

RTI Security Plugins

Release Notes

Version 6.0.1



© 2019 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
November 2019.

Trademarks

RTI, Real-Time Innovations, Connex, NDDS, the RTI logo, 1RTI and the phrase, “Your Systems. Working as one,” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

Securing a distributed, embedded system is an exercise in user risk management. RTI expressly disclaims all security guarantees and/or warranties based on the names of its products, including Connex DDS Secure, RTI Security Plugins, and RTI Security Plugins SDK. Visit www.rti.com/terms for complete product terms and an exclusive list of product warranties.

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

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	1
Chapter 2 Compatibility	3
Chapter 3 What's New in 6.0.1	
3.1 New platforms	4
3.2 Removed platforms	4
3.3 Updated OpenSSL Version	4
3.4 Support for loading private keys from an OpenSSL Engine	5
3.5 More detailed messages for OpenSSL errors	5
3.6 Increased verbosity level of benign log messages	5
3.7 Updated message logged upon authentication negotiation timeout	6
3.8 New Logging Plugin property to configure verbosity level	6
3.9 Changes related to authentication plugin configuration	6
3.10 Property key_material_key now required for Secure Persistence Service	7
3.11 Improved HMAC-only mode key derivation algorithm	7
Chapter 4 What's Fixed in 6.0.1	
4.1 Fixes Related to Discovery and Authentication	8
4.1.1 Possible segmentation fault when receiving malformed ParticipantBuiltinTopicData and using Security Plugins	8
4.1.2 Possible lack of SUBSCRIPTION_MATCHED_STATUS if a DataWriter lost liveliness with DataReader	8
4.1.3 DataWriter did not report PUBLICATION_MATCHED_STATUS for DataReaders that were inactive when it received their key material	9
4.1.4 Couldn't use "file:" prefix for CRL file	9
4.1.5 Stateless participant property may not have been correctly updated	9
4.1.6 Segmentation fault when receiving a handshake message with incorrect message_class_id length	10
4.1.7 Possible bus error or use of uninitialized value when receiving handshake message with	10

incorrect inline QoS parameter length	
4.1.8 Potential Crash upon receiving malformed authentication fragment	10
4.2 Fixes Related to Logging	11
4.2.1 Some messages marked only as ERRORS	11
4.2.2 Security messages not logged, regardless of verbosity configured at DomainParticipantFactory level	11
4.3 Other Fixes	11
4.3.1 Too much memory allocated for non-payload-encoding DataWriters	11
4.3.2 Unexpected warning when using shared memory and security	12
4.3.3 Possible deadlock risk when using TopicQueries and Security	12
4.3.4 Applications directly calling OpenSSL APIs after DomainParticipant deletion may have crashed	12
Chapter 5 Previous Release	
5.1 What's New in 6.0.0	13
5.1.1 New Platforms	13
5.1.2 Updated OpenSSL Version	13
5.1.3 Changes Related to Specification Compliance	13
5.1.3.1 Authentication and Discovery	13
5.1.3.1.1 Option to specify private key password	13
5.1.3.1.2 Added support for new participant and endpoint security information	14
5.1.3.1.3 Added new DCPSParticipantsSecure builtin topic	15
5.1.3.1.4 Support for DDS Security 1.1 AuthRequest	15
5.1.3.2 Access Control	16
5.1.3.2.1 Updated PermissionsToken class_id	16
5.1.3.2.2 Added check_remote_topic	16
5.1.3.2.3 Added functions to return security attributes	16
5.1.3.2.4 Data tagging	16
5.1.3.2.5 Support for DDS Security Topic Security Attributes	16
5.1.3.2.6 Added Support for builtin topics to get_datawriter_sec_attributes and get_datareader_sec_attributes APIs	17
5.1.3.2.7 <topics> now mandatory in permissions file	17
5.1.3.2.8 Updated matching behavior of allowed partitions condition	17
5.1.3.3 Cryptography	18
5.1.3.3.1 New protection kinds	18
5.1.4 Other Changes	18

5.1.4.1	Authentication and Discovery	18
5.1.4.1.1	Identity Certificate chaining	18
5.1.4.2	Cryptography	18
5.1.4.2.1	Support using different writer keys for protecting submessages and serialized data	18
5.1.4.2.2	WITH_ORIGIN_AUTHENTICATION protection kinds	19
5.1.4.2.3	Updated configuration property names	19
5.1.4.2.4	Option to specify file contents instead of file name	19
5.1.4.2.5	Data fragmentation support for Authentication and Key Exchange builtin topics	20
5.1.4.3	New APIs	20
5.1.4.4	Persistence Service	20
5.1.4.4.1	Changed default Persistence Service dds.data_writer.history.key_material_key ..	20
5.1.4.4.2	Improved the algorithm to derive a key used to encrypt Persistence Service's encryption key	20
5.1.4.5	Shapes Demo	21
5.1.4.5.1	Added Shapes Demo CA key to Shapes Demo resource folder	21
5.2	What's Fixed in 6.0.0	21
5.2.1	Fixes Related to Specification Compliance	21
5.2.1.1	Input parameters to Security SPI functions do not have "const"	21
5.2.1.2	Mutability of Publisher PartitionQosPolicy	21
5.2.1.3	Wrong inputs to validate_local_permissions	22
5.2.1.4	Inconsistent governance configuration incorrectly allowed	22
5.2.1.5	Wrong return code for operations disallowed by Security Plugins	22
5.2.1.6	Authentication handshake failed if plugin MinorVersion was different	22
5.2.1.7	RTI_Security_Exception type did not match DDS Security specification	23
5.2.1.8	Entity creation incorrectly succeeded when no governance rule found	23
5.2.1.9	Builtin Logging Topic not protected	23
5.2.1.10	Access Control API definitions not compliant with specification	23
5.2.1.11	Wrong output of enable_logging and log	24
5.2.1.12	Possible discovery delays when communicating with other vendors	24
5.2.2	Other Fixes	24
5.2.2.1	Potential crash in Spy, Ping, or Persistence Service when enabling Security Plugins Logging with distribution over DDS	24
5.2.2.2	Insecure random seed implementation	25
5.2.2.3	Wrong product version passed to Authentication plugin's set_remote_participant_info	25

5.2.2.4	Could not create topic if deny rule contained partition	25
5.2.2.5	Specifying "ecdh" as the shared_secret_algorithm resulted in "dh" behavior	25
5.2.2.6	OpenSSL global state not properly cleaned up when shutting down an application	25
5.2.2.7	Certificate Authority incorrectly required to be self-signed	26
5.2.2.8	Segmentation fault when trying to decode a malformed RTPS message	26
5.2.2.9	Unknown shared secret algorithm silently treated as "ecdh"	26
5.2.2.10	Data protection didn't work if fragmented samples were preallocated	26
5.2.2.11	Persistence Service failure to store encoded samples in database	27
5.2.2.12	Possible lack of SUBSCRIPTION_MATCHED_STATUS if a DataWriter lost liveliness with the DataReader	27
5.2.2.13	Communication failure when using MultiChannel and changing filter expression parameters	27
5.2.2.14	Possible deadlock risk when using TopicQueries and Security	28
Chapter 6 Known Issues		
6.1	No Support for ECDSA-ECDH with Static OpenSSL Libraries and Certicom Security Builder	29
6.2	No Support for Writing >65kB Unfragmented Samples Using Metadata or RTPS Message Protection	29
6.3	subscription_data and publication_data in check_local_datawriter_match / check_local_datareader_match Are Not Populated	30
6.4	relay_only parameter in check_remote_datareader is Not Populated	30
6.5	<subject_name> not considered a match in some situations	30
6.6	Segmentation fault when using DomainParticipantListener with Security Plugins	30
Chapter 7 Third-Party Licenses		31

Chapter 1 Supported Platforms

RTI® Security Plugins 6.0.1 is supported on the following platforms.

