IEC TS 60839-7-8

Edition 1.0 2019-05

TECHNICAL SPECIFICATION

Alarm systems –
Part 7-8: Message formats and protocols for serial data interfaces in alarm transmission systems – Requirements for common protocol for alarm transmission using the Internet protocol

IEC TS 60839-7-8:2019-05(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC TS 60839-7-8

Edition 1.0 2019-05

TECHNICAL SPECIFICATION

Alarm systems – Part 7-8: Message formats and protocols for serial data interfaces in alarm transmission systems – Requirements for common protocol for alarm transmission using the Internet protocol

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 13.320

ISBN 978-2-8322-6813-1

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

ALARM SYSTEMS –

Part 7-8: Message formats and protocols for serial data interfaces in alarm transmission systems – Requirements for common protocol for alarm transmission using the Internet protocol

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

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

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards.

IEC 60839-7-8, which is a technical specification, has been prepared by IEC technical committee 79: Alarm and electronic security systems.

The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
79/419/DTS	79/453A/RVDTS

Full information on the voting for the approval of this technical specification can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 60839 series, published under the general title *Alarm systems*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- transformed into an International Standard,
- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

ALARM SYSTEMS –

Part 7-8: Message formats and protocols for serial data interfaces in alarm transmission systems – Requirements for common protocol for alarm transmission using the Internet protocol

1 Scope

This Part of IEC 60839 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 IEC 60839-5-1.

The performance characteristics of the supervised premises equipment comply with the requirements of its associated alarm system standard and apply for transmission of all types of alarms including, but not limited to, fire, intrusion, access control and social alarms.

Compliance with this document is voluntary.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60839-5-1:2014, *Alarm and electronic security systems – Part 5-1: Alarm transmission systems – General requirements*

RFC 793:1981, *Internet standard – Transmission control protocol, DARPA Internet program, protocol specification*

NIST 800-38A:2001, *Recommendation for block cipher modes of operation: methods and techniques*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60839-5-1 apply.

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

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.2 Abbreviations

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

AES	Advanced Encryption Standard
ARC	Alarm Receiving Centre
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 object 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 IEC 62642-1 are met.

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

NOTE For further discussion of IP and UDP in alarm transmission, please see F.3.

5 Messaging

5.1 General

This clause defines the messaging layer, on top of which the alarm event data is transmitted using the existing reporting formats like 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 the RESP_CMD_NOT_SUPPORTED result value, which the receiving party can send as answer to unsupported messages.

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

5.2.2 Identifiers

The identifiers given in Table 1 below exist.

Table 1 – 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	Contributing to hashes in all messages	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. The Device ID is used when computing the hash value for each message. In combination with the encryption of the hash this is used for substitution detection.

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 in Table 2 below only to clarify the hash value calculation.

Table 2 – Basic unencrypted format of messages

Byte index	Bytes	Description	See	Group
0	4	Connection handle	5.2.4	Header
4	16	Device ID	5.2.5	
20	2	Tx Sequence number	5.2.8	
22	2	Rx Sequence number	5.2.8	
24	2	Flags	5.2.9	
26	1	Protocol version number	5.7	
27	1	Message ID	5.2.6	Message
28	2	Message length	5.2.7	
30	n	Message data	Clause 6	

The basic encrypted, transmitted format of all messages is as shown in Table 3. Note that the Device ID field is not included in the encrypted message, but its value is used to compute the message hash value i.e. the hash is calculated from the unencrypted version of the message described above.

Table 3 – 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	
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	Message
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 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 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 not a (by the installer/operator) configurable parameter, nor made visible on user interfaces. It is generated and 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 and RCT. It is used (in combination with the encryption) for substitution detection. Both SPT and RCT can verify the identity of the connected party using this field, and create a substitution alarm in case it has changed.

Within the message header, the Device ID itself is never transmitted. However Device ID is used to contribute to the message hash calculation.

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 clear text, but is needed to be known in clear text for the ARC to configure the RCT accordingly.

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

Uniqueness is assured 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 an 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.

5.2.5.3 RCT device ID

The device ID of the RCT is an ID that is unique within the receiver and never changes within the lifetime of a receiver. It represents the unique identity of the RCT.

The RCT device ID is made available to the SPT during the commissioning phase.

5.2.6 Message ID

The message IDs as used are listed in the following Table 4.

Table 4 – 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_REQ	Path supervision request	← →	1	0x45
SET_TIME_CMD	Set time command	←	1	0x47
VERSION_REQ	Protocol version request	→	1	0x48
PMTU_REQ	P-MTU	→	1	0x60
PMTU_PROBE	P-MTU probe	→	1	0x61
DTLS_COMPLETE_REQ	DTLS completed request	→	1	0x62
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_RESP	Encryption selection response	←	1	0xC2
ENCRYPT_KEY_RESP	Encryption key exchange response	← →	1	0xC3
HASH_SELECT_RESP	Hash selection response	←	1	0xC4
PATH_SUPERVISION_RESP	Path supervision response	← →	1	0xC5
SET_TIME_RESP	Set time response	→	1	0xC7
VERSION_RESP	Protocol version response	←	1	0xC8
PMTU_RESP	P-MTU response	←	1	0xE0
PMTU_PROBE_RESP	P-MTU probe response	←	1	0xE1
DTLS_COMPLETE_RESP	DTLS completed response	←	1	0xE2
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 6.4.18) 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 the 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) Both SPT and RCT choose their TX_seqs to be a random number which is used as a datagram counter, incrementing them for each sent datagram. The RX_seqs are the expected next TX_seqs from the other communication end-point. If one did see "42" as the last TX_seq coming in from the other communication end-point, one would send out "43" as next RX_seq. As the other end-point does this in the same style, the TX_seq and RX_seq operate as a mutual sequence control mechanism.

