SLOVENSKI STANDARD
SIST-TS CLC/TS 50136-9:2021

01-januar-2021

Nadomešča:
SIST-TS CLC/TS 50136-9:2017

Alarmni sistemi - Sistemi in oprema za prenos alarma - 9. del: Zahteve za skupni protokol za prenos alarma po internetnem protokolu

Alarm systems - Alarm transmission systems and equipment - Part 9: Requirements for common protocol for alarm transmission using the Internet Protocol (IP)

Alarmanlagen - Alarmübertragungsanlagen und -einrichtungen - Teil 9: Anforderungen an standardisierte Protokolle zur Alarmübertragung unter Nutzung des Internetprotokolls (IP)

Systèmes d'alarmes - Systèmes et équipements de transmission d'alarme - Partie 9 : Exigences pour le protocole commun de transmission d'alarme utilisant le protocole Internet (IP)

Ta slovenski standard je istoveten z: **CLC/TS 50136-9:2020**

ICS:

13.320	Alarmni in opozorilni sistemi	Alarm and warning systems
33.040.40	Podatkovna komunikacijska omrežja	Data communication networks

SIST-TS CLC/TS 50136-9:2021 **en**

TECHNICAL SPECIFICATION

SPÉCIFICATION TECHNIQUE

TECHNISCHE SPEZIFIKATION

CLC/TS 50136-9

November 2020

ICS 13.320; 33.040.40

Supersedes CLC/TS 50136-9:2017

English Version

Alarm systems - Alarm transmission systems and equipment - Part 9: Requirements for common protocol for alarm transmission using the Internet Protocol (IP)

Systèmes d'alarmes - Systèmes et équipements de transmission d'alarme - Partie 9 : Exigences pour le protocole commun de transmission d'alarme utilisant le protocole Internet (IP)

Alarmanlagen - Alarmübertragungsanlagen und -einrichtungen - Teil 9: Anforderungen an standardisierte Protokolle zur Alarmübertragung unter Nutzung des Internetprotokolls (IP)

This Technical Specification was approved by CENELEC on 2020-09-28.

CENELEC members are required to announce the existence of this TS in the same way as for an EN and to make the TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Ref. No. CLC/TS 50136-9:2020 E

Contents

Page

Tables

European foreword

This document (CLC/TS 50136-9:2020) has been prepared by CLC/TC 79 "*Alarm systems*".

This document supersedes CLC/TS 50136-9:2017.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

This document specifies a common IP transport protocol for alarm transmission. The published version (2017, second version) required solving both technical and security issues identified during the first actual implementations of the protocol. The working group was working closely with the early adopters of the protocol and has a very clear and complete list of issues and solutions. This revision supersedes the previous version.

EN 50136 consists of the following parts, under the general title *Alarm systems - Alarm transmission systems and equipment*:

— *Part 1: General requirements for alarm transmission systems*

— *Part 2: General requirements for Supervised Premises Transceiver (SPT)*

— *Part 3: Requirements for Receiving Centre Transceiver (RCT)*

— *Part 4: Annunciation equipment used in alarm receiving centres*

— *Part 5: (Free)*

— *Part 6: (Free)*

— *Part 7: Application guidelines*

— *Part 8: (Free)*

— *Part 9: Requirements for a common protocol for alarm transmission using the Internet Protocol (IP)*

1 Scope

This document specifies a protocol for point-to-point transmission of alarms and faults, as well as communications monitoring, between a Supervised Premises Transceiver and a Receiving Centre Transceiver using the Internet Protocol (IP).

The protocol is intended for use over any network that supports the transmission of IP data. These include Ethernet, xDSL, GPRS, WiFi, UMTS and WIMAX.

The system performance characteristics for alarm transmission are specified in EN 50136-1.

The requirements for the performance of the alarm transmission system, the SPT and the RCT are specified in the relevant parts of the EN 50136 series.

Compliance with this document is voluntary.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 50136-1:2012, *Alarm systems - Alarm transmission systems and equipment - Part 1: General requirements for alarm transmission systems*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in EN 50136-1:2012 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply.

AES Advanced Encryption Standard

ARC Alarm Receiving Centre

ATP Alarm Transmission Path

ATS Alarm Transmission System

CA X.509 Certificate Authority

CBC Cipher Block Chaining

CRC Cyclic redundancy check

DNS Domain Name System

DTLS Datagram Transport Layer Security

HL Header Length

IP Internet Protocol

IV Initialization Vector

MAC	Media Access Control
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NVM	Non-Volatile Memory
P-MTU	Path Maximum Transmission Unit
RCT	Receiver Centre Transceiver
RX	Receive
SCTP	Stream Control Transmission Protocol
SNTP	Simple Network Time Protocol
SPT	Supervised Premises Transceiver
TFTP	Trivial File Transfer Protocol
TX	Transmit
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
WS	Window Size

4 Objective

The objective of this document is to specify the protocol details (transport and application layers) for alarm transmission systems using Internet Protocol (IP), to ensure interoperability between SPTs and RCTs supplied by different manufacturers. Mechanisms to commission SPT and RCT and build mutual trust between the communicating parties are also described.

