[MS-AVEDGEA]:

Audio Video Edge Authentication Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may 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, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Copyright © 2015 Microsoft Corporation

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial version
4/25/2008	0.2		Revised and edited the technical content
6/27/2008	1.0		Revised and edited the technical content
8/15/2008	1.01		Revised and edited the technical content
12/12/2008	2.0		Revised and edited the technical content
2/13/2009	2.01		Revised and edited the technical content
3/13/2009	2.02		Edited the 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	No Change	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	No Change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	No Change	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	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Copyright © 2015 Microsoft Corporation

Date	Revision History	Revision Class	Comments
7/30/2013	4.1	Minor	Clarified the meaning of the technical content.
11/18/2013	5.0	Major	Significantly changed the technical content.
2/10/2014	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Copyright © 2015 Microsoft Corporation

Table of Contents

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation

1 Introduction

The Audio Video Edge Authentication Protocol is a proprietary protocol used by protocol clients to get security tokens needed needed to authenticate themselves with a server that implements the Traversal Using Relay NAT (TURN) Extensions protocol, as described in [MS-TURN], for use with the Interactive Connectivity Establishment (ICE) Extensions protocol, as described in [MS-ICE] and [MS-ICE2].

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

Audio/Video Edge Server (A/V Edge Server): A protocol server that implements the Traversal Using Relay NAT (TURN) Extensions Protocol, as described in [MS-TURN]. The protocol server provides connectivity to a protocol client that is behind a network entity, if the network entity provides network address translation (NAT).

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

Content-Type header: A message header field whose value describes the type of data that is in the body of the message.

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

fully qualified domain name (FQDN): An unambiguous domain name (2) that gives an absolute location in the Domain Name System's (DNS) hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

Hash-based Message Authentication Code (HMAC): A mechanism for message authentication (2) 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.

network address translation (NAT): The process of converting between IP addresses used within an intranet, or other private network, and Internet IP addresses.

SERVICE: A method that is defined by **Session Initiation Protocol (SIP)** extensions and is used by an SIP client to request a service from a server.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

SHA-1: An algorithm that generates a 160-bit hash value from an arbitrary amount of input data, as described in [RFC3174]. SHA-1 is used with the Digital Signature Algorithm (DSA) in the Digital Signature Standard (DSS), in addition to other algorithms and standards.

SHA-256: An algorithm that generates a 256-bit hash value from an arbitrary amount of input data, as described in [FIPS180-2].

shared-secret: Data that is known only to the parties that are involved in a secure communication.

Copyright © 2015 Microsoft Corporation

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group. See [RFC4346].

TURN server: An **endpoint** that receives Traversal Using Relay NAT (TURN) request messages and sends TURN response messages. The protocol server acts as a data relay, receiving data on the public address that is allocated to a protocol client and forwarding that data to the client.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

XML: The Extensible Markup Language, as described in [XML1.0].

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

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.

[IETFDRAFT-SIPSOAP-00] Deason, N., "SIP and SOAP", draft-deason-sip-soap-00, June 30 2000, http://www.softarmor.com/wgdb/docs/draft-deason-sip-soap-00.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

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

Copyright © 2015 Microsoft Corporation

1.2.2 Informative References

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

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

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

[MS-TURN] Microsoft Corporation, "Traversal Using Relay NAT (TURN) Extensions".

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, http://www.w3.org/TR/2004/REC-xml-20040204/

1.3 Overview

[MS-TURN] is used for **network address translation (NAT)** and firewall traversal. To help protocol clients traverse NATs and firewalls, a **TURN server** or servers need to be deployed at particular places in the network topology. With the security tokens supplied by this protocol, a protocol client can authenticate with these TURN servers.

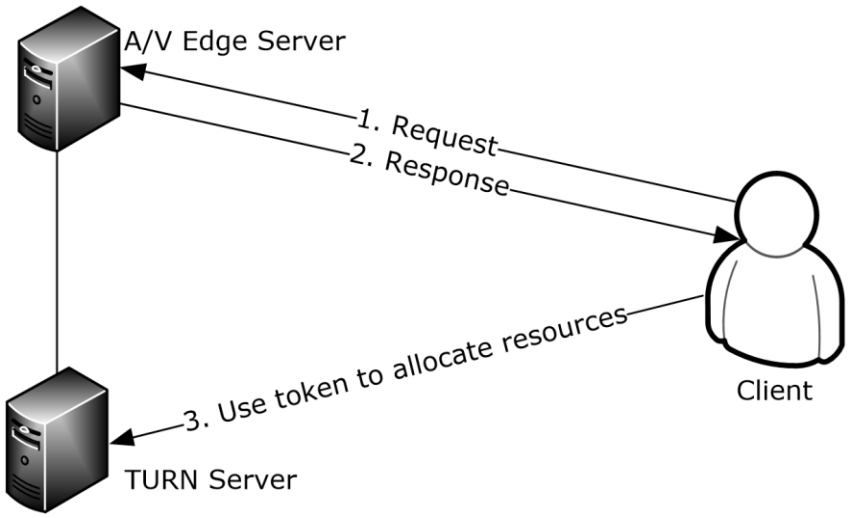

Figure 1: Protocol overview

The **Audio/Video Edge Server (A/V Edge Server)** is associated with a TURN server and is aware of the configuration details of the associated TURN server. When the protocol client needs tokens, it sends a **Session Initiation Protocol (SIP) SERVICE** request, as described in [IETFDRAFT-SIPSOAP-00], to the A/V Edge Server with the body of the message encoded in an **XML** format, as described in [XML10]. The server responds with a SIP SERVICE response message and a response code which indicates whether the response was a success or failure. If it was a success, the response contains the security tokens along with location information of the associated TURN server. If it was a failure, the response code indicates the type of failure. If there was an error with the XML body of the request, the response also contains an XML body that describes the exact cause of the problem.

The A/V Edge Server shares **shared-secret** keys with the associated TURN server and uses these keys to generate tokens. A security token consists of a user name and password. The token is valid only for a certain amount of time. If the protocol client requires the token to be valid for a shorter time interval, it can specify the length of the interval in the XML request. The server honors this value if it is less than the default duration it uses. Because the expiration time in the token is not easily decipherable, the response also includes a duration element that specifies how long the token is valid. The token also includes a hash of the identity specified by the protocol client, which the TURN server can use to implement resource management.

Copyright © 2015 Microsoft Corporation

1.4 Relationship to Other Protocols

This protocol uses [MS-SIPRE] for receiving requests and sending out responses. This protocol uses the Session Initiation Protocol (SIP) SERVICE method, as described in [IETFDRAFT-SIPSOAP-00], which is an extension to the standard SIP, to receive and send responses. The security tokens received from the A/V Edge Server are used to obtain access to the TURN server for use with the [MS-ICE] protocol.