b) Second, they can simultaneously operate as a resend-mechanism: If it is detected that a datagram is missing (because for example, the incoming TX_seq is "44", but TX_seq = 43 was expected) or the received datagram is corrupted (by checking the hash), the correct old previously sent last datagram is resent by one communication end-point and the other communication end-point will see by the old TX_seq that a re-transmission is requested.

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

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

5.2.9 Flags

The flags given in Table 5 are defined.

Table 5 – 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 Padding

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 5.3.1 (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

The methods of message validation given in Table 6 are supported.

Table 6 – Hashing IDs

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

RCTs have to 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 6.4.10 and 6.4.11.

The hashing method to be used is negotiated during session initialization, using the messages as defined in 6.4.10 and 6.4.11.

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.

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 methods given in Table 7 are supported.

Table 7 – Encryption IDs

Encryption ID	Description
0	Unencrypted May only be used for debugging purposes or in test environments.
1	AES-128
2	AES-256

RCTs have to implement all methods. SPTs shall at least implement the default method, but can implement all methods. The default method is 2 (AES-256) until explicitly updated using the messages as defined in 6.4.6 and 6.4.7.

The encryption key is valid only for one connection between an SPT and the RCT, e.g. the RCT shall keep track of all different keys as used by the SPTs connected to it.

The operation mode to be used with AES is CBC (Cipher Block Chaining) as specified in NIST Special Publication 800-38A (2001 edition). The IV (Initialization Vector) is all zeros.

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

The sole purpose of the non-encrypted mode is for implementation ease (the messaging layer can be implemented without encryption in place, and only once this is ready one can add the encryption).

5.5.2 Key exchange

The lifetime of a key is determined by the number of transmitted packets. To ensure security, key updates are triggered regularly by the RCT every n successfully transmitted packets (using the RCT's sequence counter as reference), with n being a value which is sent from the RCT to the SPT during the initial commissioning phase.

To enforce security, a key exchange is to be triggered by the RCT at least once a week or at least every 2^{16} = 65 536 successful packets (whichever comes first).

In addition to that regular pattern, both RCT and SPT can invoke additional key exchanges.

To avoid RCT and SPT getting out of synchronisation when an alarm message is triggered exactly in between an on-going session key exchange action, the RCT shall maintain the old session key until the first successful transmission of a packet with the new session key is acknowledged.

5.6 Timeouts and retries

The timeouts (after which a message will be retried) will increase with each retry as defined in RFC 793.

In addition to RFC 793, the resulting time-out value is upper-bound by the reporting time of the ATP plus/minus an evenly randomly distributed time offset of 10 %.

NOTE RFC 793 defines a learning algorithm, which tries to adapt to the available network capacity. To do so, it tries to calculate a best-guess of the network's round-trip-delay time, consisting of 90 % the time of the previously used time-out value plus 10 % the round-trip-delay time of the last packet. Times a (safety) factor of 2, this value is used as the next time-out value.

The intention is to adapt to the congestion state of the network: The more the network is congested, the larger the timeout value grows, trying to avoid a flooding of the RCT in case of a network congestion.

To avoid too long a delay of a retry, this principle is upper-bound by a maximum time-out value.

Especially in case of an invent which could still lead to all SPTs trying to re-send to their RCT in parallel, the upper limit defined by the reporting time of the ATP is changed by an evenly distributed random component.

The random component shall be based on a (pseudo)random number generator which assures randomly distributed outputs from all SPTs, even if they generate the value at the same moment of time, e.g. by taking the SPT's device ID into the random number calculation.

5.7 Version number

The version number in the message header is an unsigned numerical byte value, indicating the version of the protocol actually being used.

It defaults to "1", representing the first version of this protocol implementation. SPT and RCT shall mutually agree upon the protocol version to be used during the commissioning phase. The RCT may be configured to require a specified set of protocol versions and to refuse to communicate using other versions.

5.8 Reverse commands

To allow for an RCT to send commands to an SPT without depending on properties of the network environment in between (e.g. any forwarding- or adopted firewall rules, especially on the side of the SPTs networking equipment), a mechanism for packing reverse commands into response messages is implemented.

The approach taken is to 'piggy-pack' an embedded reverse command in the response message. This is indicated by the flag in the header of the response message (see 5.2.9).

The message ID and the message data will be added to the message as shown as Table 8.

Table 8 – Reverse commands

Byte index	Bytes	Description	What
0	HL	Header, 'Reverse command'-flag set to 1	Header
HL	1	Message ID	Response message
HL + 1	2	Message length of the response data	
HL + 3	n	Response message data	
HL + 3 + n	1	Message ID	Embedded reverse command message
HL + 4 + n	2	Message length of the reverse command	
HL + 6 + n	m	Command message data	
HL + 6 + n + m		Padding	Tail
		Hash	

The message length of the response data shall be used to determine the start position of the embedded reverse command message.

It is still possible for an RCT to send commands asynchronously (without waiting for a poll), however, depending on the network environment this command may not reach the SPT.

5.9 Initial values

The values given in Table 9 are used by the protocol until the variables are explicitly set by the corresponding configuration messages.

Table 9 – Initial values

What	Value	Description
Connection handle	0 / Number	Not set yet (DTLS) or shared secret
Hash	0	SHA-256
Encryption ID	2	AES-256
Heartbeat interval time	0	No polling
TX sequence counter	random	Starts with random number
RX sequence counter	0	No packet received yet

6 Message types

6.1 General

This clause defines the messages as used in this protocol. Note that the examples show only the message data; header, message ID and message length are not shown in the message overviews.

6.2 Path supervision

6.2.1 General

This clause describes the format of the poll message and its reply. A configuration message is used to negotiate the poll rate during commissioning. This configuration message is described in 6.4.12. The poll message itself does not include the heartbeat interval time.

Path supervision works on heartbeat traffic from the SPT to the RCT.

Any other message can implicitly function as poll message, e.g. the polling device can reset its ‘poll interval’ timer upon sending any message, and the poll monitoring device can reset its ‘timeout’ timer upon reception of any valid message from the other end.

6.2.2 Poll message

The poll message has the format shown in Table 10.

SPT ← → RCT

Table 10 – Poll message SPT ← → RCT

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL		Padding
		Hash

This message is sent by the polling device in case no messages have been sent for the heartbeat interval time as negotiated by the path supervision request/response messages (6.4.12/6.4.13) during connection setup.

6.2.3 Poll response

The poll response message has the format shown in Table 11.

RCT ← → SPT

Table 11 – Poll response RCT ← → SPT

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code [a]
		Padding
		Hash

[a] Result code can be:
RESP_ACKNOWLEDGE
RESP_POLL_REESTABLISH_CONNECTION

6.3 Event reporting

6.3.1 Event message format

6.3.1.1 General

The (alarm) event message shall always contain the actual event data. Next to this mandatory information the protocol provides the option to transmit additional information. To maintain the link between event and additional data, this data is all transmitted within one message.

To achieve this, the event message is divided into fields, each accompanied by their own length indicator.

Rationale:

– fields like ‘link’ are variable length, hence the ‘length’-bytes;

– to maintain a uniform format no distinction has been made between variable and fixed length fields.

The alarm event message has the format shown in Table 12.

SPT → RCT

Table 12 – Event message format – SPT → RCT

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Field identifier
HL + 1	2	Field length (L1)
HL + 3	L1	Field data
HL + 3 + L1	1	2nd field identifier (optional)
HL + 4 + L1	2	2nd field length (L2) (optional)
HL + 6 + L1	L2	2nd field data (optional) ... etc...
HL + 6 + L1 + L2		Padding
		Hash

The field length (L1, L2, ...) is the length of the field data (excluding field identifier and field length bytes).

The fields shown in Table 13 are defined.

Table 13 – Event message format – Fields

Field number	Description
0x00	Event field
0x01	Time event field
0x02	Time message field
0x80	Link field: IP address
0x81	Link field: IP port
0x82	Link field: URL
0x83	Link field: Filename

Field numbers above 0x80 provide a link to out-of-band additional information, like for example:

– pictures accompanying the event (IP address and port number, filename);

– audio or video streams

that are transmitted via a secondary channel. Note that the time fields can also be used to match events with the accompanying data.

These fields are explained in the next subclauses.

6.3.1.2 Event field

This field is mandatory for SPT and RCT (see Table 14):

Table 14 – Event field

Relative Byte Index	Bytes	Description
0	1	Protocol identifier: (See Annex B for definition and message layout)
1	L	Event data, for example: <SIA Account Block><SIA Event Block><SIA ASCII Block>

6.3.1.3 Time event field

This field is optional for SPT and mandatory for RCT (see Table 15):

Table 15 – Time event field

Relative Byte Index	Bytes	Description
0	8	Time format according to RFC 958 (NTP) / RFC 4330 (SNTP V4)

This field holds the timestamp on which the event occurred.

Time format is a 64 bit integer as described in RFC 958 (NTP) / RFC 4330 (SNTP V4), allowing easy local synchronization. Note that NTP basically uses a 32 bit counter of seconds since 1.January.1900, so a wrap-around will occur in 2036. Due to a 136 years "precision" in guessing the correct date (either 1900, 2036, 2172, ..) suffices to re-sync for the next 136 years. This should be easily handled by the devices, but shall be taken care by a special test-case during compliance test.

This approach is independent from daylight-saving zones and independent from time-zones, as NTP returns time based on UTC, so cross-country evaluations will be easier. Such local time adoptions against UTC (e.g. displaying time / entering time in human readable format) are thus left to the end-devices.

6.3.1.4 Time message field

This field is optional for SPT and mandatory for RCT (see Table 16):

Table 16 – Time message field

Relative Byte Index	Bytes	Description
0	8	Time format according to RFC 958 (NTP) / RFC 4330 (SNTP V4)

This field holds the timestamp on which the event message is transmitted by the SPT.

This value is to be used for life-time checking of the datagrams, i.e. harden the protocol against attackers in the sense that a datagram is accepted as being valid only if it arrived at the communication partner's end within a reasonable time (e.g. 51 h).

