In this case, if the B2BUA received an Offer like:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtcp:53020
```

It would be looking for RTCP on port 49172 but would not see any because the RTCP would be on port 53020 and after five minutes, it would tear down the call. Similarly, a B2BUA that did not understand BUNDLE yet put BUNDLE in it’s offer may be looking for media on the wrong port and tear down the call. It is worth noting that a B2BUA that generated an Offer with capabilities it does not understand is not compliant with the specifications.

A.3.1. Traffic Policing

Sometimes intermediaries do not act as B2BUA, in the sense that they don’t modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g., IP address and port) in order to control traffic gating functions, and to set traffic policing rules. There might be rules which will trigger a session to be terminated in case media is not sent or received on the ports retrieved from the SDP. This typically occurs once the session is already established and ongoing.

A.3.2. Bandwidth Allocation

Sometimes intermediaries do not act as B2BUA, in the sense that they don’t modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g., codecs and media types)

in order to control bandwidth allocation functions. The bandwidth allocation is done per "m=" line, which means that it might not be enough if media specified by all "m=" lines try to use that bandwidth. That may either simply lead to bad user experience, or to termination of the call.

A.4. Candidate Gathering

When using ICE, a candidate needs to be gathered for each port. This takes approximately 20 ms extra for each extra "m=" line due to the NAT pacing requirements. All of this gather can be overlapped with other things while for exampe a web-page is loading to minimize the impact. If the client only wants to generate TURN or STUN ICE candidates for one of the "m=" lines and then use trickle ICE [I-D.ietf-mmusic-trickle-ice] to get the non host ICE candidates for the rest of the "m=" lines, it MAY do that and will not need any additional gathering time.

Some people have suggested a TURN extension to get a bunch of TURN allocations at once. This would only provide a single STUN result so in cases where the other end did not support BUNDLE, may cause more use of the TURN server but would be quick in the cases where both sides supported BUNDLE and would fall back to a successful call in the other cases.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Harald Tveit Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no

Cullen Jennings
Cisco
400 3rd Avenue SW, Suite 350
Calgary, AB T2P 4H2
Canada

Email: fluffy@iii.ca

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

M. Thomson
E. Rescorla
Mozilla

March 13, 2017

Unknown Key Share Attacks on uses of Transport Layer Security with the Session Description Protocol (SDP)
draft-thomson-avtcore-sdp-uks-01

Abstract

Unknown key-share attacks on the use of Datagram Transport Layer Security for the Secure Real-Time Transport Protocol (DTLS-SRTP) and its use with Web Real-Time Communications (WebRTC) identity assertions are described. Simple mitigation techniques are defined.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Table of Contents

1. Introduction

The use of Transport Layer Security (TLS) [RFC5246] with the Session Description Protocol (SDP) [RFC4566] is defined in [RFC4572]. Further use with Datagram Transport Layer Security (DTLS) [RFC6347] and the Secure Real-time Transport Protocol (SRTP) [RFC3711] is defined as DTLS-SRTP [RFC5763].

In these specifications, key agreement is performed using the TLS or DTLS handshaking protocol, with authentication being tied back to the session description (or SDP) through the use of certificate fingerprints. Communication peers check that a hash, or fingerprint, provided in the SDP matches the certificate that is used in the TLS (or DTLS) handshake. This is defined in [RFC4572].

The design of DTLS-SRTP relies on the integrity of the signaling channel. Certificate fingerprints are assumed to be provided by the communicating peers and carried by the signaling channel without being subject to modification. However, this design is vulnerable to an unknown key-share (UKS) attack where a misbehaving endpoint is able to advertise a key that it does not control. This leads to the creation of sessions where peers are confused about the identify of the participants.

An extension to TLS is defined that can be used to mitigate this attack.

A similar attack is possible with sessions that use WebRTC identity (see Section 5.6 of [I-D.ietf-rtcweb-security-arch]). This issue and a mitigation for it is discussed in more detail in Section 4.

