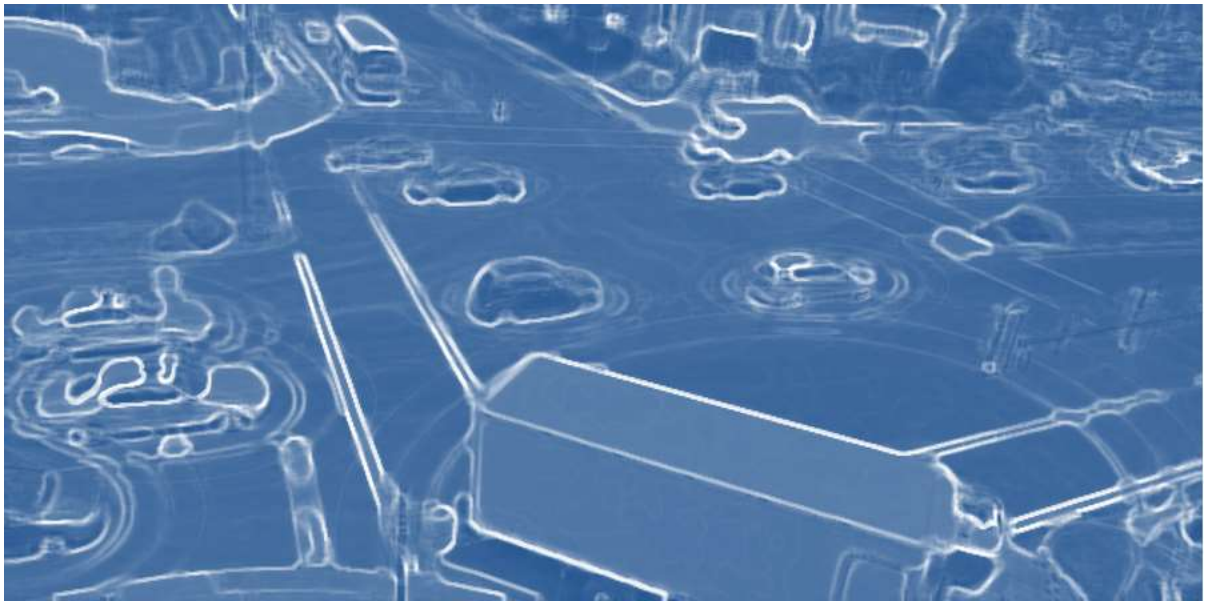

Conformance test specifications for

Wireless Access in Vehicular Environments (WAVE) — Security Services

Test Suite Structure and Test Purposes (TSS & TP)

Document Mnemonics:	WAVE-16092-TSS&TP
Revision:	[V1.1]
Revision Date:	10/10/2016

Table of Contents

1 Scope

The scope of this document provides Test Suite Structure (TSS) and Test Purposes (TP's) for WAVE Security Services as defined in IEEE 1609.2 [8]. Furthermore, the document defines a set of Test Purposes including Test Descriptions and the structure for the Test Suite. The TP's covers the Security Services requirements for BSM as specified SAE J2945/1 [1] and WSA as specified in IEEE 1609.3 [5]. The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [3] and ISO/IEC 9646-2 [4]) as well as the ETSI rules for conformance testing (ETS 300 406 [7]) are used as a basis for the test methodology.

2 References

2.1 Normative References

The following referenced documents are necessary for the application of the present document.

[1] SAE J2945/1 MAR2016: "Surface Vehicle Standard - On-board System Requirements for V2V Safety Communications"

[2] IEEE Std. 1609.12-2016 "IEEE Standard for Wireless Access in Vehicular Environments – Identifier Allocations"

[3] ISO/IEC 9646-1 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework - Part 1: General concepts".

[4] ISO/IEC 9646-2 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 2: Abstract Test Suite specification".

[5] IEEE Std 1609.3-2016 "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) — Network Services".

[6] ISO/IEC 9646-7 (1995): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

[7] ETSI ETS 300 406 (1995): "Methods for testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[8] IEEE Std. 1609.2-2016: "IEEE Standard for Wireless Access in Vehicular Environments - security Services for Applications and Management Messages".

2.2 Informative References

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EG 202 798 (V1.1.1): "Intelligent Transport Systems (ITS); Testing; Framework for conformance and interoperability testing".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in IEEE 1609.2 [8] , ISO/IEC 9646-1 [3] and in ISO/IEC 9646-7 [6] apply.

3.2 General Convention

Parameters and ***its value*** defined in SAE J2945/1 [1], IEEE 1609.12 [2], IEEE 1609.3 [5] and IEEE 1609.2 [8] used in this document are donated as **BOLD** and *ITALIC.*

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

BSM	Basic Safety Message
BI	Behaviour Invalid
BV	Behaviour Valid
CERTCH	Change Certificate
CA	Certificate Authority
EA	Enrolment Authority
ITS	Intelligent Transport Systems
IUT	Implementation Under Test
TC	Test Configuration System
TP	Test Purposes
TS	Test System
TSS	Test Suite Structure
PSID	Provider Service Identifier
PDU	Protocol Data Unit
SPDU	Secure Protocol Data Unit.
WAVE	Wireless Access in Vehicular Environments
WME	WAVE Management Entity
WSM	WAVE Short Message
WSA	WAVE Service Advertisement Message
SEND	Send message
$SPDU_{BSM}$	Represents a BSM with security credentials as per IEEE 1609.2 Standard
$SPDU_{WSA}$	Represents a WSA with security credentials as per IEEE 1609.2 Standard
SUT	System Under Test
RECV	Receive message
16092	Security Credentials

4 Prerequisites and Test Configurations

4.1 Test Configurations

This clause introduces the test configurations that is used to run the conformance testing for these definition of test purposes. These tests will be run in a lab environment in an automated fashion and controlled by the test system as shown in figure (1). The test configurations cover the various scenarios of the IEEE 1609.2 [8] test purposes.

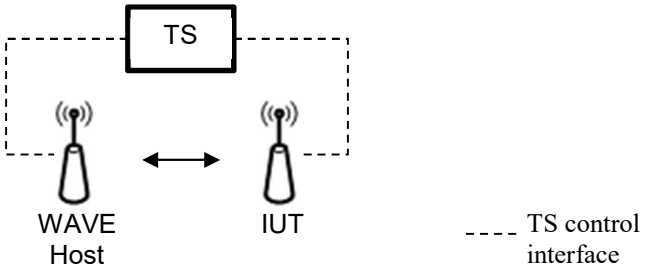

Figure 1: TC (1) Test Configuration System

4.1.1 Global Test Parameters:

Default value parameters listed in this section will be used as a global test system parameters. These values are selected based on BSM and WSA relevant security profiles as indicated in the reference column for each value.

4.1.2 $SPDU_{BSM}$ Global Test Parameters

Below are listed global test parameters / conditions that are applicable to all $SPDU_{BSM}$ test cases in this specification[1].

4.1.2.1 Value for crlSeries Parameters:

Select the default values for ***crlSeries*** according to the following table.

Table 4-1: *CrlSeries*

Parameter Name	Range of Values	Default	Reference
crlSeries	***Unit16 -*** any positive integer value in the range of (0..65535))	***1***	***[8]*** section 5.1.3

4.1.2.2 Number of ***psid*** included in the certificate:

Select the default value for ***psid*** according to the following table.

Table 4-2: *psid*

Parameter Name	Range of Values (p-encoded)	Default	Reference
psid	***1byte PSID: 0p00 to 0p7F***	***0p20*** "BSM"	[2] Section "4.1.3" Table 2
	2byte PSID: 0p80-00 to 0pBF-FF	***0p26*** "Misbehaviour for common applications."	
	3byte PSID: 0pC0-00-00 to 0pDF-FF-FF		
	4byte PSID: 0pE0-00-00-00 to 0pEF-FF-FF-FF		

4.1.2.3 ***duration*** Life Time Unit:

Select the default value for ***duration*** according to the following table.

---

[1] $SPDU_{BSM}$ will have certificates with a lifetime of a week and will be revocable. ***cracaId*** will be non-zero, ***crlSeries*** value will be 1 and ***linkageData*** is used to determine if the cert is revoked. ***reconstructionValue*** and ***r*** values will use ***compressed-y-0*** for elliptic curve point is encoding.

Table 4-3: *duration life time unit*

Parameter Name	Range of Values	Default	Reference
duration	***microseconds***	***hours***	[8] Section "D.5.2.3"
	milliseconds		
	seconds		
	minutes		
	hours		
	sixtyHours		
	years		

4.1.2.4 reconstructionValue:

Select the default value for ***reconstructionValue*** default value according to the following table.

Table 4-4: *reconstructionValue*

Parameter Name	Range of Values	Default	Reference
reconstructionValue	***x-only***	***compressed-y-0***	[8] Section "D.5.2.3"
	fill		
	compressed-y-0		
	compressed-y-1		
	uncompressed		

4.1.2.5 signature type:

Select the default value for ***signature*** according to the following table.

Table 4-5: *signature*

Parameter Name	Range of Values	Default	Reference
signature	***ecdsaNistP256Signature***	***ecdsaNistP256Signature***	[8] Section "5.3.1"
	ecdsaBrainpoolP256r1Signature		

4.1.2.6 "r" default value:

Select the default value for ***r*** according to the following table.

Table 4-6: *"r" value*

Parameter Name	Range of Values	Default	Reference
r	***x-only***	***compressed-y-0 or compressed-y-1***	[8] Section "D.5.2.3"
	fill		
	compressed-y-0		
	compressed-y-1		
	uncompressed		

4.1.2.7 Other Default values:

Select the default value for the parameter names listed on Table 4-10. The values for the parameter names listed on table 4-10 were obtained from

Table 4-7: *default values*

Parameter Name	Value	Reference
vMaxCertDigestInterval	**450 milliseconds**	[1] Section "7" Table 21
+/-DE_DSecond/2	**30 Seconds**	[1] Section "6.1.2.2.3" Table 11
vCertChangeInterval	**5 minutes**	[1] Section "7" Table 21

4.1.3 SPDU$_{WSA}$ Global Test Parameters

Below are listed global test parameters and conditions that are applicable to all SPDU$_{WSA}$ test cases in this specification[2].

4.1.3.1 *id* default value:

Select the default value for ***id*** according to the following table

Table 4-8: *id*

Parameter Name	Range of Values	Default	Reference
id	***name***	***none***	[8] Section " 5.1.3"
	binaryId		
	none		

4.1.3.2 Value for ***cracaId & crlSeries*** Parameters:

Select the default values for ***cracaId*** & ***crlSeries*** according to the following table.

