# Misbehavior Reporting Protocol for C-ITS

Joseph Kamel, Ines Ben Jemaa, Arnaud Kaiser, Pascal Urien

# Misbehavior Reporting Protocol for C-ITS

Joseph Kamel
*IRT SystemX*
Palaiseau, France
joseph.kamel@irt-systemx.fr

Ines Ben Jemaa
*IRT SystemX*
Palaiseau, France
ines.ben-jemaa@irt-systemx.fr

Arnaud Kaiser
*IRT SystemX*
Palaiseau, France
arnaud.kaiser@irt-systemx.fr

Pascal Urien
*Telecom ParisTech*
Paris, France
pascal.urien@telecom-paristech.fr

***Abstract*—Misbehavior detection is a set of mechanisms that rely on monitoring C-ITS communications to detect potentially misbehaving entities. In this paper we focus on the reporting process of Misbehavior Detection. More precisely, we propose a misbehavior report message format that enables an entity to report a detected misbehaving entity. We explain first the functional requirements of a misbehavior reporting mechanism. Then, we detail the data information that are integrated in the reports in order to provide reliable evidences to the misbehavior authority.**

***Index Terms*—Misbehavior Detection, Misbehavior Report, C-ITS, Cyber-security**

## I. Introduction

C-ITS is a promising technology that aims at improving road safety, efficiency and driving experience. Cyber-security is of paramount importance in such systems as human life is involved. The C-ITS community agreed on the use of the Public Key Infrastructure (PKI) to secure the exchanged messages in the vehicular network. Basically speaking, entities of the system (i.e. vehicles and roadside units - RSU) request digital certificates (so-called pseudonym certificates) from the PKI. They use these certificates to digitally sign the V2X sent messages. However, digital certificates do not protect the system against all security threats. Therefore, there is a need for further solutions to improve security.

Misbehavior Detection (MD) is a technology that aims at monitoring the system to detect potential misbehaving entities and prevent the system to deviate from its normal behavior. Basically speaking, the MD system operates in three steps:

1) **Misbehavior detection**: vehicles and RSU detect locally a potential misbehaving entity.
2) **Misbehavior reporting**: after detection, the vehicle/RSU sends a Misbehavior Report (MR) to the central authority (so-called Misbehavior Authority - MA) located in the cloud.
3) **Misbehavior investigation**: the MA investigates the received MRs in order to define whether the reported entity is actually misbehaving or just faulty.

Several works on misbehavior detection exist in the literature. However to the best of our knowledge most of them focus only on the first step. In this paper, we focus mainly on the second step. We believe that the reporting process is as important as the local detection process because it allows the MA to collect massive information about potential misbehaviors in the local vehicular network. Consequently, this leads the MA to build a centralized view of the misbehavior situations and to generate reliable misbehavior detection results. The choice of the data integrated to the misbehavior report is a key point that may impact the centralized misbehavior detection process in the MA. Actually, only few works define the needed data of the MR [1] [2]. These works agreed on the fact that evidences should be included in MRs as a proof of what is reported. However none of them discuss and specify what actually should be these proofs.

In this paper, we propose a MR message format and detail relevant information that should be included in it. Also, for each detected misbehavior type we propose the corresponding proofs to be included in the MR as well as a related confidence level. The latter is an indication that enables to differentiate non-forged proofs and self-forged proofs (i.e. if a proof could be forged by the reporting entity).

This paper is organized as follow: Section II presents the MR scenario we consider in this study. Section III details our proposed MR approach. Finally Section IV concludes this paper and presents future works.