2. Unknown Key-Share Attack

In an unknown key-share attack [UKS], a malicious participant in a protocol claims to control a key that is in reality controlled by some other actor. This arises when the identity associated with a key is not properly bound to the key.

In DTLS-SRTP, an endpoint is able to acquire the certificate fingerprint another entity. By advertising that fingerprint in place of one of its own, the malicious endpoint can cause its peer to communicate with a different peer, even though it believes that it is communicating with the malicious endpoint.

When the identity of communicating peers is established by higher-layer signaling constructs, such as those in SIP [RFC4474] or WebRTC [I-D.ietf-rtcweb-security-arch], this allows an attacker to bind their own identity to a session with any other entity.

By substituting the the fingerprint of one peer for its own, an attacker is able to cause a session to be established where one endpoint has an incorrect value for the identity of its peer. However, the peer does not suffer any such confusion, resulting in each peer involved in the session having a different view of the nature of the session.

This attack applies to any communications established based on the SDP "fingerprint" attribute [RFC4572].

2.1. Attack Overview

This vulnerability can be used by an attacker to create a call where there is confusion about the communicating endpoints.

A SIP endpoint or WebRTC endpoint that is configured to reuse a certificate can be attacked if it is willing to conduct two concurrent calls, one of which is with an attacker. The attacker can arrange for the victim to incorrectly believe that is calling the attacker when it is in fact calling a second party. The second party correctly believes that it is talking to the victim.

In a related attack, a single call using WebRTC identity can be attacked so that it produces the same outcome. This attack does not require a concurrent call.

2.2. Limits on Attack Feasibility

The use of TLS with SDP depends on the integrity of session signaling. Assuming signaling integrity limits the capabilities of an attacker in several ways. In particular:

1. An attacker can only modify the parts of the session signaling for a session that they are part of, which is limited to their own offers and answers.

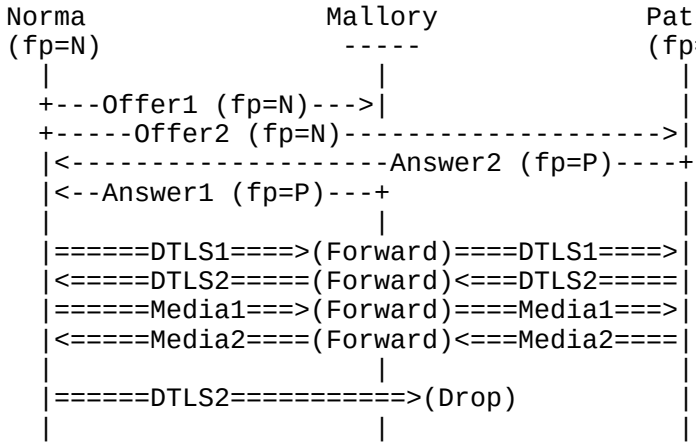
2. No entity will complete communications with a peer unless they are willing to participate in a session with that peer.

The combination of these two constraints make the spectrum of possible attacks quite limited. An attacker is only able to switch its own certificate fingerprint for a valid certificate that is acceptable to its peer. Attacks therefore rely on joining two separate sessions into a single session.

The second condition is not necessary with WebRTC identity if the victim has or is configured with a target peer identity (this is defined in [WEBRTC]). Furthermore, any identity displayed by a browser could be different to the identity used by the application, since the attack affects the browser’s understanding of the peer’s identity.

2.3. Example

In this example, two outgoing sessions are created by the same endpoint. One of those sessions is initiated with the attacker, another session is created toward another honest endpoint. The attacker convinces the endpoint that their session has completed, and that the session with the other endpoint has succeeded.



In this case, Norma is willing to conduct two concurrent sessions. The first session is established with Mallory, who falsely uses Patsy's certificate fingerprint. A second session is initiated between Norma and Patsy. Signaling for both sessions is permitted to complete.

Once complete, the session that is ostensibly between Mallory and Norma is completed by forwarding packets between Norma and Patsy. This requires that Mallory is able to intercept DTLS and media packets from Patsy so that they can be forwarded to Norma at the transport addresses that Norma associates with the first session.