Table 4-9: *cracaId & CrlSeries*

Parameter Name	Range of Values	Default	Reference
cracaId	***Octet String size(3)***	***0***	[8] Section "5.1.1.3"
crlSeries	***Integer (0 . . 65535)***	***0***	

4.1.3.3 duration Life Time Unit*:*

Select the default value for ***duration*** according to the following table.

Table 4-10: *duration life time unit*

[2] All SPDU$_{WSA}$ test cases are written with the assumption that the signer credentials (certificate) are non-revocable, because they will have short lifetimes. Certificate geographical ***region*** will be ***circularRegion*** type ***reconstructionValue*** and ***r*** values will use ***compressed-y-0*** for elliptic curve point encoding.

Parameter Name	Range of Values	Default	Reference
duration	***microseconds***	***minutes***	[8]
	milliseconds		
	seconds		
	minutes		
	hours		
	sixtyHours		
	years		

4.1.3.4 Certificate ***region*** type:

Select the default value for ***region*** according to the following table.

Table 4-11: *region*

Parameter Name	Range of Values	Default	Reference
region	***none***	***circularRegion***	[5] Annex "H" Table H.1.1.4
	identified		
	circularRegion		

4.1.3.5 reconstructionValue:

Select the default value for ***reconstructionValue*** according to the following table.

Table 4-12: *reconstructionValue*

Parameter Name	Range of Values	Default	Reference
reconstructionValue	***x-only***	***compressed-y-0***	[8] Section "D.5.2.3"
	fill	***or***	
	compressed-y-0	***compressed-y-1***	
	compressed-y-1		
	uncompressed		

4.1.3.6 ***signature*** type:

Select the default value for ***signature*** according to the following table.

Table 4-13: *signature*

Parameter Name	Range of Values	Default	Reference
signature	***ecdsaNistP256Signature***	***ecdsaNistP256Signature***	[8] Section "5.3.1"
	ecdsaBrainpoolP256r1Signature		

4.1.3.7 *"r"* default value:

Select the default value for ***r*** parameter according to the following table.

Table 4-14: *r* default value

Parameter Name	Range of Values	Default	Reference
r	***x-only***	***compressed-y-0***	[5]
	fill	***or***	Annex "H"
	compressed-y-0	***compressed-y-1***	Table H.1.1.4
	compressed-y-1		
	uncompressed		

4.2 Feature Restriction and Pre-Enrolment

4.2.1 Feature Restriction

In this clause all feature restrictions are listed:

- Encrypted PDUs are not considered
- Decrypting encrypted SPDUs are not considered.
- Peer to peer certificate distribution (P2PCD) is not considered
- Service Access Points (SAPs) are not considered.
- Certificate Revocation List (CRL) Verification Entity is not considered.

4.3 States in Initial Conditions

The description of the TP is built according to EG 202 798 [i.1].

Test purposes use a generic "Initial State" that corresponds to a state where the IUT is ready for starting the test execution. Furthermore, the IUT shall be left in this "Initial State", when the test is completed.

Being in the "Initial State" refers to the starting point of the initial device configuration. There are no pending actions, no instantiated buffers or variables, which could disturb the execution of a test.

4.3.1 Conditions for the Initial State

Figure 2 depicts the overall state diagram for a test system below.

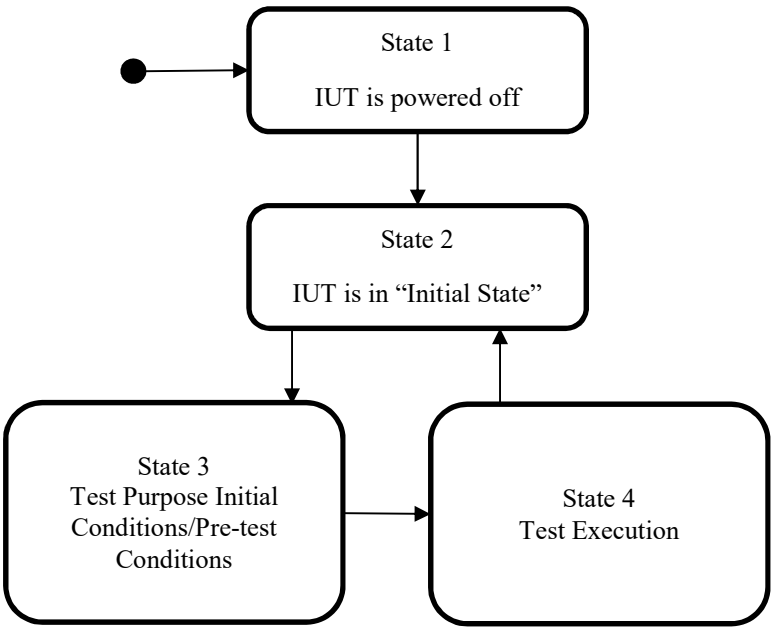

Figure 2: State Diagram

Each TP contains an initial condition. The initial condition defines the initial state in which the IUT has to be to apply the actual TP. Most of the TPs start from the "initial state" which is defined as follows:

- The IUT is powered up.
- The IUT is not transmitting or receiving messages
- The IUT is provisioned with the appropriate security credentials to enable transmission or reception of messages. That is, the IUT is configured with a valid signer credentials (certificate) as specified in SAE J2945/1 [1] and IEEE 1609.3 [5] security profiles for BSM and WSA.

Some TPs start from a different initial condition which is explicitly defined in the TP such as if an invalid behavior needs to be tested by the IUT. However, the "initial state" defined above is the starting point before the different initial conditions are established.

When the execution of the initial condition does not succeed, it leads to the assignment of an Inconclusive verdict.

5 Test Suite Structure (TSS)

5.1 Structure for security tests

The test suite is structured as a tree with the root defined as 16092. The tree is of rank 4 with the first rank is Root, 16092 second is Group, third is Sub-group and the fourth rank is the standard ISO conformance test categories. The Sub-Group (third rank) belongs to any Group member in the second rank.

5.2 Test groups

The test suite has a total of four levels. The first level is the root. The second level separates the root into various functional areas. The third level is the sub-functional areas if necessary. The fourth level is the standard ISO conformance test categories.

5.2.1 Root

The root identifies the 1609.2 protocol given in IEEE 1609.2 [8].

5.2.2 Groups

This level contains two message types identified as:

$SPDU_{BSM}$
$SPDU_{WSA}$

5.2.3 Sub-Groups

This level contains functional areas identified in Table 5-1.

Table 5-1: Functional areas

Functional areas	Description
Send/Transmit	The IUT signs and transmit WSM
Receive	The IUT receive and verifies WSM
Change Certificate	The IUT changes the signing certificate for BSM as per 2945/1 requirement

5.2.4 Categories

This level contains the standard ISO conformance test categories limited to the behaviour valid event and behaviour invalid event.

6 Test Purposes (TP)

6.1 Introduction

6.1.1 TP definition conventions

A Test Purpose (TP) is a prose description of a well-defined objective of testing. Applying to conformance testing, it focuses on a single conformance requirement or a set of related conformance requirements from the base standards [i.1]..The TP definition is built according to EG 202 798 [i.1].

The TPs are defined by the rules shown in Table 6-1.

Table 6-1: TP definition rules

Test Purpose ID	The Test Purpose ID is a unique identifier. It shall be specified according to the TP naming conventions defined in the clause below.
Test objective	Short description of test purpose objective according to the requirements from the base standard.
References	The reference indicates the sub-clauses of the reference standard specifications in which the conformance requirement is expressed.
Test Configuration	The Config Id references the test configuration selected for this TP.
PICS Selection	Reference to the PICS statement involved for selection of the TP. It may contain a Boolean expression.
Pre-Test Conditions	A list of test specific pre-conditions that need to be met by the SUT including information about equipment configuration, i.e. precise description of the initial state of the SUT required to start executing the test sequence
Test Sequence	An ordered list of equipment operation and observations. In case of a conformance test description the test sequence contains also the conformance checks as part of the observations

Event Types	
Stimulus	Corresponds to an event that enforces an IUT to proceed with a specific protocol action, like sending a message for instance.
Check	Ensures the receipt of protocol messages on reference points with valid content.
Verify	Consists of verifying that the IUT behaves according to the expected behavior (for instance the IUT behavior shows that it receives the expected message).
Configure	Corresponds to an action to modify the IUT configuration.

When a conformance test has a sequencing requirement, these are described using a format in the table 3 derived from [i.1]

6.1.2 TP Identifier naming conventions

The identifier of the TP is built according to Table 6-2.

Table 6-2:TP naming convention

Identifier	TP-<root>-<gr>-<sgr>-<x>-<nn> or TP-<root>-<gr>-<x>-<nn> when no <sgr>		
	<root> = root	16092	1609.2
	<gr> = group	$SPDU_{BSM}$	Secure Basic Safety Message
		$SPDU_{WSA}$	Secure Wave Service Advertisement message
	<sgr> =sub- group	SEND	Send Message
		RECV	Receive Message
		CERTCH	Change Certificate
	<x> = type of testing	BV	Valid Behaviour tests
		BI	Invalid Syntax or Behaviour Tests
	<nn> = sequential number		01 to 99

6.1.3 Rules for the behaviour description

The description of the TP is built according to EG 202 798 [i.1].

The base standards are not using finite state machine concept. As consequence, the test purposes use a generic "Initial State" that corresponds to a state where the IUT is ready for starting the test execution. Furthermore, the IUT shall be left in this "Initial State", when the test is completed.

Being in the "Initial State" refers to the starting point of the initial device configuration. There are no pending actions, no instantiated buffers or variables, which could disturb the execution of a test.

6.1.4 References

All Test Purposes are derived from requirements defined in 1609.2 [8]. Traceability between TPs and sub-clauses of referenced standard specifications is established in Table A- 1 for BSM and Table A-2 for WSA messages. For each PICS, a reference section from 1609.2 [8] is listed and applicable test purposes are identified in the TP ID column.

6.1.5 PICS selection and mnemonics for reference

Table A- 1 and Table A- 2 includes a subset of PICS defined in 1609.2 [8] with a traceability to TPs included in the TP ID column. Some TPs are directly derived from SAE J2945/1[1] requirements and do not refer to any PICS from 1609.2[8]. In this case the SAE J2945/1[1] requirement that is used to generate the test purpose is listed in the “Reference section” of the TP.

Table 6-3 lists mnemonic names and maps them to a subset of PICS item number. This is a partial list of PICS used in selecting of certain TPs or TPs which incorporated variances.