Table 1.1 Supported Platforms

Operating System	Version
Android™	All platforms listed in the <i>RTI Connexx® DDS Core Libraries Release Notes</i> for the same version number, except SUSE® Linux Enterprise Server platforms.
iOS®	
Linux®	
macOS®	
QNX®	All platforms listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number, except QNX Neutrino® 6.4.1.
Windows®	All platforms listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number.

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

Security Plugins is also supported on the platforms in [Table 1.2 Custom Supported Platforms](#); these are target platforms for which RTI offers custom support. If you are interested in these platforms, please contact your local RTI representative or email sales@rti.com.

Table 1.2 Custom Supported Platforms

Operating System	Version
Linux	Debian® 7 RedHawk™ Linux 6.5 Ubuntu® 18.04 LTS Wind River® Linux 8 Yocto Project® 2.5

Table 1.2 Custom Supported Platforms

Operating System	Version
QNX	QNX Neutrino 6.6

Chapter 2 Compatibility

Security Plugins 6.0.1 is interoperable with 5.2.7 and higher versions of *Security Plugins*.

This release of *Security Plugins* includes partial support for the DDS Security specification from the Object Management Group (OMG)¹.

Security Plugins 6.0.1 is API-compatible with OpenSSL® versions 1.1.0a through 1.1.1d. It is not API-compatible with previous versions to OpenSSL® 1.1.0a. Note that *Security Plugins* 6.0.1 has only been tested by RTI using OpenSSL 1.1.1d. If you need *Security Plugins* 6.0.1 to run against older versions of OpenSSL®, please contact support@rti.com.

Persistence Service databases secured with *Security Plugins* 6.0.1 are incompatible with databases generated by older versions of *Persistence Service*.

Note: For more information about these and other backward compatibility issues, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

¹<http://www.omg.org/spec/DDS-SECURITY/1.1/>

Chapter 3 What's New in 6.0.1

3.1 New platforms

This release adds support for these platforms:

- Android 9.0 (Arm v7, Arm v8 64-bit)
- macOS 10.14 (x64)
- Red Hat® Enterprise Linux 8 (x64)
- Wind River® Linux 8 on PPC e6500 CPU (Custom-supported target platform. Contact your RTI sales representative or sales@rti.com for more information.)
- Windows 10 (x86, x64) with Visual Studio® 2019
- Windows Server 2016 (x86, x64) with Visual Studio 2019
- Yocto Project® 2.5 (Custom-supported target platform. Contact your RTI sales representative or sales@rti.com for more information.)

3.2 Removed platforms

The following platforms are no longer supported:

- macOS 10.11
- Windows 7
- Windows Server 2008 R2

3.3 Updated OpenSSL Version

This release uses OpenSSL 1.1.1d (instead of 1.0.2o).

3.4 Support for loading private keys from an OpenSSL Engine

You may now delegate the loading of the `dds.sec.auth.private_key` to the `openssl_engine`. You may specify the format of the private key using the property `authentication.keyform`. The value of this property can be one of the following:

- `pem` [default]: The key is in PEM format and will be loaded as a file or a string, depending on its "file:" or "data:" prefix.
- `engine`: The key is an array of bytes and will be loaded by the engine.

3.5 More detailed messages for OpenSSL errors

Log messages for OpenSSL errors are now more detailed. In the following example, the first line has been added to the error output:

```
RTI_Security_AccessControl_verified_string_from_data_holder:OpenSSL function PKCS7_verify
failed with error: error:2107C080:PKCS7 routines:PKCS7_get0_signers:signer certificate not
found
RTI_Security_AccessControl_verified_string_from_data_holder:Document signature verification
failed. Make sure document was signed by the right permissions authority.
```

"2107C080:PKCS7 routines:PKCS7_get0_signers:signer certificate" is the output value of the function call `ERR_error_string(ERR_get_error())`.

See also https://www.openssl.org/docs/man1.0.2/man3/ERR_error_string.html.

3.6 Increased verbosity level of benign log messages

The following log messages have changed the verbosity level in which they are generated:

Message	Previous Log Level	Current Log Level
RTI_Security_Authentication_process_handshake:received unexpected handshake message, probably from a participant that lost liveness with this one before ongoing authentication completed. Once current authentication times out, communication should be restored.	ERROR	NOTICE
DDS_DomainParticipantTrustPlugins_forwardProcessHandshake:!security function process_handshake returned VALIDATION_FAILED	ERROR	STATUS_REMOTE
MIGInterpreter_parse:received unencoded rtps message. Unacceptable due to is_rtps_protected = true	ERROR	STATUS_REMOTE
unauthenticated remote participant [participant ID] denied	ERROR	WARNING

For more information, see Table 6.2 "Log Messages" in the *RTI Security Plugins User's Manual*.

3.7 Updated message logged upon authentication negotiation timeout

This release changes the message logged when an authentication negotiation times out. The new message is:

```
!DDS PRESParticipant_onCheckAuthenticationInProgressListenerEvent:  
[Local Participant: 8fb08f02 3b3136d7 1a83c45d]  
[Remote Participant: 8fb08f02 3b31bd51 757eb77a] security:  
authentication has timed out: removing remote participant state to allow  
the participants to start a new discovery and authentication process at a later point
```

3.8 New Logging Plugin property to configure verbosity level

The logging verbosity level could be configured by setting a property named **logging.log_level** and an integer value. This property is now deprecated. Alternatively, you may now specify a property named **logging.verbosity** and one of the following values (case sensitive):

- SILENT (least verbose)
- EMERGENCY
- ALERT
- CRITICAL
- ERROR
- WARNING
- NOTICE
- INFORMATIONAL
- DEBUG (most verbose)

To avoid inconsistencies, the **logging.log_level** and **logging.verbosity** properties cannot be used simultaneously.

3.9 Changes related to authentication plugin configuration

- You may now have *Security Plugins* enforce the presence of the X.509 v3 extension KeyUsage (see <https://tools.ietf.org/html/rfc5280#section-4.2.1.3>). This enforcement is determined by the property **com.rti.serv.secure.authentication.x509v3_extension_enforcement.key_usage**. See the *Security Plugins User's Manual* for more details.
- If you specify **authentication.alternative_ca_files** or **access_control.alternative_permissions_authority_files** and one of the files fails to be loaded, then *DomainParticipant* creation will now fail because your intended alternative authority is not valid. You must remove the failing file from your list.

- You may no longer specify both the pre-6.0.0 property (e.g., `com.rti.serv.secure.authentication.ca_file`) and the OMG property (e.g., `dds.sec.auth.identity_ca`) for a given attribute (e.g., Certificate Authority file). Attempting to do so will result in failure to create the *DomainParticipant*.

3.10 Property `key_material_key` now required for Secure Persistence Service

For Secure Persistence Service, setting the property `dds.data_writer.history.key_material_key` is now mandatory. You may specify either a file name or a data string using the "`file:`" or "`data:`" prefix. If the value has no prefix, the value is interpreted as a file name.

3.11 Improved HMAC-only mode key derivation algorithm

When using RTPS-HMAC-Only mode, the key derivation algorithm uses an unsigned integer called `sessionId` as one of the inputs. The `sessionId` will change as the session reaches a certain number of signed blocks. Upon creating a *DomainParticipant*, however, the `sessionId` was always initialized to zero. In order to make the derived key more difficult for an attacker to determine, the `sessionId` is now initialized to a random number.

Chapter 4 What's Fixed in 6.0.1

4.1 Fixes Related to Discovery and Authentication

4.1.1 Possible segmentation fault when receiving malformed ParticipantBuiltinTopicData and using Security Plugins

A segmentation fault may have occurred in the function `md5_process` when receiving a malformed `ParticipantBuiltinTopicData` sample while using *Security Plugins*. This problem has been resolved.

[RTI Issue ID SEC-959]

4.1.2 Possible lack of SUBSCRIPTION_MATCHED_STATUS if a DataWriter lost liveliness with DataReader

In release 6.0.0, there was a race condition in which a *DataReader* may have never reported a `SUBSCRIPTION_MATCHED_STATUS` change despite successfully matching and receiving data from a *DataWriter*. This race condition occurred if all of the following conditions were true:

- The *DataReader* set its liveliness **lease_duration** to a very small duration (on the order of milliseconds).
- The *DataReader* was communicating with a *DataWriter* with **metadata_protection_kind** or **data_protection_kind** set to a value other than NONE.
- The *DataWriter* lost liveliness with the *DataReader* between the time the *DataReader* discovered the *DataWriter* and the time the *DataReader* received key material from the *DataWriter*.
- The *DataWriter* regained liveliness with the *DataReader* before the *DataReader* received key material from the *DataWriter*.

Note: This condition is the opposite of the fourth condition in RTI Issue ID SEC-895, which was fixed in 6.0.0. (See [5.2.2.12 Possible lack of SUBSCRIPTION_MATCHED_STATUS if a DataWriter lost liveliness with the DataReader on page 27.](#))

This problem has been resolved by making sure that the SUBSCRIPTION_MATCHED_STATUS change is reported as soon as the *DataReader* receives key material from the *DataWriter*.

[RTI Issue ID SEC-911]

4.1.3 DataWriter did not report PUBLICATION_MATCHED_STATUS for DataReaders that were inactive when it received their key material

In release 6.0.0, under the following sequence of events, a *DataWriter* setting <metadata_protection_kind> or <data_protection_kind> to a value other than NONE would have never reported PUBLICATION_MATCHED_STATUS for a *DataReader*:

1. The *DataWriter* discovered the *DataReader*.
2. The *DataReader* became inactive after **writer_qos.protocol.rtps_reliable_writer.max_heartbeat_retries** Heartbeats (HBs) due to the lack of response to HBs (default behavior) or to not making progress on the NACK messages (non default - requires setting **writer_qos.protocol.rtps_reliable_writer.inactivate_nonprogressing_readers** to TRUE).
3. The *DataWriter* received key material from the *DataReader*.
4. The *DataReader* became active after starting to respond to HBs and/or making progress on the NACK messages.
5. The *DataWriter* incorrectly did not report PUBLICATION_MATCHED_STATUS for a *DataReader*.

This problem has been fixed.

[RTI Issue ID SEC-1013]

4.1.4 Couldn't use "file:" prefix for CRL file

In release 6.0.0, the property **com.rti.serv.secure.authentication.crl_file** did not accept a "file:" prefix in its value, unlike all other properties that accepted a file name as a value. (Release 6.0.0 introduced the "file:" prefix to the other properties.) This problem has been resolved. The property now accepts an optional "file:" prefix.

[RTI Issue ID SEC-893]

4.1.5 Stateless participant property may not have been correctly updated

The property **dds.participant.discovery_config.use_stateless_participant_discovery_reader** was incorrectly parsed as an integer. As a result, you had to specify a non-zero integer as the value of this property to enable the pure stateless mode for the local Simple Participant Discovery reader (disabled by default). Any non-integer value would silently fail to enable the feature.

This property is now parsed as a boolean. You can now specify “true”, “yes” or “1” as the property value to enable the pure stateless mode. You may want to use "false", "no" or "0" to keep this feature disabled (default behavior). Any other value will result in silently failing to enable the feature.

[RTI Issue ID SEC-944]

4.1.6 Segmentation fault when receiving a handshake message with incorrect message_class_id length

According to the DDS Security specification, *DomainParticipants* exchange handshake messages during authentication. These handshake messages contain a string field called **message_class_id**. If the serialized length of **message_class_id** was corrupted over the network, then the receiving *DomainParticipant* may have experienced a segmentation fault in the function **DDS_DomainParticipantGenericMessageDispatcher_dispatchSample**. This problem has been resolved by discarding the corrupted handshake message.

[RTI Issue ID SEC-936]

4.1.7 Possible bus error or use of uninitialized value when receiving handshake message with incorrect inline QoS parameter length

According to the DDS Security specification, *DomainParticipants* exchange handshake messages during authentication. These handshake messages are directed from one *DataWriter* to one *DataReader*. A directed message's destination is described in the DIRECTED_WRITE parameter of the inline QoS of the message. If the serialized length of the DIRECTED_WRITE parameter was corrupted over the network, then depending on the platform, the receiving *DomainParticipant* may have experienced either a "Bus error" crash or a "Conditional jump or move depends on uninitialized value(s)" error reported by a tool such as Valgrind. This problem has been fixed by removing these errors.

[RTI Issue ID SEC-937]

4.1.8 Potential Crash upon receiving malformed authentication fragment

In release 6.0.0, receiving a malformed fragment for the Authentication topic (for example, as a result of severe network packet corruption) may have caused a crash.

This problem has been resolved. Receiving a corrupted fragment will no longer result in a crash, but the fragment will be dropped after logging the following exception:

```
COMMENDAnonReaderService_onSubmessage:!assert data sample. Fragment table too small.
```

[RTI Issue ID SEC-974]

4.2 Fixes Related to Logging

4.2.1 Some messages marked only as ERRORS

If you used the default logging method (*Connex DDS's* builtin logging system, also known as `NDDS_Config_Logger`) in *Security Plugins*, then *Security Plugins's* Logging Plugin marked all messages as ERRORS, even if they should have been, for example, WARNINGS. (There were some exceptions to this. Some security-related messages, such as messages containing "`DDS_DomainParticipantTrustPlugins_forwardProcessHandshake`," did not pass through *Security Plugins*, but only through *Connex DDS*. These messages were marked correctly, as described in "Controlling Messages from Connex DDS" in the *RTI Connex DDS Core Libraries User's Manual*; they were not always marked as ERRORS.)

This issue has been resolved. The Logging Plugin in *Security Plugins* no longer marks all messages as ERRORS. Instead, it creates a mapping between the severity of a security event and the level at which the message is logged to give the correct verbosity when using the *Connex DDS* builtin logging system to log messages in *Security Plugins*. See the "Logging" chapter in the *RTI Security Plugins User's Manual* for more information.

[RTI Issue IDs SEC-873]

4.2.2 Security messages not logged, regardless of verbosity configured at DomainParticipantFactory level

Some security-related messages generated by *Security Plugins* were logged directly through the *Connex DDS* builtin logging system (also known as `NDDS_Config_Logger`). These messages were only logged if the log level was "exception". This means that some security-related messages were never displayed (such as some warnings related to the incorrect use of HMAC-only).

This problem has been resolved. Now the verbosity used for messages generated by the *Connex DDS* builtin logging system will be determined by the **logging.verbosity** property. See the mapping table at the end of the Logging chapter in the *Security Plugins User's Manual*.

[RTI Issue ID SEC-982]

4.3 Other Fixes

4.3.1 Too much memory allocated for non-payload-encoding DataWriters

Creating a *DataWriter* with **data_protection_kind** set to NONE resulted in 44 bytes of unnecessary memory being allocated, due to the incorrect calling of the function **RTI_Security_Cryptography_get_encoded_serialized_sample_size**. This problem has been resolved.

[RTI Issue ID SEC-892]

4.3.2 Unexpected warning when using shared memory and security

When using the Shared Memory transport along with *Security Plugins*, this message was generated at WARNING verbosity:

```
MIGGenerator_beginMessage:changing message_size_max from 65536 to 65535 due to a limitation of the Connex implementation (SEC-768). Consider reducing your message_size_max in order to force fragmentation to happen and work around this limitation.
```

This problem has been resolved by removing this warning message.

[RTI Issue ID SEC-894]

4.3.3 Possible deadlock risk when using TopicQueries and Security

This problem was fixed in release 6.0.0, but not documented then. See [5.2.2.14 Possible deadlock risk when using TopicQueries and Security on page 28](#) for details.

[RTI Issue ID SEC-1015]

4.3.4 Applications directly calling OpenSSL APIs after DomainParticipant deletion may have crashed

In 6.0.0, the destruction of all the *DomainParticipants* loading *Security Plugins* resulted in the plugins calling OpenSSL's **EVP_cleanup** and **ERR_free_strings** APIs to clean up OpenSSL state. As a result, if an application running *Connex DDS* invoked OpenSSL APIs after this cleanup took place without reinitializing OpenSSL, the application may have run into unexpected OpenSSL behavior or even a crash.

OpenSSL 1.1.0 deprecated both **EVP_cleanup** and **ERR_free_strings**, and they have no effect anymore. Since release 6.0.1 is using OpenSSL 1.1.1d, the previously described problem is now fixed.

[RTI Issue ID SEC-1031]

Chapter 5 Previous Release

5.1 What's New in 6.0.0

5.1.1 New Platforms

This release adds support for the following platforms:

Operating System	CPU	Compiler	RTI Architecture Abbreviation
Debian 7	Arm v7	gcc 4.9.3	armv7aLinux3.12gcc4.9.3cortex-a9 (Custom platform)
Ubuntu 18.04 LTS	x64	gcc 7.3.0	x64Linux4gcc7.3.0
	Arm v8 64-bit	gcc 7.3.0	armv8Linux4gcc7.3.0 (Custom platform)
Wind River Linux 8	Arm v7	gcc 5.2.0	armv7aWRLinux8gcc5.2.0 (Custom platform)

5.1.2 Updated OpenSSL Version

This release uses OpenSSL 1.0.2o (instead of 1.0.2n).

5.1.3 Changes Related to Specification Compliance

This release adds the following changes to be compliant with the Builtin Security Plugins defined in the DDS Security specification¹.

5.1.3.1 Authentication and Discovery

5.1.3.1.1 Option to specify private key password

This feature is related to the DDS Security specification properties used to configure the builtin Authentication plugin.

¹<http://www.omg.org/spec/DDS-SECURITY/1.1/>

You may now encrypt the value of the property **dds.sec.auth.private_key** and specify its decryption password as the property **dds.sec.auth.password**. See the tables at the beginning of the "Authentication" chapter in the *RTI Security Plugins Getting Started Guide* for details.

5.1.3.1.2 Added support for new participant and endpoint security information

This release adds support for the new DDS Security 1.1 participant/endpoint security matching, which allows for early detection of incompatible security configurations.

To support this feature, participants will announce new parameters as part of the participant/endpoint discovery. These discovery parameters have the following attributes:

Discovery Type	Member Name	Member Type	Parameter ID Name	Parameter ID Value
Endpoint	security_info	EndpointSecurityInfo	PID_ENDPOINT_SECURITY_INFO	0x1004
Participant	security_info	ParticipantSecurityInfo	PID_PARTICIPANT_SECURITY_INFO	0x1005

Note that while these members are propagated and used for doing discovery matching, currently they are not exposed as part of any public API. Moreover, an incompatible security_info configuration is currently not reported as part of the OFFERED_INCOMPATIBLE_QOS/REQUESTED_INCOMPATIBLE_QOS statuses. Therefore, a security_info incompatibility will not trigger **on_offered_incompatible_qos()**/**on_requested_incompatible_qos()** callbacks.

The propagation of these parameters is enabled by default. Since their propagation introduces additional restrictions for participant and endpoint matching, the following new properties have been introduced to allow for keeping the old behavior:

Property Name ¹	Property Value Description
dds.participant.discovery_config.disable_endpoint_security_info_propagation	<p>Optional</p> <p>If set to FALSE, the endpoint's security_info is propagated and a <i>DataWriter/DataReader</i> pair needs to use the same security configuration to match. If set to TRUE in both participants, contained <i>DataWriters</i> and <i>DataReaders</i> may communicate for some combinations of inconsistent metadata/data protection kinds.</p> <p>Default: FALSE</p>
dds.participant.discovery_config.disable_participant_security_info_propagation	<p>Optional</p> <p>If set to FALSE, the participant's security_info is propagated and a pair of participants needs to use the same Governance's RTPS, discovery, and liveliness configurations to match. If set to TRUE in both participants, participants may communicate for some combinations of inconsistent Governance's RTPS, discovery, and liveliness configurations.</p> <p>Default: FALSE</p>

¹These new properties do not need to be prefixed with 'com.rti.serv.secure.'

5.1.3.1.3 Added new DCPSParticipantsSecure builtin topic

This release adds a new builtin topic, “DCPSParticipantsSecure,” introduced by the DDS Security 1.1 specification. This topic is secured following the same rules as the Secure Endpoint Discovery topics (that is, it is configured through the Governance's **discovery_protection_kind** parameter).

To support this new topic, *Connex DDS* creates two new reliable endpoints (a *DataWriter* and a *DataReader*) when enabling security for a participant. The entity IDs for those endpoints are as follows:

Endpoint Name	Entity ID Definition	Entity ID Value
SPDPbuiltinParticipantsSecureWriter	ENTITYID_SPDP_BUILTIN_PARTICIPANTS_SECURE_WRITER	{{ff, 01, 01}, c2}
SPDPbuiltinParticipantsSecureReader	ENTITYID_SPDP_BUILTIN_PARTICIPANTS_SECURE_READER	{{ff, 01, 01}, c7}

When those endpoints are enabled, the following bits are set in the ParticipantBuiltinTopicData's availableBuiltinEndpoints:

Builtin Endpoint	Bit in the ParticipantBuiltinTopicData's availableBuiltinEndpoints
SPDPbuiltinParticipantsSecureWriter	(0x00000001 << 26)
SPDPbuiltinParticipantsSecureReader	(0x00000001 << 27)

Starting in this release, once a remote participant is authenticated, any changes affecting the participant discovery data must be exchanged using the “DCPSParticipantsSecure” topic. To avoid breaking backwards compatibility with previous versions of *Connex DDS*, old remote participants will still rely on the existing TrustedState mechanism described in the section Protecting Participant Discovery, in the *RTI Security Plugins Getting Started Guide*, to propagate Participant discovery updates for authenticated Participants.