The second session - between Norma and Patsy - is permitted to continue to the point where Patsy believes that it has succeeded. This ensures that Patsy believes that she is communicating with Norma. In the end, Norma believes that she is communicating with Mallory, when she is actually communicating with Patsy.

Though Patsy needs to believe that the second session is successful, Mallory has no real interest in seeing that session complete. Mallory only needs to ensure that Patsy does not abandon the session prematurely. For this reason, it might be necessary to permit the answer from Patsy to reach Norma to allow Patsy to receive a call completion signal, such as a SIP ACK. Once the second session completes, Mallory causes any DTLS packets sent by Norma to Patsy to be dropped.

For the attacked session to be sustained beyond the point that Norma detects errors in the second session, Mallory also needs to block any signaling that Norma might send to Patsy asking for the call to be abandoned. Otherwise, Patsy might receive a notice that the call is failed and thereby abort the call.

This attack creates an asymmetry in the beliefs about the identity of peers. However, this attack is only possible if the victim (Norma) is willing to conduct two sessions concurrently, and if the same certificate - and therefore SDP "fingerprint" attribute value - is used in both sessions.

2.4. Interactions with Key Continuity

Systems that use key continuity might be able to detect an unknown key-share attack if a session with the actual peer (i.e., Patsy in the example) was established in the past. Whether this is possible depends on how key continuity is implemented.

Implementations that maintain a single database of identities with an index on peer keys could discover that the identity saved for the peer key does not match the claimed identity. Such an implementation could notice the disparity between the actual keys (Patsy) and the expected keys (Mallory).

In comparison, implementations that first match based on peer identity could treat an unknown key-share attack as though their peer had used a newly-configured device. The apparent addition of a new device could generate user-visible notices (e.g., "Mallory appears to have a new device"). However, such an event is not always considered alarming; some implementations might silently save a new key.

3. Adding a Session Identifier

An attack on DTLS-SRTP is possible because the identity of peers involved is not established prior to establishing the call. Endpoints use certificate fingerprints as a proxy for authentication, but as long as fingerprints are used in multiple calls, they are vulnerable to attacks of the sort described.

The solution to this problem is to assign a new identifier to communicating peers. Each endpoint assigns their peer a unique identifier during call signaling. The peer echoes that identifier in the TLS handshake, binding that identity into the session. Including this new identity in the TLS handshake means that it will be covered by the TLS Finished message, which is necessary to authenticate it (see [SIGMA]). Validating that peers use the correct identifier then means that the session is established between the correct two endpoints.

This solution relies on the unique identifier given to DTLS sessions using the SDP "dtls-id" attribute [I-D.ietf-mmusic-dtls-sdp]. This field is already required to be unique. Thus, no two offers or answers from the same client will have the same value.

A new "sdp_dtls_id" extension is added to the TLS or DTLS handshake for connections that are established as part of the same call or real-time session. This carries the value of the "dtls-id" attribute and provides integrity protection for its exchange as part of the TLS or DTLS handshake.

3.1. The sdp_dtls_id TLS Extension

The "sdp_dtls_id" TLS extension carries the unique identifier that an endpoint selects. The value includes the "sess-id" field from the SDP that the endpoint generated when negotiating the session.

The "extension_data" for the "sdp_dtls_id" extension contains a SdpDtlsId struct, described below using the syntax defined in [RFC5246]:

```
struct {
   opaque dtls_id<1..255>;
} SdpDtlsId;
```

The "dtls_id" field of the extension includes the value of the "dtls-id" SDP attribute as defined in [I-D.ietf-mmusic-dtls-sdp] (that is, the "dtls-id-value" ABNF production). The value of the "dtls-id" attribute is encoded using ASCII [RFC0020].

Where RTP and RTCP [RFC3550] are not multiplexed, it is possible that the two separate DTLS connections carrying RTP and RTCP can be switched. This is considered benign since these protocols are often distinguishable. RTP/RTCP multiplexing is advised to address this problem.