6.1.6 Mnemonics for PICS reference

The following table lists mnemonic names and maps them to the PICS item number. This is a partial list of PICS used in selecting TPs. The complete list of PICS with traceability to TPs is included in Appendix A.

Table 6-3: **Mnemonics for PICS reference**

Mnemonic	PICS item
PIC_Generate_SignedData	[8] Annex A, S1.2.2
PIC_Generate_Using_Valid_HashAlgorithm	[8] Annex A, S1.2.2.1
PIC_Generate_Signing_With_SHA256	[8] Annex A, S1.2.2.1.1
PIC_Generate_Signed_Data_payload	[8] Annex A, S1.2.2.2
PIC_Generate_With_Payload_Containing_Data	[8] Annex A, S1.2.2.2.1
PIC_Generate_With_generationTime_In_security_headers	[8] Annex A, S1.2.2.2.3
PIC_Generate_With_generationLocation_In_security_headers	[8] Annex A, S1.2.2.2.5
PIC_Generate_Support_SignerIdentifier	[8] Annex A, S1.2.2.3
PIC_Generate_Of_Type_digest	[8] Annex A, S1.2.2.3.1
PIC_Generate_Of_Type_certificate	[8] Annex A, S1.2.2.3.2
PIC_Generate_Max_Number_Of_Certificates_In_The_chain	[8] Annex A, S1.2.2.3.2.1
PIC_Generate_Signature	[8] Annex A, S1.2.2.4
PIC_Generate_Ecdsa256_Signature	[8] Annex A, S1.2.2.4.1
PIC_Generate_Ecdsa256_Signature_Using_ NIST p256	[8] Annex A, S1.2.2.4.1.1
PIC_Generate_Signature_With_Compressed_r_value	[8] Annex A, S1.2.2.4.1.5
PIC_Generate_Support_signing_Implicit_Certificate	[8] Annex A, S1.2.2.8
PIC_Verify_Ieee1609DoT2Data_Containing_SignedData	[8] Annex A, S1.3.2
PIC_Verify_Using_Valid_HashAlgorithm	[8] Annex A, S1.3.2.1
PIC_Verify_Signing_With_SHA256	[8] Annex A, S1.3.2.1.1
PIC_Verify_Signed_Data_payload	[8] Annex A, S1.3.2.2
PIC_Verify_With_Payload_Containing_Data	[8] Annex A, S1.3.2.2.1
PIC_Verify_With_generationTime_In_security_headers	[8] Annex A, S1.3.2.2.3
PIC_Verify_With_generationLocation_In_security_headers	[8] Annex A, S1.3.2.2.5
PIC_Verify_Support_SignerIdentifier	[8] Annex A, S1.3.2.3
PIC_Verify_Of_Type_digest	[8] Annex A, S1.3.2.3.1
PIC_Verify_Of_Type_certificate	[8] Annex A, S1.3.2.3.2
PIC_Verify_Max_Number_Of_Certificates_In_The_chain	[8] Annex A, S1.3.2.3.2.1
PIC_Verify_Signature	[8] Annex A, S1.3.2.4
PIC_Verify_ecdsa256_Signature	[8] Annex A, S1.3.2.4.1
PIC_Verify_ecdsa256_Signature_Using_ NIST_p256	[8] Annex A, S1.3.2.4.1.1
PIC_Verify_Signature_With_Compressed_r_value	[8] Annex A, S1.3.2.4.1.4
PIC_Verify_SignedData_fails_if_certificate_is_not_valid	[8] Annex A, S1.3.2.5
PIC _Verify_Reject_data if_certificate_doesn't_have_proper_appPermissions	[8] Annex A, S1.3.2.5.2
PIC_Verify_Reject_data_if_generationTime_not_available	[8] Annex A,S 1.3.2.10.4
PIC_Verify_Reject_data_if_generationLocation_not_available	[8] Annex A, S1.3.2.10.5

6.1.7 Sources of TP definitions

All TPs are specified according to IEEE 1609.2 [8] and SAE J2945/1 [1]. Test purposes for 1609.2

6.1.8 Secure Protocol Data Unit for Basic Safety Messages ($SPDU_{BSM}$)

6.1.8.1 Transmission of packets

Identifier	TP-16092- $SPDU_{BSM}$-SEND-BV-01
Summary	Validate that the IUT will generate a valid $SPDU_{BSM}$ security header. Security header shall include, ***protocolVersion***, ***content***, ***signedData***, ***hashId***, ***tbsData***, ***headerInfo*** and doesn't include ***expiryTime*** nor ***generationLocation***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit more than one $SPDU_{BSM}$ per second as defined in Table 7-1	
2	Stimulus	The IUT transmits $SPDU's_{BSM}$	
3	Verify	$SPDU_{BSM}$ ***Ieee1609Dot2Data*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
4	Verify	$SPDU_{BSM}$ ***Ieee1609Dot2Data*** contains ***content*** indicating signedData	Pass/Fail
5	Verify	$SPDU_{BSM}$ ***signedData*** contains ***hashId*** indicating ***sha256***	Pass/Fail
6	Verify	$SPDU_{BSM}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
7	Verify	$SPDU_{BSM}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	Pass/Fail
8	Verify	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	Pass/Fail
9	Verify	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	Pass/Fail
10	Verify	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***expiryTime***	Pass/Fail
11	Verify	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***generationLocation***	Pass/Fail

Identifier	TP-16092- $SPDU_{BSM}$-SEND-BV-02
Summary	Validate that the $SPDU_{BSM}$ digitally signed by certificate contains a valid 1609.2 certificate data structure. The certificate shall include a valid ***signer*** info, ***toBeSigned linkageData*** information, valid ***region*** information and ***ecdsaP256Signature*** type.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit more than one BSM per second as defined in Table 7-3	
2	Stimulus	The IUT transmits $SPDU_{BSM}$	
3	Verify	$SPDU_{BSM}$ ***signer*** contains ***certificate*** indicating ***version*** value = ***0x03***	Pass/Fail
4	Verify	$SPDU_{BSM}$ ***signer*** contains ***type*** indicating ***implicit***	Pass/Fail

5	Verify	SPDU$_{BSM}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating HashedId8 (a non-zero value of size 8 octets)	Pass/Fail
6	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***id*** indicating linkageData	Pass/Fail
7	Verify	SPDU$_{BSM}$ ***linkageData*** contains ***iCert*** indicating a value of size 2 octets	Pass/Fail
8	Verify	SPDU$_{BSM}$ ***linkageData*** contains ***linkage-value*** indicating value of size 9 octets	Pass/Fail
9	Verify	SPDU$_{BSM}$ ***linkageData*** contains ***group-linkage-value*** containing ***jValue*** indicating a value of size 4 octets	Pass/Fail
10	Verify	SPDU$_{BSM}$ ***linkageData*** contains ***group-linkage-value*** containing ***value*** indicating a value of size 9 octets	Pass/Fail
11	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***cracaId*** indicating a non-zero value of size 3 octets	Pass/Fail
12	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***crlSeries*** indicating a value =***0x01***	Pass/Fail
13	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***start*** indicating ***Time32*** (a non-zero value of size 4 octets)	Pass/Fail
14	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***duration*** containing ***hours*** indicating Unit16 (a non-zero Integer value of size 2 octets)	Pass/Fail
15	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***region*** containing a sequence of ***identifiedRegion*** indicating ***countryOnly*** values ***0x7C, 0x1E4*** and ***0x348***	Pass/Fail
16	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains a sequence of ***appPermission*** with PSIDs indicating values of 0***p20*** and ***0p26***	Pass/Fail
17	Verify	SPDU$_{BSM}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** (value of size 32 octets)	Pass/Fail
18	Verify	SPDU$_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0 or compressed-y-1*** consists of octet size 32)	Pass/Fail
19	Verify	SPDU$_{BSM}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	Pass/Fail

Identifier	TP-16092- SPDU$_{BSM}$-SEND-BV-03
Summary	Validate that the SPDU$_{BSM}$ signed by certificate digest contains a valid 1609.2 data structure. The SPDU$_{BSM}$ shall include, ***protocolVersion***, ***content***, ***signedData***, ***hashId***, ***tbsData***, ***headerInfo***, ***signer***, ***ecdsaP256Signature*** and doesn't include ***expiryTime*** nor ***generationLocation***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit more than one SPDU$_{BSM}$ per second as defined in Table 7-2	
2	Stimulus	The IUT transmits SPDU's$_{BSM}$	
3	Verify	SPDU$_{BSM}$ ***Ieee1609Dot2Data*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
4	Verify	SPDU$_{BSM}$ ***Ieee1609Dot2Data*** contains ***content*** indicating ***signedData***	Pass/Fail

5	Verify	$SPDU_{BSM}$ ***signedData*** contains ***hashId*** indicating ***sha256***	Pass/Fail
6	Verify	$SPDU_{BSM}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
7	Verify	$SPDU_{BSM}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	Pass/Fail
8	Verify	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	Pass/Fail
9	Verify	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	Pass/Fail
10	Verify	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***expiryTime***	Pass/Fail
11	Verify	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***generationLocation***	Pass/Fail
12	Verify	$SPDU_{BSM}$ contains ***signer*** containing ***digest*** indicating HashedId8 (a non-zero value of size 8 octets)	Pass/Fail
13	Verify	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0 or compressed-y-1*** consists of octet size 32)	Pass/Fail
14	Verify	$SPDU_{BSM}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	Pass/Fail

Identifier	TP-16092- $SPDU_{BSM}$-SEND-BV-04
Summary	Validate that the $SPDU_{BSM}$ is digitally signed by certificate at least every ***vMaxCerDigestInterval***.
Test Configuration	TC (1)
IUT	IUT
Reference:	SAE J2945 [1] Table 10 "Security Profile for Transmitting BSMs".
PICS Selection	

Pre-test conditions

- The IUT being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit more than one $SPDU_{BSM}$ per 450 ms as defined in Table 7-3	
2	Stimulus	The IUT transmits $SPDU's_{BSM}$	
3	Verify	IUT transmitted $SPDU_{BSM}$ at TIME_1 contains ***signer*** indicating ***certificate*** where the low order 8 octets of the sha256 hash is calculated for the Certificate (ID1)	Pass/Fail
4	Verify	IUT transmitted the next $SPDU_{BSM}$ at TIME_2 (TIME_2>TIME_1)	Pass/Fail
5	Verify	IUT transmitted at TIME_2 contains ***signer*** indicating ***certificate*** where the low order 8 octets of the sha256 hash is calculated for the Certificate (ID2)	Pass/Fail
6	Verify	ID2! = ID1	Pass/Fail
7	Verify	(TIME_2 - TIME_1) ' greater than or equal to' ***vMaxCerDigestInterval***	Pass/Fail