5.1.3.1.4 Support for DDS Security 1.1 AuthRequest

This release updates the *Security Plugins* re-authentication mechanism to be compliant with the new AuthRequest mechanism described in the DDS Security 1.1 specification. The changes are as follows:

- Changed GenericMessageClassId from "com.rti.sec.auth.request" to "dds.sec.auth_request".
- Changed Token's class_id from "com.rti.sec.DDS:Auth:PKI-DH:1.0+AuthReq" to "DDS:Auth:PKI-DH:1.0+AuthReq".
- Updated **validate_remote_identity** to include AuthRequestMessageToken parameters.
- Removed **begin_auth_request** and **process_auth_request** APIs from the Authentication Plugin interface, because their logic is now part of the **validate_remote_identity** API. Note that *RTI Security Plugins SDK* still keeps those functions (now as private functions), since they are called from the **validate_remote_identity** function implementation.

Note that these changes **do not break backwards compatibility**: 6.0.0 participants will use either the old or the new re-authentication GenericMessageClassId and Token's class_id, depending on the detected remote participant's version (which is exchanged as part of participant discovery).

5.1.3.2 Access Control

5.1.3.2.1 Updated PermissionsToken class_id

The DDS Security 1.1 specification states that in the Builtin Access Control plugin, the PermissionsToken class_id shall be "DDS:Access:Permissions:1.0". *Security Plugins* has updated its PermissionsToken from "DDS:Access:Permissions" to "DDS:Access:Permissions:1.0". Note that this change does not affect compatibility because the specification states that if MajorVersion and MinorVersion are missing from the class_id, the class_id shall be interpreted as being MajorVersion 1 and MinorVersion 0. So the legacy class_id is equivalent to the new one.

5.1.3.2.2 Added check_remote_topic

The DDS Security specification describes the Access Control plugin operation **check_remote_topic()**. This function is now invoked and implemented. You will see no impact when using the builtin plugins. The function will not be invoked if either **enable_read_access_control** or **enable_write_access_control** is FALSE in the local DomainParticipant's Governance document's corresponding <topic_rule> tag. The function will return TRUE if the remote DomainParticipant's Permissions document allows a *DataWriter* or a *DataReader* of that topic-DomainParticipant combination.

5.1.3.2.3 Added functions to return security attributes

The DDS Security 1.1 specification introduces the Access Control plugin operations **return_participant_sec_attributes()**, **return_datawriter_sec_attributes()**, and **return_datareader_sec_attributes()**. These functions are now invoked and implemented. The builtin plugin implementation does nothing in these functions. If you implement a custom plugin that populates the PropertySeq within ParticipantSecurityAttributes or EndpointSecurityAttributes, then these functions should finalize the PropertySeq.

5.1.3.2.4 Data tagging

The DDS Security specification describes data tagging using the DataTagQosPolicy. *Security Plugins* now supports this policy and its usage in the Access Control plugin.

5.1.3.2.5 Support for DDS Security Topic Security Attributes

This release adds support for DDS Security 1.1 TopicSecurityAttributes and the associated **get_topic_security_attributes** API.

As part of this feature, EndpointSecurityAttributes now inherits from TopicSecurityAttributes, and some of the members of EndpointSecurityAttributes have been moved to TopicSecurityAttributes. For more information about these changes, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

5.1.3.2.6 Added Support for builtin topics to `get_datawriter_sec_attributes` and `get_datareader_sec_attributes` APIs

Previously, endpoint security attributes for builtin topic endpoints were hardcoded at the core libraries level; they were not modifiable by *Security Plugins* (or any custom security plugin).

Starting with this release, core libraries get the endpoint security attributes for the builtin topics using the `get_datawriter_sec_attributes` and `get_datareader_sec_attributes` APIs as described by the DDS Security 1.1 specification, Section 7.4.8, Securing the "Builtin Secure Endpoints."

To support this new mechanism, *Security Plugins's* `get_datawriter_sec_attributes` and `get_datareader_sec_attributes` APIs have been updated to support retrieving attributes for DDS and DDS Security builtin topics, as well as for RTI ("vendor-specific") builtin topics.

For more information about how this feature affects custom security plugins, please refer to the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

5.1.3.2.7 `<topics>` now mandatory in permissions file

The DDS Security specification's XSD schema file `omg_shared_ca_permissions.xsd` indicates that the `<topics>` element is mandatory inside a `<publish>` or `<subscribe>` element. *Security Plugins* now enforces this rule.

5.1.3.2.8 Updated matching behavior of allowed partitions condition

The DDS Security specification describes the matching behavior of the `<partitions>` section within an `<allow_rule>` of a Permissions file. In order for a *DataWriter* or *DataReader* to be matched with an "allowed partitions" condition, the DDS entity's partitions must be a subset of the partitions in the condition. This release enforces this matching rule.

To change this behavior, you may set the security plugin property `access_control.use_530_partitions` to TRUE. If TRUE, then a *DataWriter* or *DataReader* will be matched with an "allowed partitions" condition as long as at least one of the DDS entity's partitions matches one of the partitions in the condition; this is consistent with Connex 5.3.0 behavior. If FALSE, then the entity's partitions must be a subset of the condition's partitions; this is consistent with the behavior of the DDS Security specification. The default value is FALSE.

Here's an example:

DataWriter Partitions	Allowed Partitions Condition	use_530_partitions	allowed?
[A, B]	[B, C]	TRUE	yes, because B is in [B, C]
[A, B]	[B, C]	FALSE	no, because A is not in [B, C]

5.1.3.3 Cryptography

5.1.3.3.1 New protection kinds

This feature is related to the Domain Governance Document described in the DDS Security specification.

The following previously unsupported protection kinds are now supported:

- `rtps_protection_kind = ENCRYPT`
- `metadata_protection_kind = SIGN`
- `data_protection_kind = SIGN`

5.1.4 Other Changes

5.1.4.1 Authentication and Discovery

5.1.4.1.1 Identity Certificate chaining

You may now put a chain of certificates in the Identity Certificate by concatenating individual certificates and specifying the concatenated result as a single file or string. The Identity Certificate will be verified against the Identity CA using the following procedure:

- The current certificate is the first certificate in the Identity Certificate chain.
- Perform the following steps up to and including the case when the current certificate is the last certificate in the Identity Certificate chain:
 - If the current certificate is signed by the Identity CA, then the verification succeeds immediately.
 - Otherwise:
 - If a next certificate exists in the chain and the current certificate is signed by that next certificate, then the next certificate becomes the current certificate.
 - Otherwise, verification fails immediately.

5.1.4.2 Cryptography

5.1.4.2.1 Support using different writer keys for protecting submessages and serialized data

This release adds support for using different key material for protecting the submessages and serialized data encoded by a *DataWriter*.