## II. Misbehavior Reporting Scenario and Requirements

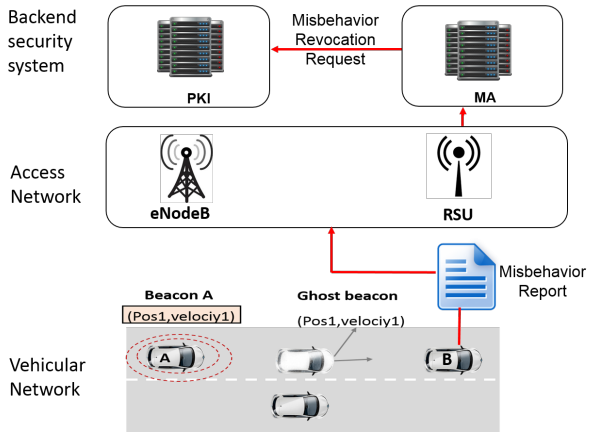


Fig. 1: An example of a Misbehavior Detection (MD) scenario

As shown in figure 1, a typical misbehavior reporting scenario occurs when a vehicle B (i,e., the reporter) detects a suspicious vehicle A (i,e., the reported) which sends fake beacons on the vehicular network. Vehicle B reports this misbehavior to the MA located in the back-end security system.

TABLE I: Misbehavior Detectors For Cooperative Awareness Messages (CAMs)

| CAM Data | Detection Level | | | |
|---|---|---|---|---|
| | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **Reference Position** | · Data Unavailable<br>· Confidence Too Large | · Position Change (PC) Too Large<br>· PC $Inc\Phi^*$ with Speed<br>· PC $Inc\Phi$ with Heading | · Position not on a Road<br>· Position overlap with other Vehicles | · Position $Inc\Phi$ with Relative Position (Lidar Radar)<br>· Position $Inc\Phi$ with Maximum Plausible Range |
| **Heading** | · Data Unavailable<br>· Confidence Too Large | · Heading Change (HC) Too Large<br>· HC $Inc\Phi$ with Speed<br>· HC $Inc\Phi$ with YawRate | · Heading $Inc\Phi$ with Road Heading | · Heading $Inc\Phi$ with Relative Heading |
| **Speed** | · Data Unavailable<br>· Confidence Too Large<br>· Speed Vaue Too High | · Speed Change (SC) Too Large<br>· SC $Inc\Phi$ with Acceleration | · Speed $Inc\Phi$ with Road Plausible Speed | · Speed $Inc\Phi$ with Relative Speed |
| **Drive Direction** | · Data Unavailable | · Direction $Inc\Phi$ with PC & Heading<br>· Direction $Inc\Phi$ with Speed | · Direction $Inc\Phi$ with Road Way | · Direction $Inc\Phi$ with Perceived Direction |
| **Vehicle Length/Width** | · Data Unavailable | – | – | · Vehicle Length and Width $Inc\Phi$ with Perceived Dimentions |
| **Longitudinal Acceleration** | · Data Unavailable<br>· Confidence Too Large<br>· Acc Value Too High | · Acceleration Change Too Large | – | · Acceleration $Inc\Phi$ with Relative Acceleration |
| **Curvature** | · Data Unavailable<br>· Confidence Too Large<br>· Curve Radius Too Small | · Curvature Change (CC) Too Large<br>· CC $Inc\Phi$ with Speed<br>· CC $Inc\Phi$ with HC<br>· CC $Inc\Phi$ with YawRate | · Curvature $Inc\Phi$ with Road Shape | · Curvature $Inc\Phi$ with Relative Curvature |
| **YawRate** | · Data Unavailable<br>· Confidence Too Large<br>· YawRate Value Too High | · YawRate Change (YC) Too Large<br>· YC $Inc\Phi$ with Speed<br>· YC $Inc\Phi$ with Curvature | – | · YawRate $Inc\Phi$ with Perceived YawRate |
| **Evidence Required** | · One Reported CAM (At least Including a full certificate) | · Multiple reported CAMs (At least one Including a full certificate) | · One Reported CAM (With a full certificate)<br>· CAMs of neighbors (With a full certificate each)<br>· Map of the Area (Already available for the MA) | · One Reported CAM (With a full certificate)<br>· Sender Sensor Information |
| **Evidence Reliability** | Total | Total | Partial | Minimal |

*$Inc\Phi$: inconsistency symbol.

The MD is based on a set of plausibility and consistency checks as listed in [3] and shown in Table I. These set of checks are performed by a vehicle when receiving a V2X message such as a Cooperative Awareness Message (CAM) or a Decentralized Environmental Notification Message (DENM)). When a vehicle detects a misbehavior, it generates a MR and sends it to the MA. Notice that the reporting is not a real time process. The report is sent to the MA when a connectivity is available via the cellular network or directly through the ITS-G5 network. The MA should proceed extensive data analysis to investigate whether a misbehavior has occurred or not in the network. Thus, a vehicle does not wait for a decision response about the reported node from the MA. Instead, it should be able to take appropriate decision locally such as blocking packet reception from the suspicious node.

The misbehavior reporting process should fit to the following requirements:

- Privacy protection: The MA should not be able to link the short term and the long term identity of the reported and the reporter entity. The reporter uses its pseudonym to communicate with the MA.
- Efficiency and minimum resource consumption: The MRs should not overload the communication channel. The reporting process should avoid sending repetitive and redundant information about the same misbehavior.
- Reliability and proof-based: The reporter should integrate the required proofs of the misbehavior: using the input data from the reporter, the MA should be able to re-compute the same misbehavior checks and get the same reported results.
- Flexibility: The MR should be extensible in order to integrate new misbehavior checks and new data proofs if needed.

## III. Proposed Misbehavior Reporting Approach

### A. Misbehavior Report Message

The proposed report format is provided on the page below in ASN.1. This format includes multiple key features:

- Reducing overhead by relating messages
- Verifying the sender with a pseudonym certificate
- Specifying the type of misbehavior
- Specifying the evidence required by misbehavior type

### B. Detailed Approach

In this analysis, we focus mainly on the CAM message. However similar approaches for the misbehavior evidence could be applied for other type of messages [4] [5]. In our

system, an ITS entity should refrain from reporting a misbehaving station that is continuously misbehaving. Instead, the station should send an initial report then wait whilst collecting evidences. After a certain period of time the entity sends a new report that includes the *RelatedReportsContainer*. This container specifies the ID of the initial report and the number of omitted reports along with the collected evidences. However, if in the meantime the reporter changes its pseudonym, the report should not include the initial report ID. This protocol would indeed prevent the linkability of the reporter pseudonyms by the MA thus ensuring the reporter privacy. Additionally, the report format requires at least one valid pseudonym certificate of the reported entity in the *ReportMessageContainer* to be valid. The detection type is specified in the *DetectionTypeContainer*. It could be on a security or semantic level. In case of a fail on the security level, an *OCTET STRING* should specify the error code (Table II). Every bit set to one infers a failed security test. This variable should include bits for all the security tests specified in the ETSI Technical Specifications [6] and [7].

TABLE II: securityDetectionErrorCode Description

| Octet ID | Bit ID | Security Reference |
|---|---|---|
| 0 | 0 | Time stamp (generation_time) |
| 0 | 1 | Region / GeographicRegion |
| 0 | 2 | Certificate validity period |
| 0 | 3 | Ascending order of hearder fields |
| 0 | 4 | Presence of AID (Application-ID) ssp list |
| 0 | 5 | No duplicate AID |
| 0 | 6 | AID in certificate are also in the parent certificate |
| 0 | 7 | Digest shall be included |
| 1 | 0 | Structure of the signature |
| 1 | 1 | The payload is present and its length is not nul |
| ... | ... | ... |

In case of a fail on the semantic level, the error code would depend on the type of the message included in *ReportedMessageContainer*. In the case of a CAM, the fail is linked to one or more data field as shown in Table I. Therefore, the *OCTET STRING* should point to the relevant data fields (Table III).