Identifier	TP-16092- $SPDU_{BSM}$-SEND-BV-05
Summary	Validate that a $SPDU_{BSM}$ containing a certificate ***digest*** is signed using a valid digital signature computed over entire payload using ***ecdsaP256Signature*** type.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions			
• The IUT is being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to transmit more than one $SPDU_{BSM}$ per second	
2	Stimulus	The IUT transmits $SPDU_{BSM}$	
3	Verify	The IUT transmitted $SPDU's_{BSM}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** where HashedId8 is referenced to pre-loaded certificate on the IUT and containing ***verificationkeyIndicator*** (KEY)	Pass/Fail
4	Verify	$SPDU_{BSM}$ Signature contains ***ecdsaP256Signature*** indicating ***r*** and ***s*** values verifiable using KEY.	Pass/Fail

Identifier	TP-16092-$SPDU_{BSM}$-SEND-BV-06
Summary	Validate that a $SPDU_{BSM}$ digitally signed by certificate contains a valid ***signature*** computed over entire payload using ***ecdsaP256Signature*** type.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions			
• The IUT being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to transmit more than one $SPDU_{BSM}$ per second as defined in Table 7-3	
2	Stimulus	The IUT transmits $SPDU_{BSM}$	
3	Verify	$SPDU_{BSM}$ ***signer*** contains certificate indicating ***type implicit***	Pass/Fail
4	Verify	$SPDU_{BSM}$ ***toBeSigned*** contains ***psid*** indicating a value=***0p20***	Pass/Fail
5	Verify	$SPDU_{BSM}$ **toBeSigned** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** (value of size 32 octets) (KEY)	Pass/Fail
6	Verify	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** and ***s*** values verifiable using (KEY)	Pass/Fail

6.1.8.2 Reception of packets

Identifier	TP-16092-$SPDU_{BSM}$-RECV-BV-01
Summary	Validate that the IUT will indicate a valid security credentials for a well-formed $SPDU_{BSM}$ security header. Security header shall include ***protocolVersion, signedData, tbsData, headerInfo*** and doesn't include ***expiryTime*** nor ***generationLocation***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT is being initialized
Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second	
2	Check	$SPDU_{BSM}$ ***Ieee1609Dot2Data*** contains ***protocolVersion*** indicating value = ***0x03***	
3	Check	$SPDU_{BSM}$ Ieee1609Dot2Data contains ***content*** indicating ***signedData***	
4	Check	$SPDU_{BSM}$ ***signedData*** contains ***hashId*** indicating ***sha256***	
5	Check	$SPDU_{BSM}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	
6	Check	$SPDU_{BSM}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	
7	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	
8	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	
9	Check	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***expiryTime***	
10	Check	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***generationLocation***	
11	Stimulate	The IUT receives $SPDU's_{BSM}$	
12	Verify	IUT indicate that the security header for $SPDU_{BSM}$ is formed correctly	Pass/Fail

Identifier	TP-16092-$SPDU_{BSM}$-RECV-BV-02
Summary	Validate that the IUT will indicate a valid security credential for a well-formed $SPDU_{BSM}$ signed by implicit certificate. The BSM shall include ***protocolVersion, signedData, tbsData, headerInfo, signer***, ***toBeSigned, linkageData, ecdsaP256Signature*** type and doesn't include ***expiryTime*** nor ***generationLocation***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second as defined in Table 7-3	
2	Check	$SPDU_{BSM}$ ***signer*** contains ***certificate*** indicating ***version*** value = ***0x03***	
3	Check	$SPDU_{BSM}$ ***signer*** contains ***type*** indicating ***implicit***	
4	Check	$SPDU_{BSM}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating ***HashedId8*** a non-zero value of size 8 octets	
5	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***id*** indicating linkageData	
6	Check	$SPDU_{BSM}$ ***linkageData*** contains ***iCert*** indicating a value of size 2 octets	
7	Check	$SPDU_{BSM}$ ***linkageData*** contains ***linkage-value*** indicating value of size 9 octets	
8	Check	$SPDU_{BSM}$ ***linkageData*** contains ***group-linkage-value*** containing ***jValue*** indicating a value of size 4 octets	
9	Check	$SPDU_{BSM}$ ***linkageData*** contains ***group-linkage-value*** containing ***value*** indicating a value of size 9 octets	
10	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***cracaId*** indicating a non-zero value of size 3 octets	
11	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***crlSeries*** indicating a value =***0x01***	

12	Check	$SPDU_{BSM}$ ***toBeSigned*** contains start indicating ***Time32*** (a non-zero value of size 4 octets)	
13	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***duration*** containing ***hours*** indicating ***Unit16*** (a non-zero Integer value of size 2 octets)	
14	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***region*** containing a sequence of ***identifiedRegion*** indicating ***countryOnly*** values ***0x7C, 0x1E4*** and ***0x348***	
15	Check	$SPDU_{BSM}$ ***toBeSigned*** contains a sequence of ***appPermission*** with PSIDs indicating values of 0***p20*** and ***0p26***	
16	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** (value of size 32 octets)	
17	Check	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0*** or ***compressed-y-1*** consists of octet size 32)	
18	Check	$SPDU_{BSM}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	
19	Stimulate	The IUT receives $SPDU_{BSM}$.	
20	Verify	IUT indicates that the $SPDU_{BSM}$ holds a valid security credentials.	Pass/Fail

Identifier		TP-16092-$SPDU_{BSM}$-RECV-BV-03	
Summary		Validate that the IUT will indicate a valid security credential for a well-formed $SPDU_{BSM}$ signed by certificate ***digest*** of known certificate. The $SPDU_{BSM}$ shall include***, protocolVersion, content, signedData, tbsData, headerInfo, signer, ecdsaP256Signature*** type and doesn't include ***expiryTime*** nor ***generationLocation***.	
Test Configuration		TC (1)	
IUT		IUT	
Reference:			
PICS Selection			
Pre-test conditions			
• The IUT is being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second as defined in Table 7-2	
2	Check	$SPDU_{BSM}$ Ieee1609Dot2Data contains ***protocolVersion*** indicating value = ***0x03***	
3	Check	$SPDU_{BSM}$ Ieee1609Dot2Data contains ***content*** indicating ***signedData***	
4	Check	$SPDU_{BSM}$ ***signedData*** contains hashId indicating ***sha256***	
5	Check	$SPDU_{BSM}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	
6	Check	$SPDU_{BSM}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	
7	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	
8	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	
9	Check	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***expiryTime***	
10	Check	$SPDU_{BSM}$ ***headerInfo*** doesn't include ***generationLocation***	
11	Check	$SPDU's_{BSM}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** (a non-zero value of size 8 octets)	
12	Check	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0*** or ***compressed-y-1*** consists of octet size 32)	

13	Check	$SPDU_{BSM}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	
14	Stimulate	IUT receives $SPDU's_{BSM}$	
15	Verify	IUT indicates that the $SPDU_{BSM}$ holds a valid security credentials.	Pass/Fail

Identifier		TP-16092-$SPDU_{BSM}$-RECV-BV-04	
Summary		Validate that the IUT will indicate a valid security credential for a $SPDU_{BSM}$ digitally signed by ***certificate***, which includes ***generationTime*** within ***+/- DE_DSecond/2*** of the current time and the BSM ***generationTime*** is earlier than the expiration time of the signing certificate.	
Test Configuration		TC (1)	
IUT		IUT	
Reference:		SAE J2945 [1] Table 11 "Security Profile for Receiving BSMs"	
PICS Selection			
Pre-test conditions			
• The IUT being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second as defined in Table 7-3	
2	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	
3	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating a TIME_1 where (CUR_TIME - DE_DSecond/2 'less or equal' TIME_1 'less or equal' CUR_TIME + DE_DSecond/2)	
4	Check	$SPDU's_{BSM}$ ***signer*** contains ***certificate*** indicating ***type implicit***	
5	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***start*** & ***duration*** indicating EXP_TIME where (CUR_TIME 'less or equal' EXP_TIME)	
6	Stimulate	The IUT receives $SPDU's_{BSM}$.	
7	Verify	IUT indicates that the $SPDU_{BSM}$ holds a valid security credentials.	Pass/Fail

Identifier		TP-16092-$SPDU_{BSM}$-RECV-BV-05	
Summary		Validate that the IUT will indicate a valid security credential for a $SPDU_{BSM}$ digitally signed by certificate ***digest*** which includes ***generationTime*** within ***+/- DE_DSecond/2*** from the current time, and the $SPDU_{BSM}$ is generated before the expiration time of the signing certificate digest pre-stored on the device.	
Test Configuration		TC (1)	
IUT		IUT	
Reference:		SAE J2945 [1] Table 11 "Security Profile for Receiving BSMs"	
PICS Selection			
Pre-test conditions			
• The IUT is being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second as defined in Table 7-2	
2	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	
3	Check	$SPDU_{BSM}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** (ID1)	
4	Stimulate	The IUT receives $SPDU's_{BSM}$	

5	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***generationTime*** indicating TIME_1 where (CUR_TIME – DE_DSecond/2 'less or equal' TIME_1 'less or equal' CUR_TIME + DE_DSecond/2)	
6	Check	$SPDU_{BSM}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** (ID1)	
7	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***start*** & ***duration*** indicating EXP_TIME where (CUR_TIME 'less or equal' EXP_TIME)	
8	Stimulate	The IUT receives SPDU's$_{BSM}$	
7	Verify	IUT indicates that the $SPDU_{BSM}$ holds a valid security credentials.	Pass/Fail

6.1.8.3 Certificate Rotation Validation

Identifier	TP-16092-$SPDU_{BSM}$-CERTCHG-BV-01
Summary	Validate that the $SPDU_{BSM}$ contains either ***certificate*** or certificate ***digest*** referencing the same certificate for (***vCertChangeInterval***) minutes and BSM changes the referenced certificate after (***vCertChangeInterval***).
Test Configuration	TC (1)
IUT	IUT
Reference:	SAE J2945 [1] section 6.3.5 "6.5.3-V2V-SECPRIV-CERTCHG-001"
PICS Selection	

