[MS-SRTP]:

Secure Real-time Transport Protocol (SRTP) Profile

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial version
4/25/2008	0.2	Minor	Revised and edited technical content
6/27/2008	1.0	Major	Revised and edited technical content
8/15/2008	1.01	Minor	Revised and edited technical content
12/12/2008	2.0	Major	Revised and edited technical content
2/13/2009	2.01	Minor	Revised and edited technical content
3/13/2009	2.02	Minor	Revised and edited technical content
7/13/2009	2.03	Major	Revised and edited the technical content
8/28/2009	2.04	Editorial	Revised and edited the technical content
11/6/2009	2.05	Editorial	Revised and edited the technical content
2/19/2010	2.06	Editorial	Revised and edited the technical content
3/31/2010	2.07	Major	Updated and revised the technical content
4/30/2010	2.08	Editorial	Revised and edited the technical content
6/7/2010	2.09	Editorial	Revised and edited the technical content
6/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.10	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	4.0	Major	Significantly changed the technical content.
4/11/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.1	Minor	Clarified the meaning of the technical content.
2/11/2013	4.1	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/30/2013	4.1	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	4.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	5.0	Major	Significantly changed the technical content.
6/30/2015	5.0	None	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	5.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2015	6.0	Major	Significantly changed the technical content.
7/15/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	7.0	Major	Significantly changed the technical content.
4/27/2018	8.0	Major	Significantly changed the technical content.
7/24/2018	9.0	Major	Significantly changed the technical content.
8/28/2018	10.0	Major	Significantly changed the technical content.
8/18/2020	10.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

The Secure Real-time Transport Protocol (SRTP) Profile specifies a subset of the Secure Real-time Transport Protocol (SRTP). This protocol provides the same functional capabilities as SRTP, which include providing confidentiality, message authentication, and replay protection to the RTP traffic and to the control traffic for RTP.

This protocol is a strict subset of SRTP and differs from it in two key aspects:

- The first key difference is that this protocol supports a strict subset of the SRTP default cryptographic transform algorithms and requires that some parameters of the encryption and authentication algorithms described in [RFC3711] be of specific values. These requirements are specified in section 3.
- The second key difference is that there is a set of "MAY, SHOULD, MUST, SHOULD NOT, MUST NOT" protocol behaviors that differ between this protocol and [RFC3711]. Section 3 enumerates these behavioral differences.

Unless explicitly noted in this document, this protocol follows standard SRTP.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Advanced Encryption Standard (AES): A block cipher that supersedes the Data Encryption Standard (DES). AES can be used to protect electronic data. The AES algorithm can be used to encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. AES is used in symmetric-key cryptography, meaning that the same key is used for the encryption and decryption operations. It is also a block cipher, meaning that it operates on fixed-size blocks of plaintext and ciphertext, and requires the size of the plaintext as well as the ciphertext to be an exact multiple of this block size. AES is also known as the Rijndael symmetric encryption algorithm [FIPS197].

AES Counter Mode: A type of counter-mode encryption that generates encryption key streams by using **Advanced Encryption Standard (AES)** cipher and successive integers.

cryptographic context: A set of cryptographic state information that is maintained in a Secure Real-Time Transport Protocol (SRTP) stream.

dual-tone multi-frequency (DTMF): In telephony systems, a signaling system in which each digit is associated with two specific frequencies. This system typically is associated with touch-tone keypads for telephones.

endpoint: A device that is connected to a computer network.

Hash-based Message Authentication Code (HMAC): A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function (for example, MD5 and SHA-1) in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

master key: A key that provides information for packet encryption and authentication in **Secure Real-Time Transport Protocol (SRTP)** and Scale Secure Real-Time Transport Protocol (SSRTP) transactions.

NULL cipher: A cipher that does not modify a **Real-Time Transport Protocol (RTP)** payload and is defined in the **Secure Real-Time Transport Protocol (SRTP)** protocol. It is used when **RTP packet** encryption is not necessary, but packet authentication is necessary.

Real-Time Transport Control Protocol (RTCP): A network transport protocol that enables monitoring of Real-Time Transport Protocol (RTP) data delivery and provides minimal control and identification functionality, as described in [RFC3550].

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

RTCP packet: A control packet consisting of a fixed header part similar to that of **RTP packets**, followed by structured elements that vary depending upon the RTCP packet type. Typically, multiple RTCP packets are sent together as a compound RTCP packet in a single packet of the underlying protocol; this is enabled by the length field in the fixed header of each RTCP packet. See [RFC3550] section 3.