In addition, the difference Time event – Time message values give rises to check whether the alarm system fulfils the over-all maximum round-trip-delay times.

6.3.1.5 Link field – IP address

This field is optional for SPT and RCT (see Table 17):

Table 17 – Link field – IP address

Relative Byte Index	Bytes	Description

0	L	IP address:
		L=4 → IPv4 address
		L=32 → IPv6 address

This field defines the IP address to which the additional info will be sent to.

6.3.1.6 Link field – IP port number

This field is optional for SPT and RCT (see Table 18):

Table 18 – Link field – IP port number

Relative Byte Index	Bytes	Description
0	2	Port number

This field defines the port number to which the additional info will be sent to.

6.3.1.7 Link field – URL

This field is optional for SPT and RCT (see Table 19):

Table 19 – Link field – URL

Relative Byte Index	Bytes	Description
0	L	URL

This field defines the URL to which the additional info will be sent to.

6.3.1.8 Link field – Filename

This field is optional for SPT and RCT (see Table 20):

Table 20 – Link field – Filename

Relative Byte Index	Bytes	Description
0	L	Filename

The filename can be used for example to identify files uploaded to a TFTP server.

6.3.2 Event response format

The event response message has the following format:

RCT → SPT. See Table 21.

Table 21 – Event response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code [a]
HL + 1		Padding
		Hash

[a] Result code can be:
RESP_ACKNOWLEDGE
RESP_NEGATIVE_ACKNOWLEDGE
RESP_EVENT_RCT_COULD_NOT_PROCESS_MESSAGE
RESP_EVENT_PROTOCOL_ID_NOT_SUPPORTED
RESP_EVENT_ACKNOWLEDGE_UNKNOWN_FIELD.

In case the SPT includes optional fields in the event message that are not supported by the RCT, the event will still be acknowledged, but with a RESP_ACKNOWLEDGE_UNKNOWN_FIELD. This is a valid acknowledge, there is no need to resend the event.

6.4 Configuration messages

6.4.1 General

This clause describes the contents of the configuration messages. For the message flow and further explanation see Clause 7.

The configuration messages are used for both commissioning methods (DTLS and 'out-of-band'), as the messaging protocol needs the same parameters independently of how the connection was established.

Most configurable parameters are unique in the SPT for each RCT it reports to, e.g.

- connection handle;
- device ID;
- encryption selection;
- session key;
- hash;
- path supervision.

In case the SPT reports to 2 RCTs, there will be 2 instances of each parameter, one for each connected RCT.

In case in the SPT the parameters of the RCT to which it shall connect are changed (e.g. change to another RCT), the SPT shall request new ones.

Other parameters (e.g. time) are one value only that is used by the SPT for all RCTs it reports to.

6.4.2 Connection handle request

The connection handle request message has the following format:

SPT → RCT. See Table 22.

Table 22 – Connection handle request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL		Padding
		Hash

This message is issued by the SPT to request a connection handle, which is a random number. The connection handle is created by the RCT instead of the SPT, as it has to be unique at the RCT, and the random generator of the RCT is usually of much better quality than the one of the SPTs. Both SPT and RCT use the same connection handle.

In case the connection is broken, a next session will have a newly generated (different) connection handle.

6.4.3 Connection handle response

The connection handle response message has the following format:

RCT → SPT. See Table 23.

Table 23 – Connection handle response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	2	Connection handle
HL + 2		Padding
		Hash

This message itself and previous messages have a connection handle with the value 0. The next message will be the first one with a valid connection handle field.

6.4.4 Device ID request

The device ID request message has the following format:

SPT → RCT. See Table 24.

Table 24 – Device ID request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Flags
HL + 1	16	Device ID
HL + 17		Padding
		Hash

This message is issued by the SPT to request a Device ID.

The following applies to allow for 2nd channel commissioning:

- when the direction and device ID flags are set, the SPT requests the RCT device ID, and stores this RCT device ID as received in the reply message in NVM;
- when the direction and device ID flags are cleared, the SPT pushes its own device ID to the RCT. See Table 25.

Table 25 – Device ID request flags

Bit	Description
0	Direction 0: Device ID push 1: Device ID request
1	Device ID 0: SPT Device ID 1: RCT Device ID
2..7	Unused

6.4.5 Device ID response

The device ID response message has the following format:

RCT → SPT. See Table 26.

Table 26 – Device ID response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	1	Flags
HL + 2	16	Device ID
HL + 18		Padding
		Hash

The next message will be the first one with a valid device ID field in the message header.

6.4.6 Encryption selection request

The Encryption selection request message has the following format:

SPT → RCT. See Table 27.

Table 27 – Encryption selection request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Flags
HL + 1	1	Encryption 1
HL + 2	1	Encryption 2 (optional) … etc …
		Padding
		Hash

This message is issued during commissioning by the SPT to indicate the encryption methods it supports – see Table 28. See 5.5 for possible encryption methods.

Table 28 – 'Master encryption selection request' flag

Bit	Description
0	Encryption selection 0: Session encryption selection request 1: Master encryption selection request
1..7	Unused

6.4.7 Encryption selection response

The encryption selection response message has the following format:

RCT → SPT. See Table 29.

Table 29 – Encryption selection response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Flags
HL + 1	1	Result code
HL + 2	1	Encryption method to be used
HL + 3		Padding
		Hash

The flags field holds the value 0.