Pre-test conditions

- The IUT being initialized
- Time is set at the moment when digest changes

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit more than one $SPDU_{BSM}$ per second	
2	Stimulus	The IUT transmits $SPDU_{BSM}$ at TIME_1	
3	Verify	SPDU's$_{BSM}$ contains **signer** containing **digest** indicating **HashedId8** (ID1) if yes go to step 5	Pass/Fail
4	Verify	$SPDU_{BSM}$ signer contains ***certificate*** where the low order 8 octets of the sha-256 hash is calculated for the certificate (ID1)	Pass/Fail
5	Verify	The IUT sends the next $SPDU_{BSM}$ at TIME_2 where (TIME_2-TIME_1) 'less' 1sec	Pass/Fail
6	Verify	SPDU's$_{BSM}$ contains **signer** containing **digest** indicating HashedId8 (ID2) if yes go to step 8	Pass/Fail
7	Verify	$SPDU_{BSM}$ **signer** contains ***certificate*** where the low order 8 octets of the **sha-256 hash** is calculated for the certificate (ID2)	Pass/Fail
8	Verify	where ID2 = ID1	Pass/Fail
9	Verify	IUT sends $SPDU_{BSM}$ at TIME_3	Pass/Fail
10	Verify	SPDU's$_{BSM}$ contains **signer** containing **digest** indicating HashedId8 (ID3) if yes go to step 12	Pass/Fail
11	Verify	$SPDU_{BSM}$ signer contains ***certificate*** where the low order 8 octets of the sha-256 hash is calculated for the certificate (ID3)	Pass/Fail
12	Verify	where ID3! = ID2	Pass/Fail
13	Verify	vCertChangeInterval 'less or equal' (TIME_3 - TIME_2) 'less or equal' vCertChangeInterval+ 30 sec	Pass/Fail

6.1.8.4 Reception of packets – invalid behaviour tests

Identifier	TP-16092-$SPDU_{BSM}$-RECV-BI-01

Summary	Validate that the IUT will indicate an invalid security credentials for a $SPDU_{BSM}$ signed by certificate digest, which failed verification due to incorrect signature.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT is being initialized

Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second	
2	Check	$SPDU_{BSM}$ ***headerInfo*** contains ***psid*** indicating value = ***0p20***	
3	Check	SPDU's$_{BSM}$ contains ***signer*** containing ***digest*** indicating HashedId8 (a non-zero value of size 8 octets)	
4	Check	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** type indicating ***r*** and ***s signature*** not verifiable using KEY	
5	Stimulate	The IUT receives SPDU's$_{BSM}$	
6	Verify	IUT indicates that the $SPDU_{BSM}$ holds an invalid security credentials	Pass/Fail

Identifier	TP-16092-$SPDU_{BSM}$-RECV-BI-02
Summary	Validate that the IUT will indicate an invalid $SPDU_{BSM}$ signed by implicit certificate which failed verification due to incorrect signature.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT being initialized

Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{BSM}$ per second	
2	Check	SPDU's$_{BSM}$ ***signer*** contains ***certificate*** indicating ***type implicit***	
3	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***psid*** indicating a value=***0p20***	
4	Check	$SPDU_{BSM}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** (value of size 32 octets) (KEY)	
5	Check	$SPDU_{BSM}$ ***signature*** contains ***ecdsaP256Signature*** type indicating ***r*** and ***s signature*** not verifiable using KEY	
6	Stimulate	The IUT receives SPDU's$_{BSM}$	
7	Verify	IUT indicates that the $SPDU_{BSM}$ holds an invalid security credentials	Pass/Fail

6.1.9 Secure Protocol Data Unit for WAVE Service Advertisements Messages ($SPDU_{WSA}$)

6.1.9.1 Transmission of packets

Identifier	TP-16092- $SPDU_{WSA}$-SEND-BV-01
Summary	Validate that the IUT will generate a correct $SPDU_{WSA}$ security header structure. That is, the WSA security header shall include ***protocolVersion, content, signedData, tbsData and headerInfo***.
Test Configuration	TC (1)

IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT is being initialized

Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to transmit one or more $SPDU_{WSA}$ per second as defined in Table 7-5	
2	Stimulus	The IUT transmits WSAs	
3	Verify	$SPDU_{WSA}$ ***Ieee1609Dot2Data*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
4	Verify	$SPDU_{WSA}$ ***Ieee1609Dot2Data*** contains ***content*** indicating ***signedData***	Pass/Fail
5	Verify	$SPDU_{WSA}$ ***signedData*** contains ***hashId*** indicating ***sha256***	Pass/Fail
6	Verify	$SPDU_{WSA}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
7	Verify	$SPDU_{WSA}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	Pass/Fail
8	Verify	$SPDU_{WSA}$ ***headerInfo*** contains ***psid*** indicating value =***0p80-07***	Pass/Fail
9	Verify	$SPDU_{WSA}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	Pass/Fail
10	Verify	$SPDU_{WSA}$ ***headerInfo*** contains ***expiryTime*** indicating a ***Time64*** (non-zero value of size 8 bytes	Pass/Fail
11	Verify	$SPDU_{WSA}$ ***headerInfo*** contains ***generationLocation*** indicating ***latitude*** (-900000000 . . 900000000) ***longitude*** (-1799999999 . . 1800000000) ***elevation*** Unit16	Pass/Fail

Identifier	TP-16092-$SPDU_{WSA}$-SEND-BV-02
Summary	Validate that the IUT will generate a correct $SPDU_{WSA}$ certificate data structure. The $SPDU_{WSA}$ shall include ***signer*** information, ***toBesigned*** data structure and a valid ***ecdsaP256Signature*** type.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT is being initialized

Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to transmit one or more $SPDU_{WSA}$ per second as defined in Table 7-6	
2	Stimulus	The IUT transmits SPDU's$_{WSA}$	
3	Verify	$SPDU_{WSA}$ ***signer*** contains ***certificate*** indicating ***version*** value= ***0x03***	
4	Verify	$SPDU_{WSA}$ ***signer*** contains ***type*** indicating ***implicit***	Pass/Fail
5	Verify	$SPDU_{WSA}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating ***HashedId8*** (a non-zero value of size 8 octets)	Pass/Fail
6	Verify	$SPDU_{WSA}$ ***toBeSigned*** contains ***id*** indicating ***none***	Pass/Fail
7	Verify	$SPDU_{WSA}$ ***toBeSigned*** contains ***cracaId*** indicating value = ***0x0***	Pass/Fail
8	Verify	$SPDU_{WSA}$ ***toBeSigned*** contains ***crlSeries*** indicating value=***0x0***	Pass/Fail

9	Verify	SPDU$_{WSA}$ ***toBeSigned*** contains ***start*** indicating ***Time32*** (a non-zero value of size 4 octets)	Pass/Fail
10	Verify	SPDU$_{WSA}$ ***toBeSigned*** contains ***duration*** containing ***minutes*** indicating ***Unit16*** (a non-zero value of size 2 bytes)	Pass/Fail
11	Verify	SPDU$_{WSA}$ ***toBeSigned*** contains ***region*** containing ***circularRegion*** indicating ***latitude*** INTEGER (-900000000..900000000) ***longitude*** INTEGER (-1799999999..1800000000) ***radius*** INTEGER (0 .. 65535)	Pass/Fail
12	Verify	SPDU$_{WSA}$ ***toBeSigned*** contains ***appPermission*** indicating ***psid*** value= ***0p80-07***	Pass/Fail
13	Verify	SPDU$_{WSA}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** (value of size 32 octets)	Pass/Fail
14	Verify	SPDU$_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (a value of ***compressed-y-0*** or ***compressed-y-1*** size of 32 octets)	Pass/Fail
15	Verify	SPDU$_{WSA}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	Pass/Fail

Identifier	TP-16092-SPDU$_{WSA}$-SEND-BV-03
Summary	Validate that the IUT will generate a well-formed SPDU$_{WSA}$ signed by certificate ***digest*** of known certificate. The SPDU$_{WSA}$ shall include***, protocolVersion, content, signedData, tbsData, headerInfo, signer, ecdsaP256Signature***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit one or more SPDU$_{WSA}$ per second as defined in Table 7-7	Pass/Fail
2	Stimulus	The IUT transmits SPDU's$_{WSA}$	Pass/Fail
4	Verify	SPDU$_{WSA}$ Ieee1609Dot2Data contains ***content*** indicating ***signedData***	Pass/Fail
5	Verify	SPDU$_{WSA}$ ***signedData*** contains hashId indicating ***sha256***	Pass/Fail
6	Verify	SPDU$_{WSA}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	Pass/Fail
7	Verify	SPDU$_{WSA}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	Pass/Fail
8	Verify	SPDU$_{WSA}$ ***headerInfo*** contains ***psid*** indicating value =***0p80-07***	Pass/Fail
9	Verify	SPDU$_{WSA}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	Pass/Fail
10	Verify	SPDU$_{WSA}$ ***headerInfo*** contains ***expiryTime*** indicating a ***Time64*** (non-zero value of size 8 bytes	Pass/Fail
11	Verify	SPDU$_{WSA}$ ***headerInfo*** contains ***generationLocation*** indicating ***latitude*** (-900000000 . . 900000000) ***longitude*** (-1799999999 . . 1800000000) ***elevation*** Unit16	Pass/Fail
12	Verify	SPDU$_{WSA}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** (a non-zero value of size 8 octets)	Pass/Fail
13	Verify	SPDU$_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0*** or ***compressed-y-1*** consists of octet size 32)	Pass/Fail

14	Verify	$SPDU_{WSA}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	Pass/Fail

Identifier	TP-16092-$SPDU_{WSA}$-SEND-BV-04
Summary	Validate that the IUT will generate $SPDU_{WSA}$ message digitally signed by ***certificate*** that contains a valid ***signature*** computed over the entire payload using ***ecdsaP256Signature*** type***.***
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit one or more $SPDU_{WSA}$ per second as defined Table 7-6	
2	Stimulus	The IUT transmits $SPDU's_{WSA}$	
3	Verify	$SPDU_{WSA}$ ***headerInfo*** contains ***psid*** indicating value =***0p80-07***	Pass/Fail
4	Verify	$SPDU_{WSA}$ ***signer*** contains ***certificate*** indicating ***version*** value = ***0x03***	Pass/Fail
5	Verify	$SPDU_{WSA}$ ***signer*** contains ***type*** indicating ***implicit***	Pass/Fail
6	Verify	$SPDU_{WSA}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating 'CERTID'	Pass/Fail
7	Verify	$SPDU_{WSA}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating (RECVAL) which creates the public key (KEY) by invoking the 1609.2 reconstruction function on (RECVAL) and the public key of the certificate stored on IUT and identified by (CERTID)	Pass/Fail
8	Verify	$SPDU_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** verifiable using (KEY)	Pass/Fail

6.1.9.2 Reception of packets

Identifier	TP-16092-$SPDU_{WSA}$-RECV-BV-01
Summary	Validate that the IUT will indicate a valid security credentials for a well-formed $SPDU_{WSA}$ security header. That is, the $SPDU_{WSA}$ shall include ***protocolVersion***, ***content***, ***signedData***, ***tbsData*** and ***headerInfo***.
Test Configuration	TC1
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to receive more than one $SPDU_{WSA}$ per second as defined in Table 7-5	
2	Check	$SPDU_{WSA}$ ***Ieee1609Dot2Data*** contains ***protocolVersion*** indicating (value = ***0x03***)	
3	Check	$SPDU_{WSA}$ ***Ieee1609Dot2Data*** contains ***content*** indicating ***signedData***	
4	Check	$SPDU_{WSA}$ ***signedData*** contains ***hashId*** indicating ***sha256***	

5	Check	$SPDU_{WSA}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	
6	Check	$SPDU_{WSA}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	
7	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***psid*** indicating value =***0p80-07***	
8	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	
9	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***expiryTime*** indicating a ***Time64*** (non-zero value of size 8 bytes	
10	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***generationLocation*** indicating ***latitude*** (-900000000 . . 900000000) ***longitude*** (-1799999999 . . 1800000000) ***elevation*** Unit16	
11	Stimulate	The IUT receives SPDU's$_{WSA}$	
12	Verify	IUT indicates that the $SPDU_{WSA}$ message holds a valid security credentials.	Pass/Fail

Identifier		TP-16092-$SPDU_{WSA}$-RECV-BV-02	
Summary		Validate that the IUT will indicate a valid security credentials for a well-formed $SPDU_{WSA}$ signed by implicit certificate. That is, the certificate data structure shall include ***signer, toBesigned*** data structure and ***ecdsaP256Signature*** type.	
Test Configuration		TC1	
IUT		IUT	
Reference:			
PICS Selection			
Pre-test conditions			
• The IUT is being initialized			
Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{WSA}$ per second as defined in Table 7-6.	
2	Check	$SPDU_{WSA}$ ***signer*** contains ***certificate*** indicating ***version*** value = ***0x03***	
3	Check	$SPDU_{WSA}$ ***signer*** contains ***type*** indicating ***implicit***	
4	Check	$SPDU_{WSA}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating ***HashedId8*** a non-zero value of size 8 octets	
5	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***id*** indicating ***none***	
6	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***cracaId*** indicating a value = ***0x0***	
7	Check	WSA ***toBeSigned*** contains ***crlSeries*** indicating a value=***0x0***	
8	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***start*** indicating ***Time32*** (a non-zero value of size 4 octets)	
9	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***duration*** containing ***minutes*** indicating ***Unit16*** (a non-zero value of size 2 bytes)	
10	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***region*** containing ***circularRegion*** indicating ***latitude*** INTEGER (-900000000..900000000) ***longitude*** INTEGER (-1799999999..1800000000) ***radius*** INTEGER (0 .. 65535)	
11	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***appPermission*** indicating ***psid*** value=***0p80-07***	
12	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating ***compressed-y-0*** or ***compressed-y-1*** (value of size 32 octets)	

13	Check	$SPDU_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (a value of ***compressed-y-0*** size of 32 octets)	
14	Check	$SPDU_{WSA}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	
15	Stimulate	The IUT receives SPDU's$_{WSA}$	
16	Verify	IUT indicates that the $SPDU_{WSA}$ message holds a valid security credentials.	Pass/Fail

Identifier	TP-16092-$SPDU_{WSA}$-RECV-BV-03
Summary	Validate that the IUT will indicate a valid security credentials for a well-formed $SPDU_{WSA}$ signed by certificate ***digest*** of known certificate. The $SPDU_{WSA}$ shall include***, protocolVersion, content, signedData, tbsData, headerInfo, signer, ecdsaP256Signature***.
Test Configuration	TC (1)
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is being initialized

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to receive more than one $SPDU_{WSA}$ per second as defined in Table 7-6.	
2	Check	$SPDU_{WSA}$ Ieee1609Dot2Data contains ***protocolVersion*** indicating value = ***0x03***	
3	Check	$SPDU_{WSA}$ Ieee1609Dot2Data contains ***content*** indicating ***signedData***	
4	Check	$SPDU_{WSA}$ ***signedData*** contains hashId indicating ***sha256***	
5	Check	$SPDU_{WSA}$ ***tbsData*** contains ***protocolVersion*** indicating value = ***0x03***	
6	Check	$SPDU_{WSA}$ ***tbsData*** contains ***content*** indicating ***unsecuredData*** (Payload Data> 0)	
7	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***psid*** indicating value = ***0p80-07***	
8	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***generationTime*** indicating a ***Time64*** (non-zero value of size 8 octets)	
9	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***expiryTime*** indicating a ***Time64*** (non-zero value of size 8 bytes	
10	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***generationLocation*** indicating ***latitude*** (-900000000 . . 900000000) ***longitude*** (-1799999999 . . 1800000000) ***elevation*** Unit16	
11		$SPDU_{WSA}$ contains ***signer*** containing ***digest*** indicating ***HashedId8*** (a non-zero value of size 8 octets)	
12	Check	$SPDU_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** (***compressed-y-0*** or ***compressed-y-1*** consists of octet size 32)	
13	Check	$SPDU_{WSA}$ ***signature*** contains opaque ***s*** indicating non-zero value of size 32 octets	
14	Stimulate	IUT receives SPDU's$_{WSA}$	
15	Verify	IUT indicates that the $SPDU_{WSA}$ message holds a valid security credentials.	Pass/Fail

6.1.9.3 Reception of packets – invalid behaviour tests

Identifier	TP-16092-$SPDU_{WSA}$-RECV-BI-01

Summary	Validate that the IUT will indicate an invalid $SPDU_{WSA}$ signed by implicit certificate, which failed verification due to incorrect signature.
Test Configuration	TC1
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions
• The IUT is being initialized

Test Sequence			
Step	**Type**	**Description**	**Verdict**
1	Configure	The IUT is configured to receive more than one $SPDU_{WSA}$ per second	
2	Check	$SPDU_{WSA}$ ***headerInfo*** contains ***psid*** indicating ***value =0p80-07***	
3		$SPDU_{WSA}$ ***signer*** contains ***certificate*** indicating ***version*** value= ***0x03***	
4	Check	$SPDU_{WSA}$ ***signer*** contains ***type*** indicating ***implicit***	
5	Check	$SPDU_{WSA}$ ***signer*** contains ***issuer*** containing ***sha256AndDigest*** indicating ***HashedId8***	
6	Check	$SPDU_{WSA}$ ***toBeSigned*** contains ***verificationKeyIndicator*** containing ***reconstructionValue*** indicating 'RECVAL' which creates the public key 'KEY' by invoking the 1609.2 reconstruction function on 'RECVAL' and the public key of the certificate stored on IUT and identified by 'DG1'	
7	Check	$SPDU_{WSA}$ ***signature*** contains ***ecdsaP256Signature*** indicating ***r*** and ***s*** not verifiable using (KEY)	
8	Stimulate	The IUT receives the $SPDU's_{WSA}$	
9	Verify	IUT indicates that the $SPDU_{WSA}$ message holds an invalid security credentials.	Pass/Fail

7 Messages and information element content

This section contains basic message structure that will be used in the TP's.

7.1 Secure Protocol Data Uunit for Basic Safety message ($SPDU_{BSM}$)

7.1.1 $SPDU_{BSM}$ defaults

The following assumptions apply to all messages defined in this section.

- All default values are listed in section 4.1
- The ASN.1 presentation in this section depicts the 1609.2 [8] secure message formats structure of WSM message.

7.1.2 $SPDU_{BSM}$ Message Details

- Table 7-1 describes 1609.2[8] security header information of BSM which includes the payload.
- Table 7-2 and Table 7-3 describes 1609.2[8] signer credentials information of BSM.
- Table 7-4 describes 1609.2[8] security signature information of BSM.

7.1.3 $SPDU_{BSM}$ Security Header information

Table 7-1: $SPDU_{BSM}$ Header Information

Information Element	Value/Remark	Comment

Ieee1609Dot2Data SEQUENCE {		
protocolVersion	***3***	
content signedData SEQUENCE {		
hashId	***sha256***	
tbsData SEQUENCE{		
payload SEQUENCE {		
data {		
protocolVersion	***3***	
content	Any valid BSM payload including 1609.3 WAVE message information.	BSM payload created according to 2945/1 and 2735 standards
}		
}		
headerInfo {		
psid	32 (PSID=***0p20***)	PSID value for BSM is 0p20
generationTime	Any valid value	
}		
}		
Require signer credentials information in Table 7-2 or Table 7-3		
Require Security Signature information in Table 7-4		

7.1.4 SPDU$_{BSM}$ Signed with Certificate Digest

Table 7-2: SPDU$_{BSM}$ Signed by Signer type of Certificate Digest

Information Element	Value/Remark	Comment
Requires BSM Security header information in Table 7-1		
signer { }	***digest***	***HashedID8***
Require Security Signature information in Table 7-4		

7.1.5 SPDU$_{BSM}$ Signed with Implicit Certificate

Table 7-3: SPDU$_{BSM}$ Signed by Signer type of Implicit Certificate

Information Element	Value/Remark	Comment
Requires BSM Security header information in Table 7-1		
signer SEQUENCE {	***certificate***	
certificate {		
version	***3***	
type	***implicit***	
issuer	***ecdsaNistP256AndDigest***	HashedID8
toBeSigned SEQUENCE{		
id {	***linkageData***	
iCert	Any valid value	
linkage-value	Any Valid value	
group-linkage-value SEQUENCE{		
jValue	Any valid value	
value	Any valid value	
}		
}		

cracaId	Any valid value	
crlSeries	***1***	
validityPeriod SEQUENCE{		
start	Any valid value	
duration hours	Any valid value	
}		
region identifiedRegion SEQUENCE {		
countryOnly	***124 (0X7C)***	
countryOnly	***484 (0X1E4)***	
countryOnly	***840 (0X348)***	
},		
appPermissions SEQUENCE{		
{		
psid	32 (PSID=***0p20***)	BSM
}		
{		
psid	38 (PSID=***0p26***)	Misbehaviour for common applications
}		
}		
verifyKeyIndicator	***reconstructionValue***	***compressed-y-0***
}		
}		
}		
Require Security Signature information in Table 7-4		

7.1.6 SPDU$_{BSM}$ Security Signature

Table 7-4: SPDU$_{BSM}$ Security Signature

Information Element	Value/Remark	Comment
Requires BSM Security header information in Table 7-1		
Require signer credentials information in Table 7-2 or Table 7-3		
signature SEQUENCE {	***ecdsa256Signature***	EccP256CurvePoint
r	***compressed-y-0*** **or** ***compressed-y-1***	Octet size of 32
s		Octet size of 32
}		

7.1.7 SPDU$_{WSA}$ Message Details

- Table 7-5 describes 1609.2[8] security header information of WSA which includes valid payload.
- Table 7-6 and Table 7-7 describes 1609.2[8] signer credentials information of WSA.
- Table 7-8 describes 1609.2[8] security signature information of WSA.