As compliance with this document is voluntary, any other alarm transmission protocol or equipment not covered by this document may be used, provided that the requirements of EN 50136-1 are met.

The protocol and its elements concern one ATP only.

This protocol is designed to run on top of UDP and is designed to support both IPv4 and IPv6.

NOTE For more information on the use of IP and UDP in alarm transmission, please refer to Annex F.

5 Messaging

5.1 General

This clause defines the messaging layer, on top of which the alarm event data are transmitted using the existing reporting formats such as, for example, Sia and Contact ID. Clause 7 defines the initial commissioning of an SPT, as well as how SPTs connect to the RCT.

The functionality of the alarm messaging and polling protocol includes:

— exchanging master and session parameters;

— (alarm) event reporting (including linking to out-of-band additional data related to events, like audio/video);

— line monitoring;

— transparent message transmission, e.g. vendor specific messages that, for example, can be used for remote commands from RCT to SPT.

It fulfils the following requirements:

— encryption, fulfilling requirements for most demanding category of EN 50136-1;

— authentication, fulfilling requirements for most demanding category of EN 50136-1;

— SPT: allows a broad range of hardware (limited demands on memory footprint as well as CPU power);

— RCT: allows support for at least 10 000 SPTs in compliance with any category in EN 50136-1, using modern general purpose server hardware;

— allow Dynamic IP addresses of the SPTs;

— allow one or more SPTs to be placed behind a NAT firewall.

5.2 Message format overview

5.2.1 General

This subclause describes the basic outline of all messages.

Each message shall be explicitly acknowledged, including line supervision messages.

Backwards compatibility is achieved by the implementation of a response with the same message id as the unknown message, however with bit 7 set, with the result code: “RESP_CMD_NOT_SUPPORTED”.

Table 1 — Backwards compatibility

Byte index	Bytes	Description
0	**HL**	**Header, Message ID and Message Length**
HL	**1**	**Result code**
		Padding
		Hash

Table 2 — Backwards compatibility result code

Description	Purpose Description
RESP_CMD_NOT_SUPPORTED	**If the message ID is not supported by the RCT**

Multi-byte values will be transmitted using network byte order (big-endian).

Examples:

1. Consider sending message: "Example message."In this case "E" is considered the most significant byte and should therefore be sent first.

2. Consider sending the encryption key (hex): "363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C" In this case 0x36 is considered the most significant byte and should therefore be sent first.

This rule will follow the recommendations of the RFC 1700.

https://tools.ietf.org/html/rfc1700

5.2.2 Identifiers

The following identifiers exist:

Table 3 — Identifiers

Description	Purpose	Present in	Encrypted	See
Connection Handle	**Look up the current symmetric encryption key**	**All messages**	**No**	**5.2.4**
Device ID	**Uniquely identify the hardware**		**N / A**	**5.2.5**

The Connection Handle is unencrypted. It is a unique number, initialized during the setup of the connection. Its sole purpose is to be able to look up the encryption key. It is valid for the communication session only.

The Device ID uniquely identifies the hardware once the connection has been established.

NOTE Device ID is not equivalent to any account code or similar ID specified by application protocol.

The Device ID shall be stored in non-volatile memory within the SPT.

The IP address is not used for identification purposes, in order to allow for the use of dynamic or translated IP addresses.

5.2.3 Message format

The basic unencrypted format of all messages is as follows. Message in this format is never transmitted. It is described here only to clarify the hash value calculation.

Table 4 — Basic unencrypted format of messages

Byte Index	Bytes	Description	See	Group
0	**4**	**Connection handle**	**5.2.4**	**Header**
4	**2**	**Tx Sequence number**	**5.2.8**	
6	**2**	**Rx Sequence number**	**5.2.8**	
8	**2**	**Flags**	**5.2.9**	
10	**1**	**Protocol version number**	**5.7**	
11	**1**	**Message ID**	**5.2.6**	
12	**2**	**Message Length**	**5.2.7**	Message
14	**n**	**Message data**	**Clause 6**	
14 + n	**n**	**Padding**	**6.4**	

The basic encrypted, transmitted format of all messages is as follows. Note that the Device ID field is not included in the encrypted message and its value is not used to compute the message hash value, i.e. the hash is calculated from the unencrypted version of the message described above.