TABLE III: semanticDetectionErrorCodeCAM Description

| Octet ID | Bit ID | Data Field |
|---|---|---|
| 0 | 0 | ReferencePosition |
| 0 | 1 | Heading |
| 0 | 2 | Speed |
| 0 | 3 | DriveDirection |
| 0 | 4 | VehicleLength |
| 0 | 5 | VehicleWidth |
| 0 | 6 | LongitudinalAcceleration |
| 0 | 7 | Curvature |
| 1 | 0 | YawRate |
| ... | ... | ... |

The error code of the *DetectionReferenceContainer* is coupled with a detection level. The levels are defined as follows:

- **Level 1:** Implausibilities within a single message
- **Level 2:** Inconsistencies between successive messages
- **Level 3:** Inconsistencies with the local environment
- **Level 4:** Inconsistencies with respect to on-board sensors

Furthermore, Table I includes the required evidence to recreate the misbehavior checks defined by detection level. This allows to determine what evidence should be included in the *EvidenceContainer* based on the error code and the detection level.

The *EvidenceContainer* could include a list of V2X Messages of the reported vehicle and of the neighbors. It could also include the information about the sender, notably in case of a Level 4 detection. The information of type *FieldofView* and *PerceivedObjects* should be defined and used similarly to the Collective Perception Message (CPM). It is also to be noted that the detection level is correlated with the reliability of the report. In the case of the CAM, the detection of Level 1 & 2 entails signed messages of the reported vehicle as evidence. This type of evidence cannot be forged. Consequently the event could be confidently reproduced by the MA. A level 3 is based on the environment and surrounding messages. The environment information (e.g. map) could be inaccurate and the surrounding messages could be forged with a sybil attack. Finally, a Level 4 is based entirely on the reporting of vehicle's sensors thus the evidence could be forged with minimal effort.

## IV. Conclusion

In this paper, we proposed a detailed misbehavior reporting protocol which provides a set of misbehavior proofs to the central misbehavior authority. This allows the misbehavior authority to reproduce the reported misbehavior detection results and to combine them with other received reports. We defined precisely the report format in ASN.1 and describe the functionalities of each field of the message. As a future work, we would like to test several reports analysis approaches in the MA and evaluate the reliability of the centralized misbehavior detection.

## Acknowledgment

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program *Investissements d'avenir*.

## References

[1] Crash Avoidance Metrics Partners (CAMP) LLC, "EE Requirements and Specifications Supporting SCMS Software Release 1.2.2," 2016.

[2] N. Bißmeyer, "Misbehavior detection and attacker identification in vehicular ad-hoc networks," Ph.D. dissertation, Technische Universität, Darmstadt, December 2014.

[3] J. Kamel, A. Kaiser, I. B. Jemaa, P. Cincilla, and P. URIEN, "Feasibility Study of Misbehavior Detection Mechanisms in Cooperative Intelligent Transport Systems (C-ITS)," in *2018 IEEE 87th Vehicular Technology Conference: VTC2018-Spring*, Porto, Portugal, Jun. 2018.

[4] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On data-centric misbehavior detection in vanets," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2011, pp. 1–5.

[5] M. Ghosh, A. Varghese, A. Gupta, A. A. Kherani, and S. N. Muthaiah, "Detecting misbehaviors in vanet with integrated root-cause analysis," *Ad Hoc Netw.*, vol. 8, no. 7, pp. 778–790, Sep. 2010.

[6] "ETSI TS 103 097 V1.3.1: Intelligent Transport Systems (ITS); Security; Security header and certificate formats," pp. 1–23, October 2017.

[7] "ETSI TS 103 096-2 V1.3.1: Intelligent Transport Systems (ITS); Testing; Conformance test specifications for ITS Security;Part 2: Test Suite Structure and Test Purposes (TSS & TP)," pp. 1–184, March 2017.

