[MS-RCMP]:

Remote Certificate Mapping 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.

Revision Summary

Date	Revision History	Revision Class	Comments
3/2/2007	1.0	Major	Updated and revised the technical content.
4/3/2007	1.1	Minor	Clarified the meaning of the technical content.
5/11/2007	1.2	Minor	Addressed EU feedback
6/1/2007	1.3	Minor	Clarified the meaning of the technical content.
7/3/2007	1.3.1	Editorial	Changed language and formatting in the technical content.
8/10/2007	1.3.2	Editorial	Changed language and formatting in the technical content.
9/28/2007	1.3.3	Editorial	Changed language and formatting in the technical content.
10/23/2007	2.0	Major	Converted document to unified format.
1/25/2008	2.0.1	Editorial	Changed language and formatting in the technical content.
3/14/2008	2.0.2	Editorial	Changed language and formatting in the technical content.
6/20/2008	2.1	Minor	Clarified the meaning of the technical content.
7/25/2008	2.2	Minor	Clarified the meaning of the technical content.
8/29/2008	2.2.1	Editorial	Changed language and formatting in the technical content.
10/24/2008	2.2.2	Editorial	Changed language and formatting in the technical content.
12/5/2008	3.0	Major	Updated and revised the technical content.
1/16/2009	3.0.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	3.0.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	3.0.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	4.0	Major	Updated and revised the technical content.
7/2/2009	5.0	Major	Updated and revised the technical content.
8/14/2009	5.0.1	Editorial	Changed language and formatting in the technical content.
9/25/2009	5.1	Minor	Clarified the meaning of the technical content.
11/6/2009	5.1.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	6.0	Major	Updated and revised the technical content.
1/29/2010	6.1	Minor	Clarified the meaning of the technical content.
3/12/2010	6.1.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	6.1.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	7.0	Major	Updated and revised the technical content.
7/16/2010	7.0	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	7.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
10/8/2010	7.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	7.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	7.1	Minor	Clarified the meaning of the technical content.
9/23/2011	7.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	8.0	Major	Updated and revised the technical content.
3/30/2012	8.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	8.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	9.0	Major	Updated and revised the technical content.
1/31/2013	9.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	10.0	Major	Updated and revised the technical content.
11/14/2013	10.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	10.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	10.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	11.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the Remote Certificate Mapping Protocol. The Remote Certificate Mapping Protocol is used by servers that authenticate users via X.509 certificates, as specified in [X509]. This protocol allows the server to use a directory, database, or other technology to map the user's X.509 certificate to a **security principal**. This protocol returns the authorization information associated with the security principal in the form of a **privilege attribute certificate (PAC)**, as specified in [MS-PAC], that represents the user's identity and group memberships. Throughout this document, **little-endian** format applies unless otherwise stated.

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:

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of **objects** in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

distinguished name (DN): A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a **domain controller (DC)** and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

domain controller (DC): The service, running on a server, that implements **Active Directory**, or the server hosting this service. The service hosts the data store for **objects** and interoperates with other **DCs** to ensure that a local change to an **object** replicates correctly across all **DCs**. When **Active Directory** is operating as Active Directory Domain Services (AD DS), the **DC** contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its forest. If the AD DS **DC** is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its forest. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When **Active Directory** is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS **DCs** can run on one server. When **Active Directory** is operating as AD DS, only one AD DS **DC** can run on one server. However, several AD LDS **DCs** can coexist with one AD DS **DC** on one server. The AD LDS **DC** contains full NC replicas of the config NC and the schema NC in its forest.

issuer name: The name of the certificate authority (CA) that signed and issued a certificate. The name is an X.509 format name, as specified in [X509].

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

object: A set of attributes (1), each with its associated values. Two attributes of an object have special significance: an identifying attribute and a parent-identifying attribute. An identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object. For the set of objects in a replica, the values of the identifying attribute are distinct. A parent-identifying attribute is a designated single-valued attribute that appears on every object; the value of this attribute identifies the object's parent. That is, this attribute contains the value of the parent's identifying attribute, or a reserved value identifying no object. For the set of objects in a replica, the values of this parent-identifying attribute define a tree with objects as vertices and child-parent references as directed edges with the child as an edge's tail and the parent as an edge's head. Note that an object is a value, not a variable; a replica is a variable. The process of adding, modifying, or deleting an object in a replica replaces the entire value of the replica with a new value. As the word replica suggests, it is often the case that two replicas contain "the same objects". In this usage, objects in two replicas are considered the same if they have the same value of the identifying attribute and if there is a process in place (replication) to converge the values of the remaining attributes. When the members of a set of replicas are considered to be the same, it is common to say "an object" as shorthand referring to the set of corresponding objects in the replicas.

principal: An authenticated entity that initiates a message or channel in a distributed system.

privilege attribute certificate (PAC): A Microsoft-specific authorization data present in the authorization data field of a ticket. The **PAC** contains several logical components, including group membership data for authorization, alternate credentials for non-Kerberos authentication protocols, and policy control information for supporting interactive logon.