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources, and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets can be contained if permitted by the encapsulation method. See [RFC3550] section 3.

RTP profile: A collection that contains payload type codes and mappings to payload formats, such as media encodings. It can also define extensions or modifications to the **Real-Time Transport Protocol (RTP)** that are specific to a particular class of applications. Typically, an application operates under only one profile.

salt: An additional random quantity, specified as input to an encryption function that is used to increase the strength of the encryption.

Secure Real-Time Transport Protocol (SRTP): A profile of **Real-Time Transport Protocol (RTP)** that provides encryption, message authentication, and replay protection to the RTP data, as described in [RFC3711].

Session Description Protocol (SDP): A protocol that is used for session announcement, session invitation, and other forms of multimedia session initiation. For more information see [MS-SDP] and [RFC3264].

session key: A symmetric key that is derived from a master key and is used to encrypt or authenticate a specific media stream by using the **Secure Real-Time Transport Protocol (SRTP)** and Scale Secure Real-Time Transport Protocol (SSRTP).

SHA-256: An algorithm that generates a 256-bit hash value from an arbitrary amount of input data.

synchronization source (SSRC): The source of a stream of **RTP packets**, identified by a 32-bit numeric SSRC identifier carried in the RTP header so as not to be dependent upon the network address. All packets from a synchronization source form part of the same timing and sequence number space, so a receiver groups packets by synchronization source for playback. Examples of synchronization sources include the sender of a stream of packets derived from a signal source such as a microphone or a camera, or an RTP mixer. A synchronization source may change its data format (for example, audio encoding) over time. The SSRC identifier is a randomly chosen value meant to be globally unique within a particular RTP session. A participant need not use the same SSRC identifier for all the RTP sessions in a multimedia session; the binding of the SSRC identifiers is provided through RTCP. If a participant generates multiple streams in one RTP session, for example from separate video cameras, each MUST be identified as a different SSRC. See [RFC3550] section 3.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

[NIST.FIPS.180-4] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", August 2015, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[RFC2104] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997, http://www.ietf.org/rfc/rfc2104.txt

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

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

1.2.2 Informative References

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

[MS-SDPEXT] Microsoft Corporation, "Session Description Protocol (SDP) Version 2.0 Extensions".

1.3 Overview

This protocol provides the same functionality as the **Secure Real-Time Transport Protocol (SRTP)** by providing confidentiality, message authentication, and replay protection to **Real-Time Transport Protocol (RTP)** traffic and to the control traffic for RTP, the **Real-Time Transport Control Protocol (RTCP)**.

This protocol is a strict subset of SRTP and differs from it in the following two key aspects. In all other cases, this protocol follows standard SRTP.

- The first key difference is that this protocol supports a subset of the SRTP default cryptographic transform algorithms, and it requires certain encryption and authentication algorithm parameters to be fixed values. For example, the **NULL cipher** transform is not supported.
- The second key difference is that there is a set of "MAY, SHOULD, MUST, SHOULD NOT, MUST NOT" protocol behaviors where this protocol differs in behavior from [RFC3711]. Section 3 enumerates these behavioral differences.

1.4 Relationship to Other Protocols

This protocol relies on **Session Description Protocol (SDP)** to exchange **master keys** and key parameters. Refer to [MS-SDPEXT] for SDP information pertinent to this protocol.

This protocol works with other **RTP profiles**; for example, **dual-tone multi-frequency (DTMF)**, as described in [MS-DTMF]. This protocol treats all other RTP profile outputs the same as audio or video data. It encrypts and authenticates after processing is performed on the sending side and authenticates and decrypts before passing **RTP packets** and **RTCP packets** on the receiving side.

The Secure Real-time Transport Control Protocol (SRTCP) is considered a sub-protocol to **SRTP**, and they are described together in [RFC3711]. The proprietary implementation of SRTCP is specified in this document in a similar way.

1.5 Prerequisites/Preconditions

This protocol has the following prerequisites:

- This protocol requires that encryption and authentication algorithms are negotiated using **SDP**, as described in [MS-SDPEXT] section 3.1.5.8.
- This protocol requires that the **master keys** are exchanged using SDP, as described in [MS-SDPEXT] section 3.1.5.8, and the keys are configured properly.
- This protocol only provides message confidentiality, authentication, and replay protection for **RTP packets** and **RTCP packets**.

1.6 Applicability Statement

This protocol is used where users require secure **RTP** traffic. This protocol is required to be used with the **SDP** extension described in [MS-SDPEXT] section 3.1.5.8 to set up the shared **master key** securely.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol transforms **RTP**/**RTCP packets** only. Refer to [MS-RTP] section 2.1 for transports that the RTP protocol uses.

2.2 Message Syntax

This protocol uses the message syntax specified in [RFC3711].

- For the **SRTP** message syntax, see [RFC3711] section 3.1.
- For the SRTCP message syntax, see [RFC3711] section 3.4.

3 Protocol Details

3.1 Endpoint Details

This protocol can be used to secure any **RTP** traffic. All behavior described here applies to both protocol client and server roles.

The following sections specify the differences between this protocol and **SRTP**, as specified in [RFC3711].

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol requires that each **endpoint** in an **SRTP** session maintains **cryptographic contexts**. A cryptographic context has two categories of parameters:

- Transform independent parameters
- Transform dependent parameters

3.1.1.1 Transform Independent Parameters

Transform independent parameters are parameters independent of what encryption and authentication algorithms are used. For example, regardless of which authentication algorithm is used, the replay checklist size is fixed to 64 entries in this protocol. For details, see [RFC3711] section 3.2.1.

This protocol does not introduce new states, but does require some states to be specific values. For details, see section 3.1.3.2.

3.1.1.2 Transform Dependent Parameters

Transform dependent parameters are parameters for specific encryption or authentication algorithms. This protocol implements the default cryptographic transform specified in [RFC3711] section 4, with exceptions specified in section 3.1.3.3. No new states are introduced.

3.1.2 Timers

None.

3.1.3 Initialization

3.1.3.1 Cryptographic Contexts

SRTP requires that each **endpoint** in an SRTP session maintain **cryptographic contexts**. For more information, see [RFC3711] section 3.2.3. This protocol maintains cryptographic contexts differently from SRTP [RFC3711].

This protocol maintains two cryptographic contexts per SRTP session:

- One for all media streams on the send direction.

- One for all media streams on the receive direction.

This protocol supports multiple media streams sharing the same SRTP session. Each media stream MUST be uniquely identified by one **Synchronization Source (SSRC)**. This protocol maintains per SSRC transform independent parameters in cryptographic contexts, as specified in section 3.1.3.2.

When sending or receiving an SRTP packet, this protocol first uses the SRTP session and direction to identify the cryptographic context, then uses the SSRC in the packet to decide the per SSRC transform independent parameters in the cryptographic context.

3.1.3.2 SRTP Parameter Settings

For information regarding **SRTP** transform independent parameters and transform dependent parameters, see [RFC3711] sections 3.2.1 and 3.2.2.

This protocol requires the following parameter settings for transform independent parameters:

- The encryption algorithm MUST be **AES Counter Mode**, and encryption MUST be used.
- The authentication algorithm MUST be **Hash-based Message Authentication Code (HMAC)**-**SHA-256** hash function ([NIST.FIPS.180-4]), and authentication MUST be used.
- The replay list size MUST be 64 entries.
- The **master key** indicator MUST be used.
- The master key indicator length MUST be one byte.
- The key derivation rate MUST be zero.
- The master key length MUST be 128-bit.
- The master **salt** key length MUST be 112-bit.
- The encryption **session key** length MUST be 128-bit.
- The encryption session salt length MUST be 112-bit.
- The authentication session key length MUST be 160-bit.
- The master key lifetime MUST be 2^{48} -1 packets for **RTP** and 2^{31} -1 for **RTCP**.
- SRTCP and SRTP MUST have the same parameter settings with the exceptions specified in [RFC3711] section 3.2.1.

This protocol maintains the following transform independent parameters per **SSRC**.

- The rollover counter
- The highest received RTP sequence number
- The replay list

For information regarding transform dependent parameters, see sections 3.1.3.3.1 and 3.1.3.3.2.

Unless explicitly noted, this protocol follows SRTP, as specified in [RFC3711], to set other mandatory parameters.

3.1.3.3 SRTP Default Cryptographic Transform

This protocol implements a subset of the default **SRTP** algorithms.

3.1.3.3.1 Message Encryption

The **SRTP** default encryption algorithms are specified in [RFC3711] section 4.1.

This protocol MUST use **AES Counter Mode**. **AES** in f8 mode or **NULL cipher** mode MUST NOT be used.

This protocol requires that the encryption algorithm MUST be AES Counter Mode with the following parameters. For parameter details, see [RFC3711] section 4.1.

