

[MS-RDPEWA]:

Remote Desktop Protocol: WebAuthn Virtual Channel Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
9/3/2022	1.0	New	Released new document.
11/8/2022	1.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	6
1.3	Overview	6
1.4	Relationship to Other Protocols	6
1.5	Prerequisites/Preconditions	6
1.6	Applicability Statement	6
1.7	Versioning and Capability Negotiation	6
1.8	Vendor-Extensible Fields	6
1.9	Standards Assignments.....	7
2	Messages.....	8
2.1	Transport	8
2.2	Message Syntax.....	8
2.2.1	WebAuthN_Channel Request Message	8
2.2.1.1	webAuthNPara Map	10
2.2.1.2	CTAPCBOR_CMD_MAKE_CREDENTIAL Request	12
2.2.1.3	CTAPCBOR_CMD_GET_ASSERTION Request	12
2.2.2	WebAuthN_Channel Response Message	13
2.2.2.1	CTAPCBOR_RPC_COMMAND_WEB_AUTHN Response Map	13
2.2.2.1.1	CTAP MakeCredential Response.....	15
2.2.2.1.2	CTAP GetAssertion Response	15
3	Protocol Details.....	17
3.1	Client and Server Details.....	17
3.1.1	Abstract Data Model.....	17
3.1.2	Timers	17
3.1.3	Initialization	17
3.1.4	Higher-Layer Triggered Events	17
3.1.5	Message Processing Events and Sequencing Rules	17
3.1.6	Timer Events.....	17
3.1.7	Other Local Events.....	17
4	Protocol Examples	18
4.1	CTAPCBOR_RPC_COMMAND_API_VERSION	18
4.1.1	Request.....	18
4.1.2	Response.....	18
4.2	CTAPCBOR_RPC_COMMAND_IUVPAA	18
4.2.1	Request.....	18
4.2.2	Response.....	18
4.3	CTAPCBOR_RPC_COMMAND_CANCEL_CUR_OP	19
4.3.1	Request.....	19
4.3.2	Response.....	19
4.4	CTAPCBOR_RPC_COMMAND_WEB_AUTHN	19
4.4.1	CTAPCBOR_CMD_MAKE_CREDENTIAL	19
4.4.1.1	Request	19
4.4.1.2	Response	20
4.4.2	CTAPCBOR_CMD_GET_ASSERTION	22
4.4.2.1	Request	22
4.4.2.2	Response	23
5	Security	24
5.1	Security Considerations for Implementers	24
5.2	Index of Security Parameters	24

6	Appendix A: Product Behavior	25
7	Change Tracking.....	26
8	Index.....	27

1 Introduction

The Remote Desktop Protocol (RDP): WebAuthn Virtual Channel Protocol provides a way for a user to do WebAuthn operations over the RDP protocol. It enables a server to send webauthn request to a client, the client can then use this request to talk to authenticators (platform as well as cross-platform) and reply with the response.

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

1.1 Glossary

This document uses the following terms:

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [\[RFC4122\]](#) or [\[C706\]](#) must be used for generating the **GUID**. See also universally unique identifier (UUID).

Remote Desktop Protocol (RDP): A multi-channel protocol that allows a user to connect to a computer running Microsoft Terminal Services (TS). RDP enables the exchange of client and server settings and also enables negotiation of common settings to use for the duration of the connection, so that input, graphics, and other data can be exchanged and processed between client and server.

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.

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.

[FIDO-CTAP] Brand, C., Czeskis, A., Ehrensvärd, J. et al. Eds., "Client to Authenticator Protocol (CTAP)", <https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.html>

[IETF-8949] Hoffman, P., "Concise Binary Object Representation (CBOR)", <https://www.ietf.org/rfc/rfc8949.txt>

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

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

[W3C-WebAuthnPKC2] Microsoft Corporation, "Web Authentication: An API for accessing Public Key Credentials Level 2", <https://www.w3.org/TR/webauthn-2/>

1.2.2 Informative References

[MSFT-WebAuthnAPIS] Microsoft Corporation, "Microsoft WebAuthn APIs", <https://github.com/microsoft/webauthn/blob/master/webauthn.h>

1.3 Overview

The Remote Desktop Protocol: WebAuthn Virtual Channel provides a way for a user to do WebAuthn operations over the RDP protocol.

More details about WebAuthn can be found in [\[W3C-WebAuthnPKC2\]](#).

The WebAuthn javascript API is handled by the browser. The browser in turn calls the system-provided APIs on Windows. The system then establishes a virtual channel to the RDP client and sends the request to it. On the RDP client side, the request is decoded and processed by talking to available authenticators via the Client-to-Authenticator protocol (CTAP) protocol. See [\[FIDO-CTAP\]](#) for more details about the CTAP protocol.

1.4 Relationship to Other Protocols

This protocol uses the [\[W3C-WebAuthnPKC2\]](#) and [\[FIDO-CTAP\]](#) protocols.

1.5 Prerequisites/Preconditions

The Remote Desktop Protocol: WebAuthn Virtual Channel operates only after the dynamic virtual channel transport is fully established.

This protocol is message-based. It assumes preservation of the packet as a whole and does not allow for fragmentation. Additionally, it assumes that no packets are lost.

1.6 Applicability Statement

This protocol is designed to be run within the context of a **Remote Desktop Protocol (RDP)** virtual channel established between a client and a server.

1.7 Versioning and Capability Negotiation

This protocol supports versioning and capability negotiation as part of the request. A client that supports this protocol does allow this virtual channel to be opened, and a client that does not support this protocol does not allow this virtual channel to be opened.