By default, *DataWriters* with metadata and data protection kinds other than NONE use the same key material for encoding both submessages and serialized data. To change this behavior, this release adds a new property to the Cryptography plugin:

- **cryptography.share_key_for_metadata_and_data_protection:** Determines whether the metadata and data encoding operations share the same key material or use different keys. Default: TRUE (they share key material).

Note that setting this property to FALSE (that is, using different keys for protecting submessages and serialized data) will break backward compatibility with older versions of *Security Plugins* when both **metadata_protection_kind** and **data_protection_kind** are set to a value other than NONE.

5.1.4.2.2 WITH_ORIGIN_AUTHENTICATION protection kinds

In the Governance Document, you may now use the protection kinds ENCRYPT_WITH_ORIGIN_AUTHENTICATION and SIGN_WITH_ORIGIN_AUTHENTICATION as explained in the DDS Security specification. WITH_ORIGIN_AUTHENTICATION adds receiver-specific Message Authentication Codes (MACs) to the encoded output. WITH_ORIGIN_AUTHENTICATION may not be used if the property **com.rti.serv.secure.cryptography.max_receiver_specific_mac**s is 0. The protection kind values ENCRYPT and SIGN no longer add receiver-specific MACs to the encoded output.

5.1.4.2.3 Updated configuration property names

This feature is related to the DDS Security specification properties used to configure the builtin Authentication and Access Control plugins.

You may now use the property names that are in the DDS Security specification. The legacy property names are still supported. The following properties are affected:

Legacy Property Name (prefix with 'com.rti.serv.secure')	New Property Name (no prefix)
authentication.ca_file	dds.sec.auth.identity_ca
authentication.private_key_file	dds.sec.auth.private_key
authentication.certificate_file	dds.sec.auth.identity_certificate
access_control.permissions_authority_file	dds.sec.access.permissions_ca
access_control.governance_file	dds.sec.access.governance
access_control.permissions_file	dds.sec.access.permissions

5.1.4.2.4 Option to specify file contents instead of file name

This feature is related to the DDS Security specification properties used to configure the builtin Authentication and Access Control plugins.

Many of the security properties required a file name as the value. You may now use the contents of the file, prefixed by "data:", as the value. For example, if the file privateKey.pem contains

```
-----BEGIN PRIVATE KEY-----
abc
def
```

```
-----END PRIVATE KEY-----
```

you previously had to specify a property with name `com.rti.serv.secure.authentication.private_key_file` and value `"privateKey.pem"`. This option is still possible, but you may alternatively specify a property with name `dds.sec.auth.private_key` and the value:

```
"data:-----BEGIN PRIVATE KEY-----\nabcdef\n-----END PRIVATE KEY-----"
```

The two `\n` characters surrounding `"abcdef"` are required. A `\n` is not required between `'c'` and `'d'`.

5.1.4.2.5 Data fragmentation support for Authentication and Key Exchange builtin topics

This release adds support for data fragmentation of the Authentication topic (`ParticipantStatelessMessage` builtin topic) and asynchronous publishing of the Key Exchange topic (`ParticipantSecureVolatileMessageSecure` builtin topic), when security is enabled. This feature addresses the scenario in which a security-enabled environment requires DDS-level fragmentation due to a hard limit on the maximum transport message size.

In the case of the Authentication builtin topic, data fragmentation is supported by default. To enable data fragmentation for the Key Exchange topic, you need to enable asynchronous publishing through `DiscoveryConfig`'s `secure_volatile_writer_publish_mode` field.

For more information, refer to *Enabling Asynchronous Publishing for the Key Exchange Topic*, in the *RTI Security Plugins Getting Started Guide*, and `PUBLISH_MODE` QoS Policy (DDS Extension), in the *RTI Connex DDS Core Libraries User's Manual*.

5.1.4.3 New APIs

New APIs are provided to get the *Security Plugins* version:

- `RTI_Security_get_build_version_string()`
- `RTI_Security_get_library_version()`

5.1.4.4 Persistence Service

5.1.4.4.1 Changed default Persistence Service `dds.data_writer.history.key_material_key`

The undisclosed non-NULL default `dds.data_writer.history.key_material_key` has changed. As a result, *RTI Persistence Service* databases protected with the old default key will not be accessible by the new *Persistence Service*.

Note that using the default key is discouraged, and you should set `dds.data_writer.history.key_material_key` to a value other than the default.

5.1.4.4.2 Improved the algorithm to derive a key used to encrypt Persistence Service's encryption key

In the builtin plugins, the key derivation algorithm applied to the `dds.data_writer.history.key_material_key` has improved. The algorithm now involves PBKDF2 (Password-Based Key Derivation Function) with SHA-512 (Secure Hash Algorithm with a 512-bit hash value) and a random salt. *Persistence Service*

now stores the random salt along with the PRSTDataWriter's encrypted key. As a result, databases protected with the old *Persistence Service* will not be accessible by the new *Persistence Service*.

5.1.4.5 Shapes Demo

5.1.4.5.1 Added Shapes Demo CA key to Shapes Demo resource folder

This release adds the *Shapes Demo* CA key file (RTI_SHAPES_DEMO_CA_KEY.pem) to the *Shapes Demo* resource\cert folder.

This file is useful for generating new signed Governance and Permissions files that can be used with the shipped *Shapes Demo* certificates.

5.2 What's Fixed in 6.0.0

This section describes bugs that have been fixed in *Security Plugins* 6.0.0.

5.2.1 Fixes Related to Specification Compliance

5.2.1.1 Input parameters to Security SPI functions do not have "const"

The header file `dds_c/dds_c_trust_plugins.h` defines the Security Service Plugin Interface (SPI) functions (for example, `DDS_Authentication_ValidateRemoteIdentityFunction`). The non-primitive input parameters of many of these functions did not have "const" preceding them. This problem has been resolved. The non-primitive input parameters now have "const" preceding them. For example:

```
typedef
DDS_ValidationResult_t (*DDS_Authentication_ValidateRemoteIdentityFunction) (
    struct DDS_AuthenticationPlugin *auth,
    DDS_IdentityHandle *remote_identity_handle, /* out */
    DDS_AuthRequestMessageToken *local_auth_request_token, /* out */
    const DDS_AuthRequestMessageToken *remote_auth_request_token,
    const DDS_IdentityHandle local_identity_handle,
    const DDS_IdentityToken *remote_identity_token,
    const struct DDS_GUID_t *remote_participant_guid,
    DDS_TrustException *exception);
```

[RTI Issue ID SEC-251]

5.2.1.2 Mutability of Publisher PartitionQosPolicy

The Publisher PartitionQosPolicy was always mutable, which did not comply with the DDS Security specification. This problem has been resolved. The Publisher PartitionQosPolicy is now immutable if the Publisher contains any *DataWriter* that meets the following two criteria:

1. The TopicSecurityAttributes for that *DataWriter* have **is_read_protected** (which corresponds to `<enable_read_access_control>` in the Governance Document) set to TRUE.

2. The *DataWriter* has the DurabilityQos policy kind set to something other than VOLATILE.

[RTI Issue ID SEC-453]

5.2.1.3 Wrong inputs to validate_local_permissions

Inputs to the **validate_local_permissions** function in the Access Control plugin were wrong. This problem has been resolved by replacing the PermissionsCredential with the DomainId_t and the DomainParticipantQos. You will see no impact when using the builtin plugins because PermissionsCredential was never used, and the two new parameters are not used.