1.5 Prerequisites/Preconditions

This protocol assumes that the TURN server associated with the A/V Edge Server has two network interfaces, one facing the Internet and the other facing the intranet as shown in the following figure.

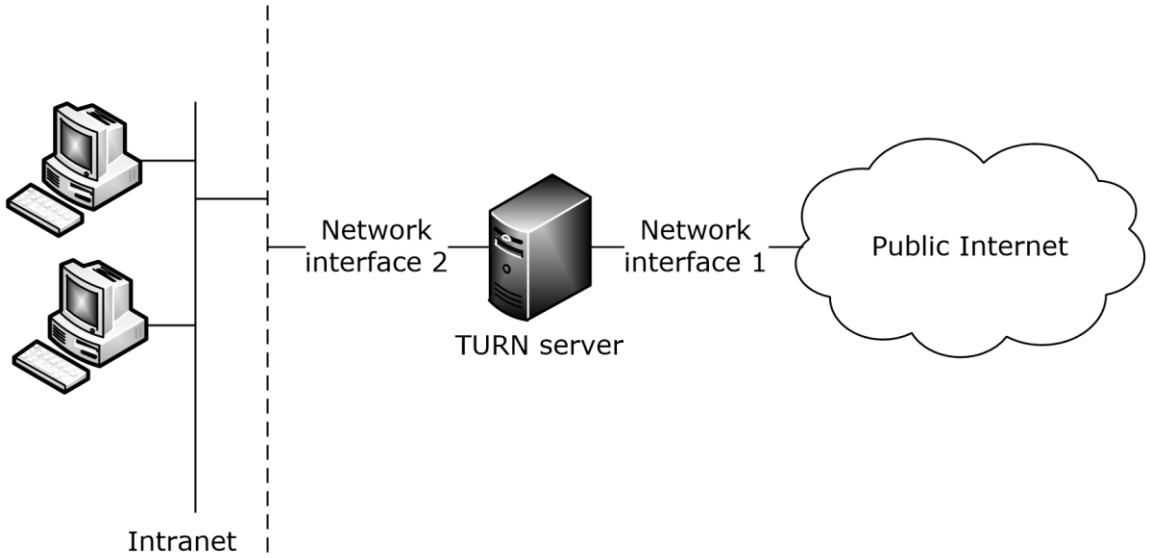

Figure 2: TURN server with two interfaces

The server implementing this protocol is assumed to be configured with the following:

- A certificate for establishing **Transport Layer Security (TLS)** connections. The certificate has a private key.
- Two shared-secret keys, known both to the A/V Edge Server and the associated TURN server.
- **User Datagram Protocol (UDP)** and **Transmission Control Protocol (TCP)** ports, on which the associated TURN server listens for protocol requests described in [MS-TURN]. The default for UDP is port 3478. The default for TCP is port 443.
- The IP address and **fully qualified domain name (FQDN)** of each network interface of the associated TURN server.
- The server version with the value "3.0".

1.6 Applicability Statement

This protocol is used to provide a protocol client with security tokens for accessing the TURN server.

1.7 Versioning and Capability Negotiation

This protocol negotiates versions in the following manner:

Copyright © 2015 Microsoft Corporation

1. The protocol client specifies the version in the XML body of the request.
2. If the server does not support the version requested by the protocol client, the request is rejected with a "Version Mismatch" error.

For more information, see section 3.1.5.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Copyright © 2015 Microsoft Corporation

2 Messages

2.1 Transport

Audio/video edge authentication protocol messages MUST be transported using SIP SERVICE messages through a connection secured with TLS.

2.2 Message Syntax

The request and response messages of this protocol MUST be SIP SERVICE messages, as specified in [IETFDRAFT-SIPSOAP-00]. Protocol requests MUST include a **Content-Type header**, "application/msrtc-media-relay-auth+xml". The schema definition specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2] for the request and response messages is documented in section 6.

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
tns	http://schemas.microsoft.com/2006/09/sip/mrasp	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2] [XMLSCHEMA2/2]

2.2.2 Request by the Client

The XML request sent by the protocol client MUST include exactly one **request** element.

2.2.2.1 request Element

2.2.2.1.1 request Element Definition

The schema definition for the **request** element is as follows:

```
<!-- REQUEST ELEMENT-->
<xs:element name="request" type="tns:requestType" />
```

2.2.2.1.2 requestType Type Definition

The schema definition for the **requestType** type is as follows:<1>

```
<!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>
      <!-- number of credentials requests will be bound within MRAS-->
      <xs:element name="credentialsRequest" type="tns:credentialsRequestType"
                  minOccurs="1"  maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required"/>
    <xs:attribute name="version" type="tns:versionType" use="required"/>
```


Copyright © 2015 Microsoft Corporation

```
    <xs:attribute name="to" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="from" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="route" type="tns:routeType" use="optional"
                  default="loadbalanced"/>
  </xs:complexType>
```

The following table describes the **requestType** type's attributes.

Attribute	Description
requestID	Type **max64CharStringType**, as defined in section 2.2.4.8. An identifier that is used to identify the request. This can be used by the protocol client to associate the response with the request, in case the protocol client sent multiple simultaneous requests to the server with a unique **requestID**.
version	Type **versionType**, as defined in section 2.2.4.12. This is the version requested by the protocol client. The version MUST be set as specified in section 3.1.5.2.1.
to	Type **mrasUriType**, as defined in section 2.2.4.11. A restricted length **Uniform Resource Identifier (URI)** type that identifies the entity to which the request needs to be sent. It MUST be a Session Initiation Protocol (SIP) URI.
from	Type **mrasUriType**, as defined in section 2.2.4.11. A restricted length URI type that identifies the entity that originated the request. It MUST be a SIP URI.
route	Type **routeType**, as defined in section 2.2.4.1. An optional attribute with a default value of "loadbalanced".

2.2.2.1.3 Child Element

The child element of the **requestType** element is defined in the following subsections.

2.2.2.1.3.1 credentialsRequest Element Definition

The schema definition for the **credentialsRequest** element is as follows:

```
<xs:element name="credentialsRequest" type="tns:credentialsRequestType" minOccurs="1"
maxOccurs="100" />
```

The following table describes the **credentialsRequest** element.

Element	Description
credentialsRequest	One or more subelements of type **credentialsRequestType**, as defined in section 2.2.2.1.3.2, MUST be present within the **requestType**. The **maxOccurs** attribute specifies the maximum number of subelements that MUST be present in the request. The schema allows 100 subelements.

2.2.2.1.3.2 credentialsRequestType Type Definition

Copyright © 2015 Microsoft Corporation

The schema definition for the **credentialsRequestType** type is as follows:

```
<!-- CREDENTIALS REQUEST TYPE-->
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0" />
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType"
                  use="required" />
  </xs:complexType>
```

The following table describes the **credentialsRequestType** type's elements.

Element	Description
identity	Type **max64kCharStringType**, as defined in section 2.2.4.9. A restricted length string type that identifies the entity to which the token SHOULD be issued. This is a required field.
location	Type **locationType**, as defined in section 2.2.4.2. This is an optional attribute that specifies the network interface on the associated TURN server for which the tokens and FQDN or IP address and port information MUST be returned.
duration	Type **positiveInteger**, as specified in [XMLSCHEMA2/2] section 3.3.25. This is an optional attribute that specifies the number of minutes for which the token needs to be valid. The default duration is 480 minutes.

The following table describes the **credentialsRequestType** type's attribute.

Attribute	Description
credentialsRequestID	Type **max64CharStringType**, as defined in section 2.2.4.8. A restricted length string type that identifies the **credentialsRequest** element within a **requestType**.

2.2.3 Response by the Server

The XML response sent by the server MUST include exactly one **response** element.

2.2.3.1 response Element

2.2.3.1.1 response Element Definition

The schema definition for the **response** element is as follows:

```
<!-- RESPONSE ELEMENT-->
<xs:element name="response" type="tns:responseType" />
```

2.2.3.1.2 responseType Type Definition

Copyright © 2015 Microsoft Corporation

The schema definition for the **responseType** type is as follows:<2>

```
<!-- RESPONSE TYPE-->
<xs:complexType name="responseType">
  <xs:sequence>
    <xs:element name="credentialsResponse" type="tns:credentialsResponseType"
                minOccurs="0" maxOccurs="100"/>
  </xs:sequence>
  <xs:attribute name="requestID" type="tns:max64CharStringType"/>
  <xs:attribute name="version" type="tns:versionType" use="required"/>
  <xs:attribute name="serverVersion" type="tns:versionType" use="optional"/>
  <xs:attribute name="to" type="tns:mrasUriType"/>
  <xs:attribute name="from" type="tns:mrasUriType"/>
  <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required"/>
</xs:complexType>
```

The following table describes the **responseType** type's attributes.

Attribute	Description
requestID	Type **max64CharStringType**, as defined in section 2.2.4.8. If the request is malformed (based on the rules in section 3.1.5.1.1), this attribute is not present. If present, this attribute MUST have the same value as the **requestID** attribute in the request.
version	Type **versionType**, as defined in section 2.2.4.12. The version MUST be set as specified in section 3.1.5.2.2.
serverVersion	Type **versionType**, as defined in section 2.2.4.12. This is an optional attribute. If present, this SHOULD be the server version with a value of "3.0"<3>.
to	Type **mrasUriType**, as defined in section 2.2.4.11. If the request is malformed (based on the rules in section 3.1.5.1.1), this attribute is not present. If present, this attribute MUST have the same value as the **to** attribute in the request.
from	Type **mrasUriType**, as defined in section 2.2.4.11. If the request is malformed (based on the rules in section 3.1.5.1.1), this attribute is not present. If present, this attribute MUST have the same value as the **from** attribute in the request.
reasonPhrase	Type **reasonPhraseType**, as defined in section 2.2.4.7. Specifies the reason for the error. A detailed description of when each error message is thrown is provided in section 3.1.5.

2.2.3.1.3 Child Element

The child element of the **responseType** element is defined in the following subsections.

2.2.3.1.3.1 credentialsResponse Element Definition

The schema definition for the **credentialsResponse** element is as follows:

```
<xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0"
maxOccurs="100" />
```


Copyright © 2015 Microsoft Corporation

The following table describes the **credentialsResponse** element.

Element	Description
credentialsResponse	For each **credentialsRequest** in the request from the protocol client, if the request succeeded, a **credentialsResponse** element MUST be included with the TURN server information and the token for the identity in the **credentialsRequest**. If the processing of the request does not succeed, there MUST be no **credentialsResponse** in the response sent by the server.

2.2.3.1.3.2 credentialsResponseType Type Definition

The schema definition for **credentialsResponseType** type is as follows:

```
<!--CREDENTIALS RESPONSE TYPE-->
  <xs:complexType name="credentialsResponseType">
    <xs:sequence>
      <xs:element name="credentials" type="tns:credentialsType" />
      <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"
/>
  </xs:complexType>
```

The following table describes the **credentialsResponseType** type's elements.

Element	Description
credentials	The type of this element is **credentialsType**, as defined in section 2.2.4.3. This element MUST contain the following subelements: ▪ A **username** element and a **password** element, which form the security token needed by the protocol client to contact the TURN server. ▪ A **duration** element that specifies how long the token is valid for. This element SHOULD contain an optional **realm** element that specifies which network segment the server belongs to. The network segment information is configured on the server and is implementation specific.
mediaRelayList	The type of this element is **mediaRelayListType**, as defined in section 2.2.4.5. Each **mediaRelay** element in **mediaRelayList** MUST contain the following subelements: ▪ A **location** element that indicates "internet" or "intranet". ▪ Either the **hostName** element or the **directIPAddress** element. If **hostName** is present, it SHOULD contain the IP address of the TURN server or the FQDN that resolves to the TURN server's IP address. If **directIPAddress** is present, it MUST contain the IP4 or IPv6<4> address of the TURN server. Each **mediaRelay** element in **mediaRelayList** SHOULD contain the following subelements: ▪ An optional **tcpPort** element that specifies the port the TURN server is using to listen for TCP.

Copyright © 2015 Microsoft Corporation

Element	Description
	▪ An optional **udpPort** element that specifies the port the TURN server is using to listen for UDP.

The following table describes the **credentialsResponseType** type's attribute.

Attribute	Description
credentialsRequestID	This MUST be the same as the **credentialsRequestID** attribute in the request.

2.2.4 Basic Data Types

This section lists the basic data types used in the XML request from the client and the response from the server.

2.2.4.1 routeType

The schema definition for the **routeType** data type is as follows:

```
<xs:simpleType name="routeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="loadbalanced"/>
    <xs:enumeration value="directip"/>
  </xs:restriction>
</xs:simpleType>
```

The **routeType** data type is a string type that can take only two possible values: "loadbalanced" or "directip".

2.2.4.2 locationType

The schema definition for the **locationType** data type is as follows:

```
<xs:simpleType name="locationType">
  <xs:restriction base="xs:string">
     <xs:enumeration value="intranet"/>
     <xs:enumeration value="internet"/>
  </xs:restriction>
</xs:simpleType>
```

The **locationType** data type is a string type that can take only two possible values: "intranet" or "internet".

2.2.4.3 credentialsType

The schema definition for the **credentialsType** data type is as follows:

```
<xs:complexType name="credentialsType">
    <xs:sequence>
        <xs:element name="username" type="tns:max64kCharStringType" />
```


Copyright © 2015 Microsoft Corporation

```
        <xs:element name="password" type="tns:max64kCharStringType" />
        <xs:element name="duration" type="xs:positiveInteger" />
        <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The **credentialsType** data type is a complex data type that contains the following elements.

Element	Description
username	The type of this element is **max64kCharStringType**, as defined in section 2.2.4.9. This element contains part of the security token sent by the server to the protocol client.
password	The type of this element is **max64kCharStringType**, as defined in section 2.2.4.9. This element contains part of the security token sent by the server to the protocol client.
duration	The type of this element is **positiveInteger**, as specified in [XMLSCHEMA2/2] section 3.3.25**.** This element contains a positive integer that specifies how long the token is valid for.
realm	The type of this element is **max64kCharStringType**, as defined in section 2.2.4.9. This optional element specifies which network segment the server belongs to.

2.2.4.4 mediaRelayType

The schema definition for the **mediaRelayType** data type is as follows:

```
<xs:complexType name="mediaRelayType">
   <xs:sequence>
      <xs:element name="location" type="tns:locationType"/>
      <xs:choice>
        <xs:element name="hostName" type="tns:hostNameType"/>
        <xs:element name="directIPAddress" type="tns:max64CharStringType"/>
       </xs:choice>
      <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0"/>
      <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The **mediaRelayType** data type is a complex data type that specifies the details of the TURN server.

Element	Description
location	Type **locationType**, as defined in section 2.2.4.2. The location of the TURN server whose **IPAddress** and **Port** information are returned to the client.
hostName	Type **hostNameType**, as defined in section 2.2.4.6. The FQDN or the IP address of the TURN server. Either this element or the **directIPAddress** element in the **mediarelayType** is present.
directIPAddress	Type **max64CharStringType**, as defined in section 2.2.4.8. The **IPAddress** of the TURN server.
udpPort	Type **unsignedShort**, as specified in [XMLSCHEMA2/2] section 3.3.23. The UDP port on which the TURN server is listening.

Copyright © 2015 Microsoft Corporation

Element	Description
tcpPort	Type **unsignedShort**, as specified in [XMLSCHEMA2/2] section 3.3.23. The TCP port on which the TURN server is listening.

2.2.4.5 mediaRelayListType

The schema definition for the **mediaRelayListType** data type is as follows:

```
<xs:complexType name="mediaRelayListType">
    <xs:sequence>
      <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

The **mediaRelayListType** data type contains a list of elements of type **mediaRelayType**, as defined in section 2.2.4.4.

2.2.4.6 hostNameType

The schema definition for the **hostNameType** data type is as follows:

```
    <xs:simpleType name="hostNameType">

   <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_\-\.]*" />
      <xs:maxLength value="255"/>
   </xs:restriction>
</xs:simpleType>
```

The **hostNameType** data type is a string type with a length restriction of 255 characters and the pattern of characters adhering to the regular expression defined previously in the schema.

2.2.4.7 reasonPhraseType

The schema definition for the **reasonPhraseType** data type is as follows:

```
<xs:simpleType name="reasonPhraseType">
   <xs:restriction base="xs:string">
     <xs:enumeration value="OK"/>
     <xs:enumeration value="Request Malformed"/>
     <xs:enumeration value="Request Too Large"/>
     <xs:enumeration value="Not Supported"/>
     <xs:enumeration value="Server Busy"/>
     <xs:enumeration value="Time Out"/>
     <xs:enumeration value="Forbidden"/>
     <xs:enumeration value="Internal Server Error"/>
     <xs:enumeration value="Other Failure"/>
     <xs:enumeration value="Version Mismatch"/>
   </xs:restriction>
</xs:simpleType>
```

The **reasonPhraseType** data type is a string type that can only take one of the values specified in the previous enumeration.

Copyright © 2015 Microsoft Corporation

2.2.4.8 max64CharStringType

The schema definition for the **max64CharStringType** data type is as follows:

```
<xs:simpleType name='max64CharStringType'>

  <xs:restriction base='xs:string'>
     <xs:maxLength value='64'></xs:maxLength>
  </xs:restriction>
 </xs:simpleType>
```

The **max64CharStringType** data type is a string type with a length restriction of 64 characters.

2.2.4.9 max64kCharStringType

The schema definition for the **max64kCharStringType** data type is as follows:

```
<xs:simpleType name='max64kCharStringType'>

  <xs:restriction base='xs:string'>
     <xs:maxLength value='64000'></xs:maxLength>

     </xs:restriction>
  </xs:simpleType>
```

The **max64kCharStringType** data type is a string type with a length restriction of 64,000 characters.

2.2.4.10 max8CharStringType

The schema definition for the **max8CharStringType** data type is as follows:

```
<xs:simpleType name='max8CharStringType'>
    <xs:restriction base='xs:string'>
      <xs:maxLength value='8'></xs:maxLength>
    </xs:restriction>
</xs:simpleType>
```

The **max8CharStringType** data type is a string type with a length restriction of eight characters.

2.2.4.11 mrasUriType

The schema definition for the **mrasUriType** data type is as follows:

```
<xs:simpleType name='mrasUriType'>
    <xs:restriction base='xs:anyURI'>
      <xs:maxLength value='10000'></xs:maxLength>
    </xs:restriction>
</xs:simpleType>
```

The **mrasUriType** data type is defined as an **anyURI** type, as specified in [XMLSCHEMA2/2] section 3.2.17, with a length restriction of 10,000 characters.

Copyright © 2015 Microsoft Corporation

2.2.4.12 versionType

The schema definition for the **versionType** data type is as follows:

```
<xs:simpleType name='versionType'>
  <xs:restriction base='xs:string'>
    <xs:pattern value="[0-9]+\.[0-9]+"></xs:pattern>
    <xs:maxLength value="5"/>
  </xs:restriction>
</xs:simpleType>
```

The **versionType** data type is a string type with a length restriction of five characters and a pattern adhering to the regular expression as defined previously in the schema.

Copyright © 2015 Microsoft Corporation

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

When a SIP SERVICE request, as described in section 2.2.2, is received by the server, the request MUST be processed based on the rules given in section 3.1.5 and a SIP SERVICE response message, as described in section 2.2.3, MUST be sent back to the client. The message contains the token information if the processing of the request succeeded or a detailed error description if the processing failed.

3.1.5 Message Processing Events and Sequencing Rules

The SIP SERVICE request described in section 2.2.2 MUST be the only message type that is accepted. The server MUST respond with a SIP SERVICE response message, as described in section 2.2.3, with the exceptions specified in section 3.1.5.1. The error codes indicate the type of the error in the request.

3.1.5.1 General Rules

When a request is received from the protocol client, it is processed based on the following rules:

- If the request message type is not SIP SERVICE, the request MUST be rejected and an **UnsupportedMessageType** error response, as defined in section 3.1.5.6, MUST be sent.
- If the **Content-Type** header of the request is not equal to "application/msrtc-media-relay-auth+xml", an **UnsupportedContentType** error response, as defined in section 3.1.5.6, MUST be sent. The SIP header of the response MUST include an **Accept** header with the value "application/msrtc-media-relay-auth+xml".
- For the previous two rules, the server does not send a XML response in its body. The error codes described in this section indicate the nature of the problem. In the checks that follow, if an error condition occurs, an XML body adhering to conditions described in section 2.2.3 MUST be sent. The **reasonPhrase** of the error message MUST be as described in the following sections.

3.1.5.1.1 Processing a Malformed Request

If the request does not adhere to schema rules, the request is malformed and MUST be rejected with a **reasonPhrase** set to "Request Malformed". If the request adheres to schema rules, except that the number of **credentialsRequest** in the XML request is greater than the **maxOccurs** attribute in the schema, the request is not considered malformed. In this case, the **reasonPhrase** in the error response MUST be "Request Too Large".

Copyright © 2015 Microsoft Corporation

3.1.5.1.2 Server Policies

The server can implement policies that restrict the request that can be sent by the protocol client. For example, the server can implement policies that restrict the number of subelements that are allowed in the request. If these policies are violated, the server MAY send an error response with a "Forbidden" **reasonPhrase**.

3.1.5.2 Version Validation

The **version** attribute in the request and response is of the type **versionType**, as defined in section 2.2.4.12.<5>

3.1.5.2.1 Validating the Version in the Request

The version in the request SHOULD be either "3.0", "2.0" or "1.0". Otherwise, the server SHOULD return an error response with a **reasonPhrase** of "Version Mismatch".

3.1.5.2.2 Setting the Version in the Response

If the server cannot read the protocol client's request because it does not adhere to the schema, the **version** in the response SHOULD be the server version.

If the protocol client's request adheres to the schema:

- If a "Version Mismatch" error response is returned to the protocol client, the highest version supported by the server that is less than the protocol client's version SHOULD be returned in the response.
- Otherwise, the **version** MUST be the version in the protocol client's request.

3.1.5.3 Checking the Attributes of the Request

Schema validation MUST be done as specified in section 3.1.5.1.1, which checks if all the attributes of the request conform to the schema rules. If schema validation succeeds, some more checks MUST be done as follows.

The version validation is done as specified in section 3.1.5.2.1. The server can include an optional **serverVersion** attribute in the response which, if present, SHOULD<6> be the same as the server version. If the client's request is valid and the version is "1.0" in the protocol client's request, the server MUST NOT include the **serverVersion** attribute.

The **from** and **to** attributes MUST be SIP URIs. If the server is not able to parse the URI, the server MUST send an error message with **reasonPhrase** with the value "Request Malformed". If these attributes are valid, the values of these attributes along with **requestID** are copied to the response. The client SHOULD use the **requestID** to correlate the server response with its request in case multiple requests are pending at the same time to the server.

3.1.5.4 Generating the credentialResponse

For each **credentialRequest** in the XML request:

- If the **duration** attribute is present in the request, the lifetime of the token MUST be calculated as the minimum of the duration specified by the protocol client and the preconfigured default lifetime value. Otherwise the preconfigured default lifetime value MUST be used.
- The tokens, **username** and **password,** are generated using the lifetime calculated as stated previously and the **identity** specified by the protocol client in the XML request. The **username** and **password** generated MUST use **base64 encoding** and be included in the XML response.

Copyright © 2015 Microsoft Corporation

- A **credentialResponse** MUST be created with the same **credentialsRequestID** as the **credentialsRequest** element in the protocol client's request. The token information MUST be in the **credentials** element and the information regarding the TURN server MUST be in the **mediaRelayList** element.
- If the **location** element was specified by the protocol client, the TURN server information related to that location only MUST be included in the **mediaRelayList** element. Otherwise, both the intranet and Internet information of the TURN server, as shown in the figure titled "TURN server with two interfaces" in section 1.5, MUST be included in the **mediaRelayList** element.
- If the **route** attribute is "directip", the server MUST return the IPv4 or IPv6 address of the TURN server's **endpoint** in the **directIPAddress** attribute in the response. If the attribute is not present or if the value is "loadbalanced", the **hostName** subelement MUST be present and SHOULD contain the IP address of the TURN server or the FQDN that resolves to the TURN server's IP address.

3.1.5.5 Populating Attributes of the Response

If the request was processed successfully without an error, the **reasonPhrase** MUST be set to "OK", and the following apply:

- The **version** attribute in the response MUST be set as specified in section 3.1.5.2.2.
- The **from**, **to**, and **requestID** attributes MUST be included and MUST be equal to the appropriate values in the XML request.

If the **reasonPhrase** is "Request Malformed", the **from**, **to**, and **requestID** attributes MAY be included in the response. Otherwise, the **from**, **to**, and **requestID** attributes MUST be included and MUST be equal to the appropriate values in the XML request.

If an unexpected server error occurs during the processing of the request, the **reasonPhrase** MUST be "Internal Server Error."

The SIP error codes that MUST be sent for the different response messages are specified in section 3.1.5.6.

3.1.5.6 Error Codes

The following table shows the SIP error codes corresponding to the different **reasonPhrase** values in the SIP responses. Some of these **reasonPhrase** values are not currently in use by the server. The unused responses are indicated in the following table with a "No" value in the column titled "In use".

Value of reasonsPhrase in the response	SIP error code	In use
OK	200	Yes
Request Malformed	400	Yes
Forbidden	403	Yes
Time Out	408	No
Request Too Large	413	Yes
Server Busy	486	No
Internal Server Error	500	Yes
Other Failure	500	No

Copyright © 2015 Microsoft Corporation

Value of reasonsPhrase in the response	SIP error code	In use
Not Supported	501	No
Version Mismatch	501	Yes

The following two error responses are used when the XML body of the response is not sent.

Response	SIP error code	In use
UnsupportedMessageType	501	Yes
UnsupportedContentType	415	Yes

3.1.5.7 Token Generation

The A/V Edge Server and the associated TURN server share two secret keys. These keys are used to create the security tokens **username** and **password**, as defined in section 2.2.4.3, which the protocol clients use to authenticate themselves with the TURN server. The server MUST include the lifetime of the token and the **identity** specified by the protocol client in the XML request while generating **username**. If the **duration** attribute is present in the request, the lifetime of the token MUST be calculated as the minimum of the duration specified by the protocol client and the configured lifetime value in the server. The server MUST encode the **username** and **password** by means of base64 encoding before including them in the XML response.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

Copyright © 2015 Microsoft Corporation

4 Protocol Examples

The following examples illustrate the protocol request-response sequence.

4.1 Version 2.0 LoadBalanced Request and Response

In this example, a client is making a version "2.0" request for the TURN server's intranet FQDN.

4.1.1 Client Request to Server

