SCMS MANAGER™
SECURITY CREDENTIAL MANAGEMENT SYSTEM

MISBEHAVIOR REPORT AND APPLICATION SPECIFICATION FOR CONNECTED VEHICLE PILOT DEPLOYMENT

VERSION 1.0

February 14, 2019

Author(s)

SCMS Manager Board of Directors

TABLE OF CONTENTS

SCMS MANAGER MISBEHAVOR REPORT AND APPLICATION SPECIFICATION FOR CONNECTED VEHICLE PILOT DEPLOYMENT

1.0 OVERVIEW AND DESIGN APPROACH

The design of the Misbehavior Report and Misbehavior Reporting Application is intended for use with PSID 0x26. This PSID supports misbehavior reports for common applications. As such, the misbehavior report design is highly modular.

The top level of the misbehavior report contains:

- A version number, which is 1 for this version of the misbehavior report
- A generation time, which is a standard 1609.2 Time32 type
- A pair (psid, contents), where
    - The psid field is the PSID for the application for which misbehavior is being reported (Note that this report format may be extended in the future to cover reports of system level misbehavior that is not application-specific, such as detection of channel-jamming; if this is done, a PSID will need to be reserved for this purpose). This is referred to as the ***misbehavior application PSID***, to distinguish it from the misbehavior reporting PSID 0x26.
    - The contents field is a PSID-specific structure.

  The ASN.1 specification uses the ASN.1 CLASS structure to ensure that a receiver associates the correct contents type with a given PSID, if that contents type is known.

For the current version of the Connected Vehicle Pilot Deployment Misbehavior Detection (CV-PD-MBD) project, only one misbehavior application PSID and associated report contents have been defined:

- PSID = 0x20, i.e. BSM sending for purposes of collision avoidance between light vehicles
- Contents:
    - Version number, which is 1 for this version of the PSID-20 misbehavior report.
    - A tuple (type, subtype, evidence) where
        - The “type” is the high-level type of the report and identifies the main evidence to be provided (for example: a single BSM from one vehicle; or multiple BSMs from one vehicle; or multiple BSMs from different vehicles).
        - The “subtype” is an identifier for a specific misbehavior and may be associated with subtype-specific parameters
        - The evidence is a type-specific structure.

      The type, subtype and evidence are bound together by an ASN.1 CLASS structure as with the (psid, contents) pair in the top-level report

For further documentation of the data structures, see the annotated ASN.1 file.

Note that a single instance of “evidence” may be evidence of multiple misbehavior subtypes. For example, an “implausible single BSM” may be implausible in multiple ways, such as having both an implausible speed and an implausible acceleration. The report format allows a report to have multiple

subtypes to support this case and to avoid the storage and transmission burden that would arise from requiring to have multiple reports referencing the same evidence.

2.0 MISBEHAVIOR TYPES AND SUBTYPES

2.1 CONFIGURATION

This section lists all the supported types and subtypes of misbehavior reporting.

The design is intended to support modifications to the configuration of the reporting software.

- For each misbehavior application PSID, the misbehavior reporting application should take the following configuration parameters:
  - Turn reporting on or off for that PSID
- For each misbehavior reporting type within a given misbehavior application PSID, the misbehavior reporting application should take the following configuration parameters:
  - Turn reporting on or off for that (PSID, type)
- For each misbehavior reporting subtype associated with a given misbehavior application PSID and type, the misbehavior reporting application should take the following configuration parameters:
  - Turn reporting on or off for that (PSID, type, subtype)
  - Adjust (subtype-specific) reporting parameters for that (PSID, type, subtype)

2.2 STORAGE AND UPLOAD BEHAVIOR

The design is also intended to support prioritization of messages for storing and for uploading.

Each misbehavior report, when generated, is assigned a (subtype-specific) weight.

Every day, at 5 am Eastern time, the weight is adjusted by multiplying it by a (subtype-specific) multiplier.

When the device gets connectivity, it uploads stored misbehavior reports in order of weight, with the reports with the highest weight uploaded first.

- An implementation need not (but may choose to) distinguish between weights that are the same to three significant (decimal) figures
- If two reports have the same weight, the device uploads the newer report first.

Reports are uploaded via the API defined [later in this document]. This API runs over TCP/IP. Once the device receives the TCP confirmation that the packet upload completed successfully for a particular report, it does not attempt to upload that report again, may delete it, and does not count it towards total storage space for reports as discussed in the next paragraph.

Each device is configured with a maximum space for storing misbehavior reports. This space is device-specific. If a report is generated such that the total amount of stored misbehavior reports would exceed this maximum space then the device deletes the stored report with the lowest weight, then the next lowest, until the total space used for reports is less than the maximum space. Note that depending on initial and current weights and the properties of the new report, it is possible that the new report is one of the ones deleted.

- An implementation need not (but may choose to) distinguish between weights that are the same to three significant (decimal) figures
- If two reports have the same weight, the device deletes the older report first.