- n_b (block cipher size) MUST be 128-bit (AES algorithm's fixed cipher block size).
- n_e (encryption key size) MUST be 128-bit.
- The Session **salt** key MUST be used and n_s MUST be 112-bit.
- SRTP_PREFIX_LENGTH MUST be 0.

3.1.3.3.2 Message Authentication and Integrity

The **SRTP** default authentication algorithm is **Hash-based Message Authentication Code (HMAC)-SHA-256** [RFC2104], as specified in [RFC3711] section 4.2. This protocol implements HMAC-SHA-256 and requires the following parameters:

- n_a (authentication key size) MUST be 160-bit.
- n_tag (authentication tag size) MUST be 80-bit.

3.1.3.4 Session Key Derivation

This protocol implements the **session key** derivation algorithm specified in [RFC3711] section 4.3.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 SRTP Packet Processing

3.1.5.1.1 Sending an SRTP Packet

This protocol implements the steps specified in [RFC3711] section 3.3, with the exception of the method used to identify the appropriate **cryptographic context** and the per **SSRC** transform independent parameters. This protocol uses the method specified in section 3.1.3.1.

This protocol requires that **RTP packets** MUST be encrypted and authenticated.

3.1.5.1.2 Receiving an SRTP Packet

This protocol implements the steps specified in [RFC3711] section 3.3, with the following exceptions:

- This protocol uses the method specified in section 3.1.3.1 to identify the **cryptographic context** and the **SSRC** to identify the transform independent parameters in the cryptographic context.
- The replay checklist size MUST be 64 entries.

- This protocol logs the number of **SRTP** failures. Individual replay check failures or authentication failures are not logged.

3.1.5.2 SRTCP Packet Processing

3.1.5.2.1 Sending an SRTCP Packet

This protocol implements the steps specified in [RFC3711] section 3.4. **RTCP packets** MUST be encrypted and authenticated.

This protocol can adjust the avg_rtcp_size or packet_size variables, as specified in [RFC3711] section 3.4.

The SRTCP index counter is shared by all media streams on the same direction in the **SRTP** session.

3.1.5.2.2 Receiving an SRTCP Packet

This protocol implements the steps specified in [RFC3711] section 3.4, with the following exceptions:

- This protocol does not honor the e-bit. All incoming **RTCP packets** MUST be encrypted regardless of the e-bit setting.
- This protocol uses the method specified in section 3.1.3.1 to identify the **cryptographic context** to use.
- The SRTCP index counter is shared by all media streams.
- The replay checklist size MUST be 64 entries.
- This protocol logs the number of SRTCP failures. Individual replay check failures or authentication failures are not logged.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

This protocol does not introduce new protocol behaviors. The test vectors in [RFC3711] apply to this protocol. For more information, see [RFC3711] Appendix B.

5 Security

5.1 Security Considerations for Implementers

- **Master keys** are randomly generated. The send and receive directions in the same **SRTP** session do not use the same master key.
- Master key exchange is done through external mechanisms in **SDP**. SDP is transferred on a secure transport, for instance Transport Layer Security TLS.
- The Initial **RTP** sequence number is randomly generated. But it cannot use a value close to 65535, because this could cause a rollover counter mismatch if there is packet loss at the beginning of session startup. For example, the server products supported by this protocol use a random value between 0 and 32767.
- SRTP cannot terminate the connection when a replay attack is detected. Some **RTP profiles** intentionally send the same packet multiple times, and the duplicated packets fail replay check. For example, **DTMF** as described in [MS-DTMF].

5.2 Index of Security Parameters

Security parameter	Section
Encryption algorithm	3.1.3.2
Authentication algorithm	3.1.3.2
Replay list size	3.1.3.2
Master key indicator length	3.1.3.2
Session key derivation rate	3.1.3.2
Master key length	3.1.3.2
Master **salt** length	3.1.3.2
Encryption session key length	3.1.3.2
Encryption session salt length	3.1.3.2
Authentication session key length	3.1.3.2
Master key lifetime	3.1.3.2
Advanced Encryption Standard (AES) cipher block size	3.1.3.3.1
SRTP cipher prefix size	3.1.3.3.1
Authentication tag size	3.1.3.3.2

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office Communications Server 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015
- Windows 10 v1511 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Microsoft Skype for Business 2019
- Microsoft Skype for Business Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
All	Glossary term SHA-1 replaced by SHA-526.	Minor

8 Index

T

V