

Recommendation

ITU-T X.1373 (03/2024)

SERIES X: Data networks, open system communications and security

Secure applications and services (II) – Intelligent transportation system (ITS) security

Secure software update capability for intelligent transportation system communication devices

ITU-T X-SERIES RECOMMENDATIONS

Data networks, open system communications and security

PUBLIC DATA NETWORKS	X.1-X.199
OPEN SYSTEMS INTERCONNECTION	X.200-X.299
INTERWORKING BETWEEN NETWORKS	X.300-X.399
MESSAGE HANDLING SYSTEMS	X.400-X.499
DIRECTORY	X.500-X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600-X.699
OSI MANAGEMENT	X.700-X.799
SECURITY	X.800-X.849
OSI APPLICATIONS	X.850-X.899
OPEN DISTRIBUTED PROCESSING	X.900-X.999
INFORMATION AND NETWORK SECURITY	X.1000-X.1099
SECURE APPLICATIONS AND SERVICES (I)	X.1100-X.1199
CYBERSPACE SECURITY	X.1200-X.1299
SECURE APPLICATIONS AND SERVICES (II)	X.1300-X.1499
Emergency communications	X.1300-X.1309
Ubiquitous sensor network security	X.1310-X.1319
Smart grid security	X.1330-X.1339
Certified mail	X.1340-X.1349
Internet of things (IoT) security	X.1350-X.1369
Intelligent transportation system (ITS) security	X.1370-X.1399
Distributed ledger technology (DLT) security	X.1400-X.1429
Application Security (2)	X.1450-X.1459
Web security (2)	X.1470-X.1489
CYBERSECURITY INFORMATION EXCHANGE	X.1500-X.1599
CLOUD COMPUTING SECURITY	X.1600-X.1699
QUANTUM COMMUNICATION	X.1700-X.1729
DATA SECURITY	X.1750-X.1799
INTERNATIONAL MOBILE TELECOMMUNICATIONS (IMT) SECURITY	X.1800-X.1839
METaverse AND DIGITAL TWIN SECURITY	X.2000-X.2199
SOFTWARE SUPPLY CHAIN SECURITY	X.2150-X.2199
ARTIFICIAL INTELLIGENCE (AI) / MACHINE LEARNING (ML) SECURITY	X.2200-X.2249

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1373

Secure software update capability for intelligent transportation system communication devices

Summary

As intelligent transportation system (ITS) technologies improve, it has become common for vehicles to communicate with other entities such as other vehicles, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Electric devices inside a vehicle such as electronic control units (ECUs), electronic toll collections (ETCs) and car navigation systems are becoming more sophisticated. As a result, software modules inside these electric devices need to be appropriately updated for the purposes of bug fixing and performance and security improvements, to avoid accidents.

In order to fulfil the above requirements, Recommendation ITU-T X.1373 provides secure software update procedures between a software update server and vehicles with appropriate security controls including in-vehicle communication messages. This Recommendation can be utilized by car manufacturers and ITS-related industries as a set of standard capabilities for best practice.

History *

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T X.1373	2017-03-30	17	11.1002/1000/13197
2.0	ITU-T X.1373	2024-03-01	17	11.1002/1000/15664

Keywords

Communication devices, denial of service attack, DoS, embedded system, hardware security module, intelligent transportation system, ITS, malware, privacy, risk analysis, V2I, V2V, vehicle-to-infrastructure, vehicle-to-vehicle, vehicle-to-X (vehicle/infrastructure) (V2X), wireless communications.

* To access the Recommendation, type the URL <https://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2024

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Recommendation.....	1
4 Abbreviations and acronyms	2
5 Conventions	3
6 Basic model of remote software updates	3
6.1 Modules of ITS environment for software updates	3
6.2 Model of software update procedure	4
7 Threats and security requirements and security capabilities	6
7.1 Threats regarding software update procedures.....	7
7.2 Security requirements	7
7.3 Security capabilities.....	7
8 Specification of secure software update procedure for outside of the vehicle	8
8.1 General message format with security functions.....	8
8.2 Protocol definition and data format	8
9 Specification of secure software update procedure for inside the vehicle.....	19
9.1 General message format with security functions.....	20
9.2 Protocol definition and data format	20
Appendix I – Threats and security requirements	28
I.1 Major modules for secure software updates	28
I.2 Threats to the modules.....	29
I.3 Security requirements for the modules.....	33
Appendix II – Examples of messages in XML format	40
Bibliography.....	43