7.1.8 SPDU$_{WSA}$ Security Header information

Table 7-5 SPDU$_{WSA}$ Header Information

Information Element	Value/Remark	Comment

Ieee1609Dot2Data SEQUENCE {		
protocolVersion	***3***	
content signedData SEQUENCE {		
hashId	***sha256***	
tbsData SEQUENCE {		
payload SEQUENCE {		
data {		
protocolVersion	***3***	
content	Valid WSA payload	
}		
}		
headerInfo SEQUENCE{		
psid	***135*** (PSID=***0p8007***)	
generationTime	Any valid value	
expiryTime	Any valid value	
generationLocation SEQUENCE {		
latitude	Any valid value	
longitude	Any valid value	
elevation	Any valid value	
}		
}		
}		
Require signer credentials information in Table 7-6 or 7-7		
Require Security Signature information in Table 7-8		
}		

7.1.9 SPDU$_{WSA}$ Signed with Implicit Certificate

Table 7-6: SPDU$_{WSA}$ Signed by Signer type of Implicit Certificate

Information Element	Value/Remark	Comment
Requires WSM Security header information in Table 7-5		
signer SEQUENCE {		
certificate {		
Version	***3***	
type	***implicit***	
issuer	***ecdsaNistP256AndDigest***	***HashedID8***
toBeSigned SEQUENCE {		
id	***none***	
cracaId	***Value = 0***	
crlSeries	***Value=0***	
validityPeriod SEQUENCE {		
start	Any valid value	
duration minutes	Any valid value	
}		
region circularRegion SEQUENCE {		
centre {		
latitude	Any valid value	
longitude	Any valid value	
}		

radius	Any valid value	
}		
appPermissions {		
{		
psid	135 (PSID=***0p8007***)	
}		
}		
verifyKeyIndicator	***reconstructionValue***	***compressed-y-0***
}		
}		
}		
Require Security Signature information in Table 7-8		

7.1.10 SPDU$_{WSA}$ Signed with Certificate Digest

Table 7-7: SPDU$_{WSA}$ Signed with Certificate digest

Information Element	Value/Remark	Comment
Requires WSA Security header information in Table 7-5		
signer { }	***digest***	***HashedID8***
Require Security Signature information in Table 7 8		

7.1.11 SPDU$_{WSA}$ Security Signature

Table 7-8: SPDU$_{WSA}$ Security Signature

Information Element	Value/Remark	Comment
Requires WSM Security header information in Table 7-5		
Require signer credentials information in Table 7-6 or Table 7-7		
signature SEQUENCE {	***ecdsa256Signature***	EccP256CurvePoint
r	***compressed-y-0 or compressed-y-1***	Octet size of 32
s	Any valid value	Octet size of 32
}		

Appendix A:

Traceability Matrix

This section of the document contains the traceability matrix for BSM and WSA security requirements. As shown below, Table A- 1 lists BSM IEEE 1609.2[8] traceability to TPs. In Page (# 39) Table A- 2 lists WSA IEEE 1609.2 traceability to TPs where PICS for WSA was derived from "IEEE 1609.2[8] security specification for WSA requirements" listed under Annex H in 1609.3[5].

The current test specification doesn't include any TP's that requires Security Credential Management System (SCMS) due to the fact that the new standard is not available and will be available in 2016. Accordingly, not all the mandatory requirements by 2945/1 is tested at this time.

Table A- 1: BSM IEEE 1609.2 PICS traceability to TPs

1609.2 PICS from [8]	Features in [8]	Reference section in [8]	Status (J2945-1 [1])	Support (J2945-1 [1])	TP ID	TP Description
S1.2.2	**Create Ieee1609Dot2 Data containing valid SignedData**	4.2.2.2.3, 5.2, 5.3.1, 5.3.3,5.3.7,6.3.4,6.3.9,9.3.9.1	S1.2:O3	Y	TP-16092-BSM-SEND-BV-01	To verify that the IUT will generate a valid signedData as per 1609.2[8] specifications
S1.2.2.1	Using a valid HashAlgorithm	6.3.5	S1.2.2:M	Y	TP-16092-BSM-SEND-BV-01	To verify that the IUT will generate a valid signedData using sha256 hash
					TP-16092-BSM-SEND-BV-03	To verify that the IUT will generate signed using certificate digest generated by hash 256
S1.2.2.1.1	Support signing with hash algorithm SHA-256	6.3.5	S1.2.2:M	Y	Refer to S1.2.2.1	
S1.2.2.2	Containing a Signed Data payload	6.3.6	S1.2.2:M	Y	TP-16092-BSM-SEND-BV-01	To verify that the IUT will generate a signedData with BSM payload is included
S1.2.2.2.1	with payload containing data	6.3.7	S1.2.2.2:O4	Y	Refer to S1.2.2.2	

S1.2.2.2.3.	… with generationTime in the security header	6.3.9, 6.3.11	S1.2.2.2: O	Y	TP-16092-BSM-SEND-BV-01	To verify that the IUT will generate BSM security header that includes generationTime
					TP-16092-BSM-SEND-BV-03	To verify that the IUT will generate BSM security header that includes generationTime signed by certificate digest
S1.2.2.3.	Support a SignerIdentifier	6.3.24	S1.2.2: M	Y	TP-16092-BSM-SEND-BV-02	To verify that the IUT will generate aBSM signed with signer type of certificate
					TP-16092-BSM-SEND-BV-03	To verify that the IUT will generate BSM signed with signer type of certificate digest
S1.2.2.3.1.	… of type digest	6.3.26	S1.2.2.3:O6	Y	Refer to S1.2.2.3	
S1.2.2.3.2.	… of type certificate	6.4.2	S1.2.2.3:O6	Y	Refer to S1.2.2.3	
S1.2.2.3.2.1	Maximum number of Certificates in the chain	5.1.2.2	S1.2.2.3.2 8:M > 8:O	1	TP-16092-BSM-SEND-BV-02	To verify that the IUT will generate BSM signed with signer type of certificate With a Maximum number of certificates in the chain is equal to 1.
S1.2.2.4.	Support a Signature	6.3.28	S1.2.2: M	Y	TP-16092-BSM-SEND-BV-05	To verify that the IUT will generate a valid signature to sign BSM message generated by signer of type certificate digest
					TP-16092-BSM-SEND-BV-06	To verify that the IUT will generate a valid signature to sign BSM messages generated by signer of type certificate
S1.2.2.4.1.	… a ecdsa256Signature	6.3.29	S1.2.2.4:M	Y	TP-16092-BSM-SEND-BV-02	To verify that the IUT will generate a valid signature using ecdsa256Signature type. Where it uses the NIST p256 algorithm to generate the compressed r value. The signer type used to sign the BSM message is certificate

					TP-16092-BSM-SEND-BV-03	To verify that the IUT will generate a valid signature using ecdsa256Signature type. Where it uses the NIST p256 algorithm to generate the compressed r value. The signer type used to sign the BSM message is digest
S1.2.2.4.1.1.	… … using NIST p256	6.3.29	S1.2.2.4.1:O7	Y	Refer to S1.2.2.4.1	
S1.2.2.4.1.4.	… … with a compressed r value	6.3.23	S1.2.2.4.1:O8	Y	Refer to S1.2.2.4.1	
S1.2.2.5.1.	Determine that the region is correct	6.4.8, 6.4.17	S1.2.2.5:O	Y	TP-16092-BSM-SEND-BV-02	To verify that the certificate region is defined as " identifiedRegion" with a minimum number of 3 countries as specified in SAE J2945/1
S1.2.2.5.1.4	Support identifiedRegion	6.4.17, 6.4.22	S1.2.2.5.1:O9	Y	Refer to S1.2.2.5.1	
S1.2.2.5.1.4.1.	Maximum number of identifiedRegions supported	6.4.17 6.4.22	S1.2.2.5.1.4: 8:M >8:O	Minimum of 3 Note: US, Canada, Mexico supported as defined by the United Nations Statistics Division, October 2013 edition	Refer to S1.2.2.5.1	
S1.2.2.5.1.4.2.	Support IdentifiedRegion of type Country Only	6.4.22, 6.4.23	S1.2.2.5.1.4:O1	Y	Refer to S1.2.2.5.1	

S1.2.2.5.2	Determine that the certificate has the proper appPermissions	6.4.8 6.4.28	S1.2.2.5:O	Y	TP-16092-BSM-SEND-BV-02	verify that the IUT will generate a signedData using implicit certificate that contains the appropriate appPermissions
S1.2.2.8.	Support signing with implicit certificate	5.3.2, 6.4.5	S1.2.2.5:O11	Y	Refer to S1.2.2.5.2	
S1.3.2.	Verify Ieee-1609Dot2Data containing SignedData	4.2.2.2.3, 5.2, 5.3.1, 5.3.3 5.3.7, 6.3.4,6.3.9	S1.3:O17	**Y**	TP-16092-BSM-RECV-BV-01	To verify that the IUT will accept a valid BSM contains signedData.
S1.3.2.1.	Using a valid HashAlgorithm		S1.3.2:M	**Y**	TP-16092-BSM-RECV-BV-01	To verify that the IUT will accept BSM message signed by a digest of type sha256
					TP-16092-BSM-RECV-BV-03	To verify that the IUT will accept BSM messages signed by a signer credential of type certificate digest using sha256
S1.3.2.1.1.	Verify signed data using Hash Algorithm SHA-256	6.3.5	S1.3.2.1:M	Y	Refer to S1.3.2.1	
S1.3.2.2.	Containing a Signed Data payload	6.3.6	S1.3.2:M	**Y**	Refer to S1.3.2	
S1.3.2.2.1.	… with payload containing data	6.3.7	S1.3.2.2:O18	**Y**	Refer to S1.3.2	
S1.3.2.2.3.	… with generation Time in the security header	6.3.9, 6.3.11	S1.3.2.2:O	**Y**	TP-16092-BSM-RECV-BV-01	To verify that the IUT will accept BSM message with the correct security header information. That is, it must contain generationTime.
S1.3.2.3.	Support a SignerIdentifier	6.3.24	S1.3.2:M	**Y**	TP-16092-BSM-RECV-BV-02	To verify that the IUT will accept BSM message signed with the correct signer credential of type certificate. .

					TP-16092-BSM-RECV-BV-03	To verify that the IUT will accept BSM message signed with the correct signer credential of type certificate digest.
S1.3.2.3.1.	… of type digest	6.3.26	S1.3.2.3:O20	**Y**	Refer to S1.3.2.3	
S1.3.2.3.2	… of type certificate	6.4.2	S1.3.2.3:O20	**Y**	Refer to S1.3.2.3	
S1.3.2.3.2.1.	… … Maximum number of Certificates in the chain	5.1.2.2	S1.3.2.3.2 1:M >1:O	1	TP-16092-BSM-RECV-BV-02	To verify that the IUT will accept a BSM message with a maximum certificate chain is equal to 1.
S1.3.2.4.	Support a Signature	6.3.28	S1.3.2:M	**Y**	TP-16092-BSM-RECV-BV-02	To verify that the IUT will accept BSM message signed by ecdsa256Signature type. Where it uses the NIST p256 algorithm to generate the compressed r value. The signer credential type used to sign the BSM message is certificate
					TP-16092-BSM-RECV-BV-03	To verify that the IUT will accept BSM message signed by ecdsa256Signature type. Where it uses the NIST p256 algorithm to generate the compressed r value. The signer credential type used to sign the BSM message is certificate digest.
S1.3.2.4.1.	… a ecdsa256Signature	6.3.29	S1.3.2.4:M	**Y**	Refer to S1.3.2.4	
S1.3.2.4.1.1.	… … using NIST p256	6.3.29	S1.3.2.4.1:O21	**Y**	Refer to S1.3.2.4	
S1.3.2.4.1.4.	… … with a compressed r value	6.3.23	S1.3.2.4.1:O22	**Y**	Refer to S1.3.2.4	
S1.3.2.10.14.	… SPDU-Crypto: Verification failure	5.3.1	S1.3.2.10:M	**Y**	TP-16092-BSM-RECV-BI-01	To verify that the IUT will reject a BSM message signed with invalid ecdsa256Signature. The signer credential of type

						certificate digest is used to sign the BSM message.
					TP-16092-BSM-RECV-BI-02	To verify that the IUT will reject a BSM message signed with invalid ecdsa256Signature. The signer credential of type certificate is used to sign the BSM message.

Table A- 2: WSA IEEE 1609.2 PICS traceability to TPs

1609.2 PICS from [8]	Features in [8]	Reference section in [8]	Status [8]	Support 1609.3[5]	TP ID	TP Description
S1.2.2	Create Ieee1609Dot2 Data containing valid SignedData	4.2.2.2.3, 4.2.2.2.3, 5.2, 5.3.1 5.3.3, 5.3.7, 6.3.4, 6.3.9, 9.3.9.1	S1.2:O 3	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid WSA signedData as per 1609.2[8]specifications
S1.2.2.1.	Using a valid HashAlgorithm	6.3.5	S1.2.2: M	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid WSA signedData using sha256 hash
S1.2.2.1.1.	Support signing with hash algorithm sha-256	6.3.5	S1.2.2: M	Y	Refer to S1.2.2.1	
S1.2.2.2.	Containing a Signed Data payload	6.3.6	S1.2.2: M	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid signedData with WSA payload is included
S1.2.2.2.1.	… with payload containing data	6.3.7	S1.2.2. 2:O4	Y	Refer to S1.2.2.2	
S1.2.2.2.3.	… with generationTime in the security headers	6.3.9, 6.3.11	S1.2.2. 2: O	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid WSA headerinfo data structure that include Generation time
S1.2.2.2.4.	… with expiryTime in the security headers	6.3.9, 6.3.11	S1.2.2. 2: O	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid WSA headerinfo data structure that includes Expiry Time.

S1.2.2.2.5.	… with generationLocation in the security headers	6.3.9, 6.3.12	S1.2.2.2: O	Y	TP-16092-WSA-SEND-BV-01	To verify that the IUT will generate a valid WSA headerinfo data structure that include Generation location.
S1.2.2.3.	Support a SignerIdentifier	6.3.24	S1.2.2: M	Y	TP-16092-WSA-SEND-BV-02	To verify that the IUT will generate WSA signed with signer type of implicit certificate
					TP-16092-WSA-SEND-BV-03	To verify that the IUT will generate WSA signed with signer type of certificate digest
S1.2.2.3.1.	… of type digest	6.3.26	S1.2.2.3:O6	Y	Refer to S1.2.2.3	
S1.2.2.3.2.	… of type certificate	6.4.2	S1.2.2.3:O6	Y	Refer to S1.2.2.3	
S1.2.2.3.2.1.	… … Maximum number of Certificates in the chain	5.1.2.2	S1.2.2.3.2 8:M >8O	1	TP-16092-WSA-SEND-BV-02	To verify that the IUT Will generate WSA signed with certificate chain =1
S1.2.2.4.	Support a Signature	6.3.28	S1.2.2: M	Y	TP-16092-WSA-SEND-BV-04	To verify that the IUT Will generate WSA signed with a valid signature. The signature will be generated using NISTp256 and using Compressed r value
S1.2.2.4.1.	… a ecdsa256Signature	6.3.29	S1.2.2.4:M	Y	Refer to S1.2.2.4.	
S1.2.2.4.1.1.	… … using NIST p256	6.3.29	S1.2.2.4.1:O7	Y	Refer to S1.2.2.4.	
S1.2.2.4.1.4.	… … with a compressed r value	6.3.23	S1.2.2.4.1:O8	Y	Refer to S1.2.2.4.	
S1.2.2.5.1.	Determine that the region is correct	6.4.8, 6.4.17	S1.2.2.5:O	Y	TP-16092-WSA-SEND-BV-02	To verify that the IUT will generated a signer of type implicit certificate that contains a valid region.
S1.2.2.8.	Support signing with implicit certificates	5.3.2, 6.4.5	S1.2.2.5:O11	Y	Refer to S1.2.2.3	
S1.3.2.	Verify Ieee1609Dot2 Data containing SignedData	4.2.2.2.3, 5.2, 5.3.1 5.3.3, 5.3.7, 6.3.4 6.3.9	S1.3:O17	Y	TP-16092-WSA-RECV-BV-01	To verify that the IUT will accept a valid WSA contains signedData.

S1.3.2.1.	Using a valid HashAlgorithm		S1.3.2: M	Y	TP-16092-WSA-RECV-BV-01	To verify that the IUT will accept WSA message signed by a digest of type sha256
					TP-16092-WSA-RECV-BV-03	To verify that the IUT will accept BSM messages signed by a signer credential of type certificate digest using sha256
S1.3.2.1.1.	Verify signed data using HashAlgorith SHA-256	6.3.5	S1.3.2. 1:M	Y	Refer to S1.3.2.1	
S1.3.2.2.	Containing a Signed Data payload	6.3.6	S1.3.2: M	Y	TP-16092-WSA-RECV-BV-01	To verify that the IUT will accept a WSA signed message containing Payload
S1.3.2.2.1.	… with payload containing data	6.3.7	S1.3.2. 2:O18	Y	Refer to S1.3.2.2	
S1.3.2.2.3.	… with generationTime in the security headers	6.3.9, 6.3.11	S1.3.2. 2:O	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA headerinfo data structure that include Generation time
S1.3.2.2.4.	… with expiryTime in the security headers	6.3.9, 6.3.11	S1.3.2. 2:O	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA headerinfo data structure that include Expiry time.
S1.3.2.2.5.	… with generationLocation in the security headers	6.3.9, 6.3.12	S1.3.2. 2:O	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA headerinfo data structure that include Generation location
S1.3.2.3.	Support a SignerIdentifier	6.3.24	S1.3.2: M	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA message signed with signer type of implicit certificate
					TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA message signed with signer type of certificate digest.
S1.3.2.3.1.	… of type digest	6.3.26	S1.3.2. 3:O20	Y	Refer to S1.3.2.3	
S1.3.2.3.2.	… of type certificate	6.4.2	S1.3.2. 3:O20	Y	Refer to S1.3.2.3.	
S1.3.2.3.2.1.	… … Maximum number of Certificates in the chain	5.1.2.2	S1.3.2. 3.2 1:M >1:O	1	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a valid WSA message signed with certificate chain = 1

S1.3.2.4.	Support a Signature	6.3.28	S1.3.2: M	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT Will accept WSA signed with a valid signature. The signature will be generated using NISTp256 and using Compressed r value
S1.3.2.4.1.	… a ecdsa256Signature	6.3.29	S1.3.2.4:M	Y	Refer to S1.3.2.4	
S1.3.2.4.1.1.	… … using NIST p256	6.3.29	S1.3.2.4.1:O21	Y	Refer to S1.3.2.4	
S1.3.2.4.1.4.	… … with a compressed r value	6.3.23	S1.3.2.4.1:O22	Y	Refer to S1.3.2.4	
S1.3.2.5.1.1.	… using a circularRegion	6.4.17, 6.4.18	S1.3.2.5.1:O23	Y	TP-16092-WSA-RECV-BV-02	To verify that the IUT will accept a WSA message signed by a signer of type implicit certificate with a region of type circular.
S1.3.2.7.	Support verifying SPDUs signed with implicit authorization certificates	5.3.2, 6.4.5	S1.3.2: O25	Y	Refer to S1.3.2.3.	
S1.3.2.10.14.	… SPDU-Crypto: Verification failure	5.3.1	S1.3.2.10:M	Y	TP-16092-WSA-RECV-BI-01	To verify that the IUT will reject a WSA message signed with invalid ecdsa256Signature. The signer credential of type certificate is used to sign the BSM message.

8 Revision History

V0.1.0	Sep 17, 2015	Initial Draft – BSM test cases
V0.2.0	Sep 30, 2015	Added test cases for WSA messages
V0.3.0	Oct 5, 2015	Updated BSM and WSA messages
V.0.4.0	Oct 23, 2015	Updated Test Cases to the new format
V.0.5.0	Dec 31, 2015	Updated TP to the new Standard Added Tractability Matrix for BSM and WSA
V.0.6.0	Feb 5, 2016	Based on peer review, multiple changes were made to the document.
V.1.0	March 23, 2016	Incorporated comments from industry reviewers
V1.1	Oct 10, 2016	Incorporated comments from CAMP reviewers.

◙ End of Document ◙