The weights for each subtype are given below in this section.

Future versions of this design may include enhancements to this algorithm – for example: different update intervals; different initial weights depending on whether there is already a misbehavior report for the same misbehavior subtype and the same received certificate; different initial weights depending on whether there is already a misbehavior report for the same misbehavior submit and a different received certificate. These enhancements are not defined in this version of this document but if desired and funded could be developed and rolled out through the deployment sites' software update processes.

2.3 PARAMETERS

2.3.1 PSID-20 MISBEHAVIOR REPORT

2.3.1.1 GENERAL

This is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20	On, off	On	Determines whether any misbehavior reports for PSID 0x20 are generated

2.3.1.2 PSID-20 MISBEHAVIOR REPORT TYPE: SINGLE VEHICLE INCONSISTENT

2.3.1.2.1 GENERAL

This report type indicates that a collection of BSMs from the same sender (i.e. signed by the same certificate) are inconsistent with each other. The evidence for this type is a sequence of BSMs signed with

the same certificate. Different subtypes indicate the exact way in which the BSMs are inconsistent. This type is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-1VInconsistent	On, off	On	Determines whether any misbehavior reports of this type are generated

2.3.1.2.2 SINGLE VEHICLE INCONSISTENT REPORT: RANDOM POSITION

This report type indicates that the position in the received BSMs appears to be random, i.e. changes in position are not consistent with the velocity. This subtype is configured by the following parameters:

Name	Possible values	Default	Notes
mbr-Psid20-1VInconsistent-RandomPosition	On, off	On	Determines whether any misbehavior reports of this subtype are generated
mbr-Psid20-1VInconsistent-RandomPosition-gpsDrift	A positive distance	1m	Two consecutive BSMs are inconsistent if [(position in the first) + (velocity in the first)*(time between)] differs from (position in the second) by more than this parameter
mbr-Psid20-1VInconsistent-RandomPosition-initialWeight	A positive whole number	1.0	Initial weight for storage per 0
mbr-Psid20-1VInconsistent-RandomPosition-DailyMultiplier	A whole number between 0 and 1	0.9	Daily weight multiplier for storage per 0

2.3.1.2.3 SINGLE VEHICLE INCONSISTENT REPORT: CONSTANT POSITION

This report type indicates that the position in the received BSMs appears to be constant, i.e. velocity is non-zero but the position has not changed. This subtype is configured by the following parameter(s):

Name	Possible values	Default	Notes
mbr-Psid20-1VInconsistent-ConstantPosition	On, off	On	Determines whether any misbehavior reports of this subtype

2.3.1.3 PSID-20 MISBEHAVIOR REPORT TYPE: IMPLAUSIBLE SINGLE BSM

2.3.1.3.1 GENERAL

This report type indicates that a single BSM is physically implausible. The evidence for this type is a single BSM. Different subtypes indicate the exact way in which the BSM is implausible. This type is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-Implausible	On, off	On	Determines whether any misbehavior reports of this type are generated

2.3.1.3.2 IMPLAUSIBLE SINGLE BSM: MAXIMUM SPEED

This report type indicates that the speed in a single BSM is physically implausible. This subtype is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-Implausible-MaxSpeed	On, off	On	Determines whether misbehavior reports of this subtype are generated
mbr-Psid20-Implausible-MaxSpeed-Threshold	A positive speed	90 m/s (approx. 200 mph)	A report is generated if the speed is greater than this threshold value
mbr-Psid20-Implausible-MaxSpeed-initialWeight	A positive whole number	1.0	Initial weight for storage per 0
mbr-Psid20-Implausible-MaxSpeed-DailyMultiplier	A whole number between 0 and 1	0.9	Daily weight multiplier for storage per 0

2.3.1.3.2 IMPLAUSIBLE SINGLE BSM: MAXIMUM ACCELERATION

This report type indicates that the acceleration in a single BSM is physically implausible. This subtype is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-Implausible-MaxAcceleration	On, off	On	Determines whether misbehavior reports of this subtype are generated

mbr-Psid20-Implausible-MaxAcceleration-Threshold	A positive acceleration	10 m/s^2	A report is generated if the acceleration is greater than this threshold value
mbr-Psid20-Implausible-MaxAcceleration-initialWeight	A positive whole number	1.0	Initial weight for storage per 0
mbr-Psid20-Implausible-MaxAcceleration-DailyMultiplier	A whole number between 0 and 1	0.9	Daily weight multiplier for storage per 0

2.3.1.3.4 IMPLAUSIBLE SINGLE BSM: ACCELERATION WITH BRAKING

This report type indicates that a single BSM indicates that the vehicle is braking but that the acceleration is not negative, i.e. that the acceleration vector makes an acute angle with the velocity vector. This subtype is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-Implausible-BrakeAcceleration	On, off	On	Determines whether misbehavior reports of this subtype are generated
mbr-Psid20-Implausible-BrakeAcceleration-initialWeight	A positive whole number	1.0	Initial weight for storage per 0
mbr-Psid20-Implausible-BrakeAcceleration-DailyMultiplier	A whole number between 0 and 1	0.9	Daily weight multiplier for storage per 0

2.3.1.4 PSID-20 MISBEHAVIOR REPORT TYPE: IMPLAUSIBLE OBSERVED BSM

2.3.1.4.1 GENERAL

This report type indicates that a received BSM is implausible based on other information available to the receiver. For example, the BSM generation location might be so far from the receiver that it's implausible that the message could have been successfully received. The evidence for this type is a BSM from the sender and a BSM from the receiver, created at approximately the same time. Different subtypes indicate the exact way in which the received BSM was judged to be implausible. This type is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-ObsImplausible	On, off	On	Determines whether any misbehavior reports of this type

2.3.1.4.2 IMPLAUSIBLE OBSERVED BSM: DISTANCE

This report type indicates that the position in a single received BSM is greater than plausible, given likely successful transmission distances to the observer. This subtype is configured by the following parameter:

Name	Possible values	Default	Notes
mbr-Psid20-ObsImplausible-Distance	On, off	Off	Determines whether misbehavior reports of this subtype are generated
mbr-Psid20-ObsImplausible-Distance-Threshold	A positive distance	3000 m	A report is generated if the distance is greater than this threshold value plus gpsDrift
mbr-Psid20-ObsImplausible-Distance-gpsDrift	A positive distance	10 m	A report is generated if the distance is greater than this value plus the threshold
mbr-Psid20-ObsImplausible-Distance-initialWeight	A positive whole number	1.0	Initial weight for storage per 0
mbr-Psid20-ObsImplausible-Distance-DailyMultiplier	A whole number between 0 and 1	0.9	Daily weight multiplier for storage per 0

NOTE: The default for this test is off, because this test is subject to malicious interference: if a man in the middle intercepts a BSM from some distance away and forwards it to a local transmitter, then the BSM could be interpreted as malicious even though the original BSM generator is innocent of any misbehavior. This test should only be configured to On if the MA develops tests to avoid revoking innocent BSM senders who may have been reported as a result of this attack.

3.0 SECURITY PROCESSING

Once the report is created per the ASN.1, it is COER-encoded to create an EndEntityMaInterfacePDU.

This COER-encoded payload is then the input to 1609.2 signing to create an Ieee1609Dot2Data of type signed. In this Ieee1609Dot2Data:

- The payload is of type data and contains an Ieee1609Dot2Data of type unsecured containing the COER-encoded EndEntityMaInterfacePDU
- In the headerInfo, the following fields are included:
    - generationTime

  All other fields are omitted.
- The SignerIdentifier is of type Certificate and contains only the end-entity certificate.

This COER-encoded Ieee1609Dot2Data is then encrypted with the encryptionKey from the MA certificate. The encryption is carried out per the Sec-EncryptedData.request primitive in 1609.2, with parameters:

- Data – the output from signing
- Data Type – *Ieee1609Dot2Data*
- Data Encryption Key Type – *ephemeral*
- Recipient Certificates – A single certificate, the MA certificate.
- EC Point Format – *compressed*
- All other parameters omitted

How the end-entity obtains the MA certificate is out of scope for this version of this document.

4.0 UPLOAD

The end-entity submits the Misbehavior Report to the RA using the following REST API over HTTPS over TCP/IP. The end-entity identifies the RA using the URL for the RA that it (the end-entity) has been provisioned with. How this provisioning happens is out of scope for this document.

Each individual report is submitted via a distinct POST command and authenticated by the EE as described below in this section.

PORT	8892
PATH	/process-misbehavior-report
HTTP Method	POST
HTTP Request Body	ASN.1 serialized *SecuredMisbehaviorReport*

PORT	8892
HTTP Request Headers	HTTP Header 'Download-Req' containing a Base64 encoded ASN.1 serialized *SecuredAuthenticatedDownloadRequest,* containing a *SignedAuthenticatedDownloadRequest,* containing a *ScopedAuthenticatedDownloadRequest,* containing an *AuthenticatedDownloadRequest* with a *filename* field containing the string “misbehavior-report” and signed by the device’s enrollment certificate.
HTTP Response Body	Empty

This is authorized similarly to other interactions with the RA that feature end-entity authentication:

- The RA acts as the server in the TLS connection. It authenticates itself to the end-entity using an X.509 certificate containing the RA URL in the subjectAltName and chaining back to a root CA known to the end entity. How the end entity establishes this root CA certificate is out of scope for this document.
- Each POST command is authenticated by a SecuredAuthenticatedDownloadRequest in the Download-Req HTTP header, as specified in the table above.

5.0 VERSION HISTORY

Version	Date	Comments
1.0	14 February 2019	Initial release