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

RPC transport: The underlying network services used by the remote procedure call (RPC) runtime for communications between network nodes. For more information, see [C706] section 2.

security principal: A unique entity that is identifiable through cryptographic means by at least one key. It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Also referred to as principal.

service principal name (SPN): The name a client uses to identify a service for mutual authentication. (For more information, see [RFC1964] section 2.1.1.) An **SPN** consists of either two parts or three parts, each separated by a forward slash ('/'). The first part is the service class, the second part is the instance name, and the third part (if present) is the service name. For example, "ldap/dc-01.fabrikam.com/fabrikam.com" is a three-part **SPN** where "ldap" is the service class name, "dc-01.fabrikam.com" is the instance name, and "fabrikam.com" is the service name. See [SPNNAMES] for more information about **SPN** format and composing a unique **SPN**.

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is:

someone@example.com (in the form of an email address). In **Active Directory**, the userPrincipalName attribute (2) of the account object, as described in [MS-ADTS].

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-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L".

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z".

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-ERREF] Microsoft Corporation, "Windows Error Codes".

[MS-KILE] Microsoft Corporation, "Kerberos Protocol Extensions".

[MS-NRPC] Microsoft Corporation, "Netlogon Remote Protocol".

[MS-PAC] Microsoft Corporation, "Privilege Attribute Certificate Data Structure".

[MS-UCODEREF] Microsoft Corporation, "Windows Protocols Unicode Reference".

[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

[RFC4556] Zhu, L., and Tung, B., "Public Key Cryptography for Initial Authentication in Kerberos", RFC 4556, June 2006, http://www.ietf.org/rfc/rfc4556.txt

[RFC5280] Cooper, D., Santesson, S., Farrell, S., et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

[X690] ITU-T, "Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, http://www.itu.int/rec/T-REC-X.690/en

1.2.2 Informative References

[GUTMANN] Gutmann, P., "X.509 Style Guide", October 2000, http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.rfc-editor.org/rfc/rfc2246.txt

[RFC2716] Aboba, B. and Simon, D., "PPP EAP TLS Authentication Protocol", RFC 2716, October 1999, http://www.ietf.org/rfc/rfc2716.txt

1.3 Overview

The Remote Certificate Mapping Protocol is used in deployments where users rely on [X509] certificates to gain access to resources. After a client authenticates itself to a server using an X.509 certificate, the server uses the Remote Certificate Mapping Protocol to contact a directory to determine the authorization information to use, such as group memberships. The Remote Certificate Mapping Protocol returns a privilege attribute certificate (PAC), as specified in [MS-PAC], that represents the user's identity and group memberships, suitable for making authorization decisions.

There are three methods by which a certificate can be associated with an account for the purposes of authorization.

1. First, the **subjectAltName** field of the X.509 certificate should be treated as a **user principal name (UPN)** and used as the key, in the database sense, to locate the account record and corresponding authorization information.

2. Second, the issuer and subject names should be taken together as a key, again in the database sense, to locate the account record.

3. Third, the **issuer name** alone should be used as the lookup key when locating the account record.

The Remote Certificate Mapping Protocol itself consists of a single request/response message pair: SSL_CERT_LOGON_REQ (section 2.2.1) and SSL_CERT_LOGON_RESP (section 2.2.2). This request/response pair is transferred by using the generic pass-through capability of the Netlogon Remote Protocol, as specified in [MS-NRPC] section 3.2.4.1. The client creates an SSL_CERT_LOGON_REQ message that contains the X.509 certificate for which the client wants to obtain the corresponding authorization information and specifies which (or all) of the methods described in the preceding paragraph should be applied. The Remote Certificate Mapping Protocol server uses attributes of this X.509 certificate and the indicated methods by the client to determine the authorization information. Assuming an account is found, the Remote Certificate Mapping Protocol server then creates and returns a PAC, as specified in [MS-PAC], that contains the authorization information in the SSL_CERT_LOGON_RESP message to the Remote Certificate Mapping Protocol client.

The Remote Certificate Mapping Protocol specification uses common fields from X.509, as specified in [X509], including **subjectName**, **subjectAltName**, and **issuerName**. An implementer of the Remote Certificate Mapping Protocol must be familiar with X.509 certificates, in particular the verification and parsing of the certificate to extract the fields listed earlier. For more information about X.509, see [GUTMANN].

1.4 Relationship to Other Protocols

Any protocol that authenticates clients based on public key certificates can make use of the Remote Certificate Mapping Protocol to obtain authorization information about the client. The protocol described in [MS-NRPC] serves as the transport for Remote Certificate Mapping Protocol messages.

The Remote Certificate Mapping Protocol can be used to implement the Secure Sockets Layer/Transport Layer Security (SSL/TLS) protocol, as specified in [RFC2246], and Extensible Authentication (EAP) TLS protocol, as specified in [RFC2716], when client authentication by means of an X.509 certificate is selected as part of the TLS handshake. In the SSL/TLS, authentication of client is optional and is only done when requested by the SSL/TLS server. If the client authentication option is chosen, the SSL/TLS client authenticates itself to the SSL/TLS server using an X.509 certificate.