The "sdp_dtls_id" extension is included in a ClientHello and either ServerHello (for TLS and DTLS versions less than 1.3) or EncryptedExtensions (for TLS 1.3). In TLS 1.3, the extension MUST NOT be included in a ServerHello.

Endpoints MUST check that the "dtls_id" parameter in the extension that they receive includes the "dtls-id" attribute value that they received in their peer’s session description. Comparison can be performed with either the decoded ASCII string or the encoded octets. An endpoint that receives a "sdp_dtls_id" extension that is not identical to the value that it expects MUST abort the connection with a fatal "handshake_failure" alert.

An endpoint that is communicating with a peer that does not support this extension will receive a ClientHello, ServerHello or EncryptedExtensions that does not include this extension. An endpoint MAY choose to continue a session without this extension in

order to interoperate with peers that do not implement this specification.

In TLS 1.3, the "sdp_dtls_id" extension MUST be sent in the EncryptedExtensions message.

4. WebRTC Identity Binding

The identity assertion used for WebRTC [I-D.ietf-rtcweb-security-arch] is bound only to the certificate fingerprint of an endpoint and can therefore be copied by an attacker along with any SDP "fingerprint" attributes.

The problem is compounded by the fact that an identity provider is not required to verify that the entity requesting an identity assertion controls the keys. Nor is it currently able to perform this validation. Note however that this verification is not a necessary condition for a secure protocol, as established in [SIGMA].

A simple solution to this problem is suggested by [SIGMA]. The identity of endpoints is included under a message authentication code (MAC) during the cryptographic handshake. Endpoints are then expected to validate that their peer has provided an identity that matches their expectations.

In TLS, the Finished message provides a MAC over the entire handshake, so that including the identity in a TLS extension is sufficient to implement this solution. Rather than include a complete identity assertion, a collision-resistant hash of the identity assertion is included in a TLS extension. Peers then need only validate that the extension contains a hash of the identity assertion they received in signaling in addition to validating the identity assertion.

Endpoints MAY use the "sdp_dtls_id" extension in addition to this so that two calls between the same parties can’t be altered by an attacker.

4.1. The webrtc_id_hash TLS Extension

The "webrtc_id_hash" TLS extension carries a hash of the identity assertion that communicating peers have exchanged.

The "extension_data" for the "webrtc_id_hash" extension contains a WebrtcIdentityHash struct, described below using the syntax defined in [RFC5246]:

```
struct {
   opaque assertion_hash[32];
} WebrtcIdentityHash;
```

A WebRTC identity assertion is provided as a JSON [RFC7159] object that is encoded into a JSON text. The resulting string is then encoded using UTF-8 [RFC3629]. The content of the "webrtc_id_hash" extension are produced by hashing the resulting octets with SHA-256 [FIPS180-2]. This produces the 32 octets of the assertion_hash parameter, which is the sole contents of the extension.

The SDP "identity" attribute includes the base64 [RFC4648] encoding of the same octets that were input to the hash. The "webrtc_id_hash" extension is validated by performing base64 decoding on the value of the SDP "identity" attribute, hashing the resulting octets using SHA-256, and comparing the results with the content of the extension.

Identity assertions might be provided by only one peer. An endpoint that does not produce an identity assertion MUST generate an empty "webrtc_id_hash" extension in its ClientHello. This allows its peer to include a hash of its identity assertion. An endpoint without an identity assertion MUST omit the "webrtc_id_hash" extension from its ServerHello or EncryptedExtensions message.

A peer that receives a "webrtc_id_hash" extension that is not equal to the value of the identity assertion from its peer MUST immediately fail the TLS handshake with an error. This includes cases where the "a=identity" attribute is not present in the SDP.

A peer that receives an identity assertion, but does not receive a "webrtc_id_hash" extension MAY choose to fail the connection, though it is expected that implementations that were written prior to the existence of this document will not support these extensions for some time.

In TLS 1.3, the "webrtc_id_hash" extension MUST be sent in the EncryptedExtensions message.

