DEUTSCHE NORM *Entwurf* Mai 2010

DIN EN 50136-1-7 (VDE 0830-5-1-7)

DIN

VDE

Diese Norm ist zugleich eine **VDE-Bestimmung** im Sinne von VDE 0022. Sie ist nach Durchführung des vom VDE-Präsidium beschlossenen Genehmigungsverfahrens unter der oben angeführten Nummer in das VDE-Vorschriftenwerk aufgenommen und in der „etz Elektrotechnik + Automation“ bekannt gegeben worden.

ICS 13.320

Einsprüche bis 2010-07-31

Entwurf

Alarmanlagen – Alarmübertragungsanlagen und -einrichtungen – Teil 1-7: Anforderungen an standardisierte Protokolle zur Alarmübertragung in Paketvermittlungsnetzwerken; Deutsche Fassung prEN 50136-1-7:2010

Alarm systems –
Alarm transmission systems and equipment –
Part 1-7: Requirements for common protocol for alarm transmission using packet switched network;
German version prEN 50136-1-7:2010

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2010-05-25 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

– vorzugsweise als Datei per E-Mail an **dke@vde.com** in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter **www.dke.de/stellungnahme** abgerufen werden
– oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevante Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 95 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Beginn der Gültigkeit

Diese Norm gilt ab ...

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments prEN 50136-1-7:2010 „Alarmanlagen – Alarmübertragungsanlagen und -einrichtungen – Teil 1-7: Anforderungen an standardisierte Protokolle zur Alarmübertragung in Paketvermittlungsnetzwerken“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und der Französischen Fassung abgeglichen ist, ist die englische Originalfassung der prEN 50136-1-7:2010 beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen Text.

Das europäische Dokument prEN 50136-1-7:2010 „Alarm systems – Alarm transmission systems and equipment – Part 1-7: Requirements for common protocol for alarm transmission using packet switched network“ wurde vom TC 79 „Alarmanlagen“ des Europäischen Komitees für Elektrotechnische Normung (CENELEC) erarbeitet und von CENELEC den Nationalen Komitees zur Stellungnahme vorgelegt.

Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen späteren „Schluss-Entwurf“ prEN nur 2 Monate beträgt und zum „Schluss-Entwurf“ prEN keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der „Entwurf“ prEN als Deutscher Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit noch vor der formellen Abstimmung berücksichtigen zu können.

Für diesen Norm-Entwurf ist das nationale Arbeitsgremium UK 713.1 „Gefahrenmelde- und Überwachungsanlagen“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Nationaler Anhang NA
(informativ)

Zusammenhang mit Europäischen und Internationalen Normen

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Eine Information über den Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ist in Tabelle NA.1 wiedergegeben.

Tabelle NA.1

Europäische Norm	Internationale Norm	Deutsche Norm	Klassifikation im VDE-Vorschriftenwerk
EN 50136-1-5	–	DIN EN 50136-1-5 (VDE 0830-5-1-5)	VDE 0830-5-1-5
CLC/TS 50136-7	–	DIN CLC/TS 50136-7 (VDE V 0830-5-7)	VDE V 0830-5-7
ETSI TS 100 900 V7.2.0 (1999-07)	–	–	–
ITU-T Rec. X.509	–	–	–

Nationaler Anhang NB
(informativ)

Literaturhinweise

DIN EN 50136-1-5 (VDE 0830-5-1-5), *Alarmanlagen – Alarmübertragungsanlagen und -einrichtungen – Teil 1-5: Anforderungen an ein paketvermittelndes Netz (Packet Switched Network PSN)*

DIN CLC/TS 50136-7 (VDE V 0830-5-7), *Alarmanlagen – Alarmübertragungsanlagen und -einrichtungen – Teil 7: Anwendungsregeln*

– Leerseite –

Entwurf
pr EN 50136-1-7

EUROPÄISCHE NORM

EUROPEAN STANDARD

NORME EUROPÉENNE

März 2010

ICS 13.320; 33.040.40

Deutsche Fassung

Alarmanlagen – Alarmübertragungsanlagen und -einrichtungen – Teil 1-7: Anforderungen an standardisierte Protokolle zur Alarmübertragung in Paketvermittlungsnetzwerken

Alarm systems –
Alarm transmission systems and equipment –
Part 1-7: Requirements for common protocol for alarm transmission using packet switched network

Dieser Europäische Norm-Entwurf wird den CENELEC-Mitgliedern zur CENELEC-Umfrage vorgelegt.

CENELEC Termin: 2010-09-03.

Er wurde von CLC/TC 79 erstellt.

Wenn aus diesem Norm-Entwurf eine Europäische Norm wird, sind die CENELEC-Mitglieder gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Dieser Europäische Norm-Entwurf wurde von CENELEC in drei offiziellen Fassungen (Deutsch, Englisch, Französisch) erstellt. Eine Fassung in einer anderen Sprache, die von einem CENELEC-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CENELEC-Mitglieder sind die nationalen elektrotechnischen Komitees von Belgien, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, der Schweiz, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, Ungarn, dem Vereinigten Königreich und Zypern.

Warnvermerk: Dieses Schriftstück hat noch nicht den Status einer Europäischen Norm. Es wird zur Prüfung und Stellungnahme vorgelegt. Es kann sich noch ohne Ankündigung ändern und darf nicht als Europäische Norm in Bezug genommen werden.

CENELEC

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

Zentralsekretariat: Avenue Marnix 17, B-1000 Brüssel

Projekt: 22700 Ref. Nr. prEN 50136-1-7:2010 D

Vorwort

Dieser Entwurf einer Europäischen Norm wurde vom Technischen Komitee CENELEC/TC 79 „Alarmanlagen“ ausgearbeitet. Er wird der formellen Abstimmung unterworfen.

Inhalt

Seite

1 Anwendungsbereich ... 5
2 Normative Verweisungen ... 5
3 Begriffe und Abkürzungen ... 5
3.1 Begriffe ... 5
3.2 Abkürzungen ... 5
4 Zweck ... 6
5 Meldungsübermittlung ... 6
5.1 Übersicht über die Meldungsformate ... 7
5.2 Auffüllen und Länge der Meldung ... 11
5.3 Hash-Verfahren ... 12
5.4 Verschlüsselung ... 12
5.5 Timeouts und Neuanfragen ... 13
5.6 Versionsnummer ... 13
5.7 Umkehrbefehle ... 13
5.8 Anfangswerte ... 14
6 Meldungstypen ... 14
6.1 Wegüberwachung ... 14
6.2 Ereignismeldungen ... 15
6.3 Konfigurationsmeldungen ... 19
7 Inbetriebnahme und Verbindungsaufbau ... 27
7.1 Inbetriebnahme ... 27
7.2 Einrichten der Verbindung ... 30
Anhang A (normativ) Ergebniscodes ... 33
Anhang B (normativ) Protokollbezeichner ... 34
Anhang C (normativ) Shared Secret ... 35
C.1 Zeichensatz für Verschlüsselung, Formatierung und Entschlüsselung ... 35
C.2 Prüfsumme für die Verschlüsselung, Formatierung und Entschlüsselung des Shared Secret ... 38
C.3 Beispiel für die Verschlüsselung, Formatierung und Entschlüsselung des Secret ... 38
C.4 Bespiel für die Entschlüsselung ... 39
Anhang D (informativ) Beispiele von Anwendungsprotokollen ... 40
D.1 Sia ... 40
D.2 Ademco Contact ID ... 40
D.3 Scancom Fast Format ... 41
Anhang E (informativ) Entwurfsgrundsätze ... 42
E.1 Informationssicherheit ... 42
E.2 Anwendung der UDP-Signalgebung ... 42
Literaturhinweise ... 43

Tabellen

Tabelle 1 – Bezeichner ... 7
Tabelle 2 – Grundformat aller unverschlüsselten Meldungen ... 8
Tabelle 3 – Grundformat aller verschlüsselten Meldungen ... 8
Tabelle 4 – Übersicht über die Message ID ... 10
Tabelle 5 – Flags ... 11
Tabelle 6 – Kennungen für Hash-Verfahren ... 12
Tabelle 7 – Verschlüsselungskennungen ... 12

Seite

Tabelle 8 – Umkehrbefehle ... 14
Tabelle 9 – Anfangswerte ... 14
Tabelle 10 – Abfragemeldung SPT ← → RCT ... 15
Tabelle 11 – Abfrageantwort RCT ← → SPT ... 15
Tabelle 12 – Format der Ereignismeldung – SPT → RCT ... 16
Tabelle 13 – Format der Ereignismeldung – Felder ... 16
Tabelle 14 – Ereignisfeld ... 17
Tabelle 15 – Zeitereignisfeld ... 17
Tabelle 16 – Zeitmeldungsfeld ... 17
Tabelle 17 – Verbindungsfeld – IP-Adresse ... 18
Tabelle 18 – Verbindungsfeld – IP-Portnummer ... 18
Tabelle 19 – Verbindungsfeld – URL ... 18
Tabelle 20 – Verbindungsfeld – Dateiname ... 18
Tabelle 21 – Format der Ereignisantwortmeldung ... 19
Tabelle 22 – Format der Anforderungsmeldung für den Verbindungsidentifikator ... 20
Tabelle 23 – Format der Antwortmeldung für den Verbindungsidentifikator ... 20
Tabelle 24 – Format der Meldung zur Anforderung einer Geräte-ID ... 20
Tabelle 25 – Flag ‚Master Device ID request' ... 21
Tabelle 26 – Format der Meldung der Antwort der Geräte-ID ... 21
Tabelle 27 – Format der Meldung zur Anforderung der Verschlüsselungsauswahl ... 21
Tabelle 28 – Flag ‚Master Encryption Selection request' ... 22
Tabelle 29 – Format der Antwortmeldung der Verschlüsselungsauswahl ... 22
Tabelle 30 – Format der Anforderungsmeldung für den Verschlüsselungscodetausch ... 22
Tabelle 31 – Flag ‚Master Key Request' ... 22
Tabelle 32 – Format der Antwortmeldung des Verschlüsselungscodeaustausches ... 23
Tabelle 33 – Format der Anforderungsmeldung zur Hashauswahl ... 23
Tabelle 34 – Format der Antwortmeldung der Hashauswahl ... 24
Tabelle 35 – Format der Anforderungsmeldung zur Wegüberwachung ... 24
Tabelle 36 – Format der Antwortmeldung der Wegüberwachung ... 24
Tabelle 37 – Format der Befehlsmeldung zur Zeiteinstellung ... 25
Tabelle 38 – Format der Antwortmeldung der Zeiteinstellung ... 25
Tabelle 39 – Format der transparente Meldung ... 25
Tabelle 40 – Format der transparente Antwort ... 25
Tabelle 41 – Format der Meldung zur Anforderung des Datagram-Verschlüsselungsprotokolls ... 26
Tabelle 42 – Format der Antwortmeldung des Datagram-Verschlüsselungsprotokolls ... 26
Tabelle 43 – Format der Anforderungsmeldung zur RCT-Parameter ... 26
Tabelle 44 – Format der Antwort der RCT-Parameter ... 27
Tabelle 45 – Ablauf der Meldungen während der Inbetriebnahme einer neuen SPT ... 28
Tabelle 46 – Fluss der Meldungen während des Einrichtens der Verbindung ... 31
Tabelle A.1 – Ergebniscodes ... 33
Tabelle B.1 – Protokollbezeichner ... 34
Tabelle C.1 – Zeichensatz ... 35

1 Anwendungsbereich

Diese Europäische Norm legt ein Protokoll für die Punkt-zu-Punkt-Übertragung von Alarmen und Störungen sowie die Überwachung der Kommunikation zwischen einer Übertragungseinrichtung und einer Empfangszentrale unter Anwendung des Internet-Protokolls (IP) fest.

Dieses Protokoll soll in jedem Netz angewendet werden, welches die Übertragung von IP-Daten unterstützt. Dazu gehören Ethernet, xDSL, GPRS, WiFi, UMTS und WIMAX.

Die System-Leistungsanforderungen an die Alarmübertragung sind in EN 50136-1 und EN 50136-1-5 festgelegt.

Die Leistungsanforderungen der Einrichtungen im überwachten Objekt müssen die Anforderungen der jeweiligen Alarmanlagen-Norm erfüllen und sie müssen für die Übertragung aller Alarmarten anwendbar sein, einschließlich, jedoch nicht beschränkt auf Brand, Einbruch, Zugangskontrolle und Personen-Hilferufe.

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

EN 50136-1:201X [1)], *Alarmanlagen – Alarmübertragungsanlagen – Teil 1: Allgemeine Anforderungen an Alarmübertragungsanlagen*

EN 50136-1-5, *Alarmanlagen – Alarmübertragungsanlagen und –einrichtungen – Teil 1-5: Anforderungen an ein paketvermittelndes Netz (Packet Switched Network PSN)*

NIST SP 800-38A, *Recommendation for Block Cipher Modes of Operation – Methods and Techniques*
NIST Special Publication 800-38A, December 2001
Available from http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

RFC793, *Transmission Control Protocol*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc793.txt.pdf

RFC958, *Network Time Protocol (NTP)*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc958.txt.pdf

RFC4330, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4330.txt.pdf

SIA DC-03-1990.01 (R2003.10), *DCS SIA Format Standard*

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach EN 50136-1:201X.

3.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Abkürzungen.

AES	hochentwickelter Verschlüsselungsstandard	(en: advanced encryption standard)
ARC	Alarmempfangsstelle	(en: alarm receiving centre)
ATS	Alarmübertragungsanlage	(en: alarm transmission system)

[1)] Im Entwurfsstadium.