Table 5 — Basic encrypted format of messages

Byte Index	Bytes	Description	See	Encrypted	Group
0	4	Connection Handle	5.2.4	No	Header
4	2	Tx Sequence number	5.2.8	Yes	Message
6	2	Rx Sequence number	5.2.8	Yes	
8	2	Flags	5.2.9	Yes	
10	1	Protocol version number	5.7	Yes	
11	1	Message ID	5.2.6	Yes	
12	2	Message Length	5.2.7	Yes	
14	n	Message data	Clause 6	Yes	
14 + n		Padding	5.3.1	Yes	Tail
	32 32	Hash – SHA-256, or Hash – RIPEMD-256	5.4	Yes	

The Connection Handle is unencrypted; the remainder of the message (from Tx Sequence Number up to and including Hash) is encrypted using the encryption method as negotiated during the commissioning stage.

Message ID's are defined in pairs: each message has its matching response. For responses the first byte of the Message Data always holds a 'Result code' as defined in normative Annex A.

All fields are described in detail in the following subclauses.

5.2.4 Connection handle

The Connection Handle is assigned (uniquely for the RCT to which a SPT reports) using the commissioning protocol. The RCT creates a unique Connection Handle and links this to the Device ID of the SPT in its internal database. This translation results in a compact, fixed length Connection Handle.

The purpose of the Connection Handle is to be able to determine the encryption key to be used to decrypt the received message, independent of the IP address of the message.

The Connection Handle is a 32-bit binary value.

For the purpose of commissioning a SPT, the RCT will present a one-time-connection-handle to the installer/operator. It will be shared/represented as a hexadecimal number (consisting of eight characters).

The final Connection Handle as in use after the commissioning phase is not a (by the installer/operator) configurable parameter, nor made visible on user interfaces. It is used internally by the SPT/RCT equipment only.

5.2.5 Device ID

5.2.5.1 General

The Device ID uniquely identifies the SPT.

Within the message header, the Device ID itself is never transmitted.

Device ID is 16 bytes long.

5.2.5.2 SPT Device ID

The Device ID of the SPT is an ID that is random to the SPT, but fixed and read-only over the lifetime of the SPT, i.e. a hardware serial number. It is unique within the SPT database in the RCT.

The Device ID is created during manufacturing time of the device; in messaging, it is never transmitted itself in cleartext, but is needed to be known in cleartext for the ARC to configure the RCT accordingly.

Thus, it is only transmitted during initial commissioning phase to the RCT.

Uniqueness is ensured by the following principles:

— Each SPT manufacturer shall use his 24 bits “Organizationally Unique Identifier” as assigned to him by the IEEE for MAC-address generation.

— Each SPT manufacturer not having such a code shall attend for such a code from IEEE.

— If an interface in the SPT makes use of a MAC address, the next 24 bits in the device ID shall be the same as the rest of MAC address specified by the manufacturer. If such interface does not exist, the manufacturer shall use another numbering scheme documented by the manufacturer.

— The manufacturer shall use non-consecutive, randomly distributed numbers for the rest of the device ID field and guarantee uniqueness for all his delivered SPT devices.

5.2.6 Message ID

The Message ID’s as used are listed in the following Table 6:

Table 6 — Message ID overview

Message name	Description	Direction SPT ←▪→ RCT	Version	Message ID
POLL_MSG	Poll message	← →	1	0x11
EVENT_MSG	Event message	→	1	0x30
CONN_HANDLE_REQ	Connection handle request	→	1	0x40
DEVICE_ID_REQ	Device ID request	→	1	0x41
ENCRYPT_SELECT_REQ	Encryption selection request	→	1	0x42
ENCRYPT_KEY_REQ	Encryption key exchange	← →	1	0x43
HASH_SELECT_REQ	Hash selection request	→	1	0x44
PATH_SUPERVISION_RE Q	Path supervision request	← →	1	0x45
SET_TIME_CMD	Set time command	←	1	0x47
VERSION_REQ	Protocol version request	→	1	0x48
DTLS_COMPLETE_REQ	DTLS completed request	→	1	0x62
RCT_IP_REQ	RCT IP parameter request	→	1	0x63
TRANSPARENT_MSG	Transparent message	← →	1	0x70
POLL_RESP	Poll Response	← →	1	0x91
EVENT_RESP	Event response	←	1	0xB0
CONN_HANDLE_RESP	Connection handle response	←	1	0xC0
DEVICE_ID_RESP	Device ID response	←	1	0xC1
ENCRYPT_SELECT_RES P	Encryption selection response	←	1	0xC2
ENCRYPT_KEY_RESP	Encryption key exchange response	← →	1	0xC3
HASH_SELECT_RESP	Hash selection response	←	1	0xC4
PATH_SUPERVISION_RE SP	Path supervision response	← →	1	0xC5
SET_TIME_RESP	Set time response	→	1	0xC7
VERSION_RESP	Protocol version response	←	1	0xC8
DTLS_COMPLETE_RESP	DTLS completed response	←	1	0xE2
RCT_IP_RESP	RCT IP parameter response	←	1	0xE3
TRANSPARENT_RESP	Transparent response	← →	1	0xF0