Recommendation ITU-T X.1373

Secure software update capability for intelligent transportation system communication devices

1 Scope

In the context of updates of software modules in the electric devices of vehicles in the Intelligent Transportation System (ITS) communication environment, this Recommendation aims to provide a procedure and the components of secure Over-The-Air (OTA) software updates for ITS communication devices. This includes a basic model of a software update procedure, with its messages and related attributes, for completing software updates with security considerations taken into account. This procedure is defined in the application layer for preventing threats, such as the tampering of and malicious intrusion to communication devices in vehicles, which are transferred using existing communication protocols. Procedures based on wired software updates through the on-board diagnostics (OBD) port or portable devices is out of the scope of this Recommendation.

The procedure is to be applied to communication devices on ITS vehicles under vehicle-to-infrastructure (V2I) communications by means of the Internet and/or ITS-dedicated networks. The procedure provides technical guidelines which can be utilized by car manufacturers and ITS-related industries as a good reference. This Recommendation does not specify any compliance requirements for software updates.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T X.509] Recommendation ITU-T X.509 (2019) | ISO/IEC 9594-8:2020, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.
- [ISO 14229-1] ISO 14229-1:2020, *Road vehicles – Unified diagnostic services (UDS) – Part 1: Specification and requirements*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 threat [b-ISO/IEC 27000]: Potential cause of an unwanted incident, which may result in harm to a system or organization.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 device: A thing made or adapted for a particular purpose such as vehicle communications, especially a piece of mechanical or electronic equipment.

NOTE – In this Recommendation, the term 'devices' is used for "communication devices" such as Electronic Control Units (ECUs).

3.2.2 module: Each of a set of standardized parts or independent units that can be used to construct a more complex structure, such as an item of vehicle.

NOTE – In this Recommendation, modules are used as an abstract representation of software components used in vehicles.

3.2.3 vehicle mobile gateway (VMG): A conceptual module which provides communication between electronic control units (ECUs) in the controller area network (CAN) (in-vehicle buses) and exterior intelligent transportation system (ITS) entities in the external network.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CAN	Controller Area Network
CD	Compact Disc
DID	Data Identifier
DoS	Denial of Service
DVD	Digital Versatile Disc
ECU	Electronic Control Unit
ETC	Electronic Toll Collection
GUID	Global User ID
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifier
IMT	International Mobile Telecommunications
IT	Information Technology
ITS	Intelligent Transportation System
MAC	Message Authentication Code
OBD	On-Board diagnostics
PC	Personal Computer
RPM	Revolutions Per Minute
RxSWIN	Rx Software Identification Number
SHA	Secure Hash Algorithm
SID	Service ID
SSL	Secure Socket Layer
UI	User Interface
URL	Uniform Resource Locator
Usvr	Update server
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-X (vehicle/infrastructure)

VMG	Vehicle Mobile Gateway
XML	extended Mark-up Language

5 Conventions

This Recommendation uses the following conventions:

The verbal form "should" indicates a requirement that is recommended, but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The verbal form "shall" indicates a requirement that must be strictly followed and from which no deviation is permitted if conformity to this Recommendation is to be claimed.

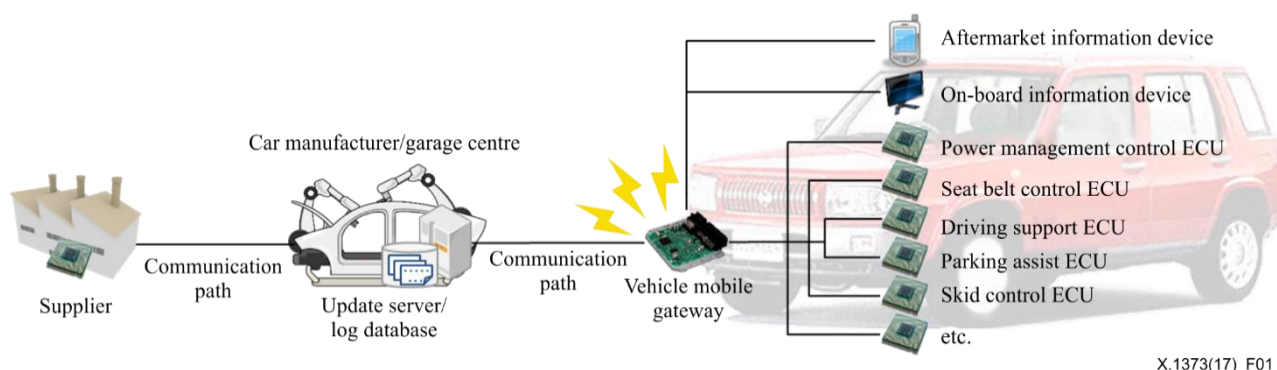
In the body of this Recommendation, the verbal form "can" sometimes appears, in which case it is to be interpreted as "is able to".

6 Basic model of remote software updates

For considerations of a practical security architecture, this clause introduces a basic model of conventional architecture for software updates, where a definition of principal modules and typical software update processes are provided.

6.1 Modules of ITS environment for software updates

Figure 1 provides a view of the principal modules around a vehicle for a remote software update in an ITS communication environment. The principal modules are information devices, electronic control units (ECUs) and a vehicle mobile gateway (VMG), which is a conceptual module on a vehicle, update server (Usvr) and log database of the car manufacturer and supplier.



X.1373(17)_F01

Figure 1 – Principal modules around a vehicle

6.1.1 User interface

The user interface (UI) is generally an on-board or an aftermarket information device with display and input devices on a vehicle. Such an information device is connected directly to other devices on a vehicle (e.g., VMG or ECU (see clause 6.1.2)), so that it can obtain and indicate various status information of the vehicle such as speed, revolutions per minute (RPM), fuel level, and so on. Particularly, in this Recommendation, a user interface is used to notify car drivers of the necessity for updates.

6.1.2 Electronic Control Unit (ECU)

ECU is a generic term for computers that control various kinds of devices in a vehicle. In the earliest years of ECU, its main functions were controls of ignition timing, injection, idling adjustment and limiter of engine in order to improve fuel efficiency and reduce gas emissions. According to the computerization of the vehicle, ECU has expanded its applications to various kinds of functions such

as power management, seat belt control, driving support, parking assist, skid control, automatic transmission, etc. In recent years, the number of ECUs within a vehicle increased from 50 to 100, and the importance of ECUs for safety control and communication is especially growing. However, since the development of ECU involves sophisticated software implementations, the recent increase of ECUs in vehicles imposes a heavy burden on car manufacturers.

6.1.3 Vehicle mobile gateway

A vehicle mobile gateway is a conceptual module logically assigned for facing "Update Server" (in clause 6.1.4), in order for the vehicle to conduct a software update. VMG can be a conceptual entity, which is implemented with a set of multiple components. For example, the connection management entity (a.k.a. "central gateway", "Head unit", "communication head unit" or "vehicle gateway (VG)") can be used for the role of VMG in this context, and any devices can also be used for the software update. A cellular network (mobile network) and a fixed wireless network are applied as a communication path between the vehicle mobile gateway and exterior ITS entities.

6.1.4 Update server and log database

An update server is located at car manufacturer centres or garage centres to gather status information of software modules from vehicles and to distribute software update modules to the vehicles. Likewise, in the most recent networked computers such as personal computers (PCs) and smart phones, one of the important functions of the update server is to entirely manage and control the software on vehicles. In order to automatically manage the software status of each vehicle, the update server should work with a log database which stores such software status information of the vehicle as evidence. Note that an update server can be developed and delivered not only by a car manufacturer but also by a supplier or third party.

6.1.5 Supplier

A vehicle is an assembly of thousands of automotive parts that are provided by automotive suppliers. On-board communication devices and ECUs are provided by those suppliers, and they are assembled by car manufacturers while considering dependencies between the various devices. Therefore, and in general, update modules for on-board communication devices are not produced beforehand by a car manufacturer but by a corresponding supplier. The provided update modules are distributed from a car manufacturer to vehicles after careful testing and evaluation at the car manufacturer.

6.2 Model of software update procedure

6.2.1 General update procedure

Figure 2 shows a typical model of the software update procedure which is initiated by the vehicle mobile gateway by checking the existence of updates. This Recommendation provides the procedure based on this model in Figure 2; however, it should not intend to limit the actual functional allocation to Figure 2.

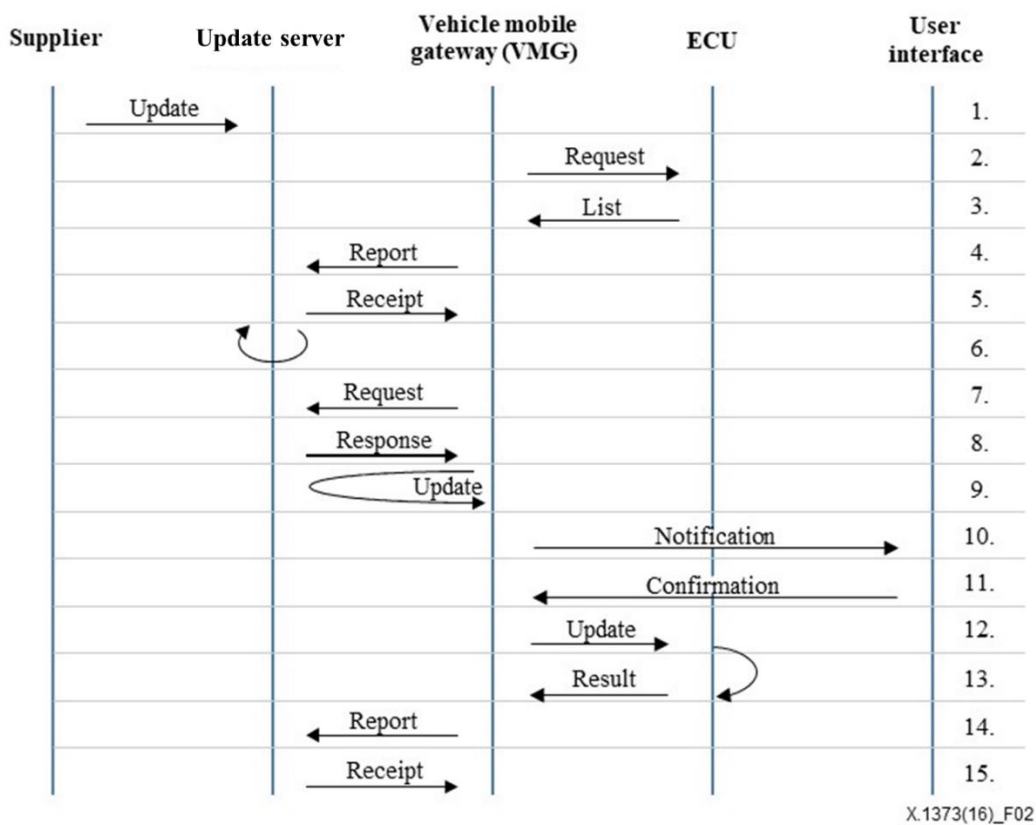


Figure 2 – Model of software update process

Steps for the update procedure are described as follows:

- 1) At the first step of the process, an update module is provided by an automotive component supplier, which occurs asynchronously with the following steps.
- 2) As the initiation of the update procedure starts, a vehicle mobile gateway requests ECUs to submit their software list.
- 3) An ECU checks its software status, generates a list of software modules and reports it to the VMG.
- 4) The VMG submits the collected list to the update server to check whether any update for the vehicle exists.
- 5) The update server sends back a receipt of the submitted list to the VMG.
- 6) According to the list, the update server inspects the status of the installed software of the vehicle and determines the necessary software updates for the ECUs.
- 7) Since this inspection may take a long time, the VMG periodically checks the necessity of the updates for the vehicle.
- 8) If there is any update, the update server sends access information such as Uniform Resource Locators (URLs) for the updates; otherwise, it sends back only an acknowledgement message.
- 9) If there is any update for the vehicle, the VMG connects to the update server to download the update modules for the vehicle.
- 10) Before applying the updates to ECUs, the VMG notifies the driver to confirm the application of the updates.

NOTE 1 – The methods for confirmation to the driver can be on a case-by-case basis, such as confirmation from the VMG or confirmation from the ECU. In addition, the confirmation can be

displayed not only by in-vehicle display but also by a device outside the vehicle such as a smartphone. In this Recommendation, the notification based on Figure 2 is only applied.

- 11) The driver confirms and accepts to apply the updates.
- 12) The VMG delivers the update files to the corresponding ECUs and requests them to apply the updates (see clause 6.2.3).
- 13) Each ECU applies the update and reports the application result to the vehicle mobile gateway.
- 14) The vehicle mobile gateway submits a report of application results to the update server.
- 15) Finally, the update server sends back a receipt of the update information. If the application of the update has failed or any remaining updates are found, the update server retries the procedure from steps 6 to 14 until the application has succeeded (see clause 6.2.2). If the application has not succeeded even though the update procedure has been retried, the failure recovery should be applied for the fail-safe (see clause 6.2.5).

NOTE 2 – It is dependent on the countries whether such notification (Step 10) should be sent to the driver before or after the actual updates. In some countries, steps 10 and 11 can be after step 13 (after updating).

6.2.2 Consideration for unbounded retries

According to the retry-until-succeed description mentioned in step 15, note that in some cases the procedure will never succeed, and eventually the description can cause the VMG to retry an unbounded number of times. To avoid this, the number of retries should be bounded by a number 'N' which can be determined based on the policy for the update procedure. How to define the policy for the update is out of the scope of this Recommendation.

6.2.3 Consideration for resource constraints

As for the practical application of the update software in a vehicle, there are components in the vehicle that do not mount enough memory resource for caching a whole update module at one time. For those modules, the stream updating technology is required by means of streaming fragmented data.

In general, for any modules in a vehicle, whatever the update system, the constraints of finite resources of devices such as memory, storage and network throughput should be carefully taken into account.

6.2.4 Consideration for failure on software downloads

During the software downloading, it can be stopped or possibly disconnected because of a limitation of connectivity or connectivity expense. If so, the vehicle retries to connect to the server to download target software. To make the system efficient, both the update server and vehicle should support the continuity of downloading target software.

6.2.5 Consideration for failure recovery

Vehicles which adopt a secure software update functionality should ensure the failure recovery or be placed into a safe state from a failed or interrupted update. For the failure recovery, different kinds of solutions such as recovering from a central or cloud server, recovering from another system's memory and memory swapping can be used. How to recover the failed device to the safe previous version is out of the scope of this Recommendation.

7 Threats and security requirements and security capabilities

This clause describes major threats, security requirements and security capabilities for secure software updates. More detailed examples of major threats and security requirements are provided in Appendix I.

7.1 Threats regarding software update procedures

7.1.1 Manipulation and corruption of software before update process

Regardless of whether the over-the-air update or local/physical update is used, the update procedure can include threats by fabricating system update programs or compromised firmware.

The software data can be manipulated and corrupted before the update process, even though the update process is intact. The software provider creates/prepares their software for the update and the software is delivered to the target systems which require the update. Therefore, there can be a serious threat that the software can be manipulated and corrupted before it is served.

7.1.2 Denial of service attacks

The services required for a software update can be disturbed by intentionally overloading various resources in the update server, network or vehicle, etc. In particular, a denial of service attack against the update server or network to prevent the rollout of critical software updates and/or the unlocking of customer-specific features can be a possible attack in the software update procedure. This can also prevent legitimate updates.

7.1.3 Compromised software update process

The entire process involved in updating software can be compromised by various types of attacks such as spoofing, exploitation of update software data and malware intrusion. The threats described in clause 7.1.3 are not only related to software updates, but are general threats in the automotive environment. It should be considered that the failure of the update process, as a consequence of this threat, will have a significant impact on the normal operation of ITS.

7.2 Security requirements

7.2.1 Requirement 1 (corresponding to clause 7.1.1)

The software for updates should be protected to prevent manipulation and corruption before the update process. Especially, tampering, rewriting, erasure and confusion are prevented for the software created before the update process, and the responsibility for creation is clear from the perspective of a third party.

7.2.2 Requirement 2 (corresponding to clause 7.1.2)

Availability of the update server and related components should be ensured against denial of service attacks for protecting update server or network.

7.2.3 Requirement 3 (corresponding to clause 7.1.3)

The entire update processes should be protected to prevent them being compromised for ensuring entity authentication, confidentiality of software data and secure system operation without malware contamination.

7.3 Security capabilities

7.3.1 Security capability for Requirement 1

The integrity capability and data source authenticity capability should be implemented for software data against manipulation and corruption of software data before the update process for ensuring the integrity and authenticity of the software data.

7.3.2 Security capability for Requirement 2

Appropriate security controls should be provided against denial of service (DoS/DDoS) attacks for the update server or network such as the detection and mitigation capability of DoS/DDoS attacks.

7.3.3 Security capability for Requirement 3

Entities' authentication capability should be implemented between the "Update server" and "Vehicle Mobile Gateway (VMG)". If the software supplier performs the update online, the entity authentication capability should also be implemented between the "Supplier" and "Server Update".

Moreover, the confidentiality capability should be implemented for data exchanged in the update processes. Furthermore, it is preferable to implement a "detection mechanism" against any injection of malware, etc.

8 Specification of secure software update procedure for outside of the vehicle

This clause describes a practical procedure and its application message formats between the update server and vehicle (VMG) for software updates with security functions. Note that clause 8 does not specify functions of message confidentiality. The confidentiality can be provided by lower layer protocols (e.g., hypertext transfer protocol secure (HTTPS) and secure tunnelling protocol, etc.).

The procedure needs to take into account the diversity of security capabilities between vehicles in the following subclauses.

8.1 General message format with security functions

This clause introduces a general message format with security functions including an authentication method of message sender and verification of the integrity of a message. In terms of an integrity and authenticity technique, the digital signature method with the public key algorithm and/or message authentication code (MAC) with shared key algorithm can be applied. In the secure software update procedure, every message should be constructed in one of the security methods in the following subclauses.

8.1.1 Digital signature method

As one implementation method, the digital signature based on [ITU-T X.509] can be applied for authentication of entities and verification of message integrity between vehicles with asymmetric crypto capability.

8.1.2 MAC method

The MAC method ensures the accuracy and consistency of data over its vehicle life-cycle and can be used in any phases for design, implementation and usage of any vehicle system which stores, processes or transfers data. Several MAC methods can be considered, such as mechanisms using a block cipher [b-ISO/IEC 9797-1] or universal hash-function [b-ISO/IEC 9797-3].

8.2 Protocol definition and data format

The application data format is dedicated to deliver messages only regarding the software update, which is enclosed in the general message format described in the previous clause. This clause firstly defines message types applied in the software update procedure, and secondly it introduces specifications of the message types. The examples of messages are provided with an extended mark-up language (XML) format for information.

8.2.1 Protocol overview

Based on the model of software update procedure described in clause 6, the messages are categorized into several types according to their objectives, as depicted in Figure 3. Procedures of in-vehicle communication are out of the scope of this Recommendation and are shown in grey in Figure 3.