CA	X.509-Zertifizierungsstelle	(en: X.509 certificate autority)
CBC	Blockchiffrenkettung	(en: cipher block chaining)
CRC	zyklische Redundanzprüfung	(en: cyclic redundancy check)
DNS	Domänen-Namensystem	(en: domain name system)
DTLS	Datagram-Verschlüsselungsprotokoll	(en: datagram transport layer security)
HL	Kopfdatenlänge	(en: header length)
IP	Internet-Protokoll	(en: internet protocol)
IV	Initialisierungsvektor	(en: initialization vector)
MAC	Medienzugriffskontrolle	(en: media access control)
MTU	Maximale Übertragungseinheit	(en: maximum transmission unit)
NAT	Adressenumsetzung für Netze	(en: network address translation)
NIST	National Institute of Standards and Technology	
NTP	Netzwerk-Zeitsynchronisationsprotokoll	(en: network time protocol)
NVM	nichtflüchtiger Speicher	(en: non-volatile memory)
P-MTU	Pfad-MTU	(en: path maximum transmission unit)
RCT	Empfangszentrale	(en: receiver centre transceiver)
RX	Empfangen	(en: receive)
SCTP	verbindungsorientiertes Transportprotokoll	(en: stream control transmission protocol)
SNTP	vereinfachtes Protokoll zur Synchronisierung von Systemzeiten vernetzter Rechner	(en: simple network time protocol)
SPT	Übertragungseinrichtung	(en: supervised premises transceiver)
TFTP	einfaches Dateiübertragungsprotokoll	(en: trivial file transfer protocol)
TX	Senden	(en: transmit)
UDP	Anwender-Datagramprotokoll	(en: user datagram protocol)
URI	einheitlicher Bezeichner für Ressourcen	(en: uniform resource identifier)
URL	einheitlicher Quellenanzeiger	(en: uniform resource locator)
UTC	koordinierte Weltzeit	(en: coordinated universal time)
WS	Fenstergröße	(en: window size)

4 Zweck

Der Zweck dieser Europäischen Norm ist die Festlegung von Einzelheiten der Protokolle (Transport- und Anwendungsschicht) für Alarmübertragungsanlagen, die das Internet-Protokoll (IP) nutzen, um damit die Fähigkeit zur Zusammenarbeit der von verschiedenen Herstellern gelieferten SPTs und RCTs sicherzustellen. Weiterhin werden die Mechanismen der Inbetriebnahme von SPT und RCT und der Aufbau eines gegenseitigen Vertrauensverhältnisses zwischen den beteiligten Parteien beschrieben.

Unter der Voraussetzung, dass die in EN 50136-1 enthaltenen Anforderungen erfüllt werden, dürfen auch alle weiteren und nicht in dieser Europäischen Norm behandelten Alarmübertragungsprotokolle oder -einrichtungen zum Einsatz kommen.

Dieses Protokoll ist für die Ausführung von UDP konzipiert und unterstützt IPv4 und IPv6.

5 Meldungsübermittlung

In diesem Abschnitt wird die Schicht der Meldungsübermittlung definiert, auf der die Daten des Alarmereignisses mit den vorhandenen Berichtsformaten, wie z. B. Sia und Contact ID, übertragen werden.

In Abschnitt 7 wird die erste Inbetriebnahme einer SPT sowie die Herstellung der Verbindung zwischen SPTs und der RCT festgelegt.

Die Funktionalität des Alarmübermittlungs- und Abfrageprotokolls umfasst:

– Austausch von Haupt- und Sitzungsparametern;
– (Alarm-)Ereignismeldung (einschließlich Verbindung zu ergänzenden Out-of-band-Daten, die mit den Ereignissen in Beziehung stehen, wie Audio/Video);
– Leitungsüberwachung;
– transparente Meldungsübertragung, z. B. verkäuferspezifische Meldungen, die beispielsweise für Fernsteuerbefehle von der RCT zur SPT benutzt werden können.

Sie erfüllen die folgenden Anforderungen:

– Verschlüsselung, erfüllt die Anforderungen der höchsten Sicherheitsklasse nach EN 50136-1;
– Authentifikation, erfüllt die Anforderungen der höchsten Sicherheitsklasse nach EN 50136-1;
– SPT: lässt ein breites Spektrum an Hardware zu (begrenzte Forderungen an Speicherplatzbedarf und CPU-Leistung);
– RCT: unterstützt gleichzeitig bis zu 10 000 SPTs nach EN 50136-1, Klasse D3, unter Anwendung moderner universeller Server-Hardware;
– erlaubt dynamische IP-Adressen der SPTs;
– lässt die Anordnung von einem oder mehr SPTs hinter einer NAT-Firewall zu.

5.1 Übersicht über die Meldungsformate

In diesem Unterabschnitt wird die grundsätzliche Gliederung aller Meldungen beschrieben.

Jede Meldung muss explizit bestätigt werden, einschließlich der Leitungsüberwachungsmeldungen.

Die Abwärtskompatibilität wird durch Implementierung des Ergebniswertes RESP_CMD_NOT_SUPPORTED erreicht, den die empfangende Seite als Antwort auf nicht unterstützte Meldungen senden kann.

Mehrbyte-Werte werden in der Netz-Bytereihenfolge übertragen (Big-Endian).

5.1.1 Bezeichner

Die folgenden Bezeichner existieren:

Tabelle 1 – Bezeichner

Beschreibung	Zweck	Vorhanden in	Verschlüsselt	Siehe
Verbindungsidentifikator (Connection Handle)	Suche des aktuellen symmetrischen Verschlüsselungsschlüssels	Alle Meldungen	Nein	5.1.3
Geräte-ID (Device ID)	Eindeutige Identifikation der Hardware	Trägt zu den Hashwerten in allen Meldungen bei	N/A	5.1.4

Der Verbindungsidentifikator ist unverschlüsselt. Es ist eine zufällig gewählte Zahl, die während der Einrichtung der Verbindung initialisiert wird. Ihr alleiniger Zweck besteht darin, nach dem Verschlüsselungscode suchen zu können. Sie ist nur für die Kommunikationssitzung gültig.

Sobald die Verbindung hergestellt ist, identifiziert die Geräte-ID eindeutig die Hardware (dies ist kein Konto-Code). Die RCT muss die Geräte-ID als Index für ihre SPT-Datenbank verwenden. Die Geräte-ID wird bei der Berechnung des Hashwertes für jede Meldung verwendet. In Kombination mit der Verschlüsselung des Hashwertes wird dies für die Substitutionserkennung verwendet.

Die Geräte-ID muss in einem nichtflüchtigen Speicher abgelegt werden.

Die IP-Adresse wird nicht für Identifikationszwecke verwendet, um damit dynamische oder umgesetzte IP-Adressen zuzulassen.

5.1.2 Meldungsformat

Im Folgenden wird das unverschlüsselte Grundformat aller Meldungen beschrieben. In diesem Format wird keine Meldung übertragen. Die Beschreibung In diesem Unterabschnitt dient nur dazu, die Berechnung des Hashwertes zu erklären.

Tabelle 2 – Grundformat aller unverschlüsselten Meldungen

Byte-Index	Bytes	Beschreibung	Siehe	Gruppe
0	4	Verbindungsidentifikator	5.1.3	Kopfdaten
4	16	Geräte-ID	5.1.4	
20	2	Tx Folgenummer	5.1.7	
22	2	Rx Folgenummer	5.1.7	
24	2	Flags	5.1.8	
26	1	Protokollversionsnummer		
27	1	Meldung-ID	5.1.5	Meldung
28	2	Länge der Meldung	5.1.6	
30	n	Daten der Meldung	Abschnitt 6	

Im Folgenden wird das verschlüsselte übertragene Grundformat aller Meldungen beschrieben. Es ist zu beachten, dass das Feld Geräte-ID nicht in der verschlüsselten Meldung enthalten ist, dessen Wert jedoch zur Berechnung des Hashwertes verwendet wird, d. h. der Hashwert wird aus der unverschlüsselten Version der vorstehend beschriebenen Meldung berechnet.

Tabelle 3 – Grundformat aller verschlüsselten Meldungen

Byte-Index	Bytes	Beschreibung	Siehe	Verschlüsselt	Gruppe
0	4	Verbindungsidentifikator	5.1.3	Nein	Kopfdaten
4	2	Tx Folgenummer	5.1.7	Ja	
6	2	Rx Folgenummer	5.1.7	Ja	
8	2	Flags	5.1.8	Ja	
10	1	Protokollversionsnummer		Ja	
11	1	Meldung-ID	5.1.5	Ja	Meldung
12	2	Länge der Meldung	5.1.6	Ja	
14	n	Daten der Meldung	Abschnitt 6	Ja	
14 + n		Füllbytes	5.2.1	Ja	Abschlussdaten
	32 32	Hash – SHA-256 oder Hash – RIPEMD-256	5.3	Ja	

Der Verbindungsidentifikator ist unverschlüsselt; das Übrige der Meldung ist nach dem Verschlüsselungsverfahren verschlüsselt, welches während der Inbetriebnahmephase vereinbart worden ist.

Die Meldung-IDs werden paarweise definiert; jede Meldung hat ihre passende Quittung. Für die Quittungen enthält das erste Datenbyte der Meldung einen „Ergebniscode“, der in Anhang A definiert ist.

In den folgenden Unterabschnitten werden alle Felder ausführlich beschrieben.

5.1.3 Verbindungsidentifikator

Der Verbindungsidentifikator wird (eindeutig für die RCT, an die eine SPT berichtet) mit dem Inbetriebnahmeprotokoll zugewiesen. Die RCT erzeugt einen eindeutigen Verbindungsidentifikator und verbindet ihn in ihrer internen Datenbank mit der Geräte-ID der SPT. Diese Umsetzung ergibt einen kompakten Verbindungsidentifikator mit fester Länge.

Der Zweck des Verbindungsidentifikators besteht darin, den Verschlüsselungscode bestimmen zu können, der – unabhängig von der IP-Adresse der Meldung – für die Entschlüsselung der empfangenen Meldung anzuwenden ist.

Der Verbindungsidentifikator ist kein (vom Errichter/Bediener) konfigurierbarer Parameter, und wird auch nicht an der Nutzerschnittstelle sichtbar gemacht. Sie wird erzeugt und intern nur von den SPT/RCT-Einrichtungen benutzt.

5.1.4 Geräte-ID

Die Geräte-ID dient der eindeutigen Kennzeichnung von SPT und RCT. Sie wird (in Kombination mit der Verschlüsselung) für die Substitutionserkennung verwendet. SPT und RCT können mit diesem Feld die Kennung des verbundenen Beteiligten überprüfen und in dem Fall, das sie verändert ist, einen Substitutionsalarm erzeugen.

Die Geräte-ID selbst wird niemals in den Kopfdaten der Meldung übertragen. Jedoch trägt die Geräte-ID zur Hashberechnung der Meldung bei.

Die Geräte-ID hat eine Länge von 16 Byte.

5.1.4.1 Geräte-ID der SPT

Die Geräte-ID der SPT ist eine ID, die für die SPT zufallsbedingt, jedoch über die Lebensdauer der SPT unveränderlich und schreibgeschützt ist, d. h. eine Hardware-Seriennummer. Sie kommt in der SPT-Datenbank in der RCT nur einmal vor.

Die Geräte-ID wird bei der Herstellung des Gerätes erzeugt; sie wird bei der Meldungsübermittlung nie selbst im Klartext übertragen, muss der ARC aber im Klartext bekannt sein, um die RCT entsprechend zu konfigurieren.

Sie wird folglich nur während der ersten Inbetriebnahmephase an die RCT übertragen.

Die Eindeutigkeit wird durch die folgenden Prinzipien gesichert:

– Jeder Hersteller muss seine aus 24 Bit bestehende „Eindeutige Herstellerkennung“ verwenden, die ihm von der IEEE für die Generierung der MAC-Adresse zugewiesen worden ist.

– Jeder SPT-Hersteller, der keinen derartigen Code besitzt, muss sich an die IEEE zur Bereitstellung eines derartigen Codes wenden.

– Wenn eine Schnittstelle in der SPT eine MAC-Adresse verwendet, müssen die nächsten 24 Bit in der Geräte-ID die gleichen wie der Rest der vom Hersteller festgelegten MAC-Adresse sein. Ist eine derartige Schnittstelle nicht vorhanden, muss der Hersteller eine anderes dokumentiertes Benummerungsschema verwenden.

– Für den Rest des Feldes der Geräte-ID muss der Hersteller nicht aufeinander folgende, zufallsverteilte Zahlen verwenden und für alle von ihm gelieferten SPT-Geräte die Eindeutigkeit zusichern.

5.1.4.2 Geräte-ID der RCT

Die Geräte-ID der RCT ist eine ID, die innerhalb des Empfängers nur einmal vorkommt und sich während der Lebensdauer eines Empfängers niemals verändert. Sie stellt eine Seriennummer des RCT dar, nur dass sie zu Beginn von der ARC eingerichtet werden kann.

Die Geräte-ID der RCT wird der SPT während der Inbetriebnahmephase zur Verfügung gestellt.

5.1.5 Meldung-ID

Die Meldung-IDs, wie sie zum Einsatz kommen, sind in der folgenden Tabelle aufgeführt.

Tabelle 4 – Übersicht über die Message ID

Name der Meldung	Beschreibung	Richtung SPT ← → RCT	Version	Meldung-ID
POLL_MSG	Abfragemeldung	→	1	0x11
EVENT_MSG	Ereignismeldung	→	1	0x30
CONN_HANDLE_REQ	Anforderung des Verbindungsidentifikators	→	1	0x40
DEVICE_ID_REQ	Anforderung der Geräte-ID	→	1	0x41
ENCRYPT_SELECT_REQ	Anforderung der Verschlüsselungsauswahl	→	1	0x42
ENCRYPT_KEY_REQ	Austausch des Verschlüsselungscodes	← →	1	0x43
HASH_SELECT_REQ	Anforderung der Hashauswahl	→	1	0x44
PATH_SUPERVISION_REQ	Anforderung der Wegüberwachung	→	1	0x45
SET_TIME_CMD	Befehl zur Zeiteinstellung	←	1	0x47
PMTU_REQ	Anforderung der P-MTU	→	1	0x60
PMTU_PROBE	Pfad-MTU-Probe	→	1	0x61
DTLS_COMPLETE_REQ	Anforderung des Datagram-Verschlüsselungs-protokolls	→	1	0x62
TRANSPARENT_MSG	Transparente Meldung	← →	1	0x70
POLL_RESP	Abfrageantwort	←	1	0x91
EVENT_RESP	Ereignisantwort	←	1	0xB0
CONN_HANDLE_RESP	Antwort des Verbindungsidentifikators	←	1	0xC0
DEVICE_ID_RESP	Antwort der Geräte-ID	←	1	0xC1
ENCRYPT_SELECT_RESP	Antwort der Verschlüsselungsauswahl	←	1	0xC2
ENCRYPT_KEY_RESP	Antwort des Verschlüsselungscodes	← →	1	0xC3
HASH_SELECT_RESP	Antwort der Hashauswahl	←	1	0xC4
PATH_SUPERVISION_RESP	Antwort der Wegüberwachung	←	1	0xC5
SET_TIME_RESP	Antwort der Zeiteinstellung	→	1	0xC7
PMTU_RESP	Antwort der Pfad-MTU	←	1	0xE0
PMTU_PROBE_RESP	Antwort der Pfad-MTU-Probe	←	1	0xE1
DTLS_COMPLETE_RESP	Antwort des Datagram-Verschlüsselungs-protokolls	←	1	0xE2
TRANSPARENT_RESP	Transparente Antwort	← →	1	0xF0

Die Meldung-ID jeder Antwort ist die gleiche Meldung-ID des entsprechenden Befehls, jedoch mit gesetztem Bit 7.