# Misbehavior Report in ASN.1 Format

```
Its-Report DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS
TimestampIts, StationType, ReferencePosition, Heading, Speed,
DriveDirection, VehicleLength, VehicleWidth, Curvature,
LongitudinalAcceleration, CurvatureCalculationMode,
YawRate, PerceivedObjectContainer,
FieldofViewContainer FROM ITS-Container {
        itu-t (0) identified-organization (4) etsi (0) itsDomain (5)
        wg1 (1) ts (102894) cdd (2) version (1)
}
EtsiTs103097Data, EtsiTs103097Certificate FROM EtsiTs103097Module {
        itu-t(0) identified-organization(4) etsi(0) itsDomain(5)
        wg5(5) ts(103097) v1(0)
};

-- The root data frame for report messages
Report::= SEQUENCE {
        reportMetadataContainer ReportMetadataContainer,
        reportContainer ReportContainer
}

ReportMetadataContainer::= SEQUENCE {
        reportID IA5String,
        generationTime TimestampIts,
        relatedReportContainer RelatedReportContainer OPTIONAL
}

RelatedReportContainer::= SEQUENCE {
        relatedReportID IA5String,
        omitedReportsNumber OmitedReportsNumber
}

ReportContainer::= SEQUENCE {
        reportedMessageContainer ReportedMessageContainer,
        detectionTypeContainer DetectionTypeContainer,
        evidenceContainer EvidenceContainer OPTIONAL
}

ReportedMessageContainer::= CHOICE {
        certificateIncludedContainer CertificateIncludedContainer,
        certificateAddedContainer CertificateAddedContainer
}

CertificateIncludedContainer::= SEQUENCE{
        reportedMessage EtsiTs103097Data
}

CertificateAddedContainer::= SEQUENCE{
        reportedMessage EtsiTs103097Data,
        reportedCertificate EtsiTs103097Certificate
}

DetectionTypeContainer::= CHOICE {
        securityDetection SecurityDetection,
        semanticDetection SemanticDetection
}

SecurityDetection::= SEQUENCE {
        securityDetectionErrorCode OCTET STRING (SIZE (0..4)),
        ...
}

SemanticDetection::= CHOICE {
        semanticDetectionReferenceCAM DetectionReferenceCAM,
        semanticDetectionReferenceDENM DetectionReferenceDENM,
        semanticDetectionReferenceCPM DetectionReferenceCPM,
        semanticDetectionReferenceSPAT DetectionReferenceSPAT,
        semanticDetectionReferenceMAP DetectionReferenceMAP,
        ...
}

DetectionReferenceCAM::= SEQUENCE{
        detectionLevelCAM DetectionLevel,
        semanticDetectionErrorCodeCAM OCTET STRING (SIZE (0..2))
}

EvidenceContainer::= SEQUENCE {
        reportedMessageContainer MessageEvidenceContainer OPTIONAL,
        neighbourMessageContainer MessageEvidenceContainer OPTIONAL,
        senderInfoContainer SenderInfoContainer OPTIONAL,
        senderSensorContainer SenderSensorContainer OPTIONAL
}

MessageEvidenceContainer::= SEQUENCE OF EtsiTs103097Data

SenderInfoContainer::= SEQUENCE {
        stationType StationType,
        referencePosition ReferencePosition,
        heading Heading,
        speed Speed,
        driveDirection DriveDirection,
        vehicleLength VehicleLength,
        vehicleWidth VehicleWidth,
        longitudinalAcceleration LongitudinalAcceleration,
        curvature Curvature,
        yawRate YawRate
}

SenderSensorContainer::= SEQUENCE OF SenderSensorChoice

SenderSensorChoice::= CHOICE{
        fieldofViewContainer FieldofViewContainer,
        perceivedObjectContainer PerceivedObjectContainer
}

DetectionLevel ::= INTEGER { level(1) } (1..4)
OmitedReportsNumber ::= INTEGER { oneReport(1) } (0..1024)

END
```

.