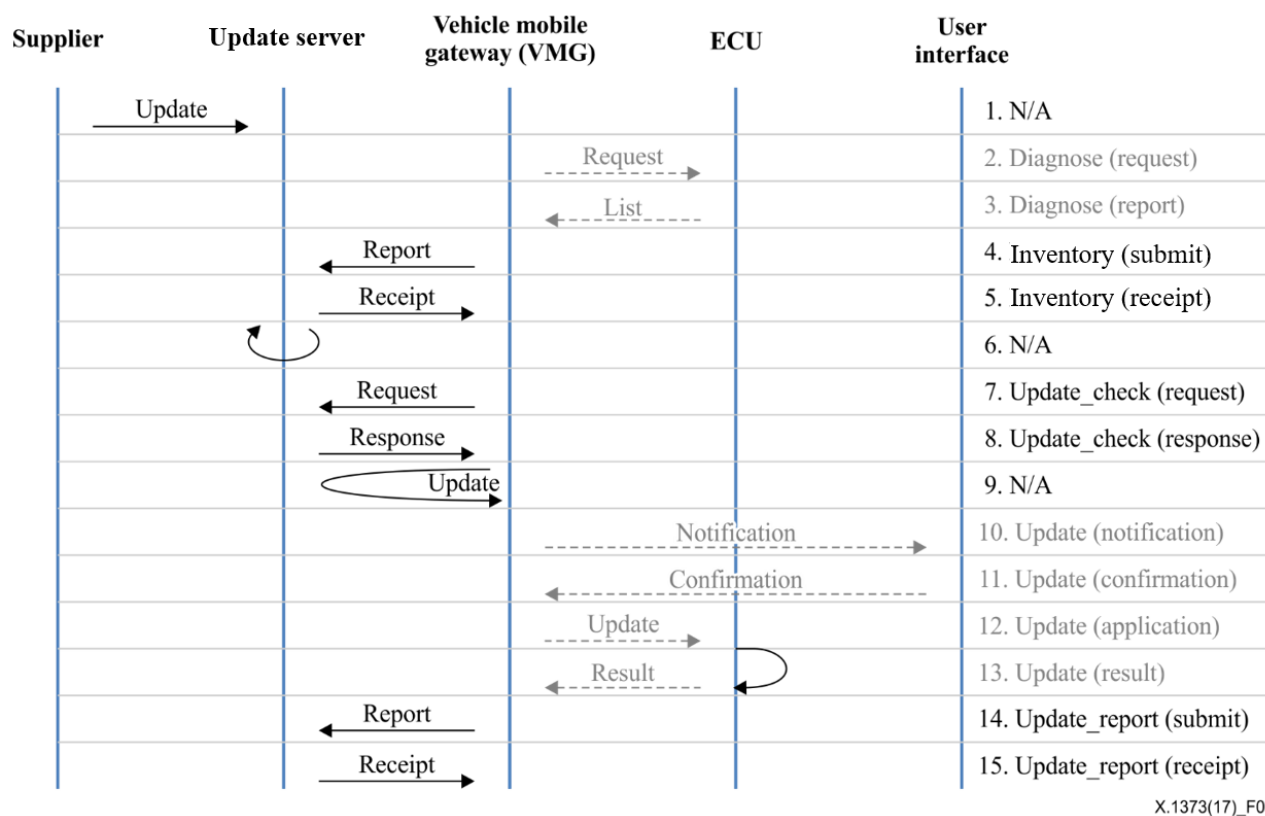


Figure 3 – Definition of message types

In steps 2, 3, 4 and 5, since the purpose of the messages are request and report of diagnoses/inventories of the software status of each ECU, the messages are categorized as "diagnose/inventory". In the same manner, the messages in steps 7 and 8 are categorized as "update_check". Steps 10, 11, 12 and 13 are "update", which are messages to confirm and apply the updates. Finally, the results of the updates are submitted by "update_report" messages in steps 14 and 15. The types, subtypes and codes of the messages are shown in Table 1.

Table 1 – Message types for outside of the vehicle

Type	Subtype	From	To	Purpose
inventory	submit	VMG	Usvr	Report of results of ECUs in a vehicle
	receipt	Usvr	VMG	Receipt for submit of inventory report
update_check	request	VMG	Usvr	Request of update module
	response	Usvr	VMG	Update module provided
update_report	submit	VMG	Usvr	Report of application of the update
	receipt	Usvr	VMG	Receipt of the report

* Usvr: Update server

8.2.2 Inventory messages

With the purpose of determining the necessary update modules for a vehicle, inventory messages are used between an update server and VMG to upload software information from vehicles to the update server. Elements specified in each structure of the message are not mandatory. Assignment of mandatory elements to be used in the message is out of the scope of this Recommendation.

8.2.2.1 Inventory (submit) message

After collecting the inventory results from the vehicle, the VMG submits a list of software information to the update server in the manufacturer (or garage). The inventory (submit) message includes the identity of the vehicle (vid) in the "Vehicle" element and a list of software information that is extracted from diagnose (report) messages. Software information can be ID, version and others of software modules.

Figure 4 specifies the structure of the inventory (submit) message and Table 2 provides elements of the message, which include the names of elements and attributes in the elements.

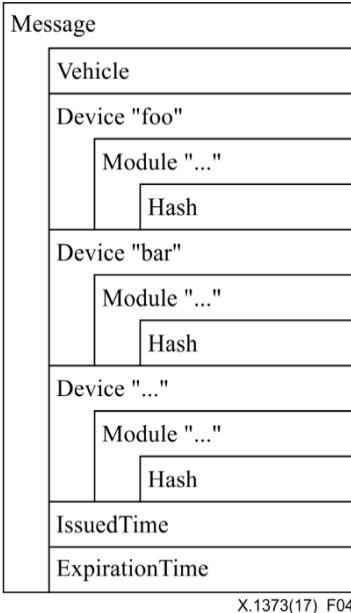


Figure 4 – Structure of inventory (submit) message

Table 2 – Elements of inventory (submit) message

Element	Attribute in element	Description
Message	–	Container of the message.
	protocol	The value to identify "protocol" used. (e.g., value=1.0)
	version	The version number of the message sender.
	type	The type is to identify the message type (in this case, type is "inventory".)
	subtype	The subtype is to identify the message subtype (in this case, subtype can be "submit".)
	sessionid	Session Identifier (ID) is a random Global User ID (GUID) associated with the inventory session. An identical session ID is applied to a set of diagnose request, report, inventory submit and receipt messages.
	trustlevel	Trust level is determined based on the security capability and safety requirement of the device that generated this message.
	ownerid	Owner ID provided by a car manufacturer/supplier.
	messageid	Message ID is a random GUID associated with an individual message.