```
SERVICE sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA
SIP/2.0

Via: SIP/2.0/TLS 10.56.65.225:7012

Max-Forwards: 70

From: <sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

To: <sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>

Call-ID: 7b25d8f0304c4655814760e624d7c3aa

CSeq: 1 SERVICE

Contact: <sip: client@contoso.com;opaque=user:epid:4WjlENtBIVSXgZyN1UZ6VgAA;gruu>

User-Agent: UCCP/2.0.6545.0 OC/2.0.6545.0 (Microsoft Office Communicator)

Proxy-Authorization: NTLM qop="auth", realm="SIP Communications Service", opaque="9574F9DA",
crand="5999c389", cnum="580", targetname="server1.contoso.com",
response="01000000064386630f99f6cb864399660"

Content-Type: application/msrtc-media-relay-auth+xml
Content-Length: 471

<request requestID="990512"
from="sip:client@contoso.com"
version="2.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <credentialsRequest credentialsRequestID="990512">
    <identity>sip:client@contoso.com</identity>
    <location>intranet</location>
    <duration>480</duration>
  </credentialsRequest>
</request>
```

4.1.2 Server Response to Client

```
SIP/2.0 200 OK

Authentication-Info: NTLM rspauth="0100000000000000C614C5BD64399660", srand="B8926199",
snum="862", opaque="9574F9DA", qop="auth", targetname="server1.contoso.com", realm="SIP
Communications Service"

Via: SIP/2.0/TLS 10.56.65.225:7012;ms-received-port=7012;ms-received-cid=19E00

FROM: "Client"<sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

TO: <sip:
relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>;tag=554ef3a784
```


Copyright © 2015 Microsoft Corporation

```
CSEQ: 1 SERVICE

CALL-ID: 7b25d8f0304c4655814760e624d7c3aa

CONTENT-LENGTH: 960

CONTENT-TYPE: application/msrtc-media-relay-auth+xml

SERVER: RTCC/3.0.0.0 Media Relay Authentication Service

ms-edge-proxy-message-trust: ms-source-type=EdgeProxyGenerated;ms-ep-fqdn=
relay.contoso.com@contoso.com;ms-source-verified-user=verified

<?xml version="1.0"?>
<response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
requestID="990512"
version="2.0"
serverVersion="3.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
from="sip:client@contoso.com"
reasonPhrase="OK"


xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp">

  <credentialsResponse credentialsRequestID="990512">
<credentials>
<username>AQAgAIaoZr4EM8gBrxTJGY83uqdEgRXUunam2c+RID/vAJeJSL4YINbAYMvRAHeANv+Zew==</username>
      <password>35yqSF/p3A8gWXFHOC9YJA2kdvY=</password>
      <duration>480</duration>
    </credentials>
    <mediaRelayList>
      <mediaRelay>
        <location>intranet</location>
        <hostName>relay.contoso.com</hostName>
        <udpPort>3478</udpPort>
        <tcpPort>443</tcpPort>
      </mediaRelay>
    </mediaRelayList>
  </credentialsResponse>
</response>
```

4.2 Version 3.0 DirectIP Request and Response

In this example, a client is making a version "3.0" request for the direct ip Internet address of a TURN server that is configured with both IPv4 and IPv6 addresses.

4.2.1 Client Request to Server

```
SERVICE sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA
SIP/2.0

Via: SIP/2.0/TLS 10.56.65.225:7012

Max-Forwards: 70

From: <sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

To: <sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>

Call-ID: 7b25d8f0304c4655814760e624d7c3aa

CSeq: 1 SERVICE

Contact: <sip: client@contoso.com;opaque=user:epid:4WjlENtBIVSXgZyN1UZ6VgAA;gruu>
```


Copyright © 2015 Microsoft Corporation

```
User-Agent: UCCP/2.0.6545.0 OC/2.0.6545.0 (Microsoft Office Communicator)

Proxy-Authorization: NTLM qop="auth", realm="SIP Communications Service", opaque="9574F9DA",
crand="5999c389", cnum="580", targetname="server1.contoso.com",
response="0100000064386630f99f6cb864399660"

Content-Type: application/msrtc-media-relay-auth+xml
Content-Length: 471

<request requestID="990512"
from="sip:client@contoso.com"
version="3.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <credentialsRequest credentialsRequestID="990512">
    <identity>sip:client@contoso.com</identity>
    <location>internet</location>
    <duration>480</duration>
    <route>directip</route>
  </credentialsRequest>
</request>
```

4.2.2 Server Response to Client

```
SIP/2.0 200 OK

Authentication-Info: NTLM rspauth="0100000000000000C614C5BD64399660", srand="B8926199",
snum="862", opaque="9574F9DA", qop="auth", targetname="server1.contoso.com", realm="SIP
Communications Service"

Via: SIP/2.0/TLS 10.56.65.225:7012;ms-received-port=7012;ms-received-cid=19E00

FROM: "Client"<sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

TO: <sip:
relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>;tag=554ef3a784

CSEQ: 1 SERVICE

CALL-ID: 7b25d8f0304c4655814760e624d7c3aa

CONTENT-LENGTH: 960

CONTENT-TYPE: application/msrtc-media-relay-auth+xml

SERVER: RTCC/3.0.0.0 Media Relay Authentication Service

ms-edge-proxy-message-trust: ms-source-type=EdgeProxyGenerated;ms-ep-fqdn=
relay.contoso.com@contoso.com;ms-source-verified-user=verified

<?xml version="1.0"?>
<response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
requestID="990512"
version="3.0"
serverVersion="3.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
from="sip:client@contoso.com"
reasonPhrase="OK"
xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp">

  <credentialsResponse credentialsRequestID="990512">
<credentials>
<username>AQAgAIaoZr4EM8gBrxTJGY83uqdEgRXUunam2c+RID/vAJeJSL4YINbAYMvRAHeANv+Zew==</username>
      <password>35yqSF/p3A8gWXFHOC9YJA2kdvY=</password>
```


Copyright © 2015 Microsoft Corporation

```
        <duration>480</duration>
      </credentials>
      <mediaRelayList>
        <mediaRelay>
          <location>internet</location>
          <directIPAddress>192.0.2.254</directIPAddress>
          <udpPort>3478</udpPort>
          <tcpPort>443</tcpPort>
        </mediaRelay>
        <mediaRelay>
          <location>internet</location>
          <directIPAddress>2001:0DB8::943c:fa53</directIPAddress>
          <udpPort>3478</udpPort>
          <tcpPort>443</tcpPort>
        </mediaRelay>
      </mediaRelayList>
    </credentialsResponse>
  </response>
```


Copyright © 2015 Microsoft Corporation

5 Security

5.1 Security Considerations for Implementers

5.1.1 Keyed Hash Function

This protocol uses the **HMAC SHA-256**<7> and HMAC **SHA-1** keyed hash functions for generating tokens.

5.1.2 Underlying Transport

Because the security tokens sent in this protocol response are in plain text, all the protocol clients communicate with the A/V Edge Server through a connection secured by TLS, as specified in section 2.1.

5.1.3 Authentication

Using this protocol, it is possible for unauthorized protocol clients to request tokens and obtain them. Also, a protocol client without proper authorization can send audio/video edge authentication protocol requests with different identities and obtain security tokens. This type of unauthorized activity precludes attempts by the TURN server to perform resource management based on protocol client identity that is present as the hash in the token. Consequently, the A/V Edge Server authenticates the protocol clients and verifies the request before distributing tokens.

5.2 Index of Security Parameters

Security Parameter	Section
The shared-secret keys between the A/V Edge Server and the TURN server.	3.1.5.7
The HMAC SHA-1 keyed hash algorithm.	5.1.1
The HMAC SHA-256 keyed hash algorithm.	5.1.1
The token generation algorithm.	3.1.5.7
A TLS certificate, if TLS is used.	2.1

Copyright © 2015 Microsoft Corporation

6 Appendix A: MS-AVEDGEA Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:tns="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      XML Schema for the MS-AVEDGEA
    </xs:documentation>
  </xs:annotation>


  <!-- REQUEST ELEMENT-->
  <xs:element name="request" type="tns:requestType" />


  <!-- RESPONSE ELEMENT-->
  <xs:element name="response" type="tns:responseType" />


  <!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>

      <xs:element name="credentialsRequest" type="tns:credentialsRequestType" minOccurs="1"
maxOccurs="100" />
    </xs:sequence>

    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required" />
   <xs:attribute name="version" type="tns:versionType" use="required" />
    <xs:attribute name="to" type="tns:mrasUriType" use="required" />
    <xs:attribute name="from" type="tns:mrasUriType" use="required" />
    <xs:attribute name="route" type="tns:routeType" use="optional" default="loadbalanced" />
  </xs:complexType>


  <!-- RESPONSE TYPE-->
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0"
maxOccurs="100" />
    </xs:sequence>

    <xs:attribute name="requestID" type="tns:max64CharStringType" />
    <xs:attribute name="version" type="tns:versionType" use="required" />
     <xs:attribute name="serverVersion" type="tns:versionType" use="optional"/>
    <xs:attribute name="to" type="tns:mrasUriType" />
    <xs:attribute name="from" type="tns:mrasUriType" />
    <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required" />
  </xs:complexType>


  <!-- CREDENTIALS REQUEST TYPE-->
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0" />
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"
/>
  </xs:complexType>

  <!--CREDENTIALS RESPONSE TYPE-->
  <xs:complexType name="credentialsResponseType">
```


Copyright © 2015 Microsoft Corporation

```
    <xs:sequence>
      <xs:element name="credentials" type="tns:credentialsType" />
      <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type=" tns:max64CharStringType" use="required"
/>
  </xs:complexType>


  <!--ROUTE TYPE-->
  <xs:simpleType name="routeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="loadbalanced" />
      <xs:enumeration value="directip" />
    </xs:restriction>
  </xs:simpleType>


  <!--LOCATION TYPE-->
  <xs:simpleType name="locationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="intranet" />
      <xs:enumeration value="internet" />
    </xs:restriction>
  </xs:simpleType>
  <!--CREDENTIALS TYPE-->
  <xs:complexType name="credentialsType">
    <xs:sequence>
        <xs:element name="username" type="tns:max64kCharStringType" />
        <xs:element name="password" type="tns:max64kCharStringType" />
        <xs:element name="duration" type="xs:positiveInteger" />
        <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!--MEDIA RELAY LIST TYPE-->
  <xs:complexType name="mediaRelayListType">
    <xs:sequence>
      <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>


  <!--MEDIA RELAY TYPE-->
  <xs:complexType name="mediaRelayType">
    <xs:sequence>
      <xs:element name="location" type="tns:locationType" />
      <xs:choice>
        <xs:element name="hostName" type="tns:hostNameType" />
        <xs:element name="directIPAddress"     type="tns:max64CharStringType" />
      </xs:choice>
      <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0" />
      <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>


  <!--DOMAIN NAME TYPE-->
  <xs:simpleType name="hostNameType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_\-\.]*" />
      <xs:maxLength value="255" />
    </xs:restriction>
  </xs:simpleType>


  <!--RESPONSE REASON PHRASE-->
  <xs:simpleType name="reasonPhraseType">
    <xs:restriction base="xs:string">
```


Copyright © 2015 Microsoft Corporation

```
        <xs:enumeration value="OK" />
        <xs:enumeration value="Request Malformed" />
        <xs:enumeration value="Request Too Large" />
        <xs:enumeration value="Not Supported" />
        <xs:enumeration value="Server Busy" />
        <xs:enumeration value="Time Out" />
        <xs:enumeration value="Forbidden" />
        <xs:enumeration value="Internal Server Error" />
        <xs:enumeration value="Other Failure" />
        <xs:enumeration value="Version Mismatch" />
      </xs:restriction>
    </xs:simpleType>


    <!--MAX 64 CHAR STRING TYPE-->
    <xs:simpleType name="max64CharStringType">
      <xs:restriction base="xs:string">
        <xs:maxLength value="64">
        </xs:maxLength>
      </xs:restriction>
    </xs:simpleType>


    <!--MAX 64k CHAR STRING TYPE-->
    <xs:simpleType name="max64kCharStringType">
      <xs:restriction base="xs:string">
        <xs:maxLength value="64000">
        </xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <!--MAX 8 CHAR STRING TYPE-->
    <xs:simpleType name="max8CharStringType">
      <xs:restriction base="xs:string">
        <xs:maxLength value="8">
        </xs:maxLength>
      </xs:restriction>
    </xs:simpleType>


    <!--mrasUri-->
    <xs:simpleType name="mrasUriType">
      <xs:restriction base="xs:anyURI">
        <xs:maxLength value="10000">
        </xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name='versionType'>
      <xs:restriction base='xs:string'>
        <xs:pattern value="[0-9]+\.[0-9]+"></xs:pattern>
  <xs:maxLength value="5"/>
      </xs:restriction>
    </xs:simpleType>


</xs:schema>
```

6.1 OCS 2007 Schema

```
  <?xml version="1.0" encoding="utf-8"?>
  <xs:schema
    targetNamespace="http://schemas.microsoft.com/2006/09/sip/mrasp"
    xmlns:tns="http://schemas.microsoft.com/2006/09/sip/mrasp"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
```


Copyright © 2015 Microsoft Corporation

```
  <xs:annotation>
    <xs:documentation xml:lang="en">
      XML Schema for the MS-AVEDGEA
    </xs:documentation>
  </xs:annotation>

  <!-- REQUEST ELEMENT-->
  <xs:element name="request" type="tns:requestType" />

  <!-- RESPONSE ELEMENT-->
  <xs:element name="response" type="tns:responseType" />

  <!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>
      <!-- number of credentials requests will be bound within MRAS-->
      <xs:element name="credentialsRequest" type="tns:credentialsRequestType" minOccurs="1"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required"/>
    <xs:attribute name="version" type="tns:max8CharStringType" use="required"/>
    <xs:attribute name="to" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="from" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="route" type="tns:routeType" use="optional" default="loadbalanced"/>
  </xs:complexType>

  <!-- RESPONSE TYPE-->
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType"/>
    <xs:attribute name="version" type="tns:max8CharStringType" use="required"/>
    <xs:attribute name="to" type="tns:mrasUriType"/>
    <xs:attribute name="from" type="tns:mrasUriType"/>
    <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required"/>
  </xs:complexType>

  <!-- CREDENTIALS REQUEST TYPE-->
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0"/>
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"/>
  </xs:complexType>

  <!-- RESPONSE TYPE-->
    <xs:complexType name="credentialsResponseType">
      <xs:sequence>
        <xs:element name="credentials" type="tns:credentialsType" />
        <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
      </xs:sequence>
      <xs:attribute name="credentialsRequestID" type="xs:string" use="required"/>
    </xs:complexType>

  <!--ROUTE TYPE-->
  <xs:simpleType name="routeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="loadbalanced"/>
      <xs:enumeration value="directip"/>
    </xs:restriction>
  </xs:simpleType>

    <!--LOCATION TYPE-->
    <xs:simpleType name="locationType">
      <xs:restriction base="xs:string">
```


Copyright © 2015 Microsoft Corporation

```
          <xs:enumeration value="intranet"/>
          <xs:enumeration value="internet"/>
        </xs:restriction>
      </xs:simpleType>

      <!--CREDENTIALS TYPE-->
      <xs:complexType name="credentialsType">
        <xs:sequence>
          <xs:element name="username" type="tns:max64kCharStringType" />
          <xs:element name="password" type="tns:max64kCharStringType" />
          <xs:element name="duration" type="xs:positiveInteger" />
          <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>

    <!--MEDIA RELAY LIST TYPE-->
    <xs:complexType name="mediaRelayListType">
      <xs:sequence>
        <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>

<!--MEDIA RELAY TYPE-->
    <xs:complexType name="mediaRelayType">
      <xs:sequence>
        <xs:element name="location" type="tns:locationType"/>
        <xs:choice>
          <xs:element name="hostName" type="tns:hostNameType"/>
          <xs:element name="directIPAddress" type="tns:max64CharStringType"/>
         </xs:choice>
        <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0"/>
        <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>

    <!--DOMAIN NAME TYPE-->
    <xs:simpleType name="hostNameType">
      <xs:restriction base="xs:string">
        <xs:pattern value="[a-zA-Z0-9 \-\.]*" />
        <xs:maxLength value="255"/>
      </xs:restriction>
    </xs:simpleType>

    <!--RESPONSE REASON PHRASE-->
    <xs:simpleType name="reasonPhraseType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="OK"/>
        <xs:enumeration value="Request Malformed"/>
        <xs:enumeration value="Request Too Large"/>
        <xs:enumeration value="Not Supported"/>
        <xs:enumeration value="Server Busy"/>
        <xs:enumeration value="Time Out"/>
        <xs:enumeration value="Forbidden"/>
        <xs:enumeration value="Internal Server Error"/>
        <xs:enumeration value="Other Failure"/>
        <xs:enumeration value="Version Mismatch"/>
      </xs:restriction>
    </xs:simpleType>

    <!--MAX 64 CHAR STRING TYPE-->
    <xs:simpleType name='max64CharStringType'>
      <xs:restriction base='xs:string'>
        <xs:maxLength value='64'></xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <!--MAX 1024 CHAR STRING TYPE-->
    <xs:simpleType name='max64kCharStringType'>
```


Copyright © 2015 Microsoft Corporation

```
      <xs:restriction base='xs:string'>
        <xs:maxLength value='64000'></xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <!--MAX 8 CHAR STRING TYPE-->
    <xs:simpleType name='max8CharStringType'>
      <xs:restriction base='xs:string'>
        <xs:maxLength value='8'></xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <!--mrasUri-->
    <xs:simpleType name='mrasUriType'>
      <xs:restriction base='xs:anyURI'>
        <xs:maxLength value='10000'></xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

  </xs:schema>
```


Copyright © 2015 Microsoft Corporation

7 Appendix B: Product Behavior

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

- Microsoft Office Communications Server 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015

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

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

<1> Section 2.2.2.1.2: Office Communications Server 2007, Office Communicator 2007: The **requestType** is the same as in the document except for the type of the version attribute which is defined as a string with eight characters.

<2> Section 2.2.3.1.2: Office Communications Server 2007, Office Communicator 2007: The **responseType** is the same as in the document except that the **serverVersion** attribute is not present and the type of the **version** attribute is "tns:max8CharStringType".

```
<xs:attribute name="version" type="tns:max8CharStringType" use="required"/>
```

<3> Section 2.2.3.1.2: Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This value is "2.0".

<4> Section 2.2.3.1.3.2: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported.

<5> Section 3.1.5.2: Office Communications Server 2007, Office Communicator 2007: The **version** attribute is defined as a string with eight characters.

<6> Section 3.1.5.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<7> Section 5.1.1: Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: behavior not supported.

Copyright © 2015 Microsoft Corporation

8 Change Tracking

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

Copyright © 2015 Microsoft Corporation

9 Index

Copyright © 2015 Microsoft Corporation

R

S

T

V

Copyright © 2015 Microsoft Corporation