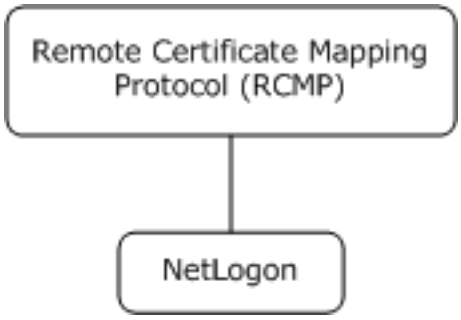


Figure 1: Protocol relationship diagram.

1.5 Prerequisites/Preconditions

The Remote Certificate Mapping Protocol requires that users have X.509 certificates available to them for authentication. The Remote Certificate Mapping Protocol also requires that a means exists to associate a certificate with a set of authorization data, commonly some form of an account database.<1>

1.6 Applicability Statement

The Remote Certificate Mapping Protocol is applicable in deployments where users have been issued X.509 certificates, as specified in [X509], and a common database for user and machine authorization information. In this type of environment, the Remote Certificate Mapping Protocol is used between the authentication step and authorization step. It enables the server that uses an authentication protocol using X.509 certificates to obtain a PAC, as specified in [MS-PAC], that represents the user's identity and group memberships, suitable for making authorization decisions.

1.7 Versioning and Capability Negotiation

The Remote Certificate Mapping Protocol does not have any versioning or capability negotiation.

1.8 Vendor-Extensible Fields

The Remote Certificate Mapping Protocol does not have any vendor-extensible fields.

1.9 Standards Assignments

There are no standards assignments in the Remote Certificate Mapping Protocol beyond the standards assignments as specified in [MS-NRPC].

2 Messages

2.1 Transport

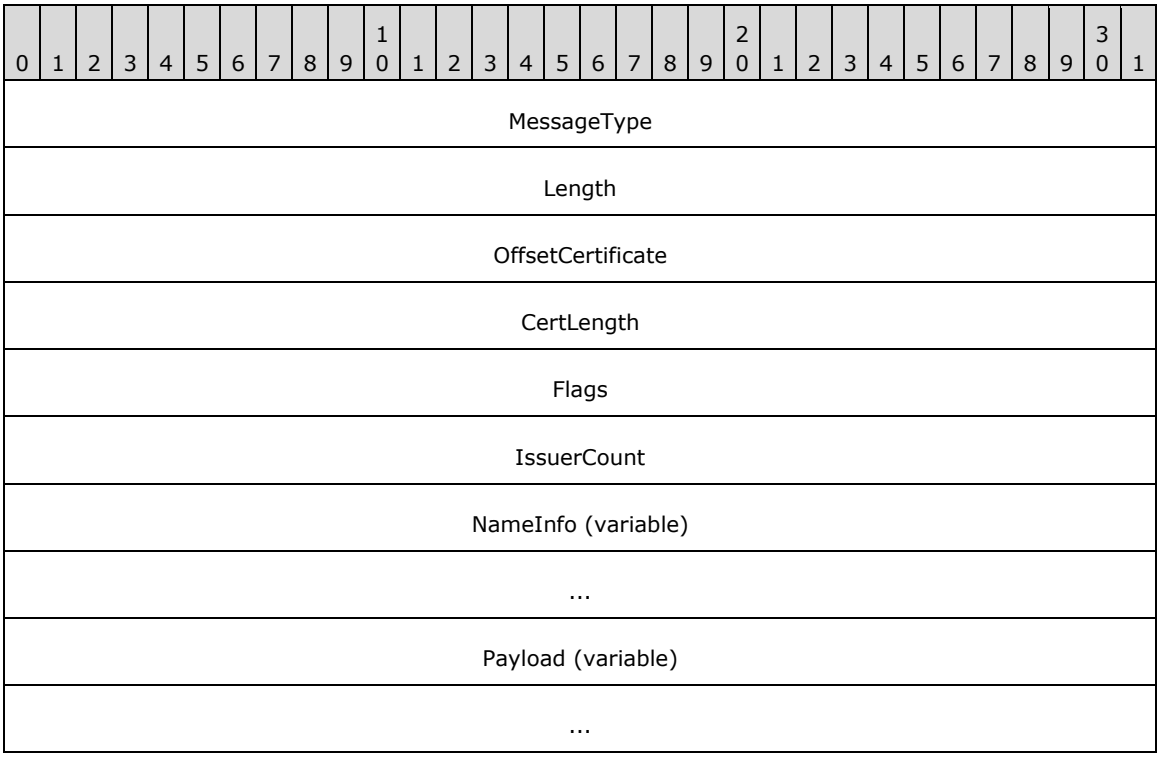
The Remote Certificate Mapping Protocol messages are embedded in Netlogon Remote Protocol messages in the logon interface. As a result, the Remote Certificate Mapping Protocol uses the Netlogon **RPC transport**, as specified in [MS-NRPC] section 2.1.

2.2 Message Syntax

Remote Certificate Mapping Protocol messages are encoded as opaque Binary Large Objects (BLOB) and transported by the generic pass-through capability of the Netlogon Remote Protocol, as specified in [MS-NRPC] section 3.2.4.1.

2.2.1 SSL_CERT_LOGON_REQ Message

The SSL_CERT_LOGON_REQ structure defines a request to map a client certificate to a security principal for the purpose of retrieving the authorization information. All member fields MUST be encoded in little-endian format.



MessageType (4 bytes): A 32-bit unsigned integer that defines the Remote Certificate Mapping Protocol message type. This member MUST be 0x00000002.

Length (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of the SSL_CERT_LOGON_REQ request message, including the variable length **NameInfo** and **Payload** sections.

OffsetCertificate (4 bytes): A 32-bit unsigned integer that defines the offset, in bytes, from the beginning of the SSL_CERT_LOGON_REQ request structure to the X.509 certificate, as specified in [X509], in the **Payload** member.

CertLength (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of the X.509 certificate in the **Payload** member.

Flags (4 bytes): A 32-bit unsigned integer that defines mapping behaviors. The value of this member is any combination of the flags as specified in the following diagram.

All other bits MUST be set to 0 by the Remote Certificate Mapping Protocol client and ignored on receipt.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D	C	B	A	0	0	0	0

Where the bits are defined as:

Value	Description
A REQ_UPN_MAPPING	When set, this indicates that the Remote Certificate Mapping Protocol client requests the Remote Certificate Mapping Protocol server to use the **subjectAltName** from the X.509 certificate in the **Payload** member to locate the authorization information, as specified in section 3.5. If not set, the **subjectAltName** extension SHOULD NOT be used during the lookup operation.
B REQ_SUBJECT_MAPPING	When set, the Remote Certificate Mapping Protocol client requests the Remote Certificate Mapping Protocol server to use the **issuer** and **subject** names from the X.509 certificate in the **Payload** member together to locate the authorization information, as specified in section 3.5. If not set, the **issuer** and **subject** fields SHOULD NOT be used during the lookup operation.
C REQ_ISSUER_MAPPING	When set, the Remote Certificate Mapping Protocol client requests the Remote Certificate Mapping Protocol server to use the **issuer** from the X.509 certificate in the **Payload** member to locate the authorization information, as specified in section 3.5. If not set, the issuer name SHOULD NOT be used during the lookup operation.
D REQ_ISSUER_CHAIN_MAPPING	When set, the Remote Certificate Mapping Protocol client requests the Remote Certificate Mapping Protocol server to use the chain of issuing authorities for the X.509 certificate in the **Payload** member to locate the authorization information, as specified in section 3.5. If not set, the chain of issuers SHOULD NOT be used during the lookup operation.

IssuerCount (4 bytes): A 32-bit unsigned integer that defines the number of **NameInfo** elements.

NameInfo (variable): An array of **IssuerOffset** and **IssuerLength** pairs, as defined in the following diagram. The issuers MUST be in the same order as the chain of issuing authorities for the X.509 certificate in the **Payload** section. That is, if the certificate was issued by A, and certificate authority A was in turn issued by B, the order would be A B.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
IssuerOffset																															
IssuerLength																															

IssuerOffset (4 bytes): A 32-bit unsigned integer that defines the byte offset from the start of the packet to an **IssuerName** in the **Payload** member.

IssuerLength (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of an **IssuerName** in the **Payload** member.

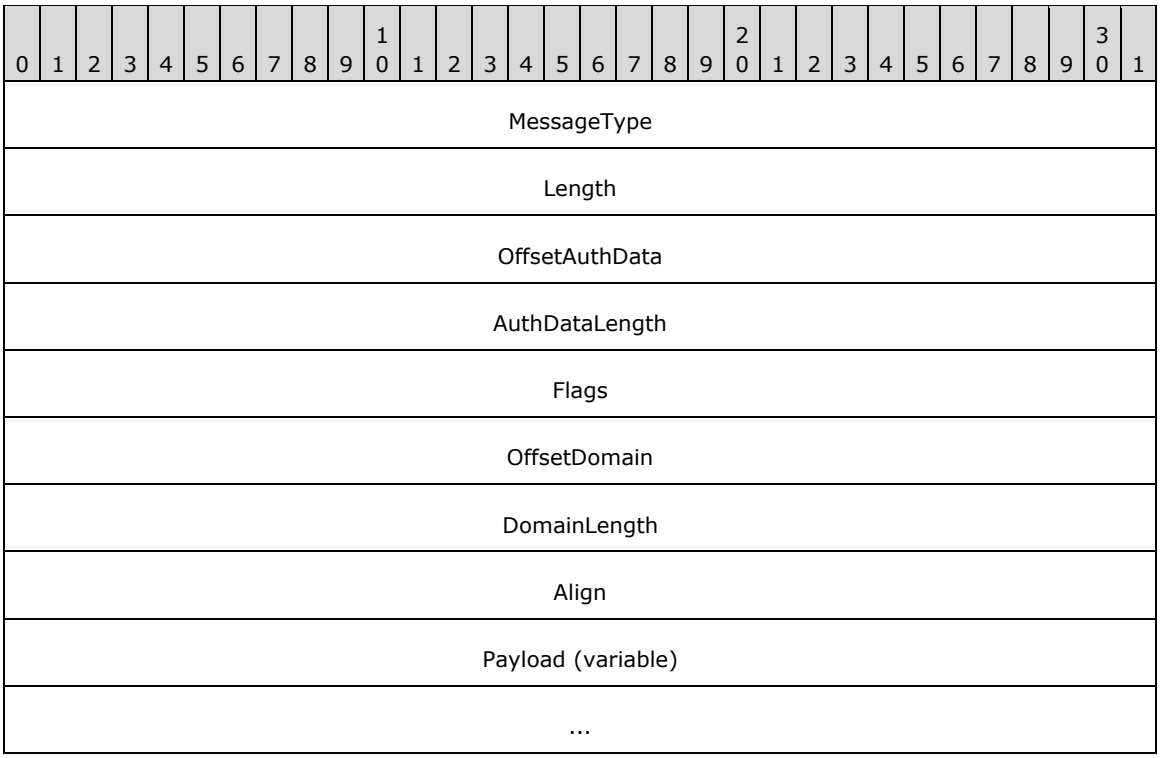
Payload (variable): A byte-array that contains the data referred to by the **OffsetCertificate** and **IssuerOffset** members. The **IssuerName** members in the **Payload** section has no guaranteed order; order is defined by the **NameInfo** array listed previously. Thus, the data may be packed into the buffer as "Issuer1, Issuer3, Certificate, Issuer2", but the **NameInfo** array would list them as "Issuer1, Issuer2, Issuer3." The actual order is specified in section 3.5.1. The number of issuer names encoded into the **Payload** section is determined by the **IssuerCount** member. Each **IssuerName** MUST be 2-byte aligned.

Certificate: The client's BER-encoded X.509 certificate referred to by the **OffsetCertificate** member. The format of an X.509 certificate is specified in ASN.1 per the X.509 standard, as specified in [X509]. BER encoding is specified in [X690].

IssuerName: The BER-encoded certificate issuer name referred to by an **IssuerOffset**. Each **IssuerName** corresponds to the **issuerName** member of an X.509 certificate in the certificate chain, as specified in [X509]. Only the issuer name is present, not the complete issuer certificate.

2.2.2 SSL_CERT_LOGON_RESP Message

The SSL_CERT_LOGON_RESP structure defines a successful response to an SSL_CERT_LOGON_REQ request. It contains the PAC that is returned to the caller. All member fields MUST be encoded in little-endian order.



MessageType (4 bytes): A 32-bit unsigned integer that defines the Remote Certificate Mapping Protocol message type. This member MUST be 0x00000002, matching SSL_CERT_LOGON_REQ.

Length (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of the SSL_CERT_LOGON_RESP response structure, including the variable **Payload** section.

OffsetAuthData (4 bytes): A 32-bit unsigned integer that defines the offset, in bytes, from the beginning of the SSL_CERT_LOGON_RESP response structure to the PAC, as specified in [MS-PAC], contained in the **Payload** field. This MUST be aligned to an 8-byte boundary.

AuthDataLength (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of the PAC, as specified in [MS-PAC], contained in the **Payload** field.

Flags (4 bytes): A 32-bit unsigned integer that MUST be 0, and ignored upon receipt. This field was intended for future expansion but was not used.

OffsetDomain (4 bytes): A 32-bit unsigned integer that defines the offset, in bytes, from the beginning of the SSL_CERT_LOGON_RESP request structure to a string of 16-bit **Unicode** characters comprising the name of the **domain** used for retrieving the authorization information. The domain name MUST be the NetBIOS name of the account domain.

DomainLength (4 bytes): A 32-bit unsigned integer that defines the length, in bytes, of the domain name referred to by the **OffsetDomain** member. The length does not include any trailing NULL character; because the string is counted, there need not be a trailing NULL.

Align (4 bytes): A 32-bit unsigned integer used to maintain 64-bit alignment. This member MUST be 0x00000000.

Payload (variable): This field contains the PAC, as specified in [MS-PAC], referred to by the **OffsetAuthData** field, and the domain name referred to by the **OffsetDomain** field.

2.3 Constants

The following constants are used in this specification.

Symbolic Name	Value	Definition
STATUS_LOGON_FAILURE	0xC000006D	A logon failure occurred.

2.4 Directory Service Schema Elements

The Remote Certificate Mapping Protocol (RCMP) accesses the Directory Service schema classes and attributes listed in the following table.

For the syntactic specifications of the following **<Class>** or **<Class><Attribute>** pairs, refer to [MS-ADTS], [MS-ADA1], [MS-ADA3].

Class	Attribute
computer	servicePrincipalName userPrincipalName
user	altSecurityIdentities servicePrincipalName userPrincipalName

3 Protocol Details

The Remote Certificate Mapping Protocol utilizes the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1, using Microsoft Unified Security Protocol Provider. The exchanged messages are SSL_CERT_LOGON_REQ and SSL_CERT_LOGON_RESP. When the account is found, the associated authorization data (for example, group memberships) is encoded as a PAC, as specified in [MS-PAC], and sent back to the Remote Certificate Mapping Protocol client. If no matching account is found, an error is returned to the client, as specified in section 3.5.2.

3.1 Abstract Data Model

The Remote Certificate Mapping Protocol requires that the server have available to it a database or directory of accounts with authorization information and associated name strings that will be used to query the database. The server will issue queries against this database based on strings extracted from the X.509 certificate.

It should be noted that a degenerate, but legal, server could map any certificate to a single set of authorization data. Or, all certificates could map to a small set of authorization data. For example, a web server could have three levels of service (bronze, silver, and gold) managed by three certificate issuers; the Remote Certificate Mapping Protocol server would then merely map the certificates based on the issuer to one of three possible authorization levels and dispense with a full database.

3.2 Timers

There are no timers for the Remote Certificate Mapping Protocol.

3.3 Initialization

There is no initialization that is specific to the Remote Certificate Mapping Protocol.

3.4 Higher-Layer Triggered Events

The Remote Certificate Mapping Protocol message exchange is triggered by a Remote Certificate Mapping Protocol client that requires user authentication via an X.509 certificate. After this authentication takes place, the Remote Certificate Mapping Protocol client sends the SSL_CERT_LOGON_REQ message to the Remote Certificate Mapping Protocol server to obtain authorization information.

3.5 Processing Events and Sequencing Rules

The Remote Certificate Mapping Protocol in itself is a stateless protocol with request/response semantics. The general model is:

- The Remote Certificate Mapping Protocol client MUST determine the validity of the certificate by whatever means appropriate to the Remote Certificate Mapping Protocol client when the Remote Certificate Mapping Protocol is used to obtain a **principal's** authorization information on the basis of which access control is performed. The Remote Certificate Mapping Protocol server has three mechanisms, as specified in sections 3.5.1, 3.5.2, and 3.5.3, for determining the authorization information.
- Upon receiving the SSL_CERT_LOGON_REQ message, if the Remote Certificate Mapping Protocol server is able to map the user's X.509 certificate to a particular account and authorization information, it MUST send an SSL_CERT_LOGON_RESP message to the Remote Certificate Mapping Protocol client. Otherwise, it MUST return an error status in the Netlogon generic passthrough function, as specified in [MS-NRPC] section 3.2.4.1.

3.5.1 Client Generation of SSL_CERT_LOGON_REQ Message

The client constructs the SSL_CERT_LOGON_REQ message by setting the user's X.509 certificate, the mapping method by which the server looks up the user's account (expressed via Flags as specified in section 2.2.1) and the X.509 certificate issuing authorities (expressed via **PayLoad** as specified in section 2.2.1). The issuing authorities are set in anchor last order. Anchor last order is defined as the leaf certification authority that issued the client's X.509 certificate is first, followed by the next certification authority in the certificate chain, and the next certification authority, and so on. The name of the root certification authority SHOULD be included in the SSL_CERT_LOGON_REQ message when the user's certificate has been directly issued by the root certification authority.

The Remote Certificate Mapping Protocol client request message is packed as a contiguous buffer and the encoded data is sent in the **LogonData** field in the NETLOGON_GENERIC_INFO structure, as specified in [MS-NRPC] section 2.2.1.4.2, via the generic passthrough capability of Netlogon, as specified in [MS-NRPC] section 3.2.4.1. The **PackageName** field in the NETLOGON_GENERIC_INFO structure, as specified in [MS-NRPC], MUST be a RPC_UNICODE_STRING structure with the string value being "Microsoft Unified Security Protocol Provider".

3.5.2 Server Processing of SSL_CERT_LOGON_REQ Message

Upon receipt of the SSL_CERT_LOGON_REQ message at the server, the server decodes the request. The server MUST examine the requested flags from the client for the **REQ_UPN_MAPPING**, **REQ_SUBJECT_MAPPING**, and **REQ_ISSUER_MAPPING** flags. These correspond to the following methods:

- Method 1: Mapping via the **userPrincipalName** attribute. The Remote Certificate Mapping Protocol client requests this mapping scheme from the Remote Certificate Mapping Protocol server by setting the **REQ_UPN_MAPPING** flag in the SSL_CERT_LOGON_REQ message. If this mapping scheme is allowed by the Remote Certificate Mapping Protocol server's local policy, the Remote Certificate Mapping Protocol server looks up the authorization information by using the **subjectAltName** field, as specified in [X509], contained in the X.509 certificate in the request.

  The DC SHOULD perform the following based on the type of certificates in the request:

  - For certificates that contain the FQDN (dNSName) in the Subject Alternative Extension ([RFC5280] section 4.2.1.6).

    The DC prefixes the FQDN with "host/" to form the **service principal name (SPN)** name form "host/machinename" and searches within the forest directory for an account that contains this SPN in the servicePrincipalName attribute ([MS-ADA3] section 2.253).

  - For certificates that contain the UPN in the Subject Alternative Extension ([RFC5280] section 4.2.1.6).

    In X.509 certificates, the UPN is encoded in the subject alternative name extension with object identifier (OID) 1.3.6.1.4.1.311.20.2.3. The character encoding is in UTF8 format if the characters are not U.S. ASCII characters. Details are specified in Appendix C of [RFC4556].

    The DC searches within the forest directory for an account containing the UPN in the userPrincipalName attribute ([MS-ADA3] section 2.349).