6.4.8 Encryption key exchange request

The encryption key exchange request message has the following format:

SPT ← → RCT. See Tables 30 and 31.

Table 30 – Encryption key exchange request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Flags
HL + 1	L	Encryption key (typically 128 or 256 bits -> 16 or 32 bytes)
HL + 1 + L		Padding
		Hash

Table 31 – ‘Master key request’ flag

Bit	Description
0	Direction 0: Key push (RCT) 1: Key request (SPT)
1	Key request 0: Session key 1: Master key
2..7	Unused

This message is issued to request an encryption key update. Both SPT and RCT can request an encryption key update. When 'direction' flag is set (request) the encryption key field is 0. The ‘key request’ flag is used only during the commission phase to exchange the new master key.

New keys are created by the RCT instead of the SPT, as they shall be generated using a cryptographically strong random number generator, and the random number generator of the RCT is usually of much better quality than the one of the SPTs.

The RCT can push a new session key to the SPT by clearing the ‘direction’ flag. The new key is in the ‘encryption key’ field. The SPT will then acknowledge by replying back this key in the Encryption key exchange response message.

6.4.9 Encryption key exchange response

The Encryption key exchange response message has the following format:

SPT ← → RCT. See Table 32.

Table 32 – Encryption key exchange response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	1	Flags
HL + 2	L	Encryption key (typically 128 or 256 bits -> 16 or 32 bytes)
HL + 2 + L		Padding
		Hash

The new key will become effective immediately, e.g. the next message is encrypted using the new key (in case ‘encryption selection’ > 0). To overcome transmission errors the RCT shall keep the previous key until a next message has successfully been received, as backup.

6.4.10 Hash selection request

The hash selection request message has the following format:

SPT → RCT. See Table 33.

Table 33 – Hash selection request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Hash 1
HL + 1	1	Hash 2 (optional) … etc.…
		Padding
		Hash

This message is issued during commissioning by the SPT to indicate the hash functions it supports. See 5.4 for possible hash functions.

6.4.11 Hash selection response

The hash selection response message has the following format:

RCT → SPT. See Table 34.

Table 34 – Hash selection response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	1	Hash to be used
HL + 2		Padding
		Hash

This is the first message that uses the newly set hash. By default SHA-256 (value 0) is used as hash function.

6.4.12 Path supervision request

The path supervision request message has the following format:

SPT → RCT. See Table 35.

Table 35 – Path supervision request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	4	Heartbeat interval time (seconds)
HL + 4	1	Push (0) or pull (1)
HL + 5		Padding
		Hash

The heartbeat interval time specifies the time until the SPT will send the next heartbeat.

The push-pull option determines the polling device:

- 0: Push: the SPT sends the poll to the RCT;
- 1: Pull: the RCT sends the poll to the SPT, which allows for load balancing.

6.4.13 Path supervision response

The path supervision response message has the following format:

RCT → SPT. See Table 36.

Table 36 – Path supervision response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code [a]
HL + 1	4	Heartbeat interval time (s)
HL + 5	1	Push (0) or pull (1)
HL + 6		Padding
		Hash

[a] Result code can be:
RESP_ACKNOWLEDGE
RESP_POLL_TOO_SLOW

6.4.14 Set time command

The set time command message has the following format:

RCT → SPT. See Table 37.

Table 37 – Set time command message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	8	Time format according to RFC 958 (NTP) / RFC 4330 (SNTP V4)
HL + 8		Padding
		Hash

This command is optional. In case events are transmitted with timestamps this command can be sent by the RCT to synchronize.

6.4.15 Set time response

The set time response message has the following format:

SPT → RCT. See Table 38.

Table 38 – Set time response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1		Padding
		Hash

6.4.16 Protocol version request

The protocol version request message has the following format:

SPT → RCT. See Table 39.

Table 39 – Protocol version request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	First supported protocol version
HL + 1	1	Second supported protocol version (optional) … etc …
		Padding
		Hash

This message is issued during commissioning and connection setup by the SPT to indicate the protocol version it supports.

6.4.17 Protocol version response

The protocol version response message has the following format:

RCT → SPT. See Table 40.

Table 40 – Protocol version response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	1	Protocol version to be used
HL + 2		Padding
		Hash

6.4.18 Transparent message

The transparent message has the format shown in Table 41.

Table 41 – Transparent message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	L	Transparent data
HL + L		Padding
		Hash

This message allows for (vendor specific) data to be transmitted between SPT and RCT. It can for example be used for configuration data or firmware uploads.

6.4.19 Transparent response

The transparent response has the format shown in Table 42.

Table 42 – Transparent response format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1	L	Transparent data
HL + 1 + L		Padding
		Hash

6.4.20 DTLS completed request

The DTLS completed request message has the following format:

SPT → RCT. See Table 43.

Table 43 – DTLS completed request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL		Padding
		Hash

This message is sent by the SPT to request the end of the DTLS session.

This message does not contain additional info.

6.4.21 DTLS completed response

The DTLS completed response message has the following format:

RCT → SPT. See Table 44.

Table 44 – DTLS completed response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL	1	Result code
HL + 1		Padding
		Hash

This message is send by the RCT as response to the DTLS completed request message.