5. Session Concatenation

Use of session identifiers does not prevent an attacker from establishing two concurrent sessions with different peers and forwarding signaling from those peers to each other. Concatenating two signaling sessions creates a situation where both peers believe that they are talking to the attacker when they are talking to each other.

Session concatention is possible at higher layers: an attacker can establish two independent sessions and simply forward any data it receives from one into the other. This kind of attack is prevented by systems that enable peer authentication such as WebRTC identity [I-D.ietf-rtcweb-security-arch] or SIP identity [RFC4474].

In the absence of any higher-level concept of peer identity, the use of session identifiers does not prevent session concatenation. The value to an attacker is limited unless information from the TLS connection is extracted and used with the signaling. For instance, a key exporter [RFC5705] might be used to create a shared secret or unique identifier that is used in a secondary protocol.

If a secondary protocol uses the signaling channel with the assumption that the signaling and TLS peers are the same then that protocol is vulnerable to attack. The identity of the peer at the TLS layer is not guranteed to be the same as the identity of the signaling peer.

It is important to note that multiple connections can be created within the same signaling session. An attacker can concatenate only part of a session, choosing to terminate some connections (and optionally forward data) while arranging to have peers interact directly for other connections. It is even possible to have different peers interact for each connection. This means that the actual identity of the peer for one connection might differ from the peer on another connection.

Information extracted from a TLS connection therefore MUST NOT be used in a secondary protocol outside of that connection if that protocol relies on the signaling protocol having the same peers. Similarly, data from one TLS connection MUST NOT be used in other TLS connections even if they are established as a result of the same signaling session.

6. Security Considerations

This entire document contains security considerations.

7. IANA Considerations

This document registers two extensions in the TLS "ExtensionType Values" registry established in [RFC5246]:

- The "sdp_dtls_id" extension has been assigned a code point of TBD; it is recommended and is marked as "Encrypted" in TLS 1.3.

- The "webrtc_id_hash" extension has been assigned a code point of TBD; it is recommended and is marked as "Encrypted" in TLS 1.3.

8. References

8.1. Normative References

[FIPS180-2] Department of Commerce, National., "NIST FIPS 180-2, Secure Hash Standard", August 2002.

[I-D.ietf-mmusic-dtls-sdp] Holmberg, C. and R. Shpount, "Using the SDP Offer/Answer Mechanism for DTLS", draft-ietf-mmusic-dtls-sdp-21 (work in progress), March 2017.

[I-D.ietf-rtcweb-security-arch] Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-12 (work in progress), June 2016.

[RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <http://www.rfc-editor.org/info/rfc20>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <http://www.rfc-editor.org/info/rfc4572>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <http://www.rfc-editor.org/info/rfc5763>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

8.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <http://www.rfc-editor.org/info/rfc4474>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <http://www.rfc-editor.org/info/rfc4648>.

[RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <http://www.rfc-editor.org/info/rfc5705>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[SIGMA] Krawczyk, H., "SIGMA: The ’SIGn-and-MAc’approach to authenticated Diffie-Hellman and its use in the IKE protocols", Annual International Cryptology Conference, Springer, pp. 400-425 , 2003.

[UKS] Blake-Wilson, S. and A. Menezes, "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", Lecture Notes in Computer Science 1560, Springer, pp. 154-170 , 1999.

[WEBRTC] Bergkvist, A., Burnett, D., Narayanan, A., Jennings, C., and B. Aboba, "WebRTC 1.0: Real-time Communication Between Browsers", W3C WD-webrtc-30160531 , May 2016.

Appendix A. Acknowledgements

This problem would not have been discovered if it weren't for discussions with Sam Scott, Hugo Krawczyk, and Richard Barnes. A solution similar to the one presented here was first proposed by Karthik Bhargavan who provided valuable input on this document. Thyla van der Merwe assisted with a formal model of the solution. Adam Roach and Paul E. Jones provided useful input.

Authors' Addresses

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com

Eric Rescorla
Mozilla

Email: ekr@rftm.com