5.1.6 Länge der Meldung

Dies ist die Länge der Daten der Meldung (außer Meldung-ID und Länge der Meldung). Dieses Feld wird verwendet:

– bei variabler Länge der Meldung (siehe z. B. 6.2.1 und 6.3.15) zur Prüfung des Endes der Daten;

– und um den Anfang eines eingebetteten Umkehrbefehls bestimmen zu können (siehe 5.7).

Bei der Berechnung des Wertes des Feldes Länge der Meldung wird ein mögliches Auffüllen niemals in Betracht gezogen.

5.1.7 Folgenummern

Die Folgenummer wird zur Bestimmung einer fehlenden oder doppelten Meldung verwendet. Beide Enden haben eine Sende-Folgenummer und eine Empfangs-Folgenummer.

Diese beiden Zähler sind an beiden Enden vorhanden (d. h., wir sprechen insgesamt von 4 Zählern), während die RX_Folgenummern für die Realisierung einer „state-full machine“-Implementierung verwendet werden.

Diese Zähler werden für die gleichzeitige Erfüllung von drei Funktionen eingesetzt:

a) am Anfang wählen SPT und RCT ihre TX-seqs (Sende-Folgenummer) als eine Zufallszahl und verwenden sie anschließend als ein Datagram-Zähler, der für jedes gesendete Datagram um eins inkrementiert wird. Die RX_seqs (Empfangs-Folgenummer) sind die erwarteten nächsten TX_seqs vom anderen Kommunikationsendpunkt. Das heißt: Wenn einer eine „42“ als die letzte TX-seq „sieht“, die von einem Kommunikationspartner kommt, würde er selbst eine „43“ als nächste RX-seq senden. Da das andere Ende dies in der gleichen Weise ausführt, funktionieren TX-seq und RX-seq als ein gegenseitiger Folgesteuermechanismus;

b) als Zweites können sie gleichzeitig als ein Rücksendemechanismus funktionieren: wenn einer feststellt, dass einem ein Datagram fehlt (z. B. weil die ankommende TX-seq den Wert „44“ hat, obwohl eine TX-seq = 43 erwartet wird) oder das eine empfangene Datagram beschädigt ist (durch Überprüfen des Hashwertes), dann wird das eigene alte und vorher gesendete letzte Datagram zurückgesendet und die andere Seite wird anhand der alten TX-seq erkennen, dass eine nochmalige Übertragung gewünscht wird;

c) dadurch, dass sie zufällig ausgewählt wurden und Teil des verschlüsselten Datenblocks sind, schließen sie alle Replay-Angriffe aus.

Bei jeder Verbindung ist jede Meldung zu bestätigen, bevor die nächste neue (keine nochmalige Übertragung) Meldung übertragen werden kann.

5.1.8 Flags

Die folgenden Flags sind definiert:

Tabelle 5 – Flags

Byte	Bit	Definition
0	0	Umkehrbefehl vorhanden: – Wert 0 = kein Umkehrbefehl enthalten – Wert 1 = Umkehrbefehl enthalten
0	1 … 7	Reserviert
1	0 … 7	Reserviert

5.2 Auffüllen und Länge der Meldung

5.2.1 Auffüllen

Das Auffüllen ist aus den beiden folgenden Gründen erforderlich:

– Erzeugen einer Meldungslänge, die ein Vielfaches der Blocklänge des verwendeten Verschlüsselungsalgorithmus ist;

– Erzeugung von Abfrage- und Alarmmeldungen, die ähnlich sind.

Das Auffüllen erfolgt mit Zusatzdaten. Die Füllbytes werden den eigentlichen Daten der Meldung angehängt bis die Gesamtlänge der Meldung eine derjenigen ist, die im nächsten Unterabschnitt festgelegt sind.

5.2.2 Länge der Meldung

Die Längen der Meldungen, die zur Erfüllung der in 5.2.1 angeführten Anforderungen (bei Anwendung einer Blocklänge von 16 Byte oder 32 Byte) verwendet werden, sind ein Kompromiss zwischen der Entstellung von Alarmereignissen und Bandbreitennutzung.

Dies führt zu Längen der Meldungen, die ein Vielfaches von 128 + 4 Byte für den Verbindungsidentifikator sind:

– 132 Bytes (4 Byte Verbindungsidentifikator + 8 × 16 Byte);

– 260 Bytes (4 Byte Verbindungsidentifikator + 16 × 16 Byte);

– usw.

5.3 Hash-Verfahren

Für die Validierung der Meldung werden die folgenden Verfahren unterstützt:

Tabelle 6 – Kennungen für Hash-Verfahren

Hash-Kennung	Beschreibung	Hashgröße in Byte
0	SHA-256	32
1	RIPEMD-256	32

Die RCTs müssen alle Verfahren ausführen können. Es ist jedoch auch möglich, eine RCT so zu konfigurieren, dass sie nicht alle Hash-Verfahren akzeptiert.

Die SPTs müssen mindestens das voreingestellte Verfahren realisieren können, dürfen jedoch auch alle Verfahren ausführen.

Das voreingestellte Verfahren ist 0 (SHA-256), solange bis es ausdrücklich mit den Meldungen aktualisiert wird, die in 6.3.9 und 6.3.10 definiert sind.

Das anzuwendende Hash-Verfahren wird während der Initialisierung der Sitzung mit den in 6.3.9 und 6.3.10 definierten Meldungen vereinbart.

Das wählbare Hash-Verfahren lässt unter Beibehaltung der Abwärtskompatibilität die zukünftige qualitätsmäßige Verbesserung der Sicherheit zu.

Der Hash ist im verschlüsselten Teil der Meldung enthalten.

5.4 Verschlüsselung

5.4.1 Verschlüsselungsverfahren

Mit Ausnahme des Verbindungsidentifikators ist die gesamte Meldung verschlüsselt. Das anzuwendende Verschlüsselungsverfahren ist bei der Inbetriebnahme zu vereinbaren. Die folgenden Verfahren werden unterstützt:

Tabelle 7 – Verschlüsselungskennungen

Verschlüsselungskennungen	Beschreibung
0	Unverschlüsselt Darf nur für Zwecke der Fehlersuche oder in Testumgebungen verwendet werden.
1	AES-128
2	AES-256

Die RCTs müssen alle Verfahren ausführen können. Die SPTs müssen mindestens das voreingestellte Verfahren realisieren können, dürfen jedoch auch alle Verfahren ausführen. Das voreingestellte Verfahren ist 1 (AES-128), solange bis es ausdrücklich mit den Meldungen aktualisiert wird, die in 6.3.5 und 6.3.6 definiert sind.

Der Verschlüsselungscode ist nur für eine Verbindung zwischen SPT und RCT gültig, z. B. muss die RCT die Übersicht über alle unterschiedlichen Schlüssel behalten, die von den angeschlossenen SPTs benutzt werden.

Die bei AES anzuwendende Betriebsart ist CBC (Blockchiffrenkettung), wie sie in der NIST-Sonderpublikation 800-38A (Ausgabe 2001) festgelegt ist. Der IV (Initialisierungsvektor) besteht nur aus Nullen.

Das wählbare Verschlüsselungsverfahren lässt unter Beibehaltung der Abwärtskompatibilität die zukünftige qualitätsmäßige Verbesserung der Sicherheit zu.

Der alleinige Zweck der unverschlüsselten Betriebsart ist die Vereinfachung der Realisierung (die Schicht der Meldungsübermittlung kann ohne vorhandene Verschlüsselung realisiert werden, und wenn diese erst einmal hergestellt ist, kann die Verschlüsselung mit hinzugenommen werden).

5.4.2 Austausch des Schlüssels

Die Lebensdauer eines Schlüssels wird durch die Anzahl der übertragenen Pakete bestimmt. Für die Sicherstellung der Sicherheit werden Aktualisierungen des Schlüssels regelmäßig von der RCT nach jeweils N erfolgreich übermittelten Paketen ausgelöst (als Bezugswert wird der Folgezähler der RCT verwendet), dabei ist N ein Wert, der von der RCT während der ersten Inbetriebnahmephase an die SPT gesendet wird.

Zur Durchsetzung der Sicherheit ist ein Austausch des Schlüssels von der RCT mindestens einmal wöchentlich auszulösen oder nach mindestens 2^{16} = 65 536 erfolgreichen Paketen (der zuerst auftretende Wert gilt).

Zusätzlich zu diesem regelmäßigen Muster können RCT und SPT zusätzliche Austausche des Schlüssels in Anspruch nehmen.

Um zu vermeiden, das die Synchronisation zwischen RCT und SPT verloren geht, wenn eine Alarmmeldung exakt während einer laufenden Sitzung des Schlüsselaustausches ausgelöst wird, muss die RCT den alten Sitzungsschlüssel behalten, bis die erste erfolgreiche Übertragung eines Paketes mit dem neuen Sitzungsschlüssel bestätigt worden ist.

5.5 Timeouts und Neuanfragen

Die Timeouts (Zeitsperren) (nach denen eine Meldung neu angefragt wird) werden sich mit jeder Neuanfrage vergrößern, wie es in RFC793 festgelegt ist.

Zusätzlich zu RFC793 ist der sich ergebende Wert der Zeitsperre begrenzt durch einen absoluten Maximalwert von 100 s plus/minus eines gleichmäßig zufällig verteilten Zeitversatzes von 10 %.

ANMERKUNG RCF793 definiert einen Lernalgorithmus, der sich an die verfügbare Kapazität des Netzes anzupassen versucht. Dazu versucht er einen besten Schätzwert der Rundreiseverzögerungszeit zu berechnen, die aus 90 % des vorigen verwendeten Timeout-Wertes plus 10 % der Rundreiseverzögerungszeit des letzten Paketes besteht.

Es wird die Absicht verfolgt, sich an den Überlastzustand des Netzes anzunähern: je mehr das Netz überlastet wird, desto stärker steigt der Timeout-Wert an, und es wird versucht ein Überschwemmen des RCT im Falle der Netzüberlastung zu vermeiden.

Zur Vermeidung einer zu langen Verzögerung einer Neuanfrage ist dieses Prinzip nach unten durch einen maximalen Timeout-Wert begrenzt.

Besonders im Fall einer Erfindung, die noch an allen SPT dazu führen würde, ein paralleles wiederholtes Senden an ihre RCT zu versuchen, wird die obere Grenze von 100 s durch eine gleichmäßig verteilte Zufallskomponente verändert.

Die Zufallszahlenkomponente muss auf einem Zufallszahlengenerator beruhen, der zufällig verteilte Ausgangsgrößen von allen SPT sichert, selbst wenn sie den Wert zum selben Zeitpunkt generieren, z. B. durch Aufnahme der Geräte-ID der SPT in die Berechnung der Zufallszahl.

5.6 Versionsnummer

Die Versionsnummer in den Kopfdaten der Meldung ist ein vorzeichenloser numerischer Bytewert, der die Version des gegenwärtig benutzten Protokolls angibt.

Sie wird standardmäßig auf „1“ gesetzt und stellt die erste Version dieser Protokollimplementation dar. SPT und RCT müssen sich während der Inbetriebnahmephase gegenseitig auf das anzuwendende Protokoll einigen. Die RCT kann so konfiguriert sein, dass sie eine festgelegte Reihe von Protokollversionen fordert und die Kommunikation mit anderen Versionen verweigert.

5.7 Umkehrbefehle

Um es einer RCT zu ermöglichen, Befehle an eine SPT zu senden, ohne von den Eigenschaften der dazwischen liegenden Netzwerkumgebung abhängig zu sein (z. B. alle umlenkenden oder übernommenen Firewall-Regeln, besonders auf der Seite der Netzwerkumgebung der SPT), wird ein Mechanismus für das Einpacken von Umkehrbefehlen in Antwortmeldungen realisiert.

Der gewählte Ansatz besteht darin, einen eingebetteten Umkehrbefehl der Antwortmeldung zusätzlich aufzuladen („Huckepack“). Dies wird durch ein Flag in den Kopfdaten der Antwortmeldung angegeben (siehe 5.1.8).

Die Meldungs-ID und die Daten der Meldung werden der Meldung wie folgt hinzugefügt:

Tabelle 8 – Umkehrbefehle

Byte-Index	Byte	Beschreibung	Was
0	HL	Kopfdaten, „Umkehrbefehl“-Flag auf 1 gesetzt	Kopfdaten
HL	1	Meldung-ID	Antwortmeldung
HL + 1	2	Meldung Länge der Antwortdaten	
HL + 3	n	Daten der Antwortmeldung	
HL + 3 + n	1	Meldung-ID	Eingebettete Umkehrbefehl-Meldung
HL + 4 + n	2	Meldung Länge des Umkehrbefehls	
HL + 6 + n	m	Daten des Umkehrbefehls	
HL + 6 + n + m		Auffüllen	Abschlussdaten
		Hash	

Die Länge der Meldung muss innerhalb der Antwortmeldung liegen, die zur Bestimmung der Anfangsposition der Meldung des eingebetteten Umkehrbefehls verwendet wird.

Es ist für eine RCT immer noch möglich, Befehle asynchron (ohne warten auf eine zyklische Abfrage) zu senden, jedoch wird der Befehl die SPT in Abhängigkeit von der Netzwerkumgebung möglicherweise nicht erreichen.

5.8 Anfangswerte

für das Protokoll werden die folgenden Werte verwendet, bis die Variablen durch die entsprechenden Konfigurationsmeldungen ausdrücklich gesetzt werden.

Tabelle 9 – Anfangswerte

Was	Wert	Beschreibung
Verbindungsidentifikator	0	Noch nicht gesetzt
Hash	0	SHA-256
Verschlüsselungs-ID	1	AES-128
Heartbeat-Intervallzeit	0	Keine zyklische Abfrage
Sendefolgezähler	Zufällig	
Empfangsfolgezähler	0	Bisher kein Paket empfangen

6 Meldungstypen

In diesem Abschnitt werden die Meldungen definiert, wie sie in diesem Protokoll verwendet werden. Es ist zu beachten, dass die Beispiele nur die Daten der Meldungen aufzeigen; Kopfdaten, Meldung-ID und Länge der Meldung werden in den Übersichten über Meldungen nicht dargestellt.

6.1 Wegüberwachung

In diesem Unterabschnitt wird das Format der Abfragemeldung und ihrer Antwort beschrieben. Die Vereinbarung der Abfragerate erfolgt während der Inbetriebnahme mit einer Konfigurationsmeldung. Diese Konfigurationsmeldung ist in 6.3.11 beschrieben. Die zyklische Abfragemeldung selbst enthält nicht das Heartbeat-Zeitintervall (Kontrollsignal-Zeitintervall).

Die Wegüberwachung arbeitet auf dem Heartbeat-Verkehr von der SPT zur RCT.

Jede weitere Meldung kann bedingungslos als Abfragemeldung arbeiten, z. B. kann das abfragende Gerät seinen Zeitgeber „Abfrageintervall“ bei Aussendung einer Meldung zurücksetzen und das Abfrageüberwachungsgerät kann seinen Zeitgeber „Timeout“ bei Empfang einer gültigen Meldung vom anderen Ende zurücksetzen.

6.1.1 Abfragemeldung

Die Abfragemeldung hat das folgende Format:

SPT ← → RCT

Tabelle 10 – Abfragemeldung SPT ← → RCT

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL		Auffüllen
		Hash

Diese Meldung wird vom abfragenden Gerät in dem Fall gesendet, wenn keine Meldungen für die Heartbeat-Intervallzeit gesendet wurden, wie es mit den Anforderungs-/Antwortmeldungen der Wegüberwachung (6.3.11/6.3.12) während des Einrichtens der Verbindung vereinbart worden ist.

6.1.2 Abfrageantwort

Die Abfrageantwort-Meldung hat das folgende Format:

RCT ← → SPT

Tabelle 11 – Abfrageantwort RCT ← → SPT

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode [a)]
		Auffüllen
		Hash
[a)] Der Ergebniscode kann sein: RESP_ACKNOWLEDGE RESP_POLL_REESTABLISH_CONNECTION		

6.2 Ereignismeldungen

6.2.1 Format der Ereignismeldung

Die (Alarm-)Ereignismeldung muss stets die eigentlichen Ereignisdaten enthalten. Neben dieser vorgeschriebenen Information enthält das Protokoll die Möglichkeit, zusätzliche Informationen zu übertragen. Damit der Zusammenhang zwischen den Ereignis- und Zusatzdaten erhalten bleibt, werden die Daten gemeinsam in einer Meldung übertragen.

Zu diesem Zweck wird die Ereignismeldung in fünf Felder aufgeteilt, zu denen jeweils die eigene Längenkennung gehört.

Begründung:

– Felder, wie „Verbindung“ sind von variabler Länge, deshalb die „Längen“-Bytes;

– für die Beibehaltung eines einheitlichen Formates wird keine Unterscheidung zwischen variablen und festen Längenfeldern vorgenommen.

Die Alarmereignismeldung hat das folgende Format:

SPT → RCT

Tabelle 12 – Format der Ereignismeldung – SPT → RCT

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Bezeichner des Feldes
HL + 1	2	Länge des Feldes (L1)
HL + 3	L1	Daten des Feldes
HL + 3 + L1	1	2. Bezeichner des Feldes (freigestellt)
HL + 4 + L1	2	2. Länge des Feldes (L2) (freigestellt)
HL + 6 + L1	L2	2. Daten des Feldes (freigestellt) … usw.
HL + 6 + L1 + L2		Auffüllen
		Hash

Die Länge des Feldes (L1, L2 …) ist die Länge der Daten des Feldes (außer den Bytes für Bezeichner des Feldes und Länge des Feldes).

Die folgenden Felder sind definiert:

Tabelle 13 – Format der Ereignismeldung – Felder

Feldnummer	Beschreibung
0x00	Ereignisfeld
0x01	Zeitereignisfeld
0x02	Zeitmeldungsfeld
0x80	Verbindungsfeld: IP-Adresse
0x81	Verbindungsfeld: IP-Port
0x82	Verbindungsfeld: URL
0x83	Verbindungsfeld: Dateiname

Feldnummern über 0x80 stellen eine Verbindung zu ergänzenden Out-of-band-Informationen bereit, wie beispielsweise

– Bilder, die das Ereignis begleiten (IP-Adresse und -Portnummer, Dateiname);

– Audio- oder Videodatenstrom,

die über einen Sekundärkanal übertragen werden. Es ist zu beachten, dass die Zeitfelder auch dazu benutzt werden können, um den Ereignissen die begleitenden Daten zuzuordnen.

Diese Felder werden in den nächsten Unterabschnitten erläutert.

6.2.1.1 Ereignisfeld

SPT: Vorgeschrieben

RCT: Vorgeschrieben

Tabelle 14 – Ereignisfeld

Relativer Byte-Index	Bytes	Beschreibung
0	1	Protokollbezeichner (siehe Anhang B zur Festlegung und Aufbau der Meldung)
1	L	Ereignisdaten, z. B.: <SIA Account Block><SIA Event Block><SIA ASCII Block>

6.2.1.2 Zeitereignisfeld

SPT: Freigestellt

RCT: Vorgeschrieben

Tabelle 15 – Zeitereignisfeld

Relativer Byte-Index	Bytes	Beschreibung
0	8	Zeitformat nach RFC958 (NTP) / RFC4330 (SNTP V4)

Dieses Feld enthält den Zeitstempel, an dem das Ereignis aufgetreten ist.

Das Zeitformat ist eine 64-Bit-Zahl, wie sie in RFC958 (NTP) / RFC4330 (SNTP V4) beschrieben wird, die eine einfache lokale Synchronisation zulässt. Es ist zu beachten, dass NTP grundsätzlich einen 32-Bit-Zähler für Sekunden seit dem 1. Januar 1900 verwendet, so dass ein Zählerumlauf im Jahr 2036 auftreten wird. Aufgrund der „136-Jahre-Präzision“ genügt für die Abschätzung das korrekte Datum (entweder 1900, 2036, 2172 usw.) für die Wiedersynchronisierung für die nächsten 136 Jahre. Dies sollte mit den Geräten leicht auszuführen sein, muss jedoch bei der Übereinstimmungsprüfung mit einer besonderen Prüfung berücksichtigt werden.

Dieser Ansatz ist unabhängig von Gebieten mit Sommerzeitregelungen und unabhängig von den Zeitzonen, weil NTP die auf UTC beruhende Zeit angibt, so dass länderübergreifende Berechnungen einfacher werden. Derartige lokale Zeitanpassungen in Bezug auf die UTC (z. B. Zeitanzeige/Zeiteingabe in einem vom Menschen lesbaren Format) werden damit den Endgeräten überlassen.

6.2.1.3 Zeitmeldungsfeld

SPT: Freigestellt

RCT: Vorgeschrieben

Tabelle 16 – Zeitmeldungsfeld

Relativer Byte-Index	Bytes	Beschreibung
0	8	Zeitformat nach RFC958 (NTP) / RFC4330 (SNTP V4)

Dieses Feld enthält den Zeitstempel, an dem die Ereignismeldung von der SPT übertragen wird.

Dieser Wert ist für die Lebensdauerprüfung der Datagrame zu verwenden, z. B. Stärken des Protokolls gegen Angreifer in dem Sinne, das ein Datagram nur dann als gültig akzeptiert wird, wenn es innerhalb einer angemessenen Dauer (z. B. 51 h) am Ende des Kommunikationspartners eintrifft.

Weiterhin gibt die Differenz der Werte von Zeitereignis und Zeitmeldung Anlass zur Überprüfung, ob die Alarmanlage die maximalen Rundreiseverzögerungszeiten erfüllt.

6.2.1.4 Verbindungsfeld – IP-Adresse

SPT: Freigestellt

RCT: Freigestellt

Tabelle 17 – Verbindungsfeld – IP-Adresse

Relativer Byte-Index	Bytes	Beschreibung
0	L	IP-Adresse: L = 4 → IPv4-Adresse L = 32 → IPv6-Adresse

Dieses Feld definiert die IP-Adresse, an die die ergänzende Information gesendet wird.

6.2.1.5 Verbindungsfeld – IP-Portnummer

SPT: Freigestellt

RCT: Freigestellt

Tabelle 18 – Verbindungsfeld – IP-Portnummer

Relativer Byte-Index	Bytes	Beschreibung
0	2	Portnummer

Dieses Feld definiert die Portnummer, an die die ergänzende Information gesendet wird.

6.2.1.6 Verbindungsfeld – URL

SPT: Freigestellt

RCT: Freigestellt

Tabelle 19 – Verbindungsfeld – URL

Relativer Byte-Index	Bytes	Beschreibung
0	L	URL

Dieses Feld definiert die URL, an die die ergänzende Information gesendet wird.

6.2.1.7 Verbindungsfeld – Dateiname

SPT: Freigestellt

RCT: Freigestellt

Tabelle 20 – Verbindungsfeld – Dateiname

Relativer Byte-Index	Bytes	Beschreibung
0	L	Dateiname

Der Dateiname kann beispielsweise dazu verwendet werden, um die Dateien anzugeben, die auf einen TFTP-Server hochgeladen werden können.

6.2.2 Format der Ereignisantwort

Die Ereignisantwortmeldung hat das folgenden Format:

RCT → SPT

Tabelle 21 – Format der Ereignisantwortmeldung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode [a)]
HL + 1		Auffüllen
		Hash

[a)] Der Ergebniscode kann sein:
RESP_ACKNOWLEDGE
RESP_NEGATIVE_ACKNOWLEDGE
RESP_EVENT_RCT_COULD_NOT_PROCESS_MESSAGE
RESP_EVENT_PROTOCOL_ID_NOT_SUPPORTED
RESP_EVENT_ACKNOWLEDGE_UNKNOWN_FIELD

Für den Fall, dass die SPT freigestellte Felder in der Ereignismeldung enthält, die nicht von der RCT unterstützt werden, wird das Ereignis noch quittiert, jedoch mit einem RESP_ACKNOWLEDGE_UNKNOWN_FIELD. Dies ist eine gültige Bestätigung und es ist nicht notwendig, das Ereignis noch einmal zu senden.

6.3 Konfigurationsmeldungen

In diesem Unterabschnitt werden die Inhalte der Konfigurationsmeldungen beschrieben. Hinsichtlich des Meldungsflusses und weiterer Erläuterungen siehe Abschnitt 7.

Die Konfigurationsmeldungen werden für die beiden Inbetriebnahmeverfahren (DLTS und Out-of-band) verwendet, weil die Meldungsprotokolle die gleichen Parameter benötigen, unabhängig davon, wie die Verbindung hergestellt worden ist.

Die meisten konfigurierbaren Parameter kommen in der SPT für jede RCT, an die sie meldet, nur einmal vor, z. B.:

– Verbindungsidentifikator;
– Geräte-ID;
– Verschlüsselungsauswahl;
– Sitzungsschlüssel;
– Hash;
– Wegüberwachung.

Für den Fall, dass die SPT an zwei RCT meldet, gibt es für jeden Parameter zwei Instanzen, und zwar eine für jede verbundene RCT.

Wenn sich in der SPT die Parameter der RCT verändern (z. B. Wechsel zu einer anderen RCT), mit der sie zu verbinden ist, muss die SPT neue Parameter anfordern.

Weitere Parameter (z. B. Zeit) sind nur ein Wert, der von der SPT für alle RCT verwendet wird, an die sie meldet.

6.3.1 Anforderung des Verbindungsidentifikators

Die Anforderungsmeldung für den Verbindungsidentifikator hat das folgende Format:

SPT → RCT

Tabelle 22 – Format der Anforderungsmeldung für den Verbindungsidentifikator

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL		Auffüllen
		Hash

Diese Meldung wird von der SPT zur Anforderung eines Verbindungsidentifikators ausgegeben, der eine Zufallszahl ist. Die Verbindungskennung wird anstelle von der SPT von der RCT erzeugt, weil sie bei der RCT nur einmal vorkommen darf und der Zufallsgenerator der RCT üblicherweise eine bessere Qualität als einer der SPTs besitzt. SPT und RCT verwenden denselben Verbindungsidentifikator.

Für den Fall, dass die Verbindung unterbrochen ist, wird die nächste Sitzung einen neu erzeugten (anderen) Verbindungsidentifikator haben.

6.3.2 Antwort des Verbindungsidentifikators

Die Antwortmeldung für den Verbindungsidentifikator hat das folgende Format:

RCT → SPT

Tabelle 23 – Format der Antwortmeldung für den Verbindungsidentifikator

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1	2	Verbindungsidentifikator
HL + 2		Auffüllen
		Hash

Diese Meldung selbst und die vorigen Meldungen haben einen Verbindungsidentifikator mit dem Wert 0. Die nächste Meldung wird die erste mit einem gültigen Verbindungsidentifikator-Feld sein.

6.3.3 Anforderung der Geräte-ID

Die Meldung zur Anforderung einer Geräte-ID hat das folgende Format:

SPT → RCT

Tabelle 24 – Format der Meldung zur Anforderung einer Geräte-ID

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Flags
HL + 1	16	Geräte-ID
HL + 17		Auffüllen
		Hash

Diese Meldung wird von der SPT ausgegeben, um eine Geräte-ID anzufordern, die eine Zufallszahl ist. Die Geräte-ID wird von der RCT anstelle der SPT erzeugt, weil sie an der RCT nur einmal vorkommen darf und weil der Zufallszahlengenerator der RCT üblicherweise eine bessere Qualität als der einer SPT besitzt. SPT und RCT verwenden die gleiche Geräte-ID.

Das Folgende gilt, um die Inbetriebnahme des 2. Kanals zu ermöglichen:

– wenn das Flag Master Device ID request gesetzt ist, fordert die SPT nur einmal eine neue Geräte_ID an und speichert diese neue Geräte-ID, wie sie sie empfangen hat, in der Antwortmeldung im NVM. Diese Geräte-ID wird auch von der RCT gespeichert;
– wenn das Flag Master Device ID request gelöscht wird und das Feld Geräte-ID nicht den Wert 0 hat, dann hat die SPT bereits eine Master Device ID und informiert die RCT darüber. Die RCT wird in ihrer Antwort die gleiche Geräte-ID zurückgeben.

Anderenfalls wird das Flag Master Device ID request gelöscht und das Feld Geräte-ID hat den Wert 0.

Flags:

Tabelle 25 – Flag ‚Master Device ID request'

Bit	Beschreibung
0	Master Device ID request
1	0: SPT-Geräte-ID 1: RCT-Geräte-ID
2 … 7	Nicht verwendet

6.3.4 Antwort der Geräte-ID

Die Meldung der Antwort der Geräte-ID hat das folgende Format:

RCT → SPT

Tabelle 26 – Format der Meldung der Antwort der Geräte-ID

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1	1	Flags
HL + 2	16	Geräte-ID
HL + 18		Auffüllen
		Hash

Die nächste Meldung wird die erste sein, die in den Kopfdaten der Meldung ein gültiges Feld Geräte-ID hat.

Die Geräte-ID in den Kopfdaten dieser Meldung selbst und den vorhergehenden Meldungen sind:

– 0 im Falls von DTLS;
– die ID des „One-Time-Pad" / Einrichten des 2. Kanals.

Die Flag-Felder behalten den Wert 0.

6.3.5 Anforderung der Verschlüsselungsauswahl

Die Meldung zur Anforderung der Verschlüsselungsauswahl hat das folgende Format:

SPT → RCT

Tabelle 27 – Format der Meldung zur Anforderung der Verschlüsselungsauswahl

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Flags
HL + 1	1	Verschlüsselung 1
HL + 2	1	Verschlüsselung 2 (freigestellt) … usw.
		Auffüllen
		Hash

Diese Meldung wird während der Inbetriebnahme von der SPT ausgegeben, um die von ihr unterstützten Verschlüsselungsverfahren anzugeben. Zu möglichen Verschlüsselungsverfahren siehe 5.4.

Flags:

Tabelle 28 – Flag ‚Master Encryption Selection request'

Bit	Beschreibung
0	Master Encryption Selection request
1 … 7	Nicht verwendet

6.3.6 Antwort der Verschlüsselungsauswahl

Die Antwortmeldung der Verschlüsselungsauswahl hat das folgende Format:

RCT → SPT

Tabelle 29 – Format der Antwortmeldung der Verschlüsselungsauswahl

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Flags
HL + 1	1	Ergebniscode
HL + 2	1	Anzuwendendes Verschlüsselungsverfahren
HL + 3		Auffüllen
		Hash

Die Flag-Felder behalten den Wert 0.

6.3.7 Anforderung des Verschlüsselungscodetausches

Die Anforderungsmeldung für den Verschlüsselungscodetausch hat das folgende Format:

SPT ← → RCT

Tabelle 30 – Format der Anforderungsmeldung für den Verschlüsselungscodetausch

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Flags
HL + 1	L	Verschlüsselungscode (üblicherweise 256 Bit → 32 Byte)
HL + 1 + L		Auffüllen
		Hash

Flags:

Tabelle 31 – Flag ‚Master Key Request'

Bit	Beschreibung
0	Key Request (SPT) / Key Push (RCT)
1	Master Key Request
2 … 7	Nicht verwendet

Diese Meldung wird ausgegeben, um eine Aktualisierung des Verschlüsselungscodes anzufordern. SPT und RCT können beide eine Aktualisierung der Verschlüsselungscodes anfordern, wobei in diesem Fall das Flag Master Key Request gesetzt wird. Wenn dieses Flag gesetzt ist, behält der Verschlüsselungscode den gegenwärtig benutzten Schlüssel.

Die SPT fordert einen neuen Schlüssel von der RCT an. Der neue Schlüssel muss von der RCT anstelle der SPT erzeugt werden, weil er von einem kryptografisch exakten Zufallszahlengenerator erzeugt werden muss, und der Zufallszahlengenerator der RCT besitzt üblicherweise eine bessere Qualität als der von einer der SPTs.

Die RCT kann einen neuen Sitzungsschlüssel durch Setzen des Flags Key Request bei der SPT einspeichern. Der neue Schlüssel ist das Feld Encryption key. Die SPT wird anschließend diesen Schlüssel bestätigen, indem sie diesen Schlüssel in der Antwortmeldung des Verschlüsselungscodeaustausches zurückgibt.

6.3.8 Antwort des Verschlüsselungscodeaustausches

Die Antwortmeldung des Verschlüsselungscodeaustausches hat das folgende Format:

SPT ← → RCT

Tabelle 32 – Format der Antwortmeldung des Verschlüsselungscodeaustausches

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1	1	Flags
HL + 2	L	Verschlüsselungscode (üblicherweise 256 Bit → 32 Byte)
HL + 2 + L		Auffüllen
		Hash

Der neue Schlüssel wird unmittelbar wirksam, so wird z. B. die nächste Meldung mit dem neuen Schlüssel verschlüsselt (für den Fall, dass der Wert von Verschlüsselungsauswahl > 0 ist). Zur Überwindung von Übermittlungsfehlern muss die RCT den vorigen Schlüssel noch als Reserve behalten, bis eine nächste Meldung erfolgreich empfangen worden ist.

6.3.9 Anforderung zur Hashauswahl

Die Anforderungsmeldung zur Hashauswahl hat das folgende Format:

SPT → RCT

Tabelle 33 – Format der Anforderungsmeldung zur Hashauswahl

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Hash 1
HL + 1	1	Hash 2 (freigestellt) … usw.
		Auffüllen
		Hash

Diese Meldung wird während der Inbetriebnahme von der SPT ausgegeben, um die von ihr unterstützten Hashes anzugeben. Zu möglichen Hashfunktionen siehe 5.3.

6.3.10 Antwort der Hashauswahl

Die Antwortmeldung der Hashauswahl hat das folgende Format:

RCT → SPT

Tabelle 34 – Format der Antwortmeldung der Hashauswahl

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1	1	Anzuwendender Hash
HL + 2		Auffüllen
		Hash

Dies ist die erste Meldung, die einen neu gesetzten Hash verwendet. Standardmäßig (nach dem Neustart) wird die Internet-Prüfsumme (Wert 1) als Hash verwendet.

6.3.11 Anforderung zur Wegüberwachung

Die Anforderungsmeldung zur Wegüberwachung hat das folgende Format:

SPT → RCT

Tabelle 35 – Format der Anforderungsmeldung zur Wegüberwachung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	4	Heartbeat-Intervallzeit (Sekunden)
HL + 4	1	Push (0) oder Pull (1)
HL + 5		Auffüllen
		Hash

Die Heartbeat-Intervallzeit legt die Dauer fest, bis die SPT den nächsten Heartbeat senden wird.

Die Push-Pull-Möglichkeit bestimmt das Abfragegerät:

– 0: Push: die SPT sendet die Abfrage an die RCT;
– 1: Pull: die RCT sendet die Abfrage an die SPT, die die Lastverteilung zulässt.

6.3.12 Antwort der Wegüberwachung

Die Antwortmeldung der Wegüberwachung hat das folgende Format:

RCT → SPT

Tabelle 36 – Format der Antwortmeldung der Wegüberwachung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode [a]
HL + 1	4	Heartbeat-Intervallzeit (s)
HL + 5	1	Push (0) oder Pull (1)
HL + 6		Auffüllen
		Hash

[a] Der Ergebniscode kann sein:
RESP_ACKNOWLEDGE
RESP_POLL_TOO_SLOW

6.3.13 Befehl zur Zeiteinstellung

Die Befehlsmeldung zur Zeiteinstellung hat das folgende Format:

RCT → SPT

Tabelle 37 – Format der Befehlsmeldung zur Zeiteinstellung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	8	Zeitformat nach RFC958 (NTP) / RFC4330 (SNTP V4)
HL + 8		Auffüllen
		Hash

Dieser Befehl ist freigestellt. Für den Fall, dass Ereignisse mit Zeitstempel übertragen werden, kann dieser Befehl zur Synchronisierung von der RCT gesendet werden.

6.3.14 Antwort der Zeiteinstellung

Die Antwortmeldung der Zeiteinstellung hat das folgende Format:

SPT → RCT

Tabelle 38 – Format der Antwortmeldung der Zeiteinstellung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1		Auffüllen
		Hash

6.3.15 Transparente Meldung

Die transparente Meldung hat das folgende Format:

Tabelle 39 – Format der transparenten Meldung

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	L	Transparente Daten
HL + L		Auffüllen
		Hash

Diese Meldung lässt (verkäuferspezifische) Daten zu, die zwischen SPT und RCT zu übertragen sind. Sie kann beispielsweise zur Konfiguration von Daten oder zum Hochladen von Firmware benutzt werden.

6.3.16 Transparente Antwort

Die transparente Antwort hat das folgende Format:

Tabelle 40 – Format der transparenten Antwort

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1	L	Transparente Daten
HL + 1 + L		Auffüllen
		Hash

6.3.17 Anforderung des Datagram-Verschlüsselungsprotokolls

Die Meldung zur Anforderung des Datagram-Verschlüsselungsprotokolls hat das folgende Format:

SPT → RCT

Tabelle 41 – Format der Meldung zur Anforderung des Datagram-Verschlüsselungsprotokolls

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL		Auffüllen
		Hash

Diese Meldung wird von der SPT zur Anforderung des Endes der Datagram-Verschlüsselungsprotokoll-Sitzung gesendet.

Diese Meldung enthält keine zusätzlichen Informationen.

6.3.18 Antwort des Datagram-Verschlüsselungsprotokolls

Die Antwortmeldung des Datagram-Verschlüsselungsprotokolls hat das folgende Format:

RCT → SPT

Tabelle 42 – Format der Antwortmeldung des Datagram-Verschlüsselungsprotokolls

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL	1	Ergebniscode
HL + 1		Auffüllen
		Hash

Diese Meldung wird von der RCT als Antwort auf die Anforderung des Datagram-Verschlüsselungsprotokolls gesendet.

Sie wird von der RCT gesendet, um die Vereinbarung der Parameter zu beenden. Nachdem sie von der RCT gesendet und von der SPT empfangen worden ist, wird die Datagram-Verschlüsselungsprotokoll-Sitzung beendet, alle von der Sitzung verwendeten Ressourcen werden frei und die weitere Kommunikation zwischen RCT und SPT erfolgt mit den vereinbarten Parametern.

6.3.19 Anforderung zur IP-Parameter der RCT

Die Anforderungsmeldung zur RCT-Parameter hat das folgende Format:

SPT → RCT

Tabelle 43 – Format der Anforderungsmeldung zur RCT-Parameter

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL		Auffüllen
		Hash

Wenn die SPT Daten austauschen soll und dabei entweder eine andere Port-Nummer für die Inbetriebnahme oder den normalen Sitzungsverkehr verwendet oder wenn für die Inbetriebnahme und Sitzung gesonderte RCTs verwendet werden oder wenn die SPT mit mehr als einer RCT Daten austauschen soll, dann kann die RCT die IP-Adresse(n) und Ports senden, die für die Sitzung zu verwenden sind. Es liegt in der Verantwortung der inbetriebnehmenden RCT, die Sitzungsparameter an alle anderen RCTs sicher zu übergeben, mit denen die SPT Daten austauschen darf. Der Mechanismus, nach dem die RCTs die Sitzungsparameter gemeinsam nutzen, ist verkäuferspezifisch und liegt außerhalb des Anwendungsbereiches dieser Protokollnorm.

Die praktische Umsetzung dieser Meldung ist für die SPT freigestellt.

6.3.20 Antwort der IP-Parameter der RCT

Die Antwort der RCT-Parameter hat das folgende Format:

RCT → SPT

Tabelle 44 – Format der Antwort der RCT-Parameter

Byte-Index	Bytes	Beschreibung
0	HL	Kopfdaten, Meldung-ID und Länge der Meldung
HL + 1	1	Ergebniscode
HL + 2	1	Feld-Bezeichner – RCT 1 IP-Adresse – siehe 6.2.1.4
HL + 3	2	Länge des Feldes (L1)
HL + 5	L1	Daten des Feldes
HL + 5 + L1	1	Feld-Bezeichner – RCT 1 Port-Nummer – siehe 6.2.1.5
HL + 6 + L1	2	Länge des Feldes (L1)
HL + 8 + L1	L2	Daten des Feldes
HL + 8 + L1 + L2	1	2. Feld-Bezeichner – RCT 2 IP-Adresse – siehe 6.2.1.4
HL + 9 + L1 + L2	2	2. Länge des Feldes (L2) (freigestellt)
HL + 11 + L1	L3	2. Daten des Feldes (freigestellt) … usw.
HL + 8 + L1 + L2 + L3		Auffüllen
		Hash

7 Inbetriebnahme und Verbindungsaufbau

7.1 Inbetriebnahme

Das Ziel des Inbetriebnahmeverfahrens besteht darin, die gegenseitige Authentifizierung der Übertragungseinrichtung (SPT) und der Empfangszentrale (RCT) zu ermöglichen.

Das Inbetriebnahmeverfahren wird auch zur Vereinbarung der folgenden Parameter verwendet:

– Haupt-Geräte-IDs von SPT und RCT;

– Haupt-Verschlüsselungscode;

– Haupt-Verschlüsselungsauswahl;

– (freigestellte) RCT-IP-Adresse(n) und Port(s), mit denen die SPT Daten austauschen sollte (damit wird ein gesonderter „Inbetriebnahme-Server“ zugelassen, der den „ersten Kontakt“ zu mehreren Empfängern herstellt. In diesem Fall muss der Inbetriebnahme-Server die Sitzungsparameter an die entsprechende RCT sicher übertragen. Der hierfür angewendete Mechanismus liegt außerhalb des Anwendungsbereiches dieses Protokolls).

Ein erfolgreiches Inbetriebnahmeverfahren baut eine Kommunikationssitzung mit einem Verbindungsidentifikator mit einem eindeutigen Bezeichner auf. Die Kommunikationssitzung dauert solange, bis eine Außerbetriebnahme stattfindet. Dabei hat besonders der Wechsel der Sitzungsschlüssel keinen Einfluss auf die Kommunikationssitzung, d. h. er führt nicht zu einer Änderung des Verbindungsidentifikators.

7.1.1 Verfahren

Für die Erlangung des „Master Set“ gibt es zwei Möglichkeiten:

– entweder generiert mit einem „Shared Secret“, mit Out-of-band-Übergabe; oder

– Verwendung von X.509-Zertifikaten und DTLS in RCT und SPT (freigestellt) (siehe 7.14).

Ungeachtet des eingesetzten Mechanismus zur Erlangung wird der Hauptschlüssel zur Verschlüsselung mit AES256 und dem Austausch der weiteren Parameter eingesetzt. Er wird auch (durch das „Lauf-“Protokoll) zur Einrichtung des (der) Sitzungsschlüssel(s) verwendet.

Der Master-Schlüssel ist ein 256-Bit-Schlüssel.

7.1.2 Meldungsfolge der Inbetriebnahme

Der „Master Set“ wird mit dem nachfolgend beschriebenen Ablauf der Meldungen ausgetauscht. Die Meldungen sind die gleichen, wobei das angewendete Inbetriebnahmeverfahren keine Rolle spielt. Der Unterschied liegt in den Maßnahmen, mit denen die Meldungen gesichert werden, entweder mit „Shared Secret“ („One-Time-Pad“-Schlüssel und -Geräte-ID), wie sie von der RCT unterstützt wird oder mit X.509/DTLS.

Der Ablauf der Meldungen während der Inbetriebnahme einer neuen SPT ist wie folgt:

Tabelle 45 – Ablauf der Meldungen während der Inbetriebnahme einer neuen SPT

SPT	Richtung	RCT	Bemerkungen
			Der Hash, mit dem begonnen wird, ist die Internet-Prüfsumme
CONN_HANDLE_REQ	→		
	←	CONN_HANDLE_RESP	
DEVICE_ID_REQ	→		Flag Master-Geräte-ID gesetzt Flag SPT-Geräte-ID gesetzt
	←	DEVICE_ID_RESP	
DEVICE_ID_REQ	→		Flag Master-Geräte-ID gesetzt Flag RCT-Geräte-ID gesetzt
	←	DEVICE_ID_RESP	
SESSION_KEY_REQ	→		Flag Master-Schlüssel-Anforderung gesetzt
	←	SESSION_KEY_RESP	
ENCRYPT_SELECT_REQ	→		Flag Master-Verschlüsselungsauswahl-Anforderung gesetzt
	←	ENCRYPT_SELECT_RESP	
			Schlüsselaktualisierung vollständig, fortsetzen mit neuem Verschlüsselungscode und neuem Verschlüsselungsverfahren
DTLS_COMPLETE_REQ	→		Nur bei Verwendung von X.509/DTLS
	←	DTLS_COMPLETE_RESP	

Die sich ergebenden Master-Parameter werden in NVM in SPT und RCT gespeichert.

Der nächste Schritt ist die Anforderung der Sitzungsparameter, wie in 7.2 festgelegt.

7.1.3 Inbetriebnahme mit Shared Secret

Die Unterstützung des Shared-Secret-Verfahrens für die Erzeugung des Master-Schlüssels ist für RCTs und SPTs verbindlich.

Bei diesem Verfahren wird die RCT ein SECRET erzeugen, welches aus 2 „Token“/„Textmeldungen“ besteht (siehe 7.1.3.1 für Anforderungen an SECRETS). Diese repräsentieren die Geräte-ID und den Verschlüsselungscode.

Die „Token“/„Textmeldungen“ werden von RCT und SPT verwendet, um die folgenden Größen zu erzeugen:

– die 16-Byte-Geräte-IDs (SPT und RCT);

– den 32-Byte-Verschlüsselungscode (AES-256).

Diese werden aus den 256-Hashes der Token erzeugt (Geräte-ID: SPT – oberste 16 Byte; RCT – untere 16 Byte).

Für die Inbetriebnahmephase ist AES-256 verbindlich. Auf Anfrage der SPT (Leistungsfähigkeit) kann er für den normalen Datenaustausch auf AES-128 geändert werden.

Diese Parameter werden nur für den Austausch des Master-Schlüssels verwendet. Wenn der Master einmal von der RCT zur SPT gesendet worden ist, dann wird die Sitzung gelöscht und niemals wieder verwendet.

Bei der Shared-Secret-Inbetriebnahme werden die „One-Time-Pad"-Geräte-ID, der Verschlüsselungscode und die Verschlüsselungsauswahl dazu verwendet, die erste Verbindung aufzubauen. Nach dem Abschluss der Inbetriebnahme wird das Token von der RCT nicht akzeptiert.

Als nächstes werden diese Parameter erneuert (SPT verwendet die Flags Master-Geräte-ID und Master-Schlüsselanforderung) und speichert es als das neue „Master Set" in einem nichtflüchtigen Speicher. Dieses neue „Master Set" wird zu Wiederverbindung nach Abschaltungen oder Stromausfall verwendet.

7.1.3.1 Übermittlung des Shared Secret über einen Out-of-band-Kanal

Die Sicherheit des Out-of-band-Kanals ist einer der Faktoren, der die Sicherheit des Paarbildungsprozesses zwischen SPT und RCT bestimmt. Da der Out-of-band-Kanal sehr wahrscheinlich vom Bediener auf der einen Seite oder auf beiden Seiten abhängt, sollte er einfach umzusetzen und tolerant für Fehler des Menschen sein. Es gelten die folgenden Anforderungen.

– Das SECRET muss durch das Managementsystem der ATS erzeugt werden, die an einer ARC betrieben wird oder nicht. Die Verarbeitungsleistung des Managementsystems übertrifft das der SPT üblicherweise um Größenordnungen und kann deshalb das SECRET mit höherer kryptografischer Qualität (Zufälligkeit) als ein kleines eingebettetes System erzeugen. Zusätzlich hat der ATS-Diensteanbieter oder ARC eine Garantie, dass der SECRET-Erzeugungsprozess mit diesen Anforderungen übereinstimmt.

– Physikalische und logische Mittel für die Übertragung des SECRET an die SPT müssen das Abfangen durch einen Dritten, ohne dass er entdeckt wird, schwierig machen. Das Wort schwierig bedeutet hier aufwendig hinsichtlich der Zeit oder der aufgewendeten Ressourcen im Vergleich zu dem Vorteil, den ein Angreifer durch Kenntnis des SECRET erlangen kann. In Abhängigkeit von der Sicherheitsstufe der zu schützenden Objekte können die folgenden Verfahren als geeignet angesehen werden:

  – der ARC-Bediener diktiert dem Außendiensttechniker das SECRET über das Telefon;
  – das SECRET wird mittels SMS übertragen;
  – das SECRET wird in einer verschlüsselten und signierten E-Mail gesendet;
  – das SECRET wird an der Leitstelle/ARC gedruckt und der Außendiensttechniker bringt es selbst zum zu schützenden Objekt;
  – das SECRET wird an der Leitstelle/ARC in die SPT programmiert und anschließend zu den zu schützenden Objekten transportiert;
  – der Außendiensttechniker erhält das SECRET über eine geschützte Website des ATS-Dienstleisters;
  – jedes weitere Verfahren, welches das Schwierigkeitskriterium erfüllt.

  Es liegt in der Verantwortung des ATS-Dienstleisters/ARC, die Sicherheit des Verfahrens zu beurteilen, die er für die Übertragung des SECRET in Bezug auf die Sicherheitsstufe der zu schützenden Objekte anwendet.

– Das SECRET darf nicht über einen Kanal gesendet werden, der für den Datenaustausch zwischen der SPT und der RCT für die Alarmmeldung und Überwachung verwendet wird.

– Das SECRET muss mit einem kryptografisch gutem Zufallszahlengenerator [2)] erzeugt werden.

– Das SECRET muss als Text dargestellt werden, der aus druckbaren Zeichen aus dem in Abschnitt C.1 beschriebenen Zeichensatz besteht.

– Um mit Tippfehlern oder anderen, üblicherweise vom Menschen verursachten, Übertragungsfehlern fertig zu werden, wird die Textdarstellung des SECRET um eine 16-Bit-Prüfsumme erweitert, die wie in Abschnitt C.2 beschrieben als gewichtete CRC berechnet und direkt der SECRET-Zeichenfolge angehangen und in der gleichen Weise wie die SECRET-Zeichenfolge verschlüsselt wird.

– Die Stärke des SECRET muss der von 128 Bit entsprechen. Da jedes der zulässigen Verschlüsselungszeichen eine Struktur aus 4 Bit darstellt und das SECRET durch eine zusätzliche 16-Bit-CRC geschützt ist, wird das gesamte SECRET durch (128 + 16) / 4 = 36 Zeichen verschlüsselt. Ein Beispiel wird in Abschnitt C.3 angegeben.

[2)] Siehe RFC4086.

7.1.4 Inbetriebnahme mit X.509-Zertifikaten und DTLS

Die Unterstützung des X.509-Mechanismus und DTLS ist für SPTs freigestellt und für RCTs verbindlich.

Die Authentifikation, die Auswahl des Verschlüsselungsverfahrens und der Schlüsselaustausch erfolgt unter Anwendung des DTLS-Protokolls mit der SPT als Client und der RCT als Server. DTLS ist eine Variante von TLS, welches die Grundmeldungen und -formate definiert. Die Geräte-ID und die freigestellten Parameter werden mit dem vereinbarten Verschlüsselungs- und Sitzungsschlüssel eingerichtet.

Die Verschlüsselungsfolge (en: cipher suite) TLS_DHE_DSS_WITH_AES_256_CBC_SHA muss verwendet werden und der Master-Schlüssel ist der mit dem DTLS-Handshake erzeugte symmetrische 256-Bit-AES-Schlüssel.

RCT-Anforderungen:

– jede RCT muss die Zertifikate für jede CA besitzen, die ein Zertifikat für jede SPT signiert hat, welche möglicherweise mit der RCT verbunden werden kann;

– die RCT muss einen Mechanismus zur Verfügung stellen, um dem System neue CA-Zertifikate hinzuzufügen, damit eine SPT von einem neuen Hersteller mit dem System verbunden werden kann, sowie einen weiteren Mechanismus, um CA-Zertifikate vom System zu löschen. Die Einzelheiten für das Einfügen/Löschen eines Zertifikates liegt außerhalb des Anwendungsbereiches dieser Europäischen Norm;

– obwohl die DTLS-Implementierung in der RCT weitere Verschlüsselungsfolgen unterstützen kann, darf für die Erzeugung des Master-Schlüssels nur TLS_DHE_DSS_WITH_AES_256_CBC_SHA angewendet werden.

SPT-Anforderungen:

– die SPT muss die Zertifikate der CAs besitzen, die die Zertifikate für die RCTs signiert haben, mit der die SPT möglicherweise verbunden werden kann. Für die SPT ist es nicht verbindlich, die Authentizität der RCT zu prüfen, es wird jedoch empfohlen;

– der einheitliche Name von SPTs entsprechend X.509-Zertifikat ist das Format „Lieferanten-Bezeichner: lieferantenspezifischer Bezeichner“ [3)]. Als Lieferanten-ID wird der registrierte Name der Internet-Domain des Lieferanten verwendet. Dies muss den SPT eindeutig identifizieren;

– das X.509-Zertifikat der SPTs muss von einer CA signiert sein, die allen RCTs bekannt ist, mit denen sie möglicherweise verbunden werden kann;

– die SPT darf nur die Verschlüsselungsfolge TLS_DHE_DSS_WITH_AES_256_CBC_SHA präsentieren, die im DTLS-Quittungsaustausch zu verwenden ist.

Bei Fertigstellung der Parametervereinbarung wird die DTLS-Sitzung abgeschlossen, Kontexte usw. werden frei und der gesamte weitere Datenaustausch erfolgt mit den vereinbarten Parametern.

7.2 Einrichten der Verbindung

Im Falle einer Wiederverbindung wird der „Master Set“, wie er während der Inbetriebnahme vereinbart worden ist, zu Beginn für die Verschlüsselung und Authentifikation der Meldungen zwischen SPT und RCT verwendet. Die ersten Schritte sind die Anforderung neuer Sitzungsparameter, die anschließend für den weiteren Datenaustausch benutzt werden.

Die Verbindungen sind üblicherweise in ständiger Bereitschaft; für den Fall, dass eine Verbindung zusammenbricht, wird die SPT versuchen, die Verbindung wieder herzustellen.

Während der Phase des Einrichtens der Verbindung werden die folgenden Parameter in der nachstehenden Reihenfolge gesetzt:

– Verbindungsidentifikator;

– Geräte-ID SPT (Authentifikation);

– Geräte-ID RCT (Authentifikation);

– Sitzungsschlüssel;

[3)] Die Länge des Lieferanten-Bezeichners und des lieferantenspezifischen Bezeichners sind zu vereinbaren und festzulegen.

– Verschlüsselungsauswahl;
– Hash;
– Wegüberwachung;
– Protokollversion, Stufe ist zwischen SPT und RCT vereinbart.

ANMERKUNG Zu Beginn sind einige Felder in den Kopfdaten noch nicht initialisiert bis die passende Konfigurationsmeldung verarbeitet worden ist. Deshalb ist es notwendig, dass die IP-Adresse der SPT sich während der Initialisierungsphase nicht verändert (und während des Austausches konstant bleiben sollte, selbst wenn die gesicherten Objekte die am stärksten einschränkende zustandsorientierte Firewall haben).

Der Fluss der Meldungen während des Einrichtens der Verbindung (zur Anforderung der Sitzungsparameter) lautet wie folgt:

Tabelle 46 – Fluss der Meldungen während des Einrichtens der Verbindung

SPT	Richtung	RCT	Bemerkungen
			Der Hash, mit dem begonnen wird, ist die Internet-Prüfsumme
CONN_HANDLE_REQ	◊		Weggelassen, wenn dies direkt nach der Inbetriebnahme folgt, wobei in diesem Fall der zuvor festgelegte Verbindungsidentifikator aktiv bleibt.
	⇓	CONN_HANDLE_RESP	
DEVICE_ID_REQ	◊		Flag Master-Geräte-ID gelöscht Flag SPT-Geräte-ID gesetzt
	⇓	DEVICE_ID_RESP	
DEVICE_ID_REQ	◊		Flag Master-Geräte-ID gelöscht Flag RCT-Geräte-ID gesetzt
	⇓	DEVICE_ID_RESP	
SESSION_KEY_REQ	◊		Flag Master-Schlüssel-Anforderung gelöscht
	⇓	SESSION_KEY_RESP	
ENCRYPT_SELECT_REQ	◊		Flag Master-Verschlüsselungsauswahl-Anforderung gelöscht
	⇓	ENCRYPT_SELECT_RESP	
			Schlüsselaktualisierung vollständig, fortsetzen mit neuem Schlüssel
HASH_SELECT_REQ	◊		
	⇓	HASH_SELECT_RESP	
PATH_SUPERVISION_REQ	◊		
	⇓	PATH_SUPERVISION_RESP	
VERSION_REQ	◊		SPT-Protokollversion
	⇓	VERSION_RESP	RCT-Protokollversion Die höchste Protokollversion, die von SPT und RCT unterstützt wird, muss von jetzt an verwendet werden. Es dürfen nur Leistungsmerkmale verwendet werden, die die vereinbarte Protokollversion unterstützt.
			Der Verbindungsaufbau ist jetzt vollständig, nach diesem Punkt darf sich auch die IP-Adresse ändern. Es darf/kann einige Zeit dauern, bevor die nächste (Abfrage-)Meldung übertragen wird.