    If successful, the Remote Certificate Mapping Protocol RCMP server constructs a PAC [MS-PAC], containing the authorization information. For more information about the initial population of the PAC structures, see the sections under [MS-KILE] section 3.3.5.6.3.

- Method 2: Mapping via the certificate's subject and issuer **distinguished names (DNs)**. The Remote Certificate Mapping Protocol client requests this mapping scheme from the Remote Certificate Mapping Protocol server, by setting the **REQ_SUBJECT_MAPPING** flag in the SSL_CERT_LOGON_REQ message. If this mapping scheme is allowed by the Remote Certificate

Mapping Protocol server's local policy, the Remote Certificate Mapping Protocol server looks up the authorization information by using the subject name and issuer name contained in the X.509 certificate in the request. To match strings, the GetWindowsSortKey algorithm (section 3.1.5.2.4) [MS-UCODEREF] with the following flags **NORM_IGNORECASE**, **NORM_IGNOREKANATYPE**, **NORM_IGNORENONSPACE** and **NORM_IGNOREWIDTH** SHOULD be used, then the CompareSortKey algorithm (section 3.1.5.2.2) ([MS-UCODEREF]) SHOULD be used to compare strings. If successful, the Remote Certificate Mapping Protocol server constructs a PAC [MS-PAC], containing the authorization information. For more information about the initial population of the PAC structures, see the sections under [MS-KILE] section 3.3.5.6.3.<2>

- Method 3: Mapping via the certificate's issuer DN. The Remote Certificate Mapping Protocol client requests this mapping scheme from the Remote Certificate Mapping Protocol server, by setting the **REQ_ISSUER_MAPPING** flag in the SSL_CERT_LOGON_REQ message. If this mapping scheme is allowed by the Remote Certificate Mapping Protocol server's local policy, the Remote Certificate Mapping Protocol server looks up the account by using the issuer name that is contained in the X.509 certificate in the request. If the additional **REQ_ISSUER_CHAIN_MAPPING** flag is set, the other issuer names from the SSL_CERT_LOGON_REQ message are also used for the search. Each name from the chain of issuers need to be used as the lookup key until a match is found, in the order from the SSL_CERT_LOGON_REQ message. If successful, the Remote Certificate Mapping Protocol server constructs a PAC [MS-PAC], containing the authorization information. For more information about the initial population of the PAC structures, see the sections under [MS-KILE] section 3.3.5.6.3.<3>

- The server SHOULD try these methods in order as listed in the previous methods; a client MUST NOT rely on the processing order. If none of the methods specified as acceptable by the client can determine the appropriate account to use, the mapping request cannot be satisfied. In this event, there is no SSL_CERT_LOGON_RESP message constructed, and the Netlogon generic passthrough method, as specified in [MS-NRPC] section 3.2.4.1, MUST return error code STATUS_LOGON_FAILURE (0xC000006D), as specified in [MS-ERREF], indicating this failure condition. There is no specific error frame or status code in the SSL_CERT_LOGON_RESP message.

If none of the requested methods are successful, the server does not generate an SSL_CERT_LOGON_RESP message, and instead only returns the error code STATUS_LOGON_FAILURE (0xC000006D) to the client via the return code of the Netlogon generic pass-through, as specified in [MS-NRPC] section 3.2.4.1.

3.5.3 Server Generation of the SSL_CERT_LOGON_RESP Message

The SSL_CERT_LOGON_RESP message is constructed by the server in the event that the certificate was associated successfully with an account, and authorization information can be retrieved. The Remote Certificate Mapping Protocol server constructs a PAC, as specified in [MS-PAC], containing the authorization information. Also, if the Remote Certificate Mapping Protocol server is not authoritative over the user's account, it uses Remote Certificate Mapping Protocol to contact the Remote Certificate Mapping Protocol server that is authoritative, and asks it to build the PAC. The response message supplies the name of the domain that contained the account used as the source of the authorization information.

The response, SSL_CERT_LOGON_RESP (section 2.2.2), is packed as a contiguous buffer and the encoded data is sent in the **LogonData** field in the NETLOGON_GENERIC_INFO structure, as specified in [MS-NRPC] section 2.2.1.4.2.

3.6 Timer Events

There are no timer events for the Remote Certificate Mapping Protocol. All timer events are associated with the Netlogon Remote Protocol specified in [MS-NRPC], which serves as the transport for Remote Certificate Mapping Protocol messages.

3.7 Other Local Events

There are no other local events that affect the operation of this protocol.