[RTI Issue ID SEC-707]

5.2.1.4 Inconsistent governance configuration incorrectly allowed

In previous releases, *Security Plugins* incorrectly allowed governance's **allow_unauthenticated_participants** to be set to TRUE while **rtps_protection_kind** was set to a value other than NONE. Now, trying to configure a Participant with this inconsistent configuration will result in a failure to create the Participant.

[RTI Issue ID SEC-726]

5.2.1.5 Wrong return code for operations disallowed by Security Plugins

The DDS Security specification adds an additional return code NOT_ALLOWED_BY_SECURITY, which shall be returned by any operation that fails because the security plugins do not allow it. This return code was incorrectly defined in *Connex DDS* as NOT_ALLOWED_BY_SEC with a value of 13. This return code has now been replaced in *Connex DDS* by NOT_ALLOWED_BY_SECURITY, whose value is 1000. Although *Connex DDS* has never used this return code, it has been replaced to be compliant with the DDS Security specification.

[RTI Issue ID SEC-737]

5.2.1.6 Authentication handshake failed if plugin MinorVersion was different

The DDS Security 1.1 specification states that in the Builtin Authentication plugin, the **validate_remote_identity()** function shall return VALIDATION_FAILED if the local and remote IdentityTokens have different values for PluginClassName or MajorVersion. *Security Plugins* was incorrectly returning VALIDATION_FAILED if the IdentityTokens had the same PluginClassName and MajorVersion but different values for MinorVersion. *Security Plugins* no longer fails the validation in this scenario.

[RTI Issue ID SEC-739]

5.2.1.7 RTI_Security_Exception type did not match DDS Security specification

In previous releases, the definition of `RTI_Security_Exception` type did not match the DDS Security specification. This problem is now resolved by using the following definition, which matches the specification:

```
typedef struct DDS_TrustException {
    char *message;
    DDS_Long code;
    DDS_Long minor_code;
} DDS_TrustException;

typedef DDS_TrustException RTI_Security_Exception;
```

[RTI Issue ID SEC-743]

5.2.1.8 Entity creation incorrectly succeeded when no governance rule found

The creation of `DomainParticipants` and `Topics` incorrectly succeeded when the Domain Governance document did not specify any rules for those entities. According to the DDS Security specification, entity creation should fail with a suitable “permissions error” if there is no governance rule for the entity. This problem has been resolved.

[RTI Issue ID SEC-750]

5.2.1.9 Builtin Logging Topic not protected

The DDS Security specification states that the Builtin Logging Topic shall use the governance XML tag `<metadata_protection_kind>SIGN</metadata_protection_kind>`. *Security Plugins* was incorrectly setting **metadata_protection_kind** to `NONE`. This problem has been resolved by setting **metadata_protection_kind** to `SIGN`. This change breaks configuration compatibility between this and previous releases when using a *DataReader* to subscribe to the Builtin Logging Topic. For details, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

[RTI Issue ID SEC-772]

5.2.1.10 Access Control API definitions not compliant with specification

The following Access Control plugin interface APIs were not compliant with the DDS Security specification:

- **check_remote_datareader** was missing the `relay_only` parameter.
- **check_local_datawriter_match** and **check_local_datareader_match** were missing publication/subscription data and incorrectly including the `tags` parameter.
- **get_datawriter_sec_attributes** and **get_datareader_sec_attributes** APIs were missing.
- **get_endpoint_sec_attributes** was not compliant with the specification.

This release includes the following changes so that the Access Control plugin interface APIs are now compliant with the DDS Security specification:

- Updated **check_remote_datareader** to include the `relay_only` parameter.
- Updated **check_local_datawriter_match** and **check_local_datareader_match** to include publication/subscription data parameters and to remove the `tags` parameter.
- Removed the **get_endpoint_sec_attributes** API from the Access Control plugin interface.
- Added **get_datawriter_sec_attributes** and **get_datareader_sec_attributes** to the Access Control plugin interface. These APIs replace **get_endpoint_sec_attributes**.

[RTI Issue ID SEC-789]

5.2.1.11 Wrong output of enable_logging and log

The output of the **enable_logging** and **log** functions in the Logging Plugin was void, which was consistent with the body of the specification but not with `dds_security_plugins_spis.idl`. **enable_logging** and **log** now return `DDS_Boolean` to be consistent with the IDL file.

[RTI Issue ID SEC-869]

5.2.1.12 Possible discovery delays when communicating with other vendors

The DDS Security specification states that the `BuiltinParticipantStatelessMessageReader` and `BuiltinParticipantVolatileMessageSecureReader` have an implied content filter with the logical expression: `"destination_participant_guid == GUID_UNKNOWN || destination_participant_guid == reader.participant_guid"`. *Security Plugins* did not apply this filter on those builtin readers. This problem may have caused severe discovery delays when communicating with other DDS Security vendors. This problem has been resolved; the filter is now applied.

[RTI Issue ID SEC-888]

5.2.2 Other Fixes

5.2.2.1 Potential crash in Spy, Ping, or Persistence Service when enabling Security Plugins Logging with distribution over DDS

Enabling *Security Plugins* Logging with distribution over DDS on *Spy*, *Ping*, or *Persistence Service* may have provoked a segfault. This problem is now resolved.

[RTI Issue ID SEC-734]

5.2.2.2 Insecure random seed implementation

Security Plugins was using a mix of time, process ID, and host ID to seed the OpenSSL random number generator. None of these values is truly random, however, so the security of the seed was weak. This problem has been resolved by using random bytes to generate the seed.

[RTI Issue ID SEC-753]

5.2.2.3 Wrong product version passed to Authentication plugin's `set_remote_participant_info`

The `DDS_TrustedParticipantInfo`'s product version that was passed to the Authentication plugin's `set_remote_participant_info` RTI Extension API was wrong; it did not match the actual Participant's product version. This problem has been resolved.

[RTI Issue ID SEC-812]

5.2.2.4 Could not create topic if deny rule contained partition

If the XML Permissions document contains a `<deny_rule>`, and the rule has both `<publish>` and `<subscribe>` tags for a given topic, then the creation of that topic should fail if and only if neither the `<publish>` tag nor the `<subscribe>` tag contains any tags other than `<topics>` and `<topic>`. Topic creation, however, failed if either `<publish>` or `<subscribe>` contained `<partitions>` and `<partition>`. This behavior was incorrect because if the rule denied certain partitions, then any other partition was not denied and topic creation should have been allowed. (But it wasn't.) This problem has been resolved: topic creation is now allowed in this case.

[RTI Issue ID SEC-822]

5.2.2.5 Specifying "ecdh" as the `shared_secret_algorithm` resulted in "dh" behavior

Specifying "ecdh" as the value of the property `authentication.shared_secret_algorithm` incorrectly resulted in behavior equivalent to specifying "dh" as the value. This problem was introduced in *Security Plugins* 5.3.0 and has now been resolved.

[RTI Issue ID SEC-824]

5.2.2.6 OpenSSL global state not properly cleaned up when shutting down an application

A security-enabled application experienced leaks of memory blocks that were still reachable. These memory blocks belonged to OpenSSL global state. One example block came from the function `ERR_get_state()`. These leaks have been fixed.

[RTI Issue ID SEC-828]

5.2.2.7 Certificate Authority incorrectly required to be self-signed

The Certificate Authority (CA), which corresponds to the property **authentication.ca_file**, was incorrectly required to be a self-signed root CA. This problem has been resolved by allowing the **ca_file** to be a subordinate CA whose certificate is signed by a superior CA.