1.8 Vendor-Extensible Fields

This protocol also uses Win32 error codes. These values are taken from the error number space as specified in [\[MS-ERREF\]](#) section 2.2. Vendors SHOULD reuse those values with their indicated meanings. Choosing any other value runs the risk of a collision in the future.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The protocol uses a channel named WebAuthN_Channel. This channel MUST be implemented using a reliable protocol, such as **TCP**. Messages written to this channel are assumed to be complete and to arrive in order.

2.2 Message Syntax

Requests and responses are encoded in Concise Binary Object Representation (CBOR) format. For more details about CBOR, see [\[IETF-8949\]](#). CBOR encoding is used because it is used in the CTAP protocol ([\[FIDO-CTAP\]](#)) to access security keys. Hence clients needing to use external security keys need to use CBOR encoding. Platform authenticators provide their own APIs to talk to use their implementations. Overall, the WebAuthn request from the relying party is encoded along with metadata about the operation in the request message.

This protocol has two messages, a request message and a response message. The two messages perform different operations depending on their content. Each message is a CBOR map (see [\[IETF-8949\]](#), section 3.1, "Major Types," for a description of a map). The messages themselves, depending on the type of request or response, will in turn contain additional maps.

The next two sections describe the request and response messages and their elements.

2.2.1 WebAuthN_Channel Request Message

The WebAuthN_Channel request message is a CBOR map using the following keys and values.

command (key type: text string (major type 3)): An unsigned integer (major type 0) indicating the RPC command type. One of the following values:

Value	Meaning
CTAPCBOR_RPC_COMMAND_WEB_AUTHN 5	Contains both registration and assertion request for the platform authenticator as well as security keys.
CTAPCBOR_RPC_COMMAND_IUVPAA 6	Corresponds to the WebAuthn IsUserVerifyingPlatformAuthenticatorAvailable API. See [W3C-WebAuthPKC2] , section 5.1.7.
CTAPCBOR_RPC_COMMAND_CANCEL_CUR_OP 7	Cancel the current webauthn request.
CTAPCBOR_RPC_COMMAND_API_VERSION 8	Get the platform authenticator API version. <1> Callers can use the version to identify what features are available on the OS so that caller can decide whether or not a request can be fulfilled.

request (key type: text string (major type 3)): A byte string (major type 2) containing details about the request. The contents vary depending on the **command** type.

For CTAPCBOR_RPC_COMMAND_API_VERSION and CTAPCBOR_RPC_COMMAND_IUVPAA, this field is not present.

For CTAPCBOR_RPC_COMMAND_CANCEL_CUR_OP, this field contains a GUID representing the current operation.

For CTAPCBOR_RPC_COMMAND_WEB_AUTHN, first byte contains the WebAuthn command type:

Value	Meaning
CTAPCBOR_CMD_MAKE_CREDENTIAL 0x01	This command is used to create a new credential for an account for a relying party (registration phase). This is done once per account.
CTAPCBOR_CMD_GET_ASSERTION 0x02	Used to authenticate the user and sign the client data using the key created previously during the registration phase. This is also called the authenticate phase. The command is exercised multiple times after the registration phase.

The second and following bytes contain a CBOR map corresponding to the WebAuthn command type in the preceding table. See sections [2.2.1.2](#) and section [2.2.1.3](#).

flags (key type: text string (major type 3)): An unsigned integer (major type 0) containing details about the request. The value is an exclusive or (XOR) of the following values:

Value	Meaning
CTAPCLT_U2F_FLAG 0x00020000	Set to indicate the request and response will use U2F. The provider should use the U2F device interface instead of the CTAP interface.
CTAPCLT_DUAL_FLAG 0x00040000	Set to indicate to first try CTAP messages and protocol. If CTAP fails, use U2F messages..
CTAPCLT_CLIENT_PIN_REQUIRED_FLAG 0x00100000	Set to force the use of a client pin for CTAPCBOR_CMD_MAKE_CREDENTIAL.
CTAPCLT_SELECT_CREDENTIAL_ALLOW_UV_FLAG 0x00080000	When set for a login get assertion, allows user verification (UV) get assertions to select the credential.
CTAPCLT_UV_REQUIRED_FLAG 0x00400000	Set to require user verification.
CTAPCLT_UV_PREFERRED_FLAG 0x00800000	Set to indicate user verification is preferred.
CTAPCLT_UV_NOT_REQUIRED_FLAG	Indicates that user verification is not required.

Value	Meaning
0x01000000	
CTAPCLT_HMAC_SECRET_EXTENSION_FLAG 0x04000000	Set to enable the hmac-secret extension for a CTAPCBOR_CMD_MAKE_CREDENTIAL request.
CTAPCLT_FORCE_U2F_V2_FLAG 0x08000000	Set to force the U2F version 2 interface to be used.

timeout (key type: text string (major type 3)): An unsigned integer (major type 0) representing the timeout (in milliseconds) for the operation.

transactionid (key type: text string (major type 3)): A byte array (major type 2); a **GUID** that is the transaction identifier.

webAuthNPara (key type: text string (major type 3)): A CBOR map (major type 5) providing parameters for authentication. See section [2.2.1.1](#) for details of the map.

2.2.1.1 webAuthNPara Map

The **webAuthNPara** is used in the WebAuthN_Channel request message to specify the parameters to use for authentication. It has the following keys and values:

wnd (key type: text string (major type 3)): An unsigned integer (major type 0) that is the window handle for the caller.

attachment (key type: text string (major type 3)): An unsigned integer (major type 0) that indicates the authenticator applicable to this operation.

Value	Meaning
WEBAUTHN_AUTHENTICATOR_ATTACHMENT_ANY 0	Use any authenticator that can satisfy the request conditions.
WEBAUTHN_AUTHENTICATOR_ATTACHMENT_PLATFORM 1	Use the platform authenticator to satisfy the request conditions. <2>
WEBAUTHN_AUTHENTICATOR_ATTACHMENT_CROSS_PLATFORM 2	Use the cross-platform roaming authenticator, such as security keys or phones, to satisfy the request conditions.

requireResident (key type: text string (major type 3)): A true or false CBOR simple value (see [\[IETF-8949\]](#), section 3.3) indicating whether or not resident credential keys are required.

preferResident (key type: text string (major type 3)): A true or false CBOR simple value (see [\[IETF-8949\]](#), section 3.3) indicating whether or not resident credential keys are preferred.

userVerification (key type: text string (major type 3)): An unsigned integer (major type 0) that represents the verification requirements. One of the following values:

Value	Meaning
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_ANY 0	User verification is not required, and any setting is acceptable to the relying party.
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_REQUIRED 1	User verification is required by the relying party.
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_PREFERRED 2	User verification is preferred by the relying party.
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_DISCOURAGED 3	User verification is discouraged by the relying party.

attestationPreference (key type: text string (major type 3)): An unsigned integer (major type 0) indicating the preferred attestation method. One of the following values:

Value	Meaning
WEBAUTHN_ATTESTATION_CONVEYANCE_PREFERENCE_ANY 0	Use any attestation conveyance preference.
WEBAUTHN_ATTESTATION_CONVEYANCE_PREFERENCE_NONE 1	No preference among attestation conveyance methods.
WEBAUTHN_ATTESTATION_CONVEYANCE_PREFERENCE_INDIRECT 2	Prefer indirect attestation conveyance.
WEBAUTHN_ATTESTATION_CONVEYANCE_PREFERENCE_DIRECT 3	Prefer direct attestation conveyance.

enterpriseAttestation (key type: text string (major type 3)): An unsigned integer (major type 0) indicating the enterprise attestation to use. One of the following values:

Value	Meaning
WEBAUTHN_ENTERPRISE_ATTESTATION_NONE 0	Enterprise attestation is not requested by the relying party.
WEBAUTHN_ENTERPRISE_ATTESTATION_VENDOR_FACILITATED 1	Enterprise attestation is requested by the relying party and authenticator can provide it if configured with this relying party.
WEBAUTHN_ENTERPRISE_ATTESTATION_PLATFORM_MANAGED 2	Enterprise attestation is requested by the relying party and the platform (OS/browser) if configured with this relying

Value	Meaning
	party can allow such attestation.

cancellationId (key type: text string (major type 3)): A byte array (major type 2); a **GUID** that is the cancellation identifier.

2.2.1.2 CTAPCBOR_CMD_MAKE_CREDENTIAL Request

This is the CBOR map used in the WebAuthN_Channel request (section 2.2.1) for the CTAPCBOR_CMD_MAKE_CREDENTIAL command. The map contains details needed for the request as defined in [FIDO-CTAP], section 5.1.

The map has up to seven fields indicated by numeric keys. Nearly all of the fields are themselves CBOR maps. Some fields are optional.

Key	Field
1	Client data hash.
2	Relying party information.
3	User account information.
4	Algorithm preference.
5	Exclude list (optional).
6	Extensions (optional).
7	Options (optional).

The following is a text representation of an example request:

```
{1: h'BB2C6711064CF3BB8C34CD2EC06398AE4F2EF60852AE6D32391AA6312C9EE609', 2: {"id": "webauthntest.azurewebsites.net", "icon": "https://example.com/rpIcon.png", "name": "WebAuthn Test Server"}, 3: {"id": h'626F62406578616D706C652E636F6D', "icon": "https://example.com/userIcon.png", "name": "bob@example.com", "displayName": "Bob Smith"}, 4: [{"alg": -7, "type": "public-key"}, {"alg": -35, "type": "public-key"}, {"alg": -36, "type": "public-key"}, {"alg": -257, "type": "public-key"}, {"alg": -8, "type": "public-key"}], 6: {"credProtect": 2}, 7: {"rk": true}}
```

See [FIDO-CTAP], section 5.1, for details.

2.2.1.3 CTAPCBOR_CMD_GET_ASSERTION Request

This is the CBOR map used in the WebAuthN_Channel request (section 2.2.1) for the CTAPCBOR_CMD_GET_ASSERTION command. The map contains details needed for the request as defined in [FIDO-CTAP], section 5.2.

The map has up to five fields, indicated by numeric keys. Nearly all of the fields are themselves CBOR maps. Some fields are optional.

Key	Field
1	Relying party identifier.

Key	Field
2	Client data hash.
3	Credential include list (optional).
4	Extensions (optional).
5	Exclude list (optional).

The following is a text representation of an example request:

```
{1: "webauthntest.azurewebsites.net", 2:
h'71416126685DFB9B2776D1B26AD709605951061F3692A7AD025F919C4881FD39', 3: [{"id":
h'19308CB37A700C8454D6C914D48A95E187D71B71B64F9AE4ACA7D09C89BCD4F9', "type": "public-key",
"transports": 23}, {"id":
h'FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9
784BB', "type": "public-key", "transports": 23}], 5: {"up": true}}
```

See [FIDO-CTAP], section 5.2, for details.

2.2.2 WebAuthN_Channel Response Message

The WebAuthN_Channel response message is a 32-bit value followed by bytes containing data depending on the RPC command:

hresult (unsigned integer): A 32-bit HRESULT for the RPC command. See [MS-ERREF] section 2.1.

response (byte string): Variable length string of bytes depending on the RPC command. The following are the forms the **response** takes for a given RPC command:

RPC Command	Response Type
CTAPCBOR_RPC_COMMAND_WEB_AUTHN 5	A CBOR map (major type 5). See section 2.2.2.1 .
CTAPCBOR_RPC_COMMAND_IUVPA 6	A 4-byte Boolean value in little endian format. 1 for true and 0 for false.
CTAPCBOR_RPC_COMMAND_CANCEL_CUR_OP 7	No additional bytes.
CTAPCBOR_RPC_COMMAND_API_VERSION 8	A 4-byte unsigned integer in little endian format giving the API version.

2.2.2.1 CTAPCBOR_RPC_COMMAND_WEB_AUTHN Response Map

The WebAuthN_Channel response message is a CBOR map using the following keys and values.

deviceInfo (key type text string (major type 3)): A CBOR map (major type 5) containing information about the device. The map uses the following keys and values:

maxMsgSize (key type text string (major type 3)): An unsigned 32-bit integer (major type 0) providing the maximum message size the authenticator can process.

maxSerializedLargeBlobArray (key type text string (major type 3)): An unsigned 32-bit integer (major type 0) providing the maximum size serialized large blob the authenticator can process.

providerType (key type text string (major type 3)): A text string (major type 3) representing the provider type. Used only for information purposes. One of the following values:

Value	Meaning
CTAPHID_PROVIDER_TYPE L"Hid"	An Hid provider.
CTAPNFC_PROVIDER_TYPE L"Nfc"	An Nfc provider.
CTAPBLE_PROVIDER_TYPE L"Ble"	A Ble provider.
WEBAUTHN_PLATFORM_PROVIDER_TYPE L"Platform"	A platform provider.

providerName (key type text string (major type 3)): A text string (major type 3) representing the provider's name. Used only for information.

devicePath (key type text string (major type 3)): A text string (major type 3) providing the path to the authenticator.

Manufacturer (key type text string (major type 3)): A text string (major type 3) representing the manufacturer of the authenticator.

Product (key type text string (major type 3)): A text string (major type 3) representing the product name of the authenticator.

aaGuid (key type text string (major type 3)): A 16 byte byte-string (major type 2) containing the Authenticator Attestation GUID (AAGUID) for the authenticator.

residentKey (key type text string (major type 3)): A true or false CBOR simple value that indicates whether a resident credential was created.

uvStatus (key type text string (major type 3)): An unsigned integer (major type 0) giving the verification status of the operation. One of the following values:

Value	Meaning
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_ANY 0	Use any verification requirement.
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_REQUIRED 1	A verification is required.

Value	Meaning
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_PREFERRED 2	A verification is preferred.
WEBAUTHN_USER_VERIFICATION_REQUIREMENT_DISCOURAGED 3	Verification is discouraged.

uvRetries (key type text string (major type 3)): An unsigned integer (major type 0) representing the number of verification retries available for the authenticator.

Status (key type text string (major type 3)): An unsigned integer (major type 0) giving the status of the overall operation.

Response (key type text string (major type 3)): Contains the response to the individual operation corresponding to the individual WebAuthn command. The response consists of a single byte indicating success or failure. A value of 0x00 indicates success. Any other value is an error. See [\[FIDO-CTAP\]](#), section 6.3 for values and meaning. The single-byte code is followed by a CBOR map containing details of the response. See section [2.2.2.1.1](#) and section [2.2.2.1.2](#) for the response maps.

2.2.2.1.1 CTAP MakeCredential Response

This is the CBOR map used in the CTAPCBOR_RPC_COMMAND_WEB_AUTHN response map for a response to a request to make a credential (see section [2.2.2.1](#)).

See [\[FIDO-CTAP\]](#), section 6.2, for details of the map keys, values, and data types.

The following is a text representation of an example response:

```
{1: "packed", 2:
h'E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C2C500000003D8522D9F575B48668
8A9BA99FA02F35B0030FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D9
7F581B422370AC2C9784BBA5010203262001215820FFE2DC7BB7DFD9C8C268C45DD34383E21988D0F18870AB0E115
0278B8F8BEA742258202DE56C3BCFA03520409164829D13A8D4CCF82894FEF2D48BC75F0064F9DD1AFA16B637265
6450726F7465637402', 3: {"alg": -7, "sig":
h'304402203F97BADFB8FD74ECE32AE298E65FF32F930B5F1F430741C1FAC0BF91FA37029802200E2391E5BAE6620
85958C67B76FC7ED0483C6F4EAC14A8A2F9945E2041C38754', "x5c":
[h'308202D8308201C0A003020102020900FF876C2DAF7379C8300D06092A864886F70D01010B0500302E312C302A
0603550403132359756269636F2055324620526F6F742043412053657269616C203435373230303633313020170D3
134303830313030303030305A180F32303530303930343030303030305A306E310B30090603550406130253453112
3010060355040A0C0959756269636F20414231223020060355040B0C1941757468656E74696361746F72204174746
573746174696F6E3127302506035504030C1E59756269636F205532462045452053657269616C2037363230383734
32333059301306072A8648CE3D020106082A8648CE3D0301070342000425F123A048283FC5796CCF887D99489FD93
5C24198C4B5D8D5B2C2BFD7DD5D15AFE45B7070776567D5B5B0B23E04560B5BEA77B483B1F6491E53A3F2BEE6A39A
A38181307F3013060A2B0601040182C40A0D0104050403050506302206092B0601040182C40A020415312E332E362
E312E342E312E34313438322E312E393013060B2B0601040182E51C0201010404030205203021060B2B0601040182
E51C01010404120410D8522D9F575B486688A9BA99FA02F35B300C0603551D130101FF04023000300D06092A86488
6F70D01010B0500038201010052B06949DBAAD1A64C1BA9EBC198B317EC31F9A37363BA5161B342E3A49CAD504F34
E7428BB896E9CFD28D03AD10CE325A06838E9B6C4ECB17AD40D090A16C9E7C34498332FF853B62747E8FCDF00DAE6
2756E57BD40B16D677907A835C0435A2EBCE9B0B9069CA122BF9D964A73206AF74FF3C00144EBFF3DE7C7758D3147
C8C2F9FE87C12F2A9675A2046B01076361A99721871FA78FB0DE2945B579F9166C48AD2FD50C3CE56C8221A75083F
656119394368FF17D2C920C63A09F01ED2501146B7DF1AB3970A2A32938FA9A517AF471085E160B3CA79764231746
BA6ABBA68E0D13CE259796BCD2A03AD83C74E15331328EAB438E6A4197CB12EC6FD1E388' ] }
```

2.2.2.1.2 CTAP GetAssertion Response

This is the CBOR map used in the CTAPCBOR_RPC_COMMAND_WEB_AUTHN response map for a response to a request to get an assertion (see section [2.2.2.1](#)).

See [\[FIDO-CTAP\]](#), section 6.2, for details of the map keys, values, and data types.

The following is a text representation of an example response:

```
{1: {"id":  
h'FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9  
784BB', "type": "public-key"}, 2:  
h'E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C20500000009', 3:  
h'304602210099C54C2075B88279DE38944F5492E21DA2F5E1969BCCDD99A97F39BDDD844A78022100BF091FBE452  
42DF484D7396D5304A570A78F2E5576ECC8AB24107E51968AD1C4', 4: {"id":  
h'626F62406578616D706C652E636F6D'}}
```


3 Protocol Details

3.1 Client and Server Details

The protocol has two main operations: MakeCredential (section [2.2.1.2](#)) and GetAssertion (section [2.2.1.3](#)). The MakeCredential operation registers a credential on the authenticator for the replying party. The GetAssertion operation authenticates the user to the relying party using the authenticator.

This protocol is designed to be closer to the WebAuthn layer rather than the CTAP layer. Individual fields in the request map to the WebAuthn options defined at the WebAuthn layer.

Once a request is received by the client, the client determines which authenticator can satisfy the request. The process of determining which authenticator supports particular capabilities and whether it can satisfy a request are defined in the CTAP specification. See [\[FIDO-CTAP\]](#), sections 4 and 5.4.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following sections provide examples of requests and responses for different commands.

4.1 CTAPCBOR_RPC_COMMAND_API_VERSION

4.1.1 Request

Complete request (a CBOR map):

```
A467636F6D6D616E640565666C616773006774696D656F7574006D7472616E73616374696F6E4964500000000000  
00000000000000000000
```

Textual representation of the CBOR encoding:

```
{"command": 5, "flags": 0, "timeout": 0, "transactionId":  
h'00000000000000000000000000000000'}
```

4.1.2 Response

Full response including HRESULT and API Version:

```
0x00000004
```

The preceding response indicates a successful call (0x00000000 as first 32 bits) and that the API version is 4.

4.2 CTAPCBOR_RPC_COMMAND_IUVPAA

4.2.1 Request

Full request (a CBOR map):

```
0xA467636F6D6D616E640665666C616773006774696D656F7574006D7472616E73616374696F6E49645000000000  
00000000000000000000
```

Textual representation of the CBOR encoding:

```
{"command": 6, "flags": 0, "timeout": 0, "transactionId":  
h'00000000000000000000000000000000'}
```

4.2.2 Response

Full response including HRESULT and IUVPAA:

```
0x00000001
```


Textual representation of the response following the HRESULT:

```
{"deviceInfo": {"maxMsgSize": 1200, "maxSerializedLargeBlobArray": 1024, "providerType": "Hid", "providerName": "MicrosoftCtapHidProvider", "devicePath": "\\\\.\\hid#vid 1050&pid 0402&7&31130694&1&0000#{4d1e55b2-f16f-11cf-88cb-001111000030}", "manufacturer": "Yubico", "product": "YubiKey FIDO", "aaGuid": "h'9FD52D85B57664888A9BA99FA02F35B'", "residentKey": true, "uvStatus": 1, "uvRetries": 3}, "status": 0, "response": "h'00A301667061636B65640258C2E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C2C500000003D8522D9F575B486688A9BA99FA02F35B0030FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9784BBA5010203262001215820FFE2DC7BB7DFD9C8C268C45DD34383E21988D0F18870AB0E1150278B8F8BEA742258202DE56C3BCFA03520409164829D13A8D4CCF82894FEF2D48BC75F0064F9DDB1AFA16B6372656450726F746563740203A363616C6726637369675846304402203F97BADFB8FD74ECE32AE298E65FF32F930B5F1F430741C1FAC0BF91FA37029802200E2391E5BAE662085958C67B76FC7ED0483C6F4EAC14A8A2F9945E2041C3875463783563815902DC308202D8308201C0A003020102020900FF876C2DAF7379C8300D06092A864886F70D01010B0500302E312C302A0603550403132359756269636F2055324620526F6F742043412053657269616C203435373230303633313020170D3134303830313030303030305A180F323035303930343030303030305A306E310B300906035504061302534531123010060355040A0C0959756269636F20414231223020060355040B0C1941757468656E74696361746F72204174746573746174696F6E3127302506035504030C1E59756269636F205532462045452053657269616C203736323038373432333059301306072A8648CE3D020106082A8648CE3D0301070342006025F1D130101FF04023000300D06092A864886F70D01010B0500038201010052B06949DBAAD1A64C1BA9EBC198B317EC31F9A37363BA5161B342E3A49CAD504F34E7428BB896E9CFD28D03AD10CE325A06838E9B6C4ECB17AD40D090A16C9E7C34498332FF853B62747E8FCDF00DAE62756E57BD40B16D677907A835C0435A2EBCE9B0B9069CA122BF9D964A73206AF74FF3C00144EBFF3DE7C7758D3147C8C2F9FE87C12F2A9675A2046B01076361A99721871FA78FB0DE2945B579F9166C48AD2FD50C3CE56C8221A75083F656119394368FF17D2C920C63A09F01ED2501146B7DF1AB3970A2A32938FA9A517AF471085E160B3CA79764231746BA6ABBA68E0D13CE259796BCD2A03AD83C74E15331328EAB438E6A4197CB12EC6FD1E388'}
```

Inner Authenticator response after first byte indicating success(0x00):

```
A301667061636B65640258C2E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C2C500000003D8522D9F575B486688A9BA99FA02F35B0030FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9784BBA5010203262001215820FFE2DC7BB7DFD9C8C268C45DD34383E21988D0F18870AB0E1150278B8F8BEA742258202DE56C3BCFA03520409164829D13A8D4CCF82894FEF2D48BC75F0064F9DDB1AFA16B6372656450726F746563740203A363616C6726637369675846304402203F97BADFB8FD74ECE32AE298E65FF32F930B5F1F430741C1FAC0BF91FA37029802200E2391E5BAE662085958C67B76FC7ED0483C6F4EAC14A8A2F9945E2041C3875463783563815902DC308202D8308201C0A003020102020900FF876C2DAF7379C8300D06092A864886F70D01010B0500302E312C302A0603550403132359756269636F2055324620526F6F742043412053657269616C203435373230303633313020170D3134303830313030303030305A180F3230353039303430303030305A306E310B300906035504061302534531123010060355040A0C0959756269636F20414231223020060355040B0C1941757468656E74696361746F72204174746573746174696F6E3127302506035504030C1E59756269636F205532462045452053657269616C203736323038373432333059301306072A8648CE3D020106082A8648CE3D0301070342000425F123A048283FC5796CCF887D99489FD935C24198C4B5D8D5B2C2BFD7DD5D15AFE45B7070776567D5B5B0B23E04560B5BEA77B483B1F6491E53A3F2BEE6A39AA38181307F3013060A2B0601040182C40A0D01040504030506302206092B0601040182C40A020415312E332E362E312E342E312E34313438322E312E393013060B2B0601040182E51C02010404030205203021060B2B0601040182E51C01010404120410D8522D9F575B486688A9BA99FA02F35B300C0603551D130101FF04023000300D06092A864886F70D01010B0500038201010052B06949DBAAD1A64C1BA9EBC198B317EC31F9A37363BA5161B342E3A49CAD504F34E7428BB896E9CFD28D03AD10CE325A06838E9B6C4ECB17AD40D090A16C9E7C34498332FF853B62747E8FCDF00DAE62756E57BD40B16D677907A835C0435A2EBCE9B0B9069CA122BF9D964A73206AF74FF3C00144EBFF3DE7C7758D3147C8C2F9FE87C12F2A9675A2046B01076361A99721871FA78FB0DE2945B579F9166C48AD2FD50C3CE56C8221A75083F656119394368FF17D2C920C63A09F01ED2501146B7DF1AB3970A2A32938FA9A517AF471085E160B3CA79764231746BA6ABBA68E0D13CE259796BCD2A03AD83C74E15331328EAB438E6A4197CB12EC6FD1E388
```

Textual representation of the inner Authenticator response:

```
{1: "packed", 2: "h'E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C2C500000003D8522D9F575B486688A9BA99FA02F35B0030FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9784BBA5010203262001215820FFE2DC7BB7DFD9C8C268C45DD34383E21988D0F18870AB0E115
```

```
0278B8F8BEA742258202DE56C3BCFA03520409164829D13A8D4CCF82894FEF2D48BC75F0064F9DDB1AFA16B637265
6450726F7465637402', 3: {"alg": -7, "sig":
h'304402203F97BADFB8FD74ECE32AE298E65FF32F930B5F1F430741C1FAC0BF91FA37029802200E2391E5BAE6620
85958C67B76FC7ED0483C6F4EAC14A8A2F9945E2041C38754', "x5c":
[h'308202D8308201C0A003020102020900FF876C2DAF7379C8300D06092A864886F70D01010B0500302E312C302A
0603550403132359756269636F2055324620526F6F742043412053657269616C203435373230303633313020170D3
134303830313030303030305A180F32303530303930343030303030305A306E310B30090603550406130253453112
3010060355040A0C0959756269636F20414231223020060355040B0C1941757468656E74696361746F72204174746
573746174696F6E3127302506035504030C1E59756269636F205532462045452053657269616C2037363230383734
32333059301306072A8648CE3D020106082A8648CE3D0301070342000425F123A048283FC5796CCF887D99489FD93
5C24198C4B5D8D5B2C2BFD7DD5D15AFE45B7070776567D5B5B0B23E04560B5BEA77B483B1F6491E53A3F2BEE6A39A
A38181307F3013060A2B0601040182C40A0D0104050403050506302206092B0601040182C40A020415312E332E362
E312E342E12E34313438322E312E393013060B2B0601040182E51C0201010404030205203021060B2B0601040182
E51C01010404120410D8522D9F575B486688A9BA99FA02F35B300C0603551D130101FF04023000300D06092A86488
6F70D01010B0500038201010052B06949DBAAD1A64C1BA9EBC198B317EC31F9A37363BA5161B342E3A49CAD504F34
E7428BB896E9CFD28D03AD10CE325A06838E9B6C4ECB17AD40D090A16C9E7C34498332FF853B62747E8FCDF00DAE6
2756E57BD40B16D677907A835C0435A2EBCE9B0B9069CA122BF9D964A73206AF74FF3C00144EBFF3DE7C7758D3147
C8C2F9FE87C12F2A9675A2046B01076361A99721871FA78FB0DE2945B579F9166C48AD2FD50C3CE56C8221A75083F
656119394368FF17D2C920C63A09F01ED2501146B7DF1AB3970A2A32938FA9A517AF471085E160B3CA79764231746
BA6ABBA68E0D13CE259796BCD2A03AD83C74E15331328EAB438E6A4197CB12EC6FD1E388']}] }
```

4.4.2 CTAPCBOR_CMD_GET_ASSERTION

4.4.2.1 Request

Full request, a CBOR map:

```
0xA667636F6D6D616E640565666C6167731A008400006774696D656F75741A000493E06D7472616E73616374696F6
E496450F0D8CF821A912D42B1F083439A32C4C0677265717565737458E202A401781E776562617574686E74657374
2E617A75726577656273697465732E6E657402582071416126685DFB9B2776D1B26AD709605951061F3692A7AD025
F919C4881FD390382A3626964582019308CB37A700C8454D6C914D48A95E187D71B71B64F9AE4ACA7D09C89BCD4F9
64747970656A7075626C69632D6B65796A7472616E73706F72747317A36269645830FFE2DC7BB7DFD9C8C268C45DD
339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9784BB64747970656A7075626C69
632D6B65796A7472616E73706F7274731705A1627570F56C776562417574684E50617261A863776E641A0001036C6
A6174746163686D656E74006F726571756972655265736964656E74F46E7072656665725265736964656E74F47075
736572566572696669636174696F6E02756174746573746174696F6E507265666572656E63650075656E746572707
26973654174746573746174696F6E006E63616E63656C6C6174696F6E4964501D8CEE3C00000000000000000000
00
```

Textual representation of the request:

```
{"command": 5, "flags": 8650752, "timeout": 300000, "transactionId":
h'F0D8CF821A912D42B1F083439A32C4C0', "request":
h'02A401781E776562617574686E746573742E617A75726577656273697465732E6E657402582071416126685DFB9
B2776D1B26AD709605951061F3692A7AD025F919C4881FD390382A3626964582019308CB37A700C8454D6C914D48A
95E187D71B71B64F9AE4ACA7D09C89BCD4F964747970656A7075626C69632D6B65796A7472616E73706F72747317A
36269645830FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B42
2370AC2C9784BB64747970656A7075626C69632D6B65796A7472616E73706F7274731705A1627570F5',
"webAuthNPara": {"wnd": 66412, "attachment": 0, "requireResident": false, "preferResident":
false, "userVerification": 2, "attestationPreference": 0, "enterpriseAttestation": 0,
"cancellationId": h'1D8CEE3C000000000000000000000000'}}
```

Inner GetAssertion request details following the command type(0x02):

```
A401781E776562617574686E746573742E617A75726577656273697465732E6E657402582071416126685DFB9B277
6D1B26AD709605951061F3692A7AD025F919C4881FD390382A3626964582019308CB37A700C8454D6C914D48A95E1
87D71B71B64F9AE4ACA7D09C89BCD4F964747970656A7075626C69632D6B65796A7472616E73706F72747317A3626
9645830FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370
AC2C9784BB64747970656A7075626C69632D6B65796A7472616E73706F7274731705A1627570F5
```

Textual representation of the preceding GetAssertion request:

```
{1: "webauthntest.azurewebsites.net", 2:
h'71416126685DFB9B2776D1B26AD709605951061F3692A7AD025F919C4881FD39', 3: [{"id":
h'19308CB37A700C8454D6C914D48A95E187D71B71B64F9AE4ACA7D09C89BCD4F9', "type": "public-key",
"transports": 23}], {"id":
h'FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9
784BB', "type": "public-key", "transports": 23}], 5: {"up": true}}
```

4.4.2.2 Response

Full response including the HRESULT and the CBOR map:

```
0x0000A36A646576696365496E666FAB6A6D61784D736753697A651904B0781B6D617853657269616C697A65644C6
1726765426C6F6241727261791904006C70726F766964657254797065634869646C70726F76696465724E616D6578
184D6963726F736F66744374617048696450726F76696465726A6465766963655061746878525C5C3F5C686964237
669645F31303530267069645F30343032233726333131333036393426312630303030237B34643165353562322D66
3136662D313163662D383863622D3030313131313030303033307D6C6D616E7566616374757265726659756269636
F6770726F647563746C597562694B6579204649444F66616147756964509F2D52D85B57664888A9BA99FA02F35B78
1A63726564656E7469616C4C697374496E646578506C75734F6E65026875765374617475730169757652657472696
57303667374617475730068726573706F6E736558D100A401A26269645830FFE2DC7BB7DFD9C8C268C45DD339BB4F
187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9784BB64747970656A7075626C69632D6B6
579025825E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C205000000090358483046
02210099C54C2075B88279DE38944F5492E21DA2F5E1969BCCDD99A97F39BDD844A78022100BF091FBE45242DF48
4D7396D5304A570A78F2E5576ECC8AB24107E51968AD1C404A16269644F626F62406578616D706C652E636F6D
```

Textual representation of the CBOR map following the HRESULT:

```
{"deviceInfo": {"maxMsgSize": 1200, "maxSerializedLargeBlobArray": 1024, "providerType":
"Hid", "providerName": "MicrosoftCtapHidProvider", "devicePath":
"\\\\\\?\\hid#vid 1050&pid 0402#7&31130694&1&0000#{4d1e55b2-f16f-11cf-88cb-001111000030}"},
"manufacturer": "Yubico", "product": "YubiKey FIDO", "aaGuid":
h'9F2D52D85B57664888A9BA99FA02F35B', "credentialListIndexPlusOne": 2, "uvStatus": 1,
"uvRetries": 3}, "status": 0, "response":
h'00A401A26269645830FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D
97F581B422370AC2C9784BB64747970656A7075626C69632D6B6579025825E45329D03A2068D1CAF7F7BB0AE954E6
B0E6259745F32F4829F750F05011F9C20500000009035848304602210099C54C2075B88279DE38944F5492E21DA2F
5E1969BCCDD99A97F39BDD844A78022100BF091FBE45242DF484D7396D5304A570A78F2E5576ECC8AB24107E5196
8AD1C404A16269644F626F62406578616D706C652E636F6D'}
```

Inner Authenticator response details following the first byte indicating success(0x00):

```
A401A26269645830FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F5
81B422370AC2C9784BB64747970656A7075626C69632D6B6579025825E45329D03A2068D1CAF7F7BB0AE954E6B0E6
259745F32F4829F750F05011F9C20500000009035848304602210099C54C2075B88279DE38944F5492E21DA2F5E19
69BCCDD99A97F39BDD844A78022100BF091FBE45242DF484D7396D5304A570A78F2E5576ECC8AB24107E51968AD1
C404A16269644F626F62406578616D706C652E636F6D
```

Textual representation of the preceding map:

```
{1: {"id":
h'FFE2DC7BB7DFD9C8C268C45DD339BB4F187E4F33B96B2B02551FFBC264BD325BC9345A27D97F581B422370AC2C9
784BB', "type": "public-key"}, 2:
h'E45329D03A2068D1CAF7F7BB0AE954E6B0E6259745F32F4829F750F05011F9C20500000009', 3:
h'304602210099C54C2075B88279DE38944F5492E21DA2F5E1969BCCDD99A97F39BDD844A78022100BF091FBE452
42DF484D7396D5304A570A78F2E5576ECC8AB24107E51968AD1C4', 4: {"id":
h'626F62406578616D706C652E636F6D'}}
```

5 Security

5.1 Security Considerations for Implementers

For information, see [\[W3C-WebAuthPKC2\]](#), section 13.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

The terms "earlier" and "later", when used with a product version, refer to either all preceding versions or all subsequent versions, respectively. The term "through" refers to the inclusive range of versions. Applicable Microsoft products are listed chronologically in this section.

Windows Client

- Windows 10 v1809 operating system
- Windows 10 v1903 operating system
- Windows 10 v1909 operating system
- Windows 10 v2004 operating system
- Windows 10 v20H2 operating system
- Windows 10 v21H1 operating system
- Windows 10 v21H2 operating system
- Windows 11 operating system
- Windows 11, version 22H2 operating system

Windows Server

- Windows Server v1809 operating system
- Windows Server v1903 operating system
- Windows Server v1909 operating system
- Windows Server v2004 operating system
- Windows Server v20H2 operating system
- Windows Server 2022 operating system

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

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

[<1> Section 2.2.1](#): This is the Microsoft Windows API version on Windows systems. See WebAuthNGetApiVersionNumber in [\[MSFT-WebAuthnAPIS\]](#).

[<2> Section 2.2.1.1](#): For example, Windows Hello on Windows systems.

7 Change Tracking

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

8 Index

A

Abstract data model
[server](#) 17
[Applicability](#) 6

C

[Capability negotiation](#) 6
[Change tracking](#) 26

D

Data model - abstract
[server](#) 17

F

[Fields - vendor-extensible](#) 6

G

[Glossary](#) 5

H

Higher-layer triggered events
[server](#) 17

I

[Implementer - security considerations](#) 24
[Index of security parameters](#) 24
[Informative references](#) 6
Initialization
[server](#) 17
[Introduction](#) 5

M

Message processing
[server](#) 17
Messages
[transport](#) 8
[WebAuthN Channel Request Message](#) 8
[WebAuthN Channel Response Message](#) 13

N

[Normative references](#) 5

O

Other local events
[server](#) 17
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 24
[Preconditions](#) 6

[Prerequisites](#) 6
[Product behavior](#) 25

R

[References](#) 5
[informative](#) 6
[normative](#) 5
[Relationship to other protocols](#) 6

S

Security
[implementer considerations](#) 24
[parameter index](#) 24
Sequencing rules
[server](#) 17
Server
[abstract data model](#) 17
[higher-layer triggered events](#) 17
[initialization](#) 17
[message processing](#) 17
[other local events](#) 17
[overview](#) 17
[sequencing rules](#) 17
[timer events](#) 17
[timers](#) 17
[Standards assignments](#) 7

T

Timer events
[server](#) 17
Timers
[server](#) 17
[Tracking changes](#) 26
[Transport](#) 8
Triggered events - higher-layer
[server](#) 17

V

[Vendor-extensible fields](#) 6
[Versioning](#) 6

W

[WebAuthN Channel Request Message message](#) 8
[WebAuthN Channel Response Message message](#) 13