This is sent by the RCT to end the parameter negotiation. After this is sent by the RCT and received by the SPT, the DTLS session is closed, all resources used by the session are freed and further communication between the RCT and SPT is done using the negotiated parameters.

6.4.22 RCT IP parameter request

The RCT parameter request message has the following format:

SPT → RCT. See Table 45.

Table 45 – RCT IP parameter request message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL		Padding
		Hash

If the SPT is to communicate either using a different port number for commissioning and 'normal' session traffic, or if separate commissioning and session RCTs are used, or if the SPT is to communicate with more than one RCT, then the RCT can send the IP address(es) and port(s) to be used for the session. It is the responsibility of the commissioning RCT to securely pass the session parameters to any other RCTs to which the SPT may have to communicate. The mechanism by the RCTs share the session parameters is vendor specific and outside the scope of this protocol document.

Implementation of this message is optional for the SPT.

6.4.23 RCT IP parameter response

The RCT IP parameter response message has the following format:

RCT → SPT. See Table 46.

Table 46 – RCT IP parameter response message format

Byte index	Bytes	Description
0	HL	Header, message ID and message length
HL + 1	1	Result code
HL + 2	1	Field identifier – RCT 1 IP address – see 6.3.1.5
HL + 3	2	Field length (L1)
HL + 5	L1	Field data
HL + 5 + L1	1	Field identifier – RCT 1 port number – see 6.3.1.6
HL + 6 + L1	2	Field length (L1)
HL + 8 + L1	L2	Field data
HL + 8 + L1 + L2	1	2nd field identifier (optional) – RCT 2 IP address – see 6.3.1.5
HL + 9 + L1 + L2	2	2nd field Length (L2) (optional)
HL + 11 + L1	L3	2nd field Data (optional) ... etc...
HL + 8 + L1 + L2 + L3		Padding
		Hash

7 Commissioning and connection setup

7.1 Commissioning

7.1.1 General

The objective of the commissioning procedure is to enable the supervised premises transceiver and the receiving centre transceiver to mutually authenticate each other.

Further, the commissioning procedure is used to negotiate the parameters:

– connection handle;
– device IDs of SPT and RCT;
– master encryption key;
– master encryption selection;
– (optional) RCT IP address(es) and port(s) with which the SPT should communicate (this allows for a separate 'commissioning server' to handle the 'initial contact' for multiple receivers. In this situation the commissioning server will have to securely transfer the session parameters to the appropriate RCT. The mechanism for doing this is outside the scope of this protocol).

A successful commissioning procedure establishes a communication session with a connection handle as unique identifier. The communication session lasts until a re-commissioning takes place. Especially, the change of session keys does not have an impact upon the communication session, i.e. it does not lead to any change in the connection handle.

7.1.2 Procedures

There are two options for obtaining the ‘master set’. Either:

– generated using a ‘shared secret’ passed out-of-band, or
– using X.509 certificates and DTLS in both RCT and SPT (optionally) (see 7.1.5)

Irrespective of the mechanism used to obtain it, the master key is then used to encrypt, using AES256, the exchange of the other parameters. It is also used (by the 'running' protocol) to establish the session key(s).

The master key is a 256 bit key.

7.1.3 Commissioning message sequence

The ‘master set’ is exchanged using the message flow as described below. The messages are the same, irrespective of the commissioning procedure in use. The difference is in the method in which the messages are secured, either using the ‘shared secret’ (‘one-time-pad’ key and device ID) as provided by the RCT, or using X.509/DTLS.

The message flow during the commissioning of a new SPT is as shown in Table 47.

Table 47 – Message flow during the commissioning of a new SPT

SPT	Direction	RCT	Remarks
VERSION_REQ	→		
	←	VERSION_RESP	
CONN_HANDLE_REQ	→		
	←	CONN_HANDLE_RESP	New connection handle generated by RCT, and stored to NVM
DEVICE_ID_REQ	→		SPT device ID
	←	DEVICE_ID_RESP	
DEVICE_ID_REQ	→		RCT device ID
	←	DEVICE_ID_RESP	
ENCRYPT_SELECT_REQ	→		
	←	ENCRYPT_SELECT_RESP	

SPT	Direction	RCT	Remarks
ENCRYPT_KEY_REQ	→		
	←	ENCRYPT_KEY_RESP	
			Key update complete, proceed using new encryption key and method
DTLS_COMPLETE_REQ	→		Only when using X.509/DTLS
	←	DTLS_COMPLETE_RESP	

The resulting master parameters are stored in NVM on both SPT and RCT.

Note that initially some fields in the header will be uninitialized until the matching ***configuration*** message is processed. Therefore it is essential that the IP address of the SPT does not change during this commissioning phase (it should remain constant throughout the exchange even if the secured premises have the most restrictive stateful firewall).

A detailed overview can be found in D.1.

The next step is to request the session parameters as specified in 7.2.

7.1.4 Commissioning using shared secret

7.1.4.1 General

Support for the shared secret procedure for generating the master key is mandatory in both RCTs and SPTs.

For this procedure, the RCT will generate a shared secret which consists of the connection handle and the encryption key.

Shared secret consists of:

– the 4 byte connection handle;

– the 32 byte (AES-256) encryption key.

For the commissioning stage AES-256 is mandatory. On request of the SPT (performance) this can be changed to AES-128 for normal communication.

The parameters will be used only for the exchange of the master key. Once the master has been successfully sent from the RCT to SPT, the session will be deleted and never re-used.

Next, these parameters are renewed, and stored into non-volatile memory as the new ‘master set’. This new ‘master set’ will be used to reconnect after disconnections or power failures.

7.1.4.2 Transferring the shared secret via out-of-band channel

The security of the out-of-band channel is one of the factors that determine the security of the pairing process between SPT and RCT. As the out-of-band channel is very likely to rely on human operator at one or both sides it should also be simple to implement and tolerant of human error. The following requirements are applicable:

– the shared secret shall be generated by the management system of the ATS, which may or may not be operated by an ARC. The processing power of the management system typically exceeds that of the SPT by orders of magnitude and therefore can generate shared secret of better cryptographic quality (randomness) than a small embedded system. In addition the ATS service provider or ARC has a guarantee that the shared secret generation process is compliant with these requirements;

– physical and logical means of shared secret transfer to the SPT shall make it difficult for a third party to intercept it without being detected. The word difficult means expensive in terms of time or resources in comparison to the gain the attacker may obtain by knowing the shared secret. The following methods may be considered appropriate depending on the security level of protected premises:
  - ARC operator dictates the shared secret to the field technician over the phone;
  - the shared secret is transmitted using SMS;
  - the shared secret is sent in an encrypted and signed e-mail;
  - the shared secret is printed at the management centre / ARC and the field technician brings it to the protected premises himself;
  - the shared secret is programmed into SPT at the management centre / ARC and then transported to the protected premises;
  - the shared secret is obtained by the field technician from a secured web site of the ATS service provider;
  - any other method meeting the difficulty criterion;

    It is the responsibility of the ATS service provider / ARC to judge the security of the method it uses to transfer the shared secret vs. the security level of the protected premises.
– the shared secret shall not be sent over a channel which is used for communication between the SPT and RCT for alarm reporting and monitoring;
– the shared secret shall be generated using cryptographically strong random number generator (see RFC 4086);
– to cope with potential typos and other human typical transmission errors, the text representation of the shared secret is extended by a 16 bit checksum, calculated as CRC as described in C.2, directly appended to the shared secret string.

7.1.5 Commissioning using X.509 certificates and DTLS

Support for the X.509 mechanism and DTLS is optional for SPTs and mandatory for RCTs.

The authentication, cipher selection and key exchange are performed using the DTLS protocol with the SPT as client and RCT as server. DTLS is a variation of TLS, which defines the base messages and formats. The connection handle and optional parameters are set using the cypher and session key negotiated.

The cipher suite TLS_DHE_DSS_WITH_AES_256_CBC_SHA shall be used and the master key is the 256 bit AES symmetric key created by the DTLS handshake.

RCT requirements:

– each RCT shall hold the certificates for every CA which has signed a certificate for any SPT which can potentially connect to the RCT;
– RCT shall provide mechanism to add new CA certificates to the system to allow SPT from a new manufacturer to be connected to the system, as well as a mechanism to delete CA certificates from the system. The details of the insertion/removal of the certificate is outside the scope of this document;
– while the DTLS implementation in the RCT may support other cipher suites, only TLS_DHE_DSS_WITH_AES_256_CBC_SHA shall be used for generating the master key.

SPT requirements:

- the SPT shall hold the certificates of the CAs which have signed the certificates for the RCTs to which the SPT may potentially connect. It is not mandatory for the SPT to validate the authenticity of the RCT but it is recommended that it do so;
- the common name of SPTs X.509 certificate shall be in the format “supplier identifier: supplier specific identifier“. Registered Internet domain name of the supplier is used as the supplier ID. This shall uniquely identify the SPT;
- the SPTs X.509 certificate shall be signed by a CA which is known to all the RCTs to which it could potentially connect;
- the SPT shall only present the cipher suite TLS_DHE_DSS_WITH_AES_256_CBC_SHA to be used in the DTLS handshake.

On completion of the parameter negotiation, the DTLS session is terminated, contexts etc., freed and all further communication takes place using the negotiated parameters.

7.2 Connection setup

In case of a reconnect, the ‘master set’ as negotiated during commissioning will be initially be used for encryption and authentication the messages between SPT and RCT. The first steps are to request new session parameters that are then used for further communication.

Typically connections are permanent 24/7, in case a connection breaks the SPT will attempt to re-establish the connection.

During the connection setup stage, the following parameters are set in the order per below:

- protocol version level mutually agreed by SPT and RCT;
- encryption selection;
- session key;
- hash;
- path supervision.

The message flow during connection setup (to request the session parameters) is as shown in Table 48.

Table 48 – Message flow during connection setup

SPT	Direction	RCT	Remarks
			The hash to start with is the Internet checksum
VERSION_REQ	→		SPT protocol version
	←	VERSION_RESP	RCT protocol version The highest protocol version supported by both SPT and RCT shall be used from now on. Only features supported by agreed protocol version shall be used.
ENCRYPT_SELECT_REQ	→		
	←	ENCRYPT_SELECT_RESP	
ENCRYPT_KEY_REQ	→		Session key
	←	ENCRYPT_KEY_RESP	
HASH_SELECT_REQ	→		
	←	HASH_SELECT_RESP	

SPT	Direction	RCT	Remarks
PATH_SUPERVISION_REQ	→		
	←	PATH_SUPERVISION_RESP	
			Connection setup is now complete, IP address is allowed to change after this point. It may/will take some time before the next (poll) message is transmitted.
POLL_MSG	→		First poll send after the poll interval.
	←	POLL_RESP	

For further details refer to the example in D.2.

