

## [MS-RDPELE]: Remote Desktop Protocol: Licensing Extension

This topic lists the Errata found in [MS-RDPELE] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.



Errata are subject to the same terms as the Open Specifications documentation referenced.

Errata below are for Protocol Document Version [V15.0 – 2020/03/04](#).

Errata Published*	Description
2020/07/06	<p>In Section 3.2.1.10 Encryption Keys, revised description removing undefined field.            Changed from:</p> <p>The server uses the 128-bit licensing encryption key to encrypt and to decrypt licensing information message data obtained in Server Platform Challenge messages (section 2.2.2.4), in Client HWID (section 2.2.2.3 and 2.2.2.3.1), and in Client Platform Challenge messages (section 2.2.2.5), as specified in section 5.1.2.</p> <p>Changed to:</p> <p>The server uses the 128-bit licensing encryption key (section 5.1.2) to encrypt the EncryptedPlatformChallenge field in the Server Platform Challenge message (section 2.2.2.4), and decrypt the EncryptedHWID field in the Client License Information (section 2.2.2.3) and Client Platform Challenge (section 2.2.2.5) messages.</p> <p>In Section 4.2 CLIENT NEW LICENSE REQUEST, revised 'pBlob' to 'blobData'.            Changed from:</p> <pre> 41 64 6d 69 6e 69 73 74 -\ 72 61 74 6f 72 00      -/ ClientUserName::pBlob 0x14a: ClientMachineName (2 + 2 + 7 = 0xb bytes) 10 -\ 00 -/ ClientMachineName::wBlobType = BB CLIENT MACHINE NAME  07 -\ 00 -/ ClientMachineName::wBlobLen = 7 bytes  52 4f 44 45 4e 54 00 -&gt; ClientMachineName::pBlob           </pre> <p>Changed to:</p> <pre> 41 64 6d 69 6e 69 73 74 -\ 72 61 74 6f 72 00      -/ ClientUserName::blobData 0x14a: ClientMachineName (2 + 2 + 7 = 0xb bytes) 10 -\ 00 -/ ClientMachineName::wBlobType = BB_CLIENT_MACHINE_NAME  07 -\ 00 -/ ClientMachineName::wBlobLen = 7 bytes  52 4f 44 45 4e 54 00 -&gt; ClientMachineName::blobData           </pre>
2020/07/06	In Section 1.3.3, Licensing PDU Flows, changed from:

Errata Published*	Description
	<p>If the target machine is a personal terminal server, whether the client sends the license or not, the server always sends a license error message with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. Also, in the case that the client sends a license, the server does not validate it. The licensing protocol is complete at this point.</p> <p>Changed to:</p> <p>If the target machine is a personal terminal server, whether the client sends the license or not, the server always sends a Licensing Error Message (section 2.2.2.8) with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. Also, in the case that the client sends a license, the server does not validate it. The licensing protocol is complete at this point.</p> <p>In Section 2.2.2.1.1, Product Information (PRODUCT_INFO), changed from:</p> <p>The Product Information packet contains the details of the product license that is required for connecting to the terminal server. The client uses this structure together with the scope list to search for and identify an appropriate license in its license store. Depending on the outcome of the search, the client sends a Client New License Request (section 2.2.2.2), Client License Information packet (section 2.2.2.3), or license error message (section 2.2.2.8) to the server.</p> <p>Changed to:</p> <p>The Product Information packet contains the details of the product license that is required for connecting to the terminal server. The client uses this structure together with the scope list to search for and identify an appropriate license in its license store. Depending on the outcome of the search, the client sends a Client New License Request (section 2.2.2.2), Client License Information packet (section 2.2.2.3), or Licensing Error Message (section 2.2.2.8) to the server.</p> <p>In Section 2.2.2.8, Licensing Error Message (LICENSE_ERROR_MESSAGE), changed from:</p> <p>The license error message specified in [MS-RDPBCGR] section 2.2.1.12.1.3 can be used by both client and server.</p> <p>Changed to:</p> <p>The Licensing Error Message specified in [MS-RDPBCGR] section 2.2.1.12.1.3 can be used by both client and server.</p> <p>In Section 3.1.5.2 Sending Licensing Error Messages, changed from:</p> <p>Both the client and the server can send a license error message (section 2.2.2.8). Whenever an error message is sent, the message type in the licensing preamble MUST be set to ERROR_ALERT (0xFF). For the PDU, see [MS-RDPBCGR] section 2.2.1.12.1.3</p> <p>The client and the server MUST also set the appropriate state transition value in the dwStateTransition field in the PDU. This is used to determine the next action to take. For state transitions, see Processing License Error Messages.</p> <p>Changed to:</p> <p>Both the client and the server can send a Licensing Error Message (section 2.2.2.8). Whenever an error message is sent, the message type in the Licensing Preamble (section 2.2.1.2) MUST be set to ERROR_ALERT (0xFF). For the PDU, see [MS-RDPBCGR] section 2.2.1.12.1.3</p>

Errata Published*	Description
	<p>The client and the server MUST also set the appropriate state transition value in the dwStateTransition field in the PDU. This is used to determine the next action to take. For state transitions, see Processing Licensing Error Messages (section 3.1.5.3).</p> <p>In Section 3.1.5.3 Processing Licensing Error Message, changed from:</p> <p>Both the server and the client can send a license error message (section 2.2.2.8) and indicate a state transition with the error code. Possible state transitions include the following:</p> <p>Changed to:</p> <p>Both the server and the client can send a Licensing Error Message (section 2.2.2.8) and indicate a state transition with the error code. Possible state transitions include the following:</p> <p>In Section 3.2.5.2 Processing Client New License Requests, changed from:</p> <p>In case of a personal terminal server, no processing is done on the server side, and the server sends a license error message with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>Changed to:</p> <p>In case of a personal terminal server, no processing is done on the server side, and the server sends a Licensing Error Message (section 2.2.2.8) with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>In Section 3.2.5.3 Processing Client License Information (2 instances), changed from:</p> <p>In the case of a personal terminal server, the sent license is cached by the server, and then the server sends a license error message with error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>&amp;</p> <p>Case 1: The client presents a valid permanent license that does not require an upgrade. The server MUST send a license error message (section 2.2.2.8) with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>Changed to:</p> <p>In the case of a personal terminal server, the sent license is cached by the server, and then the server sends a Licensing Error Message (section 2.2.2.8) with error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>&amp;</p> <p>Case 1: The client presents a valid permanent license that does not require an upgrade. The server MUST send a Licensing Error Message with the error code STATUS_VALID_CLIENT and the state transition code ST_NO_TRANSITION. The licensing protocol is complete at this point.</p> <p>In Section 3.2.5.5 Processing Client Platform Challenge Responses, changed from:</p> <p>In this case, if the server's grace period has not been exceeded, the server responds as if the client presented a valid license by sending a license error message (section 2.2.2.8) with an error code of STATUS_VALID_CLIENT (0x00000007) and a state transition code of ST_NO_TRANSITION (0x00000002), ending the licensing protocol.</p>

Errata Published*	Description
	<p>If the server's grace period has been exceeded, it sends a license error message (section 2.2.2.8) with error code ERR_NO_LICENSE_SERVER (0x00000006) and a state transition of ST_TOTAL_ABORT (0x00000001). The licensing protocol is aborted.</p> <p>&amp;</p> <p>In this case, if the grace period has not been exceeded, the server responds as if the client presented a valid license by sending a license error message (section 2.2.2.8) with an error code of STATUS_VALID_CLIENT (0x00000007) and a state transition code of ST_NO_TRANSITION (0x00000002), ending the licensing protocol.</p> <p>If the server's grace period has been exceeded, it sends a license error message (section 2.2.2.8) with an error code of ERR_INVALID_CLIENT (0x00000008) and a state transition of ST_TOTAL_ABORT (0x00000001). The licensing protocol is aborted.</p> <p>Changed to:</p> <p>In this case, if the server's grace period has not been exceeded, the server responds as if the client presented a valid license by sending a Licensing Error Message (section 2.2.2.8) with an error code of STATUS_VALID_CLIENT (0x00000007) and a state transition code of ST_NO_TRANSITION (0x00000002), ending the licensing protocol.</p> <p>If the server's grace period has been exceeded, it sends a Licensing Error Message with error code ERR_NO_LICENSE_SERVER (0x00000006) and a state transition of ST_TOTAL_ABORT (0x00000001). The licensing protocol is aborted.</p> <p>&amp;</p> <p>In this case, if the grace period has not been exceeded, the server responds as if the client presented a valid license by sending a Licensing Error Message with an error code of STATUS_VALID_CLIENT (0x00000007) and a state transition code of ST_NO_TRANSITION (0x00000002), ending the licensing protocol.</p> <p>If the server's grace period has been exceeded, it sends a Licensing Error Message with an error code of ERR_INVALID_CLIENT (0x00000008) and a state transition of ST_TOTAL_ABORT (0x00000001). The licensing protocol is aborted.</p> <p>In Section 3.2.5.8 Handling Out of Sequence or Unrecognized Messages, changed from:</p> <p>If the server receives a message that is not expected according to the Licensing PDU Flow, or a malformed or an unrecognized message, the server MUST send a License Error message (section 2.2.2.8) with an error code of ERR_INVALID_CLIENT and a state transition code of ST_TOTAL_ABORT.</p> <p>Changed to:</p> <p>If the server receives a message that is not expected according to the Licensing PDU Flow (section 1.3.3), or a malformed or an unrecognized message, the server MUST send a Licensing Error Message (section 2.2.2.8) with an error code of ERR_INVALID_CLIENT and a state transition code of ST_TOTAL_ABORT.</p> <p>In Section 3.2.5.9 Handling Invalid MACs, changed from:</p> <p>If the MAC generated over decrypted fields of a message does not match the MAC contained in the message, the server MUST send a License Error message (section 2.2.2.8) with an error code of ERR_INVALID_MAC and a state transition code of ST_TOTAL_ABORT.</p> <p>Changed to:</p> <p>If the MAC generated over decrypted fields of a message does not match the MAC contained in the message, the server MUST send a Licensing Error Message (section 2.2.2.8) with an error code of ERR_INVALID_MAC and a state transition code of ST_TOTAL_ABORT.</p> <p>In Section 3.3.5.1 Processing Server License Requests, changed from:</p>

Errata Published*	Description
	<ul style="list-style-type: none"> <li>●If the server certificate does not authenticate correctly, the client MUST return a license error message (section 2.2.2.8) with an error code of ERR_INVALID_SERVER_CERTIFICATE (0x01) and a state transition of ST_TOTAL_ABORT (0x01). The server MUST then end the licensing protocol.</li> <li>●The client searches its license store to find a CAL that matches the Product Information packet provided in the Server License Request. If the client finds a matching license, it MUST respond with a Client License Information message.</li> <li>●If the client does not find a license matching the product information provided in the Server License Request, it MUST request a new license by sending the Client New License Request message.</li> <li>●The client MAY also choose to end the licensing protocol by sending a license error message (section 2.2.2.8) with an error code of ERR_NO_LICENSE (0x02) and a state transition of ST_TOTAL_ABORT (0x01).</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>●If the Server Certificate (section 2.2.1.4) does not authenticate correctly, the client MUST return a Licensing Error Message (section 2.2.2.8) with an error code of ERR_INVALID_SERVER_CERTIFICATE (0x01) and a state transition of ST_TOTAL_ABORT (0x01). The server MUST then end the licensing protocol.</li> <li>●The client searches its license store to find a CAL that matches the Product Information (section 2.2.2.1.1) packet provided in the Server License Request. If the client finds a matching license, it MUST respond with a Client License Information (section 2.2.2.3) message.</li> <li>●If the client does not find a license matching the Product Information provided in the Server License Request, it MUST request a new license by sending the Client New License Request (section 2.2.2.2) message.</li> <li>●The client MAY also choose to end the licensing protocol by sending a Licensing Error Message with an error code of ERR_NO_LICENSE (0x02) and a state transition of ST_TOTAL_ABORT (0x01).</li> </ul> <p>In Section 3.3.5.9 Handling Invalid MACs, changed from:</p> <p>If the MAC generated over decrypted fields of a message does not match the MAC contained in the message, the client MAY send a License Error message (section 2.2.2.8) with an error code of ERR_INVALID_MAC and a state transition code of ST_TOTAL_ABORT. The client then MUST disconnect the RDP connection.</p> <p>Changed to:</p> <p>If the MAC generated over decrypted fields of a message does not match the MAC contained in the message, the client MAY send a Licensing Error Message (section 2.2.2.8) with an error code of ERR_INVALID_MAC and a state transition code of ST_TOTAL_ABORT. The client then MUST disconnect the RDP connection.</p> <p>In Section 6 Appendix A: Product Behavior, changed from:</p> <p>&lt;16&gt; Section 3.1.5.3: In Windows XP, the RDP connection is not disconnected on receiving ST_TOTAL_ABORT as the state transition in the license error message (section 2.2.2.8).</p> <p>Changed to:</p> <p>&lt;16&gt; Section 3.1.5.3: In Windows XP, the RDP connection is not disconnected on receiving ST_TOTAL_ABORT as the state transition in the Licensing Error Message (section 2.2.2.8).</p> <p>In Section 2.2.2 Licensing PDU, revised descriptions for the values of LicensingMessage.</p>

Errata Published*	Description																						
	<p>Changed from:</p> <p>LicensingMessage (variable): A variable-length licensing message whose structure depends on the value of the bMsgType field in the preamble structure. The following table lists possible values for bMsgType and the associated licensing message (this table also appears in [MS-RDPBCGR] section 2.2.1.12.1.1).</p> <p>Sent by the server.</p> <table border="1" data-bbox="402 451 1414 865"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>LICENSE_REQUEST 0x01</td> <td>The Licensing PDU is a License Request PDU, and the LicensingMessage contains a Server License Request.</td> </tr> <tr> <td>PLATFORM_CHALLENGE 0x02</td> <td>The Licensing PDU is a Platform Challenge PDU, and the LicensingMessage contains a Server Platform Challenge.</td> </tr> <tr> <td>NEW_LICENSE 0x03</td> <td>The Licensing PDU is a New License PDU, and the LicensingMessage contains a Server New License structure.</td> </tr> <tr> <td>UPGRADE_LICENSE 0x04</td> <td>The Licensing PDU is an Upgrade License PDU, and the LicensingMessage contains a Server Upgrade License structure. Sent by the client.</td> </tr> </tbody> </table> <table border="1" data-bbox="402 907 1414 1297"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>LICENSE_INFO 0x12</td> <td>The Licensing PDU is a License Info PDU, and the LicensingMessage contains a Client License Information structure.</td> </tr> <tr> <td>NEW_LICENSE_REQUEST 0x13</td> <td>The Licensing PDU is a New License Request PDU, and the LicensingMessage contains a Client New License Request structure.</td> </tr> <tr> <td>PLATFORM_CHALLENGE_RESPONSE 0x15</td> <td>The Licensing PDU is a Platform Challenge Response PDU, and the LicensingMessage contains a Client Platform Challenge Response structure. Sent by either the client or the server.</td> </tr> </tbody> </table> <table border="1" data-bbox="402 1339 1414 1528"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ERROR_ALERT 0xFF</td> <td>The Licensing PDU is a Licensing Error Message PDU, and the LicensingMessage contains a license error message (section 2.2.2.8) structure.</td> </tr> </tbody> </table> <p>Changed to:</p> <p>LicensingMessage (variable): A variable-length licensing message whose structure depends on the value of the bMsgType field in the preamble structure. The following table lists possible values for bMsgType and the associated licensing message (this table also appears in [MS-RDPBCGR] section 2.2.1.12.1.1).</p> <p>Sent by the server.</p>	Value	Meaning	LICENSE_REQUEST 0x01	The Licensing PDU is a License Request PDU, and the LicensingMessage contains a Server License Request.	PLATFORM_CHALLENGE 0x02	The Licensing PDU is a Platform Challenge PDU, and the LicensingMessage contains a Server Platform Challenge.	NEW_LICENSE 0x03	The Licensing PDU is a New License PDU, and the LicensingMessage contains a Server New License structure.	UPGRADE_LICENSE 0x04	The Licensing PDU is an Upgrade License PDU, and the LicensingMessage contains a Server Upgrade License structure. Sent by the client.	Value	Meaning	LICENSE_INFO 0x12	The Licensing PDU is a License Info PDU, and the LicensingMessage contains a Client License Information structure.	NEW_LICENSE_REQUEST 0x13	The Licensing PDU is a New License Request PDU, and the LicensingMessage contains a Client New License Request structure.	PLATFORM_CHALLENGE_RESPONSE 0x15	The Licensing PDU is a Platform Challenge Response PDU, and the LicensingMessage contains a Client Platform Challenge Response structure. Sent by either the client or the server.	Value	Meaning	ERROR_ALERT 0xFF	The Licensing PDU is a Licensing Error Message PDU, and the LicensingMessage contains a license error message (section 2.2.2.8) structure.
Value	Meaning																						
LICENSE_REQUEST 0x01	The Licensing PDU is a License Request PDU, and the LicensingMessage contains a Server License Request.																						
PLATFORM_CHALLENGE 0x02	The Licensing PDU is a Platform Challenge PDU, and the LicensingMessage contains a Server Platform Challenge.																						
NEW_LICENSE 0x03	The Licensing PDU is a New License PDU, and the LicensingMessage contains a Server New License structure.																						
UPGRADE_LICENSE 0x04	The Licensing PDU is an Upgrade License PDU, and the LicensingMessage contains a Server Upgrade License structure. Sent by the client.																						
Value	Meaning																						
LICENSE_INFO 0x12	The Licensing PDU is a License Info PDU, and the LicensingMessage contains a Client License Information structure.																						
NEW_LICENSE_REQUEST 0x13	The Licensing PDU is a New License Request PDU, and the LicensingMessage contains a Client New License Request structure.																						
PLATFORM_CHALLENGE_RESPONSE 0x15	The Licensing PDU is a Platform Challenge Response PDU, and the LicensingMessage contains a Client Platform Challenge Response structure. Sent by either the client or the server.																						
Value	Meaning																						
ERROR_ALERT 0xFF	The Licensing PDU is a Licensing Error Message PDU, and the LicensingMessage contains a license error message (section 2.2.2.8) structure.																						

Value	Meaning
LICENSE_REQUEST 0x01	The Licensing PDU is a License Request PDU, and the LicensingMessage contains a SERVER_LICENSE_REQUEST (section 2.2.2.1) structure.
PLATFORM_CHALLENGE 0x02	The Licensing PDU is a Platform Challenge PDU, and the LicensingMessage contains a SERVER_PLATFORM_CHALLENGE (section 2.2.2.4) structure.
NEW_LICENSE 0x03	The Licensing PDU is a New License PDU, and the LicensingMessage contains a SERVER_NEW_LICENSE (section 2.2.2.7) structure.
UPGRADE_LICENSE 0x04	The Licensing PDU is an Upgrade License PDU, and the LicensingMessage contains a SERVER_UPGRADE_LICENSE (section 2.2.2.6) structure. Sent by the client.

Value	Meaning
LICENSE_INFO 0x12	The Licensing PDU is a License Info PDU, and the LicensingMessage contains a CLIENT_LICENSE_INFO (section 2.2.2.3) structure.
NEW_LICENSE_REQUEST 0x13	The Licensing PDU is a New License Request PDU, and the LicensingMessage contains a CLIENT_NEW_LICENSE_REQUEST (section 2.2.2.2) structure.
PLATFORM_CHALLENGE_RESPONSE 0x15	The Licensing PDU is a Platform Challenge Response PDU, and the LicensingMessage contains a CLIENT_PLATFORM_CHALLENGE_RESPONSE (section 2.2.2.5) structure. Sent by either the client or the server.

Value	Meaning
ERROR_ALERT 0xFF	The Licensing PDU is a Licensing Error Message PDU, and the LicensingMessage contains a LICENSE_ERROR_MESSAGE (section 2.2.2.8) structure.

In Section 2.2.2.9.1 Licensed Product Info (LICENSED\_PRODUCT\_INFO), revised 'AdjustedProductdOffset' to 'AdjustedProductIdOffset'.

Changed from:

AdjustedProductdOffset	AdjustedProductIdByteCount
------------------------	----------------------------

Changed to:

AdjustedProductIdOffset	AdjustedProductIdByteCount
-------------------------	----------------------------

Errata Published*	Description
	<p>In Section 3.1.5.2 Sending Licensing Error Message, revised title.</p> <p>Changed from: 3.1.5.2 Sending License Error Messages</p> <p>Changed to: 3.1.5.2 Sending Licensing Error Messages</p> <p>In Section 3.1.5.3 Processing Licensing Error Message, revised title.</p> <p>Changed from: 3.1.5.3 Processing License Error Messages</p> <p>Changed to: 3.1.5.3 Processing Licensing Error Messages</p> <p>In Section 3.2.1.6 Platform Challenge, removed 'Encrypting License Data' from description.</p> <p>Changed from: The platform challenge is a random string generated by the server. This string is encrypted (see Encrypting Licensing Data (section 5.1.3)) with the licensing encryption key using RC4 and sent in the EncryptedPlatformChallenge field of the Server Platform Challenge message. It is created at the beginning of the licensing protocol and destroyed when the licensing protocol is completed.</p> <p>Changed to: The Platform Challenge is a random string generated by the server. This string is encrypted (section 5.1.3) with the licensing encryption key using RC4 and sent in the EncryptedPlatformChallenge field of the Server Platform Challenge (section 2.2.2.4) message. It is created at the beginning of the licensing protocol and destroyed when the licensing protocol is completed.</p> <p>In Section 3.3.1.9 Client Hardware Identification, revised 'PlatformID' to 'PlatformId'.</p> <p>Changed from: The content and format of the PlatformID field of Client Hardware Identification are the same as the PlatformID field of the Client License Information and Client New License Request messages. This ties a particular Client Hardware Identification to the client's operating system. The other 16 bytes (fields Data1 through Data4) of the Client Hardware Identification are intended to be hardware-specific. Clients SHOULD attempt to use operating system-specific or hardware-specific information that is easily and consistently retrievable. Examples include hard-wired processor IDs, Ethernet addresses of nonremovable Ethernet cards, and disk subsystem serial numbers. The client SHOULD cache the Client Hardware Identification for later retrieval after it is generated.</p> <p>Changed to: The content and format of the PlatformId field of Client Hardware Identification (section 2.2.2.3.1) are the same as the PlatformId field of the Client License Information (section 2.2.2.3) and Client New License Request (section 2.2.2.2) messages. This ties a particular Client Hardware Identification to the client's operating system. The other 16 bytes (fields Data1 through Data4) of the Client Hardware Identification are intended to be hardware specific. Clients SHOULD attempt to use operating system-specific or hardware-specific information that is easily and consistently retrievable. Examples include hard-wired processor IDs, Ethernet addresses of nonremovable Ethernet cards, and disk subsystem serial numbers. The client SHOULD cache the Client Hardware Identification for later retrieval after it is generated.</p>



Errata Published*	Description
	<p>In Section 4.1 Server License Request (SERVER_LICENSE_REQUEST), revised title.</p> <p>Changed from:</p> <p>4.1 SERVER LICENSE REQUEST</p> <p>Changed to:</p> <p>4.1 Server License Request (SERVER_LICENSE_REQUEST)</p> <p>In Section 4.2 Client New License Request (CLIENT_NEW_LICENSE_REQUEST), revised title.</p> <p>Changed from:</p> <p>4.2 CLIENT NEW LICENSE REQUEST</p> <p>Changed to:</p> <p>4.2 Client New License Request (CLIENT_NEW_LICENSE_REQUEST)</p> <p>In Section 4.2 Client New License Request (CLIENT_NEW_LICENSE_REQUEST), revised '_PREAMBE' to '_PREAMBLE'.</p> <p>Changed from:</p> <pre> 0x00: LICENSE_PREAMBLE (4 bytes) 13 -&gt; LICENSE_PREAMBE::bMsgType = NEW_LICENSE_REQUEST </pre> <p>Changed to:</p> <pre> 0x00: LICENSE_PREAMBLE (4 bytes) 13 -&gt; LICENSE_PREAMBLE::bMsgType = NEW_LICENSE_REQUEST </pre> <p>In Section 4.3 Client License Information (CLIENT_LICENSE_INFO), revised title.</p> <p>Changed from:</p> <p>4.3 CLIENT LICENSE INFO</p> <p>Changed to:</p> <p>4.3 Client License Information (CLIENT_LICENSE_INFO)</p> <p>In Section 4.3 Client License Information (CLIENT_LICENSE_INFO), revised '_PREAMBE' to '_PREAMBLE'.</p> <p>Changed from:</p> <pre> 0x00: LICENSE_PREAMBLE (4 bytes) 12 -&gt; LICENSE_PREAMBE::bMsgType = CLIENT LICENSE INFO 83 -&gt; LICENSE_PREAMBE::bVersion = 0x80   0x3 </pre> <p>Changed to:</p>

Errata Published*	Description
	<pre> 0x00: LICENSE_PREAMBLE (4 bytes)                                 12 -&gt; LICENSE_PREAMBLE::bMsgType = CLIENT_LICENSE_INFO  83 -&gt; LICENSE_PREAMBLE::bVersion = 0x80   0x3 </pre> <p>In Section 4.4 Server Platform Challenge (SERVER_PLATFORM_CHALLENGE), revised title.</p> <p>Changed from:</p> <p>4.4 SERVER PLATFORM CHALLENGE</p> <p>Changed to:</p> <p>4.4 Server Platform Challenge (SERVER_PLATFORM_CHALLENGE)</p> <p>In Section 4.5 Client Platform Challenge Response (CLIENT_PLATFORM_CHALLENGE_RESPONSE), revised title.</p> <p>Changed from:</p> <p>4.5 CLIENT PLATFORM CHALLENGE RESPONSE</p> <p>Changed to:</p> <p>4.5 Client Platform Challenge Response (CLIENT_PLATFORM_CHALLENGE_RESPONSE)</p> <p>In Section 4.6 Server New License (SERVER_NEW_LICENSE), revised title.</p> <p>Changed from:</p> <p>4.6 SERVER NEW LICENSE</p> <p>Changed to:</p> <p>4.6 Server New License (SERVER_NEW_LICENSE)</p> <p>In Section 4.7 Server Upgrade License (SERVER_UPGRADE_LICENSE), revised title.</p> <p>Changed from:</p> <p>4.7 SERVER UPGRADE LICENSE</p> <p>Changed to:</p> <p>4.7 Server Upgrade License (SERVER_UPGRADE_LICENSE)</p>
2020/07/06	<p>In Section 2.2.2.9, X.509 Certificate Extensions, revised certificate object identifier from 'OID' to 'szOID'.</p> <p>Changed from:</p> <ul style="list-style-type: none"> <li>●"1.3.6.1.4.1.311.18.4" (OID_HYDRA_CERT_VERSION)</li> <li>●"1.3.6.1.4.1.311.18.2" (OID_MANUFACTURER)</li> <li>●"1.3.6.1.4.1.311.18.5" (OID_LICENSED_PRODUCT_INFO)</li> <li>●"1.3.6.1.4.1.311.18.6" (OID_MS_LICENSE_SERVER_INFO)</li> <li>●"1.3.6.1.4.1.311.18.7" (OID_PRODUCT_SPECIFIC_OID)</li> </ul> <p>Changed to:</p> <ul style="list-style-type: none"> <li>●"1.3.6.1.4.1.311.18.4" (szOID_PKIX_HYDRA_CERT_VERSION)</li> </ul>

<b>Errata Published*</b>	<b>Description</b>
	<ul style="list-style-type: none"><li>●"1.3.6.1.4.1.311.18.2" (szOID_PKIX_MANUFACTURER)</li><li>●"1.3.6.1.4.1.311.18.5" (szOID_PKIX_LICENSED_PRODUCT_INFO)</li><li>●"1.3.6.1.4.1.311.18.6" (szOID_PKIX_MS_LICENSE_SERVER_INFO)</li><li>●"1.3.6.1.4.1.311.18.7" (szOID_PKIS_PRODUCT_SPECIFIC_OID)</li></ul>

\*Date format: YYYY/MM/DD