Tabelle 46 *(fortgesetzt)*

SPT	Richtung	RCT	Bemerkungen
POLL_MSG	◊		
	⇓	POLL_RESP	Erste Abfrage gesendet nach dem Abfrage-intervall.

Die SPT beginnt mit dem „Master Set“, den sie in einem NVM hat. Die Flags Master-Geräte-ID, Master-Verschlüsselungsauswahl-Anforderung und Master-Schlüsselanforderung werden gelöscht. Der „Master Set“ wird somit nur dazu verwendet, um neue Sitzungsparameter zu erhalten. Üblicherweise ändert sich nur der Sitzungsschlüssel. Die Geräte-ID kennzeichnet SPT und RCT und verändert sich nicht.

Anhang A
(normativ)

Ergebniscodes

Tabelle A.1 – Ergebniscodes

Bytes	Antwort auf	Wert
RESP_ACKNOWLEDGE	Alle	0x00
RESP_NEGATIVE_ACKNOWLEDGE	Allee	0x01
RESP_EVENT_RCT_COULD_NOT_PROCESS_MESSAGE	Ereignismeldungen	0x10
RESP_EVENT_PROTOCOL_ID_NOT_SUPPORTED	Ereignismeldungen	0x11
RESP_EVENT_ACKNOWLEDGE_UNKNOWN_FIELD	Ereignismeldungen	0x12
RESP_POLL_TOO_SLOW	Anforderung der Wegüberwachung	0x20
RESP_POLL_REESTABLISH_CONNECTION	Abfragemeldungen	0x21
RESP_CMD_NOT_SUPPORTED	Befehle	0x30
RESP_DEVICE_ID_UNKNOWN	Anforderung Geräte-ID	0x31
RESP_UNKNOWN	Alle	0xFF

Anhang B
(normativ)

Protokollbezeichner

In der folgenden Tabelle sind die möglichen Protokollbezeichner für das Anwendungsschichtprotokoll zusammengefasst, die von dem in dieser Europäischen Norm definierten Protokoll geführt werden.

Jede kompatible Implementierung dieses Protokolls muss mindestens zwei Arten der Meldungsübermittlung unterstützen:

– Transparente Meldungen für seriell verbundene AE und/oder AS;
– Sia DC-03 Meldungsstrukturen für AS-Signale, die über Kontaktstifteingänge verbunden sind;
– Sia DC-03 Meldungsstrukturen für Meldungen, die intern durch SPT und/oder RCT erzeugt werden.

Tabelle B.1 – Protokollbezeichner

Protokoll-ID	Protokoll
01	Sia DC-03 Meldungen, wie beschrieben in SIA DC-03-1990.01(R2003.10), Abschnitt 5 und Anhang 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	Herstellerspezifisch
255	Transparent, serielle Übertragung des empfangenen Inhalts im Datenfeld

Ein Hersteller, der Meldungen übertragen will, die zu keinem der aufgeführten Anwendungsprotokolle passen, muss den Protokollbezeichner 254 verwenden. Ein gegenwärtig nicht zugewiesener ProtokollBezeichner darf nach einer späteren Überarbeitung dieser Europäischen Norm zugewiesen werden.

Anhang C
(normativ)

Shared Secret

C.1 Zeichensatz für Verschlüsselung, Formatierung und Entschlüsselung

Bei der Verschlüsselung und Formatierung des Masterschlüssel-Secret in ein Zeichenkettenformat, welches für Menschen lesbar ist, dürfen nur die Zeichen des folgenden Zeichensatzes entsprechend der Beschreibung verwendet werden.

– Die Verschlüsselung des Schlüsselwertes selbst (und der angehängten Prüfsumme) darf nur mit den Zeichen {0, …, 9} + {A, …, F} + {G, H, K, M, N, Q, …, U, W, X, …, Z} + {+, #} erfolgen. Jedes dieser Zeichen repräsentiert einen 4-Bit-Wert, wie er in der Zeichensatztabelle angegeben ist (siehe Tabelle C.1).

– Die Formatierung des Schlüsselwertes zur Verbesserung der Lesbarkeit für Menschen muss mit den aufgeführten Trennzeichen {-} oder {Leerezeichen} erfolgen. Die Trennzeichen dürfen zur Verbesserung der Lesbarkeit frei bei der Formatierung verwendet werden (z. B. Anordnung in 4-Zeichen-Blöcken, jeder Block ist durch Trennungsstriche von den anderen getrennt); bei der Entschlüsselung wird das Vorhandensein von Trennzeichen innerhalb der Schlüsselzeichenkette vollständig ignoriert.

– Alle weiteren Zeichen werden nicht ausdrücklich für die Verschlüsselung des Schlüssels verwendet. Wenn sie während der Entschlüsselung auftreten, dann ist einer der beiden Fälle zu berücksichtigen:
  – wenn dem Zeichen ein 4-Bit-Wert in der Zeichensatztabelle zugewiesen worden ist, dann wird ein Zweideutigkeitsfehler angenommen und der zugewiesene Wert wird stillschweigend verwendet;
  – wenn dem Zeichen kein 4-Bit-Wert in der Zeichensatztabelle zugewiesen worden ist, dann wird das Ende der Zeichenkette angenommen und die interne Schlüsselverarbeitung kann starten.

Die Zeichendarstellung ist in der folgenden Tabelle definiert.

Tabelle C.1 – Zeichensatz

Zeichen	Zeichencode (hexadezimal)	Dargestellter Wert (binär)	Bemerkungen
0	0x30	0000	
1	0x31	0001	
2	0x32	0010	
3	0x33	0011	
4	0x34	0100	
5	0x35	0101	
6	0x36	0110	
7	0x37	0111	
8	0x38	1000	
9	0x39	1001	
A	0x41	1010	
B	0x42	1011	
C	0x43	1100	
D	0x44	1101	
E	0x45	1110	
F	0x46	1111	
G	0x47	0000	

Tabelle C.1 *(fortgesetzt)*

Zeichen	Zeichencode (hexadezimal)	Dargestellter Wert (binär)	Bemerkungen
H	0x48	0001	
K	0x4B	0010	
M	0x4D	0011	
N	0x4E	0100	
P	0x50	0101	
R	0x52	0110	
S	0x53	0111	
T	0x54	1000	
U	0x55	1001	
W	0x57	1010	
X	0x58	1011	
Y	0x59	1100	
Z	0x5A	1101	
*	0x2A	1110	
#	0x23	1111	
Leerzeichen	0x20		Trennzeichen
!	0x21	0001	Dargestellt als „1“
"	0x22		
$	0x24		
%	0x25		
&	0x26		
'	0x27		
(	0x28		
)	0x29		
+	0x2B		
,	0x2C		
-	0x2D		Trennzeichen
.	0x2E		
/	0x2F		
:	0x3A		
;	0x3B		
<	0x3C		
=	0x3D		
>	0x3E		
?	0x3F		
I	0x49	0001	Dargestellt als „1“
J	0x4A	0001	Dargestellt als „1“
L	0x4C	0001	Dargestellt als „1“
O	0x4F	0000	Dargestellt als „0“
Q	0x51	1001	Dargestellt als „9“

Tabelle C.1 *(fortgesetzt)*

Zeichen	Zeichencode (hexadezimal)	Dargestellter Wert (binär)	Bemerkungen
V	0x56	1001	Dargestellt als „U“
a	0x61	1010	
b	0x62	1011	
c	0x63	1100	
d	0x64	1101	
e	0x65	1110	
f	0x66	1111	
g	0x67	0000	
h	0x68	0001	
i	0x69	0001	Dargestellt als „1“
j	0x6A	0001	Dargestellt als „1“
k	0x6B	0010	
l	0x6C	0001	Dargestellt als „1“
m	0x6D	0011	
n	0x6E	0100	
o	0x6F	0000	Dargestellt als „0“
p	0x70	0101	
q	0x71	1001	Dargestellt als „9“
r	0x72	0110	
s	0x73	0111	
t	0x74	1000	
u	0x75	1001	
V	0x76	1001	Dargestellt als „U“
w	0x77	1010	
x	0x78	1011	
y	0x79	1100	
z	0x7A	1101	

Tabelle C.1 *(fortgesetzt)*

Zeichen	Zeichencode (hexadezimal)	Dargestellter Wert (binär)	Bemerkungen
ANMERKUNG 1 Die Kleinbuchstaben werden identisch wie Großbuchstaben behandelt. Übermittlungsprobleme bei Klein-/Großbuchstaben (wie beim Buchstabieren der Schlüsselzeichenkette durch Sprache über eine Telefonleitung) werden damit zu einer gültigen Decodierung des Schlüssels führen. ANMERKUNG 2 Alle Zeichen, die zu Zweideutigkeiten führen können (entweder bei Kleinbuchstaben, Großbuchstaben oder sogar in gemischten Fällen) werden neu abgebildet, um so den gleichen Binärwert darzustellen. Z. B. „l“ (Kleinbuchstabe „L“, „i“ (Kleinbuchstabe „l“) und „j“ (Kleinbuchstabe „J“) werden alle als das grafisch ähnliche Zeichen „1“ (Ziffer Eins) dargestellt. ANMERKUNG 3 Jeder der 4-Bit-Binärwerte hat zwei zulässige Zeichen für die Decodierung (z. B. der Wert „1010“ kann entweder als „A“ oder als „W“ decodiert werden). Beide Darstellungen sind äquivalent und gestatten eine bessere Lesbarkeit. Es ist zu beachten, dass die Werte so gewählt wurden, um damit eine Darstellung zu ermöglichen, die vollständig aus Ziffern und Buchstaben besteht, aber auch eine alternative Decodierung unterstützt, die nur aus Symbolen besteht, die ebenfalls über DTMF-Töne übermittelt werden können („0“, …, „9“, „*“, „#“). ANMERKUNG 4 Die Buchstabencodierung mit Zeichen („0“, …, „9“, „A“, …, „F“) wurde ausgewählt, damit die sich ergebende codierte Zeichenkette identisch mit der hexadezimalen Schreibweise des SECRET-Schlüssels in der Byteordnung des Netzwerks ist (siehe Beispiel in Abschnitt C.3). ANMERKUNG 5 Alle Zeichencodes wurden so ausgewählt, dass sie sich mit dem gut bekannten ASCII-Zeichensatz [4] und dem GSM-7-Bit-Alphabet [5] auf einen gemeinsamen Nenner bringen lassen.			

C.2 Prüfsumme für die Verschlüsselung, Formatierung und Entschlüsselung des Shared Secret

Um vor der Anwendung von Shared Secrets mögliche Fehler darin zu erkennen, werden die CRC-16-CCITT-Prüfsummen verwendet. In diesem Abschnitt werden Beispiele für das Prüfsummenverfahren angegeben.

Redundant.

Die CRC-16-CCITT-Berechnung ist durch die folgenden Parameter definiert:

– Polynom: 0x1021

– CRC-Startwert: 0xffff

C.3 Beispiel für die Verschlüsselung, Formatierung und Entschlüsselung des Secret

BEISPIEL Verschlüsselung und Formatierung:

Secret-Schlüssel k = 72101108108111032069078032119111114108 (dezimal)

Erster Schritt: Umformen von k in Binärdarstellung:

k = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001 0101`
`0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101 0111 1100`
(Binärdarstellung, zur besseren Lesbarkeit in Halbbytes angeordnet)

Zweiter Schritt: Bilden der Internet-Prüfsumme von k:

k = `0x36 3e 2b 16 8d bb 5a 95 7d 5f 2b f4 25 a4 5d 7c` (in Hexadezimaldarstellung, um die Darstellung der Byteordnung zu sehen)

---

[4] US-amerikanischer Standardcode für den Informationsaustausch (en: American Standard Code for Information Interchange – ASCII), definiert ist ANSI X3.4-1968.

[5] Digitales zellulares Funkkommunikationssystem (Phase 2+); Alphabete und sprachspezifische Informationen (GSM 03.38, Version 7.2.0, Ausgabe 1998), Abschnitt 6.2.1, „Default alphabet“, ETSI TS 100 900 V7.2.0 (1999-07).

CRC16(k) = 0x3 + 2*0x6 + 0x3 + 2*0xe + 0x2 + 2*0xb + 0x1 +2*0x6 + 0x8 +2*0xd + 0xb +2*0xb + 0x5 + 2*0xa + 0x9 +2*0x5 + 0x7 + 2*0xd + 0x5 + 2*0xf + 0x2 + 2*0xb + 0xf + 2*0x4 + 0x2 + 2*0x5 + 0xa + 2*0x4 + 0x5 + 2*0xd + 0x7 + 2*0xc = 0x0689 (hexadezimal) = 0000 0110 1000 1001 (binär) = 401 (dezimal) = 0x0191 (hex) = 0000 0001 1001 0001 (binär)

Dritter Schritt: Aufsuchen einer Verschlüsselung von k:

363E2B168DBB5A957D5F2BF425A45D7C

ANMERKUNG 1 Aufgrund der gewählten Verschlüsselung des Alphabets mit {„0“, …, „9“, „A“, …, „F“} führt die Verwendung dieser Zeichen bei der Verschlüsselung von k in einer verschlüsselten Zeichenkette zu einer Äquivalenz mit der hexadezimalen Aschreibweise von k in der Byteordnung des Netzwerks, wie es im Beispiel dargestellt ist.

ANMERKUNG 2 Es gibt viele weitere gültige Verschlüsselungen, z. B. MRM*KXHRTZXXPWUPSZP#KX#NKPWNPZSY.

Vierter Schritt: Anwendung von Trennzeichen zur Verbesserung der Lesbarkeit (freigestellt).