Annex A
(normative)

Result codes

The result codes are listed in the following table – see Table A.1.

Table A.1 – Result codes

Bytes	Response to	Value
RESP_ACKNOWLEDGE	All	0x00
RESP_NEGATIVE_ACKNOWLEDGE	All	0x01
RESP_EVENT_RCT_COULD_NOT_PROCESS_MESSAGE	Event messages	0x10
RESP_EVENT_PROTOCOL_ID_NOT_SUPPORTED	Event messages	0x11
RESP_EVENT_ACKNOWLEDGE_UNKNOWN_FIELD	Event messages	0x12
RESP_POLL_TOO_SLOW	Path supervision request	0x20
RESP_POLL_REESTABLISH_CONNECTION	Poll messages	0x21
RESP_CMD_NOT_SUPPORTED	Commands	0x30
RESP_DEVICE_ID_UNKNOWN	Device ID request	0x31
RESP_UNKNOWN	All	0xFF

Annex B
(normative)

Protocol identifiers

The following Table B.1 summarizes the possible protocol identifiers for application layer protocol carried by the protocol defined in this document.

Each compatible implementation of this protocol shall support at least two types of messaging:

- transparent messages for serially connected AE and / or AS;
- Sia DC-03 message structures for AS signals connected by pin inputs and for messages generated internally by SPT and / or RCT.

Table B.1 – Protocol identifiers

Protocol ID	Protocol
01	Sia DC-03 messages as described in SIA DC-03-1990.01(R2003.10), Chapter 5 and Annex A
02	Ademco Contact ID
03	Scancom FF
04	VdS 2465
05	CEI ABI 79 5/6
06	SurGard
07	F1COM
08	SOS Access v4
…	
254	Manufacturer specific
255	Transparent, transmitting serially received content in the data field

A manufacturer wishing to send messages that do not fit any of the listed application protocols shall use protocol identifier 254. Any currently unallocated protocol identifier may be allocated in a later revision of this document.

Annex C
(normative)

Shared secret

C.1 Formatting of the shared secret

When encoding and formatting the shared secret into a string format, readable for human beings, it shall be represented in ASCII hexadecimal format, in Network byte order. E.g. the characters as used are {'0',..,'9'} + {'A',..,'F'}.

Formatting of the key value to improve the readability for humans shall use one of the explicitly named separator symbols {‘-‘} or {space}.

NOTE 1 During formatting, the separator symbols can be freely used to improve readability (e.g. grouping in four character blocks, each block separated by hyphens from each other); during decoding, the occurrence of separator symbols inside of the key string is ignored completely.

NOTE 2 Lower case letters are treated identical to upper case letters, i.e. Lower/upper case transmission problems (like spelling the key string by voice over a telephone line) will lead to a valid decoding of the key.

NOTE 3 Each part of the shared secret has a checksum appended.

Example of encryption key (256-bit with CRC) as part of the shared secret:

363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-0689

Example of connection handle (with CRC) as part of the shared secret:

7D30-FA26-8238

C.2 Checksum for shared secret formatting

CRC-16-CCITT checksums are used to detect possible errors in shared secrets before they are used. This clause provides examples of the checksum procedure.

The CRC-16-CCITT calculation is defined by the following parameters:

– Polynomial:	0x1021
– Initial crc value:	0xffff

C.3 Example of secret encoding and formatting

Example encoding and formatting

Step 1: Create random key

```
secret key k =  0x36 3e 2b 16 8d bb 5a 95 7d 5f 2b f4 25 a4 5d 7c
                24 e3 c1 b9 2f 4b a0 13 ee 6a d9 b2 3f 91 f5 63
                (in hex, to see byte order representation)
```

Step 2: Calculate CRC

CRC16(k) = 0x4A97 (hexadecimal)

Step 3: Present key in ascii hexadecimal format, (optionally) use separators to improve readability:

363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-24E3-C1B9-2F4B-A013-EE6A-D9B2-3F91-F563

Step 4: Append encoding of CRC16(k) to k, optionally using separators:

363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-24E3-C1B9-2F4B-A013-EE6A-D9B2-3F91-F563-4A97

Annex D
(informative)

Examples of messaging sequences

D.1 Commissioning

The following Table D.1 demonstrates the commissioning messaging sequence.

Devices example:

– SPT device ID	88998899
– RCT device ID	66776677

Shared secret example:

– One-time-key	12341111
– One-time-connection-handle	56781111

The example device IDs, keys and handles as shown above are illustrative only, and do not represent the actual format of these parameters.

Table D.1 – Commissioning messaging sequence

SPT	Dir	RCT	Example message data	Remarks	Conn Handle	TX Seq	RX Seq	Key	Encrypt	Device ID SPT	Device ID RCT
				The hash to start with is SHA-256							
VERSION_ REQ	→		First supported protocol version: 1	TX sequence number randomly chosen by SPT RX sequence number is not known yet	56781111	42	0	12341111	AES-256	0	0
	←	VERSION_ RESP	Result code: RESP_ACKNOWLEDGE Version: 1		56781111	17	43	12341111	AES-256	0	0