The Message ID of any Response is the same as the Message ID of the corresponding Command, but with bit 7 set.

5.2.7 Message length

This is the length of the Message Data (excluding Message ID and Message Length). This field is used:

— in variable length messages (see for example 6.3.1 and 6.5.19) to check for the end of data;

— to be able to determine the start of an embedded reverse command (see 5.8).

Possible padding is never considered when calculating the value of message length field.

5.2.8 Sequence numbers

The sequence number is used to determine if a message is missing or duplicated. Both ends have a transmit sequence number and a receive sequence number.

These two counters exist at both ends (e.g. we are speaking about 4 counters in total), whereas the RX_Sequence counters are used to realize a “state-full machine” implementation.

These counters are used to fulfil three simultaneous functions:

a) Initially, both the SPT and RCT choose their TX_seqs to be a random number, then they use it as a datagram counter, incrementing them for each sent datagram by one. The RX_seqs are the expected next TX_seqs from the other communication end-point. That is: If one did see “42” as the last TX_seq coming in from the communication partner, oneself would send out “43” as next RX_seq. As the other end does this in the same style, the TX_seq and RX_seq function as a mutual sequence control mechanism.

b) Second, they can simultaneously function as a resend-mechanism: If one detected that one missed a datagram (because for example, the incoming TX_seq is “44”, but one expected TX_seq = 43) or the one got is corrupt (by checking the hash), one just resends the own old previously sent last datagram and the other side will see by the old TX_seq that one wants to get a re-transmission.

c) Being chosen randomly and being part of the encrypted data block, they rule out replay attacks.

For each connection, every message shall be acknowledged before the next new (not retransmission) message may be transmitted.

5.2.9 Flags

The following flags are defined:

Table 7 — Flags

Byte	Bit	Definition
0	**0**	**Reverse command included in response:** **– value 0 = no reverse command included,** **– value 1 = reverse command included**
0	**1…7**	**Reserved**
1	**0…7**	**Reserved**

5.3 Padding and message length

5.3.1 General

Padding is required for the following two reasons:

— create a message length which is a multiple of the block length of the encryption algorithm as used;

— make poll and alarm messages look alike.

Padding is done using random or pseudo-random data. Random bytes are appended to the actual messages data until the total message length is one of those as specified in the next clause.

5.3.2 Message length

The message lengths as used fulfil the requirements as mentioned in 5.4.2 (using a 16 or 32 byte block length), and are a compromise between obfuscation of alarm events and bandwidth usage.

This results message lengths that are a multiple of 128 + 4 bytes for the Connection Handle:

– 132 bytes (4 bytes Connection Handle + 8 × 16 bytes);

– 260 bytes (4 bytes Connection Handle + 16 × 16 bytes);

– etc.

5.4 Hashing

5.4.1 General

The following methods of message validation are supported:

Table 8 — Hashing ID's

Hash ID	Description	Hash size in bytes
0	SHA-256	32
1	RIPEMD-256	32

RCTs shall implement all methods. However, it is permissible to configure a RCT not to accept all hash methods.

SPTs shall at least implement the default method, but can implement all methods.

The default method is 0 (SHA-256) until explicitly updated using the messages as defined in 6.5.11 and 6.5.12.

The hashing method to be used is negotiated during session initialization, using the messages as defined in 6.5.11 and 6.5.12.

The selectable hashing method allows for an upgrade of security in the future while maintaining backwards compatibility.

The hash is included in the encrypted part of the message.

Faults processing is defined for following conditions.

5.4.2 Invalid hash – transmitter response

In the event of invalid hash the transmitter shall work in same way as if no response was received.

5.4.3 Invalid hash - receiver response

In the event of invalid hash the receiver shall ignore the message.

5.5 Encryption

5.5.1 General

Except for the Connection Handle, the entire message is encrypted. The encryption method to be used has been negotiated during Commissioning. The following methods are supported: