RTI Security Plugins

Release Notes

Version 5.3.1

Printed in U.S.A. First printing.
March 2018.

Trademarks

Copy and Use Restrictions

Technical Support

Real-Time Innovations, Inc.
232 E. Java Drive
Sunnyvale, CA 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: https://support.rti.com/

Contents

Chapter 1 Supported Platforms

RTI® Security Plugins 5.3.1 is supported on the following platforms.

Table 1.1 Supported Platforms for Security Plugins 5.3.1

Operating System	Version
Android™	Android 2.3 - 4.4, 5.0, 5.1
iOS®	iOS 8.2
Linux® (ARM® CPU)	NI Linux 3 Raspbian Wheezy 7.0 Ubuntu 16.04 LTS
Linux (Intel® CPU)	CentOS 6.0, 6.2, 6.3, 6.4, 7.0 Red Hat® Enterprise Linux 5.0-5.2, 5.4, 5.5, 6.0 - 6.5, 6.7, 6.8, 7.0 Ubuntu® 12.04 LTS, Ubuntu 14.04 LTS, Ubuntu 16.04 LTS Wind River® Linux 7 RedHawk™ Linux 6.5 (custom support)
Mac® OS X®	OS X 10.10- 10.13
QNX® (target only)	QNX Neutrino® 6.5, 6.5 SP1, 7.0 QNX Neutrino 6.6 (custom support)
Solaris™	Solaris 2.10
Windows®	Windows 7, 8, 8.1, 10 Windows Server 2008 R2, 2012 R2, 2016

See the *RTI Connext® DDS Core Libraries Platform Notes* for more information.

Chapter 2 Compatibility

Security Plugins 5.3.1 is interoperable with 5.2.7 and higher versions of Security Plugins.

Security Plugins 5.3.1 is compatible with OpenSSL® 1.0.2n.

2.1 Incompatibilities in 5.3.1 and Higher with Previous Versions

2.1.1 Configuration Incompatibilities (5.3.1 and Higher)

2.1.1.1 Change to Precedence in Permissions Document

In Connext DDS 5.3.0 and below, if a Permissions Document contained conflicting allow and deny rules for a given domain-topic combination, then the rule that took effect was the one that appeared last in the document. In this release, the rule that takes effect by default is the one that appears first in the document.

To change this behavior, you may set the security plugin property access_control.use_530_permissions_rules_precedence to true. (If true, then the last rule will take precedence, which is consistent with Connext 5.3.0 behavior. If false, then the first rule will take precedence, which is consistent with the intended behavior of the DDS Security specification. The default value is false.)

To avoid the question of precedence, rewrite your Permissions Document to eliminate conflicting rules.

2.1.2 API Incompatibilities (5.3.1 and Higher)

Changes in DDS Types and definitions:

- Changed Token to DDS_Token.

2.2 Incompatibilities in 5.3.0 and Higher with Previous Versions

2.2.1 API Incompatibilities (5.3.0 and Higher)

Changes in DDS Types and definitions:

- Changed **encode_datawriter_submessage**, **encode_datareader_submessage**, and **encode_rtps_message** to take a DDS_InlineList * instead of a void ** for the list of receiver CryptoHandles.
- Updated Logging Plugin log levels to match the DDS Security specification:
  - DDS_LOGGING_FATAL_LEVEL changed to DDS_LOGGING_EMERGENCY_LEVEL
  - DDS_LOGGING_SEVERE_LEVEL changed to DDS_LOGGING_ALERT_LEVEL
  - DDS_LOGGING_ERROR_LEVEL changed to DDS_LOGGING_CRITICAL_LEVEL
  - DDS_LOGGING_WARNING_LEVEL changed to DDS_LOGGING_ERROR_LEVEL
  - DDS_LOGGING_NOTICE_LEVEL changed to DDS_LOGGING_WARNING_LEVEL
  - DDS_LOGGING_INFO_LEVEL changed to DDS_LOGGING_NOTICE_LEVEL
  - DDS_LOGGING_DEBUG_LEVEL changed to DDS_LOGGING_INFORMATIONAL_LEVEL
  - DDS_LOGGING_TRACE_LEVEL changed to DDS_LOGGING_DEBUG_LEVEL

2.3 Incompatibilities in 5.2.7 and Higher with Previous Versions

This section describes configuration and wire incompatibilities with previous releases that have been introduced starting with 5.2.7.

2.3.1 Configuration Incompatibilities (5.2.7 and Higher)

2.3.1.1 Changes to Behavior of enable_[join/read/write]_access_control

The governance attributes **enable_join_access_control**, **enable_read_access_control**, and **enable_write_access_control** have changed their behavior. See the Access Control section in the *RTI Security Plugins Getting Started Guide* for details.

2.3.2 Wire Incompatibilities (5.2.7 and Higher)

2.3.2.1 New Format of Subject Name Used During Authentication Handshake

Release 5.2.7 and higher changes the format of the certificate subject name used during authentication handshaking. This change was made to be compliant with the latest DDS Security specification regarding computation of the **adjusted_participant_key**. It breaks wire compatibility between DDS 5.2.7 and previous releases.

2.3.2.2 New RSA Signature and Verification Used During Authentication Handshake

Release 5.2.7 and higher changes the padding and mask generation function of the RSA digital signature generation and verification used during authentication handshaking. This change was made to be compliant with the latest DDS Security specification. It breaks wire compatibility between DDS 5.2.7 and previous releases when using an RSA private key and certificate.

2.3.2.3 New Derivation of Key Exchange Key Material

Release 5.2.7 and higher changes the derivation of key exchange key material. This change was made to be compliant with the latest DDS Security specification Table 52. It breaks wire compatibility between DDS 5.2.7 and previous releases.

2.3.2.4 New Length of master_sender_key in CryptoTokens

Release 5.2.7 and higher changes the length of the **master_sender_key** sent in the CryptoTokens of an aes-128 key. This change was made to be compliant with the latest DDS Security specification Table 54. It breaks wire compatibility between Connext DDS 5.2.7 and previous releases when using aes-128-gcm.

2.3.2.5 New Encryption Algorithm for Key Exchange

Release 5.2.7 and higher changes the encryption algorithm for key exchange from aes-128-gcm to aes-256-gcm. This change was made to disambiguate the latest DDS Security specification Table 52 transformation_kind. It breaks wire compatibility between Connext DDS 5.2.7 and previous releases.

2.4 Incompatibilities in 5.2.6 and Higher with Previous Versions

This section describes configuration, wire, and API incompatibilities with previous releases that have been introduced starting with 5.2.6.

2.4.1 Configuration Incompatibilities (5.2.6 and Higher)

2.4.1.1 Changed Permissions and Governance Document Definitions to be Compliant with DDS Security Specification

Prior to release 5.2.6, Permissions and Governance documents were not compliant with the DDS Security specification. Starting with 5.2.6, these files are fully compliant with the specification. The following list describes the old and new behavior:

1. Topic and partition expressions (declared with the <topic> and <partition> tags) were placed directly inside the <publish> and <subscribe> rule tags.

   Starting with this release, <topic> and <partition> tags must be inside the <topics> and <partitions> tags respectively, and more than one <topic> or <partition> tag is supported.

2. Only the first <topic> and <partition> tags were actually parsed. Any additional <topic> and <partition> tags would be ignored.

   The behavior of multiple <topic> and <partition> tags is now aligned with the specification. The relationship among multiple <topic> and <partition> tags is of logical OR for tags of the same kind and logical AND between <topic> and <partition> tags.
3. The <relay> tag, although described in the specification, was not allowed by the parser.

   The <relay> tag is now allowed but ignored.
4. The date-time expression format in the <not_before> and <not_after> tags was not compliant with the specification.

   In previous releases, the date and time format accepted by <not_before> and <not_after> was YYYYMMDDhh. Starting with this release, the expected format is ISO-8601 combined date and time YYYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm
5. Boolean values of Governance documents were unconstrained XSD strings.

   In previous releases, multiple strings were accepted as valid boolean values (yes, no, 1, 0, TRUE and FALSE in a case-insensitive manner), but the specification states that these values must be xs:-boolean (1, 0, true and false, always in lower case). This behavior is now fixed and aligned with the specification.

2.4.1.2 Removed Support for rsa and aes-128-ctr

The value **rsa** is no longer an option for the property **authentication.shared_secret_algorithm**.

The value **aes-128-ctr** is no longer an option for the property **cryptography.encryption_algorithm**.

2.4.1.3 Replaced Possible Values of authentication.shared_secret_algorithm

The values **ecdsa-ecdh** and **dsa-dh** have been replaced with **ecdh** and **dh**, respectively. Their meanings remain the same.

2.4.1.4 Generic Security Profile Moved from BuiltinQosLibExp to BuiltinQosLib

Starting with 5.2.6 release, Generic.Security is defined under the QoS profile library BuiltinQosLib instead of BuiltinQosLibExp.

2.4.2 Wire Incompatibilities (5.2.6 and Higher)

2.4.2.1 New Way To Compute Inline QoS Keyhash in Security Plugins 5.2.6 and Higher

Release 5.2.6 and higher introduces changes in the way KeyHash inline QoS is computed for RTPS messages where the payload is encrypted.

The change applies to Data and DataFrag RTPS messages in which the payload is encrypted.

The change is that the KeyHash inline QoS associated with these messages is always computed as the 128 bit MD5 Digest (IETF RFC 1321) applied to the CDR Big-Endian encapsulation of all the Key fields in sequence independently of the length of the serialized key.

This change was made to be compliant with the latest DDS Security specification and it breaks wire compatibility between DDS 5.2.6 and previous releases.

2.4.2.2 GUID_t has Replaced BuiltinTopicKey_t in Security Plugins 5.2.6 and Higher

Release 5.2.6 and higher replaces BuiltinTopicKey_t with GUID_t in the APIs and on the wire for DDS security. For example, the definition of the ParticipantGenericMessage has changed from:

```
struct ParticipantGenericMessage {
    MessageIdentity message_identity;
    MessageIdentity related_message_identity;
    BuiltinTopicKey_t destination_participant_key;
    BuiltinTopicKey_t destination_endpoint_key;
    BuiltinTopicKey_t source_endpoint_key;
    GenericMessageClassId message_class_id;
    DataHolderSeq message_data;
};
```

To:

```
struct ParticipantGenericMessage {
    MessageIdentity message_identity;
    MessageIdentity related_message_identity;
    GUID_t destination_participant_key;
    GUID_t destination_endpoint_key;
    GUID_t source_endpoint_key;
    GenericMessageClassId message_class_id;
    DataHolderSeq message_data;
};
```

This change was done to minimize wire compatibility issues with previous releases when security is not enabled. However, the change breaks wire compatibility when enabling security.

Note: The DDS Security specification still uses BuiltinTopicKey_t. This will be fixed in future versions of the specification.

2.4.2.3 New RTPS Wire Protocol Representation in Security Plugins 5.2.6 and Higher

Release 5.2.6 and higher introduces changes in the RTPS Wire Protocol representation when using the Security Plugins. In particular, it removes the Secure Submessage (RTPS_SECURE_SUB_MSG (0x30)) and adds the following new submessages:

- Secure body submessage: RTPS_SECURE_BODY = 0x30
- Secure prefix submessage: RTPS_SECURE_PREFIX = 0x31

- Secure postfix submessage: RTPS_SECURE_POSTFIX = 0x32
- Secure RTPS prefix submessage: RTPS_SECURE_RTPS_PREFIX = 0x33
- Secure RTPS postfix submessage: RTPS_SECURE_RTPS_POSTFIX = 0x34

2.4.2.4 New Secure Liveliness Behavior in Security Plugins 5.2.6 and Higher

Release 5.2.6 and higher introduces changes in the way Secure Liveliness works.

In previous releases, Liveliness assertions for AUTOMATIC_LIVELINESS_QOS and MANUAL_BY_PARTICIPANT_LIVELINESS_QOS were done using the Secure channel only if the local Participant's **domain_rule'sliveliness_protection_kind** was not NONE and the remote participant was using Security Plugins.

Starting with 5.2.6, this behavior changed so that it matches the DDS Security specification behavior for other builtin topics: The Secure Liveliness channel will be used in the following cases:

- When using AUTOMATIC_LIVELINESS_QOS: If the TopicRule associated with a *DataWriter* has (a) **enable_liveliness_protection** to set to true, OR (b) **enable_liveliness_protection** not set and **enable_discovery_protection** set to true.
- When using MANUAL_BY_PARTICIPANT_LIVELINESS_QOS: Announcements are sent using both non-secure and secure channels if available.

The way received liveliness assertions are interpreted has also changed. Before 5.2.6, there was no filtering in the received liveliness updates. Starting in 5.2.6, *DataReaders* only consider liveliness updates received through a channel (secure or non-secure) that matches the configuration of **enable_liveliness_protection** for the *associated DataWriter*.

2.4.2.5 Authentication Handshaking

The authentication handshake messages have been updated to comply with the DDS Security specification sections 9.3.2.3.1 - 9.3.2.3.3. The following changes were made:

- Changed class_id from "DDS:Auth:ChallengeReq:ecdsa-ecdh", etc., to “DDS:Auth:PKI-DH:1.0+Req”.
- Populated adjusted_participant_key.
- Populated c.dsign_algo and c.kagree_algo.
- Moved c.id from string_properties to binary_properties.
- Moved dh1 and dh2 from (Reply and Final) to (Request and Reply).
- Removed Domain parameters p and g from the DH handshake.
- Populated and verified c.pdata.

- Populated the other side's challenge.
- Updated signature calculation.

The builtin Security Plugins do not send any of the IdentityToken, PermissionsToken, or HandshakeMessageToken properties that the specification states are optional to send.

2.4.2.6 Cryptographic Key Exchange and Transformations

The cryptographic key exchange and transformations have been updated to comply with the DDS Security specification sections 9.5.2 - 9.5.3. The following changes were made:

- Changed CryptoTokens wire representation from a list of binary_properties ("MasterKey", "MasterSalt", etc.) to the big-endian serialization of KeyMaterial_AES_GCM_GMAC.
- Populated transformation_kind, which replaces the "EncryptionAlgorithmKind" binary_property.
- Updated key material and transformation computations to match the DDS Security specification sections 9.5.3.3.2 - 9.5.3.3.6, including removal of MasterSessionSalt.
- Replaced HMAC-SHA256 with the GMAC variant of the configured cryptography.encryption_algorithm.
- Added SecureDataHeader to encoded serialized payload output.
- Added SecureDataBody.secure_data.length to encoded output.

2.4.2.7 Changes in RTI Endpoint Discovery parameter 0x8018 in Security Plugins 5.2.6 and Higher

Security Plugins 5.2.6 introduced changes in the mapping of the EndpointSecurity attributes to the Endpoint Discovery Parameter 0x8018 with respect to release 5.2.5.

Secure Endpoints will now perform consistency checks for the endpoint security attributes. This will allow for early detection of incompatible secure configurations in the Governance configuration that may have resulted in serialization/deserialization errors and/or no communication.

This feature is implemented by using the RTI endpoint discovery parameter 0x8018, which is a bitmask that includes the information of the EndpointSecurityAttributes of section 8.4.2.5 of the DDS Security specification, plus additional information about the payload protection kind and the liveliness protection:

EndpointSecurityAttribute	Bitmask Value
is_access_protected	1
is_discovery_protected	2
is_submessage_protected	4

EndpointSecurityAttribute	Bitmask Value
is_payload_signed	8
is_payload_encrypted	10
is_liveliness_protected	20

Note that this parameter is sent with the endpoint discovery data, but it is not a new member of the PublicationBuiltinTopicData or SubscriptionBuiltinTopicData structures and is therefore not exposed via the Connext DDS APIs. Also note that this parameter is specific to RTI and not part of any standard: if the parameter is not present in the discovery information received for a remote endpoint, no consistency checks will be done.

2.4.2.8 New Crypto Tokens GMCLASSIDs in Security Plugins 5.2.6 and Higher

Starting in release 5.2.6, the GMCLASSIDs used for crypto tokens have been changed to match those defined in the DDS Security specification. In particular, the new values are:

- Participant Crypto Tokens: "dds.sec.participant_crypto_tokens"
- DataWriter Crypto Tokens: "dds.sec.datawriter_crypto_tokens"
- DataReader Crypto Tokens: "dds.sec.datareader_crypto_tokens"

2.4.2.9 Support for discovery_protection_kind in Security Plugins 5.2.6 and Higher

For discovery to succeed between two Endpoints belonging to different Participants when using Secure Endpoint Discovery, their configurations for **discovery_protection_kind** must be consistent. Consequently, applications using the Security Plugins that have inconsistent **discovery_protection_kind** configurations may stop communicating and the configuration must be updated.

2.4.3 API Incompatibilities (5.2.6 and Higher)

Security Plugins 5.2.6 introduces multiple API changes:

- Renamed dds_c/dds_c_trustPlugins.h to dds_c/dds_c_trust_plugins.h.
- Renamed RTI_SECURITY_BUILTIN_PLUGIN_NAME to RTI_SECURITY_BUILTIN_PLUGIN_PROPERTY_NAME.
- Renamed DDS_GMCLASSID_SECURITY definitions to DDS_GMCLASSID_TRUST.
- Removed SecurityPluginSuite_create from security/security_default.h; replaced with RTI_Security_PluginSuite_create.

- Moved DDS_SecurityExceptionCode from dds_c/dds_c_trust_plugins.h to security/security_default.h.
- Removed struct DDS_SecurityException from dds_c/dds_c_trust_plugins.h; replaced with DDS_SecurityException under security/security_default.h.
- Moved DDS_SECURITY_EXCEPTION_INITIALIZER from dds_c/dds_c_trust_plugins.h to security/security_default.h.
- Removed struct DDS_ParticipantSecurityAttributes from dds_c/dds_c_trust_plugins.h; replaced with DDS_ParticipantSecurityAttributes under security/security_default.h.
- Moved DDS_PARTICIPANT_SECURITY_ATTRIBUTES_DEFAULT from dds_c/dds_c_trust_plugins.h to security/security_default.h.
- Removed struct DDS_EndpointSecurityAttributes from dds_c/dds_c_trust_plugins.h; replaced with DDS_EndpointSecurityAttributes under security/security_default.h.
- Moved DDS_ENDPOINT_SECURITY_ATTRIBUTES_DEFAULT from dds_c/dds_c_trust_plugins.h to security/security_default.h.
- Moved DDS_LOGGING_CRYPTOGRAPHY_CLASS and DDS_LOGGING_SECURITY_CLASS from dds_c/dds_c_trust_plugins.h to security/security_logging.h.
- Moved DDS_SecureSubmessageCategory_t from dds_c/dds_c_trust_plugins.h to security/security_cryptography.h.
- Removed all the occurrences of DDS_BuiltinTopicKey_t (defined as DDS_UnsignedLong[4]) in the Security Plugins APIs; replaced with DDS_GUID_t (defined as DDS_Octet[16]).
- Added DDS_RETCODE_NOT_ALLOWED_BY_SEC to DDS_ReturnCode_t.
- Added the following APIs to Authentication Plugin:
  - begin_auth_request()
  - process_auth_request()
  - set_local_participant_trusted_state()
  - verify_remote_participant_trusted_state()
  - get_max_signature_size()
  - private_sign()
  - verify_private_signature()

As a consequence of the changes in KeyHash inline QoS computation described in section 2.4.2.1, the code produced by previous versions of *RTI Code Generator* needs to be regenerated if you are using an IDL containing keyed types. That is, if you used an IDL containing keyed types in *Code Generator* 5.2.x or lower, then you must regenerate that code using the version of rtiddsgen provided with this release.

2.5 Incompatibilities in 5.2.5 and Higher with Previous Versions

This section describes configuration and on-the-wire incompatibilities with previous releases that have been introduced starting with 5.2.5.

2.5.1 Configuration Incompatibilities (5.2.5 and Higher)

2.5.1.1 Domain Governance File uses <allow_unauthenticated_participants> in Security Plugins 5.2.5 and Higher

The <allow_unauthenticated_join> tag is no longer supported.

In previous releases, the XML tag <allow_unauthenticated_join> was used in the Governance file. To align with the DDS Security specification, Section 9.4.1.2.2, this tag has been replaced by <allow_unauthenticated_participants>.

2.5.1.2 Removed Support for <domain_id> in Governance and Permissions Documents in Security Plugins 5.2.5 and Higher

The <domain_id> tag is no longer supported. Domain sets are now specified using domain ranges with the new <domains> tag.

In previous releases, the XML tag <domain_id> was used in both Governance and Permissions files to specify the domain ID in which the settings within these files apply. This tag has been replaced by the new <domains> tag, which can be used to specify individual domain IDs, domain ranges, open domain ranges, and combinations thereof.

Example: Individual domain IDs

```
<!-- Domains 0 and 1 -->
<domains>
    <id>0</id>
    <id>1</id>
</domains>
```

Example: Domain Ranges

```
<!-- All domains between 3 and 10, inclusive -->
<domains>
    <id_range>
        <min>3</min>
        <max>10</min>
    </id_range>
</domains>
```

Example: Open Domain Ranges

```
<domains>
    <!-- Domain 10 and above -->
    <id_range>
        <min>10</min>
    </id_range>

    <!-- Domains from 0 to 5, inclusive -->
    <id_range>
        <max>5</max>
    </id_range>
</domains>
```

2.5.1.3 New Root Tag in Permissions Document in Security Plugins 5.2.5 and Higher

The <permissions> root tag of the Permissions document is no longer supported. It has been replaced by the <dds> tag. The <permissions> tag is still mandatory and must be nested inside <dds>.

2.5.2 Wire Incompatibilities (5.2.5 and Higher)

2.5.2.1 EndpointSecurityAttributes now Sent with Endpoint Discovery in Security Plugins 5.2.5 and Higher

In previous releases, the PublicationBuiltinTopicData and SubscriptionBuiltinTopicData included a parameter, 0x0077, whose boolean value indicated the use of encryption. 0x0077 violated the RTPS specification because it is within the range of non-vendor-specific parameters, even though it really was vendor-specific. In release 5.2.5, this parameter has been replaced by an unsigned long parameter 0x8018, which is a bitmask of the EndpointSecurityAttributes of section 8.4.2.5 of the DDS Security specification.

EndpointSecurityAttribute	Bitmask Value
is_access_protected	1
is_discovery_protected	2
is_submessage_protected	4
is_payload_protected	8

Note that this parameter is sent with the endpoint discovery data, but it is not a new member of the PublicationBuiltinTopicData or SubscriptionBuiltinTopicData structures and is therefore not exposed via RTI Connext DDS APIs. Also note that this parameter is specific to RTI and not part of any standard. It is used to support scenarios in which matching endpoints have different security attributes (e.g., a writer is encrypting data but a matched reader is not encrypting ACKNACKs).

2.5.2.2 DataHolder Definition Aligns with DDS Security Specification in Security Plugins 5.2.5 and Higher

The DataHolder structure is now aligned with the definition in the DDS Security specification section 7.2.3.1. In Security Plugins releases prior to 5.2.5, the DataHolder contained additional members. This structure is used for authentication handshaking and key exchange.

2.5.2.3 IdentityToken now Sent with ParticipantBuiltinTopicData in Security Plugins 5.2.5 and Higher

In previous releases, the ParticipantBuiltinTopicData included a parameter 0x0078 whose boolean value indicated the usage of security features. In Security Plugins 5.2.5, this parameter has been replaced by the IdentityToken parameter as defined in the DDS Security specification Table 10 (parameter 0x1001). Note that this parameter is sent with the ParticipantBuiltinTopicData during participant discovery, but it is not a new member of the ParticipantBuiltinTopicData structure and is therefore not exposed via Connext DDS APIs.

2.5.2.4 Changed ParticipantBuiltinTopicData availableBuiltinEndpoints Values to Match DDS Security Specification in Security Plugins 5.2.5 and Higher

This release changes the ParticipantBuiltinTopicData availableBuiltinEndpoints' values assigned to Security Plugins endpoints. The new values match the ones defined in the DDS Security specification:

Builtin Endpoint	Bit in ParticipantBuiltinTopicData availableBuiltinEndpoints
SEDPbuiltinPublicationsSecureWriter SEDPbuiltinPublicationsSecureReader	(0x00000001 << 16) (0x00000001 << 17)
SEDPbuiltinSubscriptionsSecureWriter SEDPbuiltinSubscriptionsSecureReader	(0x00000001 << 18) (0x00000001 << 19)
BuiltinParticipantMessageSecureWriter BuiltinParticipantMessageSecureReader	(0x00000001 << 20) (0x00000001 << 21)
BuiltinParticipantStatelessMessageWriter BuiltinParticipantStatelessMessageReader	(0x00000001 << 22) (0x00000001 << 23)
BuiltinParticipantVolatileMessageSecureWriter BuiltinParticipantVolatileMessageSecureReader	(0x00000001 << 24) (0x00000001 << 25)

Chapter 3 What's New in 5.3.1

3.1 New Platforms

This release adds support for the following platforms in Security Plugins 5.3.1. These platforms have been added since 5.3.0:

Operating System	CPU	Compiler	RTI Architecture Abbreviation
OS X 10.13	x64	clang 9.0	x64Darwin17clang9.0
QNX Neutrino 7.0	ARMv8	qcc 7.0.0 with LLVM default libraries	armv8QNX7.0.0qcc_cxx5.4.0
		qcc 7.0.0 with GNU C++ libraries	armv8QNX7.0.0qcc_gpp5.4.0
	x64	qcc 7.0.0 with LLVM default libraries	x64QNX7.0.0qcc_cxx5.4.0
		qcc 7.0.0 with GNU C++ libraries	x64QNX7.0.0qcc_gpp5.4.0
RedHawk™ Linux 6.5 (Available through custom support)	i86	gcc 4.9.2	i86RedHawk6.5gcc4.9.2
	x64	gcc 4.9.2	x64RedHawk6.5gcc4.9.2
Ubuntu® 16.04 LTS	ARMv8 (AArch64)	gcc 5.4.0	armv8Linux4.4gcc5.4.0
Wind River® Linux 7	x64	gcc 4.9.1	x64WRLinux7gcc4.9.1
Windows 10 Windows Server 2008 R2 Windows Server 2016	x86	Visual Studio 2017 Update 2	i86Win32VS2017
	x64	Visual Studio 2017 Update 2	x64Win64VS2017

Chapter 4 What's Fixed in 5.3.1

This section describes bugs that have been fixed in Security Plugins 5.3.1. These fixes have been made since 5.3.0.

4.1 Fixes Related to Specification Compliance

4.1.1 Wrong Precedence of Conflicting Permissions Rules

In Connext 5.3.0 and below, if a Permissions Document contained conflicting allow and deny rules for a given domain-topic combination, then the rule that took effect was the one that appeared last in the document. This order of precedence is not the intention of the DDS Security specification. In this release, the rule that takes effect by default is the one that appears first in the document.

To change this behavior, you may set the security plugin property access_control.use_530_permissions_rules_precedence to true. (If true, then the last rule will take precedence, which is consistent with Connext 5.3.0 behavior. If false, then the first rule will take precedence, which is consistent with the intended behavior of the DDS Security specification. The default value is false.)

To avoid the question of precedence, rewrite your Permissions Document to eliminate conflicting rules.

[RTI Issue ID SEC-792]

4.2 Other Fixes

4.2.1 Incorrect Value for BUILTIN_ENDPOINT_SET when not Using Security

In releases 5.1.1 and 5.3.0, the BUILTIN_ENDPOINT_SET propagated as part of Participant discovery incorrectly set the flags for builtin secure endpoint discovery endpoints when not using security. This problem has been resolved.

[RTI Issue ID SEC-729]

4.2.2 DDS Namespace Prefix was Missing in Token Type

In previous releases, the type Token was incorrectly missing the DDS namespace prefix. This problem is now resolved.

[RTI Issue ID SEC-732]

4.2.3 Potential Crash when Matching Secure Endpoints Belonging to Removed Remote Participant

There was a rare race condition that may have resulted in a crash. In particular, this issue may have occurred while matching secure endpoints for a remote participant at the same time the remote participant left the system. This problem has been resolved.

[RTI Issue ID SEC-741]

4.2.4 Memory Corruption when Writing >65kB Unfragmented Samples and Using Metadata or RTPS Message Protection

Memory corruption occurred when all of the following conditions were true:

- **metadata_protection_kind** = SIGN or ENCRYPT or **rtps_protection_kind** = SIGN or ENCRYPT
- **message_size_max** > 65535. This is possible when using the TCP transport.
- The user wrote unfragmented samples of size greater than 65kB but less than **message_size_max**.

Memory corruption occurred in the functions **encode_datawriter_submessage()**, **encode_datareader_submessage()**, and **encode_rtps_message()**, and the application crashed with a general protection fault. This problem has been resolved. Instead of corrupting memory, these conditions will result in a warning message, followed by a failure to write the large sample. In order to write the large sample, you must set **message_size_max** to be smaller than the message size, so the sample can be put in fragments smaller than 65 kB. See 6.2 No Support for Writing >65kB Unfragmented Samples Using Metadata or RTPS Message Protection on page 34.

[RTI Issue ID SEC-767]

4.2.5 Data Fragmentation not Working for Certain Values of message_size_max when Using Security

RTPS message serialization may have failed when fragmentation was required. This was only an issue when using security, only for certain **message_size_max** values, and only if the RTPS message or

submessage needed to be encrypted or signed. When this problem triggered, a message similar to the following was logged:

```
MIGGeneratorContext_flush:Secure DDS does not support fragmenting
encoded messages that would be unfragmented if unencoded.
Please either increase or decrease your transport message_size_max by 204,
but do not increase it to more than 65507. If you are using batching,
please reduce your max_data_bytes.
COMMENDSrWriterService_agentFunction:!addDataFrag
```

This problem has been resolved.

[RTI Issue ID SEC-770]

4.2.6 Changes in Reliable Remote Reader Liveliness Not Reported when Using Security

There was an issue that may have prevented remote reader liveliness changes from being reported. For example, this issue may have prevented the **on_publication_matched()** and **on_liveliness_changed()** callbacks from being triggered upon remote reader discovery. This problem, which only affected reliable readers using security, has been resolved.

[RTI Issue ID SEC-773]

4.2.7 Security Log File did not Immediately Show Log Messages

When directing security log messages to a file, the log file did not show log messages as soon as they were generated. The log messages were flushed to the file only after either a certain number of log messages were generated or the DomainParticipant was deleted. This problem has been resolved.

[RTI Issue ID SEC-774]

4.2.8 Rare Segmentation Fault when Deleting DomainParticipant that Distributed Security Log Messages

If a *DomainParticipant* was using Security Plugins and distributing log messages with the built-in logging topic, a segmentation fault may have occurred in rare cases in the function **RTI_Security_LoggingQueue_lock()** when deleting the *DomainParticipant*. This problem has been resolved.

[RTI Issue ID SEC-778]

4.2.9 Unbounded Memory Growth when New Participants Joined and Left System

There was an issue that may have caused unbounded memory growth if new participants were continuously joining and leaving the system. This problem, which only affected participants with security enabled, has been resolved.

[RTI Issue ID SEC-782]

4.2.10 Failed Re-Authentication May Have Prevented Communication

In previous releases, a failure during participant re-authentication (for example, as a result of network corruption), may have prevented communication from recovering until participants were destroyed and re-created. This problem has now been resolved.

[RTI Issue ID SEC-795]

4.2.11 Missing security_authentication.h Header

In previous releases, the security_authentication.h header was missing from Security Plugins and Security SDK bundles. While the absence of this header was inconsistent with the rest of Security Plugins, it had no impact on the functionality of the plugins, and it did not affect the Security Plugins SDK build. This problem has now been resolved: for consistency, the security_authentication.h header has now been added to Security Plugins and Security PluginsSDK bundles.

[RTI Issue ID SEC-806]

4.2.12 Keyed Sample Lost When Decoding Serialized Data Failed

When a secure *DataReader* using a **data_protection_kind** other than NONE receives a keyed sample, the *DataReader* may fail to process the sample due to an error decoding the sample. When this happens, the *DataReader* does not deliver the sample to the application, and the sample is rejected as expected.

In previous releases, secure reliable *DataReaders* were incorrectly skipping samples rejected in this scenario, instead of requesting a repair. As a result, the rejected sample was never repaired. Note that this problem had no impact on subsequent samples, which could still be received with no issues.

Unless there was network corruption, this problem was unlikely. If network corruption had occurred, the problem could have been prevented by setting **metadata_protection_kind** or **rtps_protection_kind** to SIGN or ENCRYPT.

This problem has been resolved: secure reliable *DataReaders* now request repairs of samples rejected in this scenario.

[RTI Issue ID SEC-809]

Chapter 5 Previous Release

5.1 What's New in 5.3.0

This section highlights improvements in 5.3.0. These enhancements have been made since 5.2.3.

5.1.1 RTI Security Plugins now Wire Aligned with DDS Security Specification

Security Plugins are now wire aligned with the OMG DDS-Security specification, which allows for forward compatibility with future versions of Security Plugins.

For a detailed list of the specification compliance fixes included in this release, please see 5.2.1 Fixes Related to Specification Compliance on page 26. For a detailed list of the wire-representation changes introduced in this release, please see Compatibility (Chapter 2 on page 2).

5.1.2 Authentication and Discovery

5.1.2.1 Changes in Authentication Behavior

This release introduces multiple changes to Authentication behavior for specification compliance, robustness, and scalability. The list of changes is as follows:

- Authentication will no longer fail upon getting a VALIDATION_FAILED return code from the Authentication plugins. Instead, the Authentication state machine will remain in the same state it was in before getting the VALIDATION_FAILED. This way, the local participant can complete the Authentication process with the remote participant if the proper handshake is received at a later point.
- Authentication will only fail if the Authentication timeout expires. Please see the section on Authentication in the *RTI Security Plugins Getting Started Guide* for more information about how to configure the Authentication timeout.

- Upon a failed Authentication, the remote participant will no longer be ignored. Instead, the remote participant will be removed from the local participant. This way, the remote participant will have a chance of authenticating later.
- Receiving a duplicate to an already received Handshake Message will no longer trigger a new call to the Authentication plugin. Instead, the Authentication state machine will remain in the same state.

5.1.2.2 Support for Securely Re-Authenticating Remote Participants

This releases introduces a secure re-authentication capability as an extension to the DDS Security specification.

Current DDS Security specification does not define a mechanism for re-authenticating with a Participant for which the local Participant had established a previous authentication. This is needed to be able to recover communication in scenarios where there is asymmetric liveliness loss.

Asymmetric liveliness loss occurs when one of the Participants loses liveliness with the other Participant, and therefore cleans up all the associated state, while the other Participant still keeps the authenticated state. Asymmetric liveliness loss will lead to communication not recovering.

Using re-authentication, Participants that have run into an asymmetric liveliness loss scenario are able to establish a new authenticated session and re-discovery all the involved endpoints, allowing communication to recover.

For more information about this feature, please see the section on Authentication in the *Security Plugins Getting Started Guide*.

5.1.2.3 New Properties for Tuning Authentication

This release introduces two new properties for tuning Authentication:

- **dds.participant.trust_plugins.authentication_timeout.sec**: This property controls the maximum time in seconds that an ongoing authentication can remain without completing. After this timeout expires, the authentication process is canceled, and associated resources are released.
- **dds.participant.trust_plugins.authentication_request_delay.sec**: This property controls the delay in seconds before sending an authentication_request to the remote participant. See the section on Authentication in the *Security Plugins Getting Started Guide* for more details about the authentication_request mechanism for re-authenticating remote participants.

5.1.2.4 New Return Code to Replace Deprecated DDS_VALIDATION_FINAL_MESSAGE

This release introduces a new return code, DDS_VALIDATION_OK_FINAL_MESSAGE. You should use this instead of the now deprecated DDS_VALIDATION_FINAL_MESSAGE return code. While this release supports both definitions, future releases may drop support for DDS_VALIDATION_FINAL_MESSAGE.

5.1.2.5 Participant Discovery Mutability Support for Authenticated Participants

This release introduces a secure way to propagate participant discovery updates as an extension to the DDS Security specification.

Current DDS Security specification does not define a mechanism for validating received participant discovery upon authentication completion, neither it defines a way for securely propagating participant discovery updates after authentication is completed. These two mechanisms are needed to be able to support secure participant discovery updates (e.g., for supporting changes in IP addresses after authenticating a participant).

The *Security Plugins* now support these two mechanisms, for more information about them, please refer to the section on *Protecting Participant Discovery* in the *RTI Security Plugins Getting Started Guide*.

5.1.2.6 New Pure Stateless Mode for Participant Discovery Reader

This release introduces a new pure stateless mode for the local Simple Participant Discovery reader. This mode allows Simple Participant Discovery to be robust against Sequence Number Attacks. This mode is disabled by default. It can be enabled by setting the Participant QoS's **dds.participant.discovery_config.use_stateless_participant_discovery_reader** property to true.

5.1.2.7 IdentityToken and PermissionsToken now Sent with ParticipantBuiltinTopicData

In previous releases, the ParticipantBuiltinTopicData included a parameter, 0x0078, whose boolean value indicated the usage of security features. In this release, this parameter has been replaced by the IdentityToken parameter as defined in the DDS Security specification Table 10 (parameter 0x1001).

Additionally, in this release the PermissionsToken parameter is now sent as part of participant discovery as defined in the DDS Security specification Table 10 (parameter 0x1002).

Note that these parameters are sent with the ParticipantBuiltinTopicData during participant discovery, but they are not new members of the ParticipantBuiltinTopicData structure and are therefore not exposed via Connext DDS APIs.

5.1.2.8 Early Detection of Inconsistent Secure Configuration for Endpoints

Secure Endpoints will now perform consistency checks for the endpoint security attributes. This will allow for early detection of incompatible secure configurations in the Governance configuration that may have resulted in serialization/deserialization errors and/or no communication.

For more information, see 2.4.2.7 Changes in RTI Endpoint Discovery parameter 0x8018 in Security Plugins 5.2.6 and Higher on page 8.

5.1.2.9 Increased Local Stateless and Secure Volatile Reader Max. Samples

In order to speed up discovery times, this release increases the maximum samples for the local stateless and secure volatile readers from 4 samples to unlimited.

5.1.3 Access Control

5.1.3.1 Support for Domain Ranges in Domain Governance and DomainParticipant Permissions Files

In previous releases, the XML tag <domain_id> was used in both Governance and Permissions files to specify the domain ID in which the settings within these files apply. This tag has been replaced by the new <domains> tag, which can be used to specify individual domain IDs, domain ranges, open domain ranges, and combinations thereof.

Example: Individual domain IDs

```
<!-- Domains 0 and 1 -->
<domains>
    <id>0</id>
    <id>1</id>
</domains>
```

Example: Domain Ranges

```
<!-- All domains between 3 and 10, inclusive -->
<domains>
    <id_range>
        <min>3</min>
        <max>10</min>
    </id_range>
</domains>
```

Example: Open Domain Ranges

```
<domains>
    <!-- Domain 10 and above -->
    <id_range>
        <min>10</min>
    </id_range>

    <!-- Domains from 0 to 5, inclusive -->
    <id_range>
        <max>5</max>
    </id_range>
</domains>
```

5.1.3.2 Support for Lists of Alternative CA and Permissions Authority Files

This release supports specifying a comma-separated list of alternative CA certificates (through the new **com.rti.serv.secure.authentication.alternative_ca_files** property) and a comma-separated list of alternative Permissions authority certificates (through the new **com.rti.serv.secure.access_control.alternative_permissions_authority_files** property).

If the verification of a file fails with the main certificate (ca_file or permissions_authority_file), it will be retried with all of the corresponding alternative certificates. If none of the alternative certificates can be used to verify the file, the verification process will fail.

5.1.3.3 Support for Extensible DomainParticipant Permissions Documents

The validation of remote DomainParticipant Permissions documents was strict (i.e., Connext DDS did not allow unsupported XML tags). Starting with 5.3.0, all unexpected tags found in remote DomainParticipant Permissions documents will be ignored by the local *DomainParticipant*.

5.1.4 Cryptography

5.1.4.1 New Encryption Algorithm for Key Exchange

This release changes the encryption algorithm for key exchange from aes-128-gcm to aes-256-gcm. This change was made following the proposal under OMG's issue DDSSEC11-53 to disambiguate the latest DDS Security specification Table 52 transformation_kind.

This change breaks wire compatibility between Connext DDS 5.2.7 and previous releases, as described in 2.3.2.5 New Encryption Algorithm for Key Exchange on page 4.

5.1.4.2 Receiver-Specific MACs

This release introduces support for receiver-specific Message Authentication Codes (MACs) as defined in the DDS Security specification Section 9.5.2.5, with background information in Section 7.1.1.3.

You can configure the maximum number of receiver-specific MACs using the Cryptography Plugin property **max_receiver_specific_macs**. For more information, see the *RTI Security Plugins Getting Started Guide* (Table 8.1 Properties for Configuring Cryptography).

5.1.4.3 Received Endpoint Discovery now Protected

In previous releases, the **enable_discovery_protection** setting only affected the endpoint discovery information sent to other participants. **enable_discovery_protection** had no effect on the received endpoint discovery. While this behavior was consistent with current DDS-SECURITY specification, it left endpoint discovery vulnerable to outsiders injecting unauthorized endpoint discovery traffic if **rtps_protection_kind** was set to NONE.

Starting with this release, enabling discovery protection for a topic will also protect received endpoint discovery. This way, if a topic is configured with **enable_discovery_protection** set to true, that topic's discovery information received through the non-secure endpoint discovery topics will be dropped.

5.1.4.4 Partial Support for Domain Rule's discovery_protection_kind

This release adds partial support for the Governance Domain Rule's **discovery_protection_kind** field. In particular, this field now supports two values:

- NONE: RTPS submessages for Secure Endpoint Discovery topics will be sent unprotected.
- ENCRYPT: RTPS submessages for Secure Endpoint Discovery topics will be sent encrypted.

For discovery to succeed between two Endpoints belonging to different Participants when using Secure Endpoint Discovery, the configuration for **discovery_protection_kind** must be consistent.

5.1.5 Logging

5.1.5.1 Decreased Verbosity for Messages Reporting Failed Submessage Decoding

In previous releases, the logging message for reporting a failed submessage decode had EXCEPTION level (in core libraries) and SEVERE level (in the logging plugin). This release decreases the verbosity for those messages from EXCEPTION to REMOTE and from SEVERE to ERROR, respectively.

5.1.5.2 Decreased Verbosity for Messages Reporting Denied Remote Participant

In previous releases, the logging message for reporting a denied remote participant had ERROR level (in the logging plugin). This release decreases the verbosity for those messages from ERROR to WARNING.

5.1.6 New Name for RTI Security Plugins Library

The name for the RTI Security Plugins library changed from **rtisecurity** to **nddssecurity**.

5.1.7 Support for RTPS-HMAC-Only Protection Mode

RTPS-HMAC-Only Protection Mode allows RTPS messages to be signed with a user-provided HMAC key while disabling all other security features (authentication, access control, and encryption). To set up the behavior of the RTPS-HMAC-Only mode, the following properties have been added (assuming you use **com.rti.serv.secure** as the alias to load the plugin):

- **com.rti.serv.secure.hmac_only.enabled** (boolean, optional): Enables or disables the HMAC-only mode (default: false)
- **com.rti.serv.secure.hmac_only.cryptography.key** (string, mandatory): Sets the static HMAC key used to compute message signatures. The HMAC key can be either a plain text string or an arbitrary binary string:
  - Plain text HMAC keys are case sensitive and must start with the prefix **str:** (e.g.: str:Some secret key string)

  - Binary HMAC keys must be provided as a sequence of upper- or lower-case hexadecimal digits prefixed by **hex:** (e.g.: hex:1489a95de3873df5).
- **com.rti.serv.secure.hmac_only.cryptography.max_blocks_per_session** (integer, optional): For signing RTPS messages, HMAC-only mode uses a key derived from the HMAC key and a sessionId that is serialized as part of the signed RTPS message representation. This property sets the number of message blocks that can be signed with the same sessionId. The current message block size is fixed at 32 bytes (default: 0xffffffffffffffff)

Empty keys (either string or binary) are not allowed. The maximum HMAC key size is bounded by the maximum property size, controlled by the DomainParticipant resource limit **participant_property_string_max_length**.

For more information, see the section on *RTPS-HMAC-Only Mode* in the *Security Plugins Getting Started Guide*.

5.1.8 Secure Service Request Built-in Channel

This release introduces a new Secure Service Request built-in channel for securely sending Topic Queries and Locator Reachability Response messages.

The rules for using this channel depend on the usage:

- Topic Queries are sent over the secure channel if the associated *DataReader* is configured with **enable_discovery_protection** set to true in the Governance file.
- Locator Reachability Response messages are sent over the secure channel if both the local and remote participant are using security.

For more information, see Section 8.1 Related Governance Attributes in the *Security Plugins Getting Started Guide*.

5.1.9 Generic Security Profile Moved from BuiltinQosLibExp to BuiltinQosLib

Starting with this release, **Generic.Security** is now defined under the QoS profile library **BuiltinQosLib**.

5.1.10 Platforms on Legacy Operating Systems

The following legacy operating systems have reached end-of-life from their corresponding vendors. Please contact RTI support or your account manager if you require version 5.3 to run on these platforms:

- CentOS 5.x
- Red Hat Enterprise Linux 5.x

- Wind River Linux 4

5.2 What's Fixed in 5.3.0

This section describes bugs that have been fixed in Security Plugins 5.3.0. These fixes have been made since 5.2.3.

5.2.1 Fixes Related to Specification Compliance

5.2.1.1 Wrong ParticipantBuiltinTopicData availableBuiltinEndpoints Values Used by Security Plugins

In 5.1.1.4 and earlier releases, ParticipantBuiltinTopicData included wrong values for availableBuiltinEndpoints. This has been fixed and now the correct values are sent. See 2.5.2.4 Changed ParticipantBuiltinTopicData availableBuiltinEndpoints Values to Match DDS Security Specification in Security Plugins 5.2.5 and Higher on page 13 for more details.

[RTI Issue ID SEC-298]

5.2.1.2 Wrong Root Tag in Permissions Document

In 5.1.1.4 and earlier releases, the root tag of the permissions document was <permissions>, but according to the DDS-SECURITY specification, the root tag should be <dds>, and <permissions> should appear as a nested element under <dds> tag. This problem has been resolved.

[RTI Issue ID SEC-300]

5.2.1.3 Wrong PID Used by Security Plugins

In 5.1.1.4 and earlier releases, the PublicationBuiltinTopicData and SubscriptionBuiltinTopicData included a parameter 0x0077 whose boolean value indicated the use of encryption. 0x0077 violated the RTPS specification because it is within the range of non-vendor-specific parameters, even though it really was vendor-specific.

In this release, this parameter has been replaced by an unsigned long parameter 0x8018, which is a bitmask of the EndpointSecurityAttributes of section 8.4.2.5 of the DDS Security specification. See Chapter 2 Compatibility on page 2.

[RTI Issue ID SEC-400]

5.2.1.4 Wrong Message ID and Related Fields Sent in Security Plugins Builtin Channel Messages

In 5.1.1.4 and earlier releases, the message identity and related message-identity fields in the ParticipantGenericMessage samples sent on Security Plugins builtin channels were not properly populated. This problem has been resolved.

[RTI Issue ID SEC-406]

5.2.1.5 RTI Security Plugins Liveliness Channel did not Match DDS Security Specification

In 5.1.1.4 and earlier releases, the Secure Liveliness channel behavior was not consistent with the DDS Security specification. In particular, it was inconsistent with the behavior for other builtin topics as the secure endpoint discovery topics. This problem has been resolved. For more information, see 2.4.2.4 New Secure Liveliness Behavior in Security Plugins 5.2.6 and Higher on page 7.

[RTI Issue ID SEC-462]

5.2.1.6 Permissions and Governance Documents not Compliant with DDS Security Specification

In 5.1.1.4 and earlier releases, Permissions and Governance documents were not compliant with the DDS Security specification. Starting with 5.2.6, these files are fully compliant with the specification. For more information, see 2.4.1.1 Changed Permissions and Governance Document Definitions to be Compliant with DDS Security Specification on page 4.

[RTI Issue ID SEC-596]

5.2.1.7 Builtin Logging Plugin not Compliant with DDS Security Specification

In 5.1.1.4 and earlier releases, the only members in the Builtin Logging IDL type were **source_guid**, **log_level**, **message**, and **category**. Starting with version 1.0 of the DDS Security specification (section 9.6 Builtin Logging Plugin), the IDL type was modified to be compatible with the syslog specification (RFC-5424). The builtin logging plugin has been updated accordingly.

[RTI Issue ID SEC-607]

5.2.1.8 Wrong Behavior when Allowing for Unauthenticated Participants

In 5.1.1.4 and previous releases, a local participant with **allow_unauthenticated_participants** set to true may have discovered and communicated with secure endpoints belonging to unauthenticated participants. This was not compliant with the DDS Security specification, which disallows discovering secure endpoints belonging to not fully authenticated participants. This problem has been resolved.

[RTI Issue ID SEC-633]

5.2.1.9 Governance Attributes enable_read/write_access_control not Enforced on Remote Endpoints

In 5.1.1.4 and earlier releases, the Access Control governance attributes **enable_read_access_control** and **enable_write_access_control** had no effect on the enforcement of permissions on remotely discovered DataReaders and DataWriters. Those permissions were enforced if and only if **enable_join_access_control** was set to TRUE.

Now, **if enable_read_access_control** is set to TRUE for a given topic, the local permissions are enforced on locally created DataReaders, and the remote permissions are enforced on remotely discovered DataReaders. Similar logic applies to **enable_write_access_control** and DataWriters. **enable_join_access_control** no longer affects the enforcement of permissions on DataReaders and DataWriters.

[RTI Issue ID SEC-660]

5.2.1.10 Wrong RSA Signature and Verification

In 5.1.1.4 and earlier releases, when using an RSA private key and RSA certificate, the digital signature and verification were not compliant with the following sentence from the DDS Security specification:

> The digital signature shall be computed using the RSASSA-PSS algorithm specified in PKCS #1 (IETF 3447) RSA Cryptography Specifications Version 2.1 [44], using SHA256 as hash function, and MGF1 with SHA256 (mgf1sha256) as mask generation function.

This problem has been resolved.

[RTI Issue ID SEC-667]

5.2.1.11 Log Levels of Security Logging Plugin Messages did not Match Specification

In previous releases, log levels of Security Logging Plugin messages did not match the specification. This did not affect interoperability, as the integer representation of the log levels remained unchanged. This problem has been resolved. For more information, please see 2.2.1 API Incompatibilities (5.3.0 and Higher) on page 3.

[RTI Issue ID SEC-686]

5.2.1.12 Liveliness Not Interoperable with Other Vendors when Using Security Plugins

In previous releases, there was an issue that impacted interoperability with other vendors for endpoints enabling liveliness protection.

This problem, which did not affect interoperability among applications running different versions of Security Plugins, has been resolved.

[RTI Issue ID SEC-689]

5.2.1.13 Missing Bits in ParticipantBuiltinTopicData availableBuiltinEndpoints Values when Using Security Plugins

In previous releases, ParticipantBuiltinTopicData's availableBuiltinEndpoints mask was incorrectly missing the bits for Secure Volatile and Participant Stateless endpoints. This has been resolved and now the correct values are sent.

[RTI Issue ID SEC-709]

5.2.2 Other Fixes in 5.3.0

5.2.2.1 Potential Crash when Receiving Security Plugins Handshake Messages

In 5.1.1.4 and earlier releases, there may have been a segmentation fault upon receiving a Security Plugins handshake message. This problem, which was very unlikely to occur, has been resolved.

[RTI Issue ID SEC-439]

5.2.2.2 Potential Communication Loss when using Non-Robust Custom Authentication Plugin

The previous release of Security Plugins, 5.1.1.4, introduced an issue that may have caused communication loss when using a non-robust custom authentication plugin. In particular, this issue may have occurred when receiving out-of-order authentication handshake messages. This problem has been resolved.

[RTI Issue ID SEC-441]

5.2.2.3 Writer AUTOMATIC and MANUAL_BY_PARTICIPANT Liveliness did not work Between Secure and Non-Secure Participants

In 5.1.1.4 and earlier releases, there was an issue that prevented the exchange of liveliness samples between non-secure participants and secure participants configured with **allow_unauthenticated_participants=true**. As a consequence, AUTOMATIC and MANUAL_BY_PARTICIPANT Liveliness did not work in this scenario. This problem has been resolved.

[RTI Issue ID SEC-459]

5.2.2.4 Security Plugins Errors not Logged Using Logging Infrastructure

In 5.1.1.4 and earlier releases, the Security Plugins used **printf()** for logging some errors, instead of using the Connext DDS logging infrastructure. This prevented you from being able to configure advanced logging functionalities, such as logging those errors to an output file. This problem has been resolved.

[RTI Issue ID SEC-465]

5.2.2.5 Builtin Logging Plugin did not Distribute all Log Security-Related Messages

In 5.1.1.4 and earlier releases, due to thread-safety and concurrency restrictions, the Builtin Logging plugin was not able to distribute all log security-related messages. Starting with this release, the Builtin Logging Plugin DataWriter runs in a separate thread, fixing this issue.