4 Protocol Examples

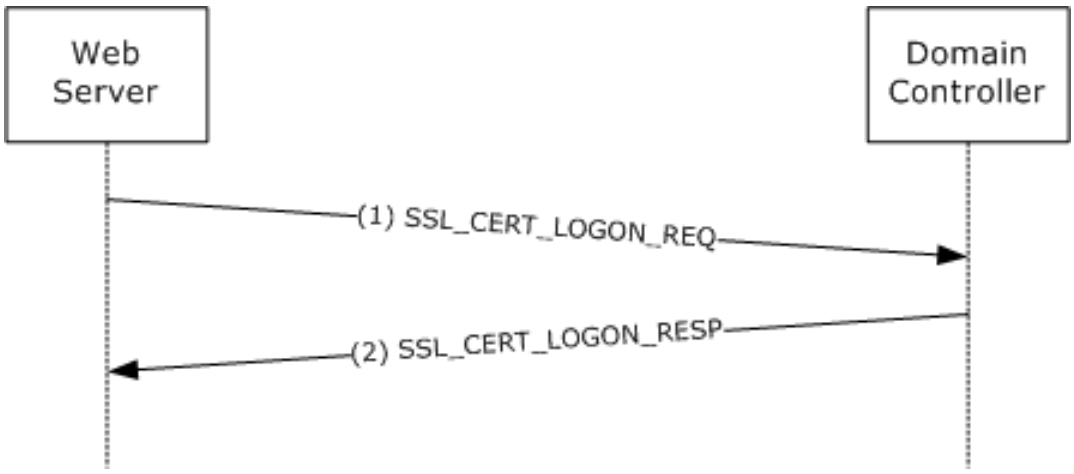


Figure 2: Obtaining a PAC that corresponds to an X.509 certificate

1. A web server requires clients to authenticate via an X.509 certificate. During the Transport Layer Security (TLS) handshake, the client sends the user's X.509 certificate to the server and proves knowledge of the corresponding private key. On completing the handshake, the server side of the TLS implementation builds the SSL_CERT_LOGON_REQ message (which contains the user's X.509 certificate) and sends it to the Remote Certificate Mapping Protocol server, in this example, located on a **domain controller**.

2. The Remote Certificate Mapping Protocol server on the domain controller parses the incoming request and uses the X.509 certificate attributes to look up the user's account in **Active Directory**. On a successful lookup, the domain controller generates the SSL_CERT_LOGON_RESP message, which includes the user's PAC, as specified in [MS-PAC], and sends the message back via the protocol described in [MS-NRPC]. On receiving this message, the server will generate a Windows access token ([MS-DTYP] section 2.5.2) for the client, which it can then use to access resources on the user's behalf.

5 Security

5.1 Security Considerations for Implementers

The Remote Certificate Mapping Protocol enables a user with an X.509 certificate and corresponding private key to gain access to resources based on group information associated with a given Active Directory account. Prior to performing the Remote Certificate Mapping Protocol, the Remote Certificate Mapping Protocol client has to first authenticate the user using the X.509 certificate because the authorization information returned by the Remote Certificate Mapping Protocol server enables the user to gain access to various resources.

The Remote Certificate Mapping Protocol itself does not have any built-in security mechanisms to provide authentication and assure the confidentiality and integrity of the Remote Certificate Mapping Protocol client/Remote Certificate Mapping Protocol server message exchange. Instead, it relies on security mechanisms, as specified in [MS-RPCE], used to protect Netlogon **remote procedure call (RPC)**, as specified in [MS-NRPC], that transport Remote Certificate Mapping Protocol request/response messages.

5.2 Index of Security Parameters

There are no security parameters for the Remote Certificate Mapping Protocol. All associated security parameters are described in [MS-NRPC], which provides all security for Remote Certificate Mapping Protocol messages.

6 Appendix A: Product Behavior

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

Note: Some of the information in this section is subject to change because it applies to an unreleased, preliminary version of the Windows Server operating system, and thus may differ from the final version of the server software when released. All behavior notes that pertain to the unreleased, preliminary version of the Windows Server operating system contain specific references to Windows Server 2016 Technical Preview as an aid to the reader.

- Windows NT operating system
- Windows 2000 operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- Windows 10 operating system
- Windows Server 2016 Technical Preview operating system

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 1.5: Windows 2000 and subsequent versions of Windows, according to the applicability list at the beginning of this section, use Active Directory as the database for the authorization. Active Directory uses the distinguished name (DN) of the security principal **object** and the **userPrincipalName** and **altSecurityIdentities** attributes on the security principal objects for establishing the relationship between the certificates and the authorization information, as specified in [MS-ADTS].

<2> Section 3.5.2: The Windows 2000 domain controller and all subsequent versions of Windows domain controllers, according to the applicability list at the beginning of this section, extract the issuer DN and the subject DN from the X.509 certificate, and constructs the following string: "<I>[value of issuer]<S>[value of the subject DN]". The resulting string is evaluated against the

altSecurityIdentities attribute of all user and machine account objects, and the scope of this evaluation is the entire forest. The altSecurityIdentities attribute is a multiple-value attribute.

<3> Section 3.5.2: The Windows 2000 domain controller and all subsequent versions of Windows domain controllers, according to the applicability list at the beginning of this section, extract the issuer DN from the X.509 certificate, and constructs the following string: "<I>[value of issuer]". The resulting string is evaluated against the altSecurityIdentities attribute of all user and machine account objects, and the scope of this evaluation is the entire forest. Note that the altSecurityIdentities attribute is a multiple-value attribute.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- The removal of a document from the documentation set.

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 **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix A: Product Behavior	Updated product applicability list to include Windows 10 operating system.	Y	Content update.

8 Index

T

V