[RTI Issue ID SEC-839]

5.2.2.8 Segmentation fault when trying to decode a malformed RTPS message

According to the DDS Security specification, an encoded RTPS message must have an SRTPS_POSTFIX submessage. When trying to decode an RTPS message with an SRTPS_PREFIX but no SRTPS_POSTFIX, a segmentation fault would occur in the function **MIGRtpsTrustSubmessage_deserializePostfix**. This problem has been resolved by discarding such an RTPS message.

[RTI Issue ID SEC-842]

5.2.2.9 Unknown shared secret algorithm silently treated as "ecdh"

Setting the property **authentication.shared_secret_algorithm** to a value other than the supported values of **dh** or **ecdh** was equivalent to setting the property to **ecdh**. Similarly, discovering a *DomainParticipant* that was using an unsupported **shared_secret_algorithm** was equivalent to discovering a *DomainParticipant* that was using **ecdh**. These problems have been resolved by treating these scenarios as failure conditions. A *DomainParticipant* that attempts to use an unsupported **shared_secret_algorithm** will fail to be created or authenticated.

[RTI Issue ID SEC-856]

5.2.2.10 Data protection didn't work if fragmented samples were preallocated

A *DataReader* failed to receive samples in the following scenario:

- **data_protection_kind** was set to a value other than NONE.
- A *DataWriter* wrote fragmented samples whose actual serialized size was very close to or equal to the maximum serialized size.
- **reader_qos.reader_resource_limits.dynamically_allocate_fragmented_samples** was set to **DDS_BOOLEAN_FALSE**.

This problem has been resolved.

[RTI Issue ID SEC-871]

5.2.2.11 Persistence Service failure to store encoded samples in database

RTI Persistence Service did not work with security (**data_protection_kind** = ENCRYPT) if the actual serialized sample size was close or equal to the maximum serialized sample size. *Persistence Service* generated these errors when trying to store the samples in the database:

```
sample cannot be stored in database. Increase database_sample_buffer_max_size
```

This problem has been resolved.

[RTI Issue ID SEC-885]

5.2.2.12 Possible lack of SUBSCRIPTION_MATCHED_STATUS if a DataWriter lost liveliness with the DataReader

There was a race condition in which a *DataReader* may have never reported a SUBSCRIPTION_MATCHED_STATUS change despite successfully matching and receiving data from a *DataWriter*. This race condition occurred if all of the following conditions were true:

- The *DataReader* set its liveliness **lease_duration** to a very small duration (on the order of milliseconds).
- The *DataReader* was communicating with a *DataWriter* with **metadata_protection_kind** or **data_protection_kind** set to a value other than NONE.
- The *DataWriter* lost liveliness with the *DataReader* between the time the *DataReader* discovered the *DataWriter* and the time the *DataReader* received key material from the *DataWriter*.
- The *DataWriter* regained liveliness with the *DataReader* after the *DataReader* received key material from the *DataWriter*.

This problem has been resolved by making sure that the SUBSCRIPTION_MATCHED_STATUS change is reported as soon as the *DataWriter* becomes alive once again to the *DataReader*.

[RTI Issue ID SEC-895]

5.2.2.13 Communication failure when using MultiChannel and changing filter expression parameters

A *DataReader* failed to receive samples from a *DataWriter* in the following scenario:

- The *DataWriter* set **metadata_protection_kind** or **data_protection_kind** to a value other than NONE.
- The *DataWriter* used the MultiChannelQosPolicy to create channels 1 and 2.
- The *DataReader* used the ContentFilteredTopic API to subscribe to channel 1.
- After some time, the *DataReader* changed its filter expression parameters to subscribe to channel 2.

- The *DataWriter* wrote a sample that was sent to channel 2.
- The *DataReader* failed to receive this sample and generated this log message at the ENTITIES category and the STATUS_REMOTE verbosity:

```
key material not yet received. Dropping data...
```

This problem has been resolved. The *DataReader* will now receive the sample.

[RTI Issue ID SEC-896]

5.2.2.14 Possible deadlock risk when using TopicQueries and Security

Under the following sequence of events, a *DataWriter* setting <metadata_protection_kind> or <data_protection_kind> to a value other than NONE would have generated a "deadlock risk" error message:

1. The *DataWriter* discovered a *DataReader*.
2. The *DataReader* became inactive after **writer_qos.protocol.rtps_reliable_writer.max_heartbeat_retries** Heartbeats (HBs) due to the lack of response to HBs (default behavior) or to not making progress on the NACK messages (non default - requires setting **writer_qos.protocol.rtps_reliable_writer.inactivate_nonprogressing_readers** to TRUE).
3. The *DataWriter* received a TopicQuery from the *DataReader*.
4. The *DataWriter* received key material from the *DataReader*.
5. The *DataReader* became active after starting to respond to HBs and/or making progress on the NACK messages.
6. The *DataWriter* incorrectly did not dispatch the TopicQuery. Instead, the *DataWriter* incorrectly generated error messages similar to the following:

```
REDASWorker_enterExclusiveArea:worker rCo47750##02Rcv deadlock risk: cannot enter
0xb37ae40 of level 10 from level 30
REDACursor_modifyReadWriteArea:!enter worker (rCo47750##02Rcv)'s exclusive area
PRESPPServiceRemoteTopicQueryIterator_getQuery:!modify pres remote topicQuery
PRESPPService_dispatchMatchingTopicQueries:!get query from table
```

This problem has been fixed.

Note: Although this problem was fixed in 6.0.0, it was not initially documented in 6.0.0.

[RTI Issue ID SEC-1015]

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** > 65536. 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 subscription_data and publication_data in check_local_datawriter_match / check_local_datareader_match Are Not Populated

When calling `check_local_datawriter_match` / `check_local_datareader_match`, *Connex DDS* does not set the `subscription_data` and `publication_data` parameters. While this issue has no impact on the DDS Security builtin plugins, it could affect a custom plugin relying on those parameters.

[RTI Issue ID SEC-758]

6.4 relay_only parameter in check_remote_datareader is Not Populated

When calling `check_remote_datareader`, *Connex DDS* does not set the `relay_only` parameter. While this issue has no impact on the DDS Security builtin plugins, it could affect a custom plugin relying on this parameter.

[RTI Issue ID SEC-852]

6.5 <subject_name> not considered a match in some situations

If the order of the <subject_name> attributes in the permissions document (i.e., the order of the C, L, CN, ST elements) is not in forward or reverse order relative to the Subject field in the identity_certificate, then the <subject_name> will not be considered a match with the identity_certificate, and you will get the following error:

```
[CREATE Participant]RTI_Security_AccessControl_get_grant_from_certificate:XML file doesn't contain a grant for subject name
```

[RTI Issue ID SEC-1000]

6.6 Segmentation fault when using DomainParticipantListener with Security Plugins

While enabling a *DomainParticipant*, a segmentation fault may occur in the function `DDSDomainParticipantListener_forward_onPublicationMatched()` when using a *DomainParticipantListener* with the *Security Plugins* enabled.

[RTI Issue ID SEC-1029]

Chapter 7 Third-Party Licenses

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

Copyright (c) 1998-2015 The OpenSSL Project. All rights reserved.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).