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

Fünfter Schritt: Aufsuchen einer Verschlüsselung für CRC16(k):

0191

Sechster Schritt: Anhängen der Verschlüsselung von CRC16(k) an k, wahlweise mit Trennzeichen:

363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-0191

C.4 Bespiel für die Entschlüsselung

Empfangene Zeichenkette s = „363E-2B**l**6-8DBB-5A95-7D5F-2BF4-25A4-5D7C-**o**191“

ANMERKUNG 1 Die Ziffer „1“ wurde durch den Kleinbuchstaben „L“ und die Ziffer „0“ wurde durch den Kleinbuchstaben „O“ ersetzt, um Tippfehler/Lesefehler eines Menschen zu simulieren.

Erster Schritt: Umwandeln jedes Zeichens von s in den entsprechenden 4-Bit-Wert. Wenn ein Zeichen gefunden wird, welches keinen zugewiesenen 4-Bit-Wert besitzt: wenn es ein Trennzeichen ist, dann ignorieren. Anderenfalls wird das Ende der Zeichenkette angenommen und mit dem nächsten Schritt fortgesetzt.

s = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001 0101`
`0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101 0111 1100`
`0000 0001 1001 0001` (In diesem Beispiel dienen die Leerzeichen nur der besseren Lesbarkeit.)

ANMERKUNG 2 Die Tippfehler wurden automatisch wieder angepasst, um die Halbbyte-Werte zu korrigieren.

Zweiter Schritt: Überprüfen der richtigen Länge von s, es sollten 32 + 4 = 36 decodierte Halbbytes insgesamt sein. Falls nicht: Fehlerzustand angeben, dass die empfangene Schlüsselzeichenkette ein falsches Format hat.

Dritter Schritt: Aufteilen von s in seinen Schlüsselteil k (erste 32 Halbbytes) und seinen CRC-Teil crc (letzte 4 Halbbytes).

k = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001 0101`
`0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101 0111 1100`
(Binärdarstellung, zur besseren Lesbarkeit in Halbbytes angegeben.)

crc = 0000 0001 1001 0001 (Binärdarstellung, zur besseren Lesbarkeit in Halbbytes angegeben.)

Vierter Schritt: Umwandeln von k und crc in ihre numerischen Werte:

k = 72101108108111032069078032119111114108 (dezimal)

crc = 401 (dezimal)

Fünfter Schritt: Berechnen von CRC(k) und Prüfen auf Übereinstimmung mit crc. Bei Gleichheit ist k als Secret-Schlüssel zu verwenden, anderenfalls ist ein Fehlerzustand anzugeben, dass die empfangene Schlüsselzeichenkette einen Tippfehler enthält.

CRC16(k) = 0x0191 (hexadezimal, Berechnung exakt wie vor im Verschlüsselungsbeispiel.

0x0191 (hex) = 401 (dezimal) = crc, somit wird die Übermittlung als korrekt angesehen, k darf als Schlüssel verwendet werden.

Anhang D
(informativ)

Beispiele von Anwendungsprotokollen

D.1 Sia

Die folgenden Sia-Blöcke müssen in einer Alarmmeldung mit dem Sia-Protokollbezeichner vorhanden sein:

– # Konto-Block;

– N Neues-Ereignis-Block; oder

– O Altes-Ereignis-Block.

Die Meldung darf zusätzlich den folgenden Block enthalten:

A ASCII-Text

Wenn die Kombination #N, #O, #NA, #OA in der Alarmmeldung enthalten ist, dann wird die Meldung vom Empfänger bestätigt. Das Kopfdatenbyte der Meldung und die Spaltenparität werden NICHT in das SIA-Meldungsformat an die RCT aufgenommen; die Meldung ist bereits an der SPT-Seite überprüft worden und die Unversehrtheit der Meldung wird durch die Hashfunktion sichergestellt.

Ein weiterer Block, wie & (Ursprung), L (Mithören), X (Erweitert), @ (Konfiguration) usw. dürfen in der Meldung vorhanden sein, werden jedoch nicht zwangsläufig vom Empfänger verarbeitet.

Die Blöcke werden durch ein „|“ Zeichen getrennt. Somit wird eine gültige Meldung etwa wie folgt aussehen:

#1234|NCL001|ACenelecMember

#1234|OBA012|AFrontdoor

Im Ereignisblock dürfen sämtliche Änderungen und Textzusätze vorkommen, die in SIA DC-03-1990.01 (R2003.10) festgelegt sind.

D.2 Ademco Contact ID

Die Ademco-Contact-ID-Meldungen, manchmal als POINT ID bezeichnet, haben zwischen SPT und RCT den folgenden Aufbau:

AAAAMTQXYZGGCCC

Dabei ist

AAAA der Kontocode [4 … 6] Ziffern;

MT der Meldungstyp (18 oder 98);

Q der Kennzeichner, Wert 1, 3 oder 6;

XYZ der Ereigniscode;

GG die Gruppennummer;

CCC die Nummer des Melderbereiches.

Die RCT muss prüfen, ob die Länge der Meldung innerhalb des Bereiches [15 … 17] liegt und ob MT gleich 18 oder 98 ist. Der Kontocode muss eine Länge von mindestens 4 Ziffern und maximal 6 Ziffern haben.

Die Ziffern des Kontocodes müssen im Bereich von [„0“, …, „9“] (0x30, …, 0x39) liegen. Der Meldungstyp und der Kennzeichner haben feste Werte, wie vorstehend definiert. Alle weiteren Ziffern müssen im Bereich [„0“, …, „9“ + „“B“ , …, „F“] liegen.

Der Wert der Prüfsumme darf NICHT in der Meldung enthalten sein.

BEISPIEL 123418113101015

Das Konto 1234 berichtet einen Einbruchsalarm an der Grenze von Zone 15 von Teilbereich 1.

Die Länge des Kontocodes [4, 5 oder 6 Ziffern] wird durch die Gesamtlänge der Meldung bestimmt.

D.3 Scancom Fast Format

Die Meldungen im Scancom Fast Format können 8, 16 oder 24 Kanäle enthalten und auch 1 bis 6 Kontoziffern. Das korrekte Format kann vom Empfänger durch Überprüfung der Länge der empfangenen Nachrichtengröße bestimmt werden.

Aufbau einer Scancom-Meldung mit 8 Kanälen:

AAAACCCCCCCCS

Dabei ist

AAAA der Kontocode;

C der Status des Signals (Werte: 1, 2, 3, 4, 5, 6);

S der Systemkanal (Werte: 7, 8, 9).

Der Kontocode kann zwischen 1 … 6 Dezimalziffern variieren.

Die Anzahl der Kanäle kann 8, 16 oder 24 sein.

Der Systemkanal ist immer eine Ziffer.

Die Länge der 8-Kanal-Meldung kann betragen: 10 bis 15 Ziffern.

Die Länge der 16-Kanal-Meldung kann betragen: 18 bis 23 Ziffern.

Die Länge der 24-Kanal-Meldung kann betragen: 26 bis 31 Ziffern.

Alle Bytes müssen im Bereich „0“, …, „9“ liegen.

Der Empfänger wird die Meldung bestätigen, wenn die Größe erwartet wird (innerhalb der vorstehend angegebenen Werte) und alle Bytes haben Werte im richtigen Bereich „0“, …, „9“.

Anhang E
(informativ)

Entwurfsgrundsätze

Dieser Anhang dient der Verdeutlichung einiger Grundsätze, die beim Entwurf dieser Protokollnorm angewendet worden sind.

Der Leser dieser Norm sollte beachten, dass die vorliegende Europäischen Norm etwas von anderen Europäischen Normen abweicht, die sich mit einem anderen Aspekt der Alarmübertragung beschäftigen. Diese Europäische Norm verfolgt die Absicht, im Gegensatz zu anderen, einen exakten Entwurf zu beschreiben, um die Fähigkeit zur Zusammenarbeit zu erreichen und nicht nur die Anforderungen an das Betriebsverhalten zu beschreiben.

E.1 Informationssicherheit

Informationssicherheit ist ein Hauptthema beim Entwurf von Alarmübertragungsanlagen und -geräten. Das Ziel dieser Europäischen Norm besteht darin, einen hohen Grad der Informationssicherheit zu erreichen und gleichzeitig die praktische Ausführung und den Gebrauch kompatibler Geräte so anwendungsfreundlich wie möglich zu halten. Wo es möglich war, wurden bekannte und bewährte Algorithmen und Methoden neuen herstellereigenen Entwürfen vorgezogen.

Es hat sich herausgestellt, dass der schwierigste Teil das Entwerfen einer Inbetriebnahmephase ist, die sowohl sicher und auch noch praktikabel ist. Eine absolute Anforderung besteht darin, die Auswirkung der Beeinträchtigung eines Objekts auf nur dieses eine Objekt zu begrenzen. Dies ist bei vielen über IP arbeitenden Alarmübertragungsprotokollen nicht der Fall.

E.2 Anwendung der UDP-Signalgebung

Als Basis für dieses Protokoll wurde die UDP-Signalgebung ausgewählt, weil sie auf fast jeder Plattform zur Verfügung steht und eine viel bessere Kontrolle über die Übertragung als TCP zulässt. Bei der Signalübertragung ist es wichtig, das Verhalten des Stapelspeichers für den Datenaustausch so präzise wie möglich vorhersagen zu können. Dies wird mit der Anwendung von UDP erreicht.

Die Anwendung von SCTP könnte bei einer späteren Überarbeitung dieser Europäischen Norm in Betracht gezogen werden, aber es ist gegenwärtig für nicht so viele Plattformen wie UDP verfügbar.

Literaturhinweise

CLC/TS 50136-7, *Alarmanlagen – Alarmübertragungsanlagen und –einrichtungen – Teil 7: Anwendungsregeln*

ANSI X3.4:1968, *USA Standard Code for Information Interchange*
American National Standards Institute: New York (1968)

ETSI TS 100 900 V7.2.0 (1999-07), *Digital cellular telecommunications system (Phase 2+) (GSM); Alphabets and language-specific information* (GSM 03.38 version 7.2.0 Release 1998)

ITU-T Recommendation X.509, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*

RFC1071, *Computing the Internet Checksum*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc1071.txt.pdf

RFC1191, *Path MTU Discovery*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc1191.txt.pdf

RFC4086, *Randomness Requirements for Security*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4086.txt.pdf

RFC4347, *Datagram Transport Layer Security*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4347.txt.pdf

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
prEN 50136-1-7

March 2010

ICS 13.320; 33.040.40

English version

Alarm systems - Alarm transmission systems and equipment - Part 1-7: Requirements for common protocol for alarm transmission using packet switched network

To be completed

Alarmanlagen - Alarmübertragungsanlagen und -einrichtungen - Teil 1-7: Anforderungen an standardisierte Protokolle zur Alarmübertragung in Paketvermittlungsnetzwerken

This draft European Standard is submitted to CENELEC members for CENELEC enquiry.
Deadline for CENELEC: 2010-09-03.

It has been drawn up by CLC/TC 79.

CENELEC

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

Central Secretariat: Avenue Marnix 17, B - 1000 Brussels

1 # Foreword

2 This draft European Standard was prepared by the Technical Committee CENELEC TC 79, Alarm
3 systems. It is submitted to the CENELEC enquiry.

Contents

1 Scope **5**
2 Normative references **5**
3 Terms, definitions and abbreviations **5**
3.1 Terms and definitions 5
3.2 Abbreviations 6
4 Objective **7**
5 Messaging **7**
5.1 Message format overview 7
5.2 Padding and message length 12
5.3 Hashing 13
5.4 Encryption 13
5.5 Timeouts and retries 14
5.6 Version number 15
5.7 Reverse commands 15
5.8 Initial values 16
6 Message types **16**
6.1 Path supervision 16
6.2 Event reporting 17
6.3 Configuration messages 21
7 Commissioning and connection setup **30**
7.1 Commissioning 30
7.2 Connection setup 35
Annex A (normative) **Result codes** **37**
Annex B (normative) **Protocol Identifiers** **38**
Annex C (normative) **Shared secret** **39**
C.1 Character set for secret encoding, formatting and decoding 39
C.2 Checksum for shared secret encoding, formatting and decoding 42
C.3 Example of secret encoding, formatting and decoding 43
C.4 Example decoding 43
Annex D (informative) **Examples of application protocols** **45**
D.1 Sia 45
D.2 Ademco Contact ID 45
D.3 Scancom fast format 46
Annex E (informative) **Design principles** **47**
E.1 Information security 47
E.2 Use of UDP signalling 47
Bibliography **48**

43 **Tables**

44 Table 1 – Identifiers 8
45 Table 2 – Basic unencrypted format of messages 8
46 Table 3 – Basic encrypted format of messages 9
47 Table 4 – Message ID overview 11
48 Table 5 – Flags 12
49 Table 6 – Hashing ID's 13
50 Table 7 – Encryption ID's 13
51 Table 8 – Reverse commands 15
52 Table 9 – Initial values 16
53 Table 10 – Poll message SPT ← → RCT 16
54 Table 11 – Poll response RCT ← → SPT 17
55 Table 12 – Event message format – SPT → RCT 17
56 Table 13 – Event message format – Fields 18
57 Table 14 – Event field 18
58 Table 15 – Time event field 18
59 Table 16 – Time message field 19
60 Table 17 – Link field – IP Address 19
61 Table 18 – Link field – IP Port number 20
62 Table 19 – Link field – URL 20
63 Table 20 – Link field – Filename 20
64 Table 21 – Event response message format 20
65 Table 22 – Connection handle request message format 21
66 Table 23 – Connection handle response message format 22
67 Table 24 – Device ID request message format 22
68 Table 25 – 'Master Device ID request' flag 23
69 Table 26 – Device ID response message format 23
70 Table 27 – Encryption selection request message format 24
71 Table 28 – 'Master Encryption Selection request' flag 24
72 Table 29 – Encryption selection response message format 24
73 Table 30 – Encryption key exchange request message format 25
74 Table 31 – 'Master Key request' flag 25
75 Table 32 – Encryption key exchange response message format 25
76 Table 33 – Hash selection request message format 26
77 Table 34 – Hash selection response message format 26
78 Table 35 – Path supervision request message format 27
79 Table 36 – Path supervision response message format 27
80 Table 37 – Set time command message format 27
81 Table 38 – Set time response message format 28
82 Table 39 – Transparent message format 28
83 Table 40