Additionally, you can configure the behavior of the Builtin Logging Thread with the following properties (assuming you used **com.rti.serv.secure** as the alias to load the plugin):

- **com.rti.serv.secure.logging.distribute.enable:** Replaces the previous com.rti.serv.secure.logging.distribute, controls whether security-related log messages should be distributed using DDS. Boolean. Default: false.
- **com.rti.serv.secure.logging.distribute.profile:** QoS Library and QoS profile used to create logging-related entities (*Publisher*, *Topic*, and *DataWriter*). Must be a string of the format QosLibraryName::QosProfileName. String. Default: empty string (uses default QoS profile).
- **com.rti.serv.secure.logging.distribute.writer_history_depth:** History depth (in samples) of the logging *DataWriter*. Integer. Default: 64.
- **com.rti.serv.secure.logging.distribute.writer_timeout:** Number of milliseconds to wait before giving up trying to write a log message. This property overwrites the **max_blocking_time** QoS of the logging *DataWriter*. Integer. Default: 5 seconds.
- **com.rti.serv.secure.logging.distribute.queue.size:** Size of the logging thread queue, in bytes. Integer. Default: 50688.
- **com.rti.serv.secure.logging.distribute.queue.message_count_max:** Maximum number of log messages in the logging queue. Integer. Default: 64.
- **com.rti.serv.secure.logging.distribute.queue.message_size_max:** Maximum serialized size of a log message in the logging queue. Integer. Default: 792.
- **com.rti.serv.secure.logging.distribute.thread.message_threshold:** Number of bytes to preallocate for the logging message string in the logging thread, beyond which dynamic allocation will occur. Integer. Default: 256.
- **com.rti.serv.secure.logging.distribute.thread.plugin_method_threshold:** Number of bytes to preallocate for the plugin method string in the logging thread, beyond which dynamic allocation will occur. Integer. Default: 256.
- **com.rti.serv.secure.logging.distribute.thread.message_threshold:** Number of bytes to preallocate for the plugin class string in the logging thread, beyond which dynamic allocation will occur. Integer. Default: 256.

All of the above properties are optional.

[RTI Issue ID SEC-487]

5.2.2.6 Potential Decryption Failure or Segmentation Fault when Using Batching

In 5.1.1.4 and earlier releases, a potential race condition may have caused decryption failures or, in rare cases, a segmentation fault. This issue only affected scenarios using Batching, with DDS_ BatchQosPolicy's **thread_ safe_write** set to false. This problem has been resolved.

[RTI Issue ID SEC-511]

5.2.2.7 Potential Decryption Failure or Segmentation Fault when Remote Endpoint Left the System

In 5.1.1.4 and earlier releases, a potential rare race condition may have caused submessage decryption failures or a segmentation fault. In particular, this issue may have occurred after a remote endpoint left the system. This problem has been resolved.

[RTI Issue ID SEC-513]

5.2.2.8 Incorrect Number of Publications Reported when Using Secure Endpoints and Multichannel

In 5.1.1.4 and earlier releases, there was an issue that may have caused the DataReader to report an incorrect number of matched publications when using the Security Pluginsand Multichannel in the remote DataWriter. This problem has been resolved.

[RTI Issue ID SEC-588]

5.2.2.9 Secure Volatile Channel not Secure when Communicating with Local Participant

In 5.1.1.4 and earlier releases, the Secure Volatile Channel did not encrypt the samples sent to the local Participant. Consequently, crypto tokens for the local Participant and Endpoints may have been sent on the network unprotected. This issue only occurred if there was at least one pair of matching secure *DataWriters* and *DataReaders* in the same Participant. This problem has been resolved.

[RTI Issue ID SEC-624]

5.2.2.10 Potential Incorrect Publication/Subscription Matched Status when Endpoints Leave and Join the System

In 5.1.1.4 and earlier releases, there was an issue that may have caused the content of Publication/Subscription Matched Status to be incorrect. In particular, this issue may have been triggered in a scenario where multiple secure endpoints leave and join the system. This problem has been resolved.

[RTI Issue ID SEC-654]

5.2.2.11 No Communication between Secure Endpoints that had Incompatible QoS upon Initial Discovery

In 5.1.1.4 and earlier releases, there was an issue that prevented communication between compatible Secure Endpoints that were incompatible when they discovered each other for the first time. This problem has been resolved.

[RTI Issue ID SEC-657]

5.2.2.12 Wrong Log Level When Using a Logging Device

In 5.1.1.4 and earlier releases, when using the Security Plugins in conjunction with an NDDS_Config_LoggerDevice, the levels of the log messages were incorrect. This problem has been resolved.

[RTI Issue ID SEC-666]

5.2.2.13 Not Safe to Call DDS Functions within on_publication_matched(), on_subscription_matched(), on_liveliness_changed()

In 5.1.1.4 and earlier releases, it was not safe to call to DDS functions within **on_publication_matched()**, **on_subscription_matched()**, and **on_liveliness_changed()** when security was enabled. In particular, calling DDS functions may have triggered a deadlock. This problem has been resolved.

[RTI Issue ID SEC-673]

5.2.2.14 Data Fragment Submessages were not Encrypted

In 5.1.1.4 and earlier releases, when using fragmentation of large data samples, Security Plugins did not encrypt most data fragment submessages, even if you configured **metadata_protection_kind** to ENCRYPT. This problem did not affect the **data_protection_kind** setting. This problem has been resolved.

[RTI Issue ID SEC-675]

5.2.2.15 Unnecessary Traffic for Non-Secure Builtin Endpoints when Not Allowing Unauthenticated Participants

In 5.2.6 and 5.2.7, a secure Participant with **allow_unauthenticated_participants** set to false may have exchanged unnecessary traffic with unsecure Participants. This traffic was associated with non-secure liveliness and non-secure service request builtin topics. This issue did not affect discovery or user data endpoints. This problem has been resolved.

[RTI Issue ID SEC-695]

5.2.2.16 Segmentation Fault when Creating Secure DomainParticipants in Multiple Threads on QNX Systems

On QNX systems, the creation of Security-enabled *DomainParticipants* was not thread-safe and may have led to a segmentation fault in the function **RTIOsapiSemaphore_take()**. This problem has been resolved for all RTI Security Plugins architectures.

[RTI Issue ID SEC-700]

Chapter 6 Known Issues

6.1 No Support for ECDSA-ECDH with Static OpenSSL Libraries and Certicom Security Builder

If you are using the Certicom® Security Builder® engine, you cannot use the ecdsa-ecdh shared secret algorithm together with static OpenSSL libraries. If you want to use ecdsa-ecdh with Certicom Security Builder, you must use dynamic OpenSSL libraries. Attempting to use ecdsa-ecdh with static OpenSSL libraries and Certicom Security Builder will cause the following errors during participant discovery:

```
Authentication_compute_sharedsecret:failed to provide remote DP public key
Authentication_process_handshake:key generation fail
Authentication_get_shared_secret:empty secret
PRESParticipant_authorizeRemoteParticipant:!security function get_shared_secret
```

6.2 No Support for Writing >65kB Unfragmented Samples Using Metadata or RTPS Message Protection

The following use case is not supported:

- **metadata_protection_kind** = SIGN or ENCRYPT or **rtps_protection_kind** = SIGN or ENCRYPT
- **message_size_max** > 65535. This is possible when using the TCP transport.
- The user is writing unfragmented samples of size greater than 65kB but less than **message_size_max**.

In order to write the large sample, you must set **message_size_max** to be smaller than the message size, so the sample can be put in fragments smaller than 65 kB.

[RTI Issue ID SEC-768]

6.3 Spy and Ping do not Support Security Plugins' Distributed Logging

Spy and Ping do not support enabling the distribution of security-related log messages through the builtin DDS:Security:LogTopic topic.

[RTI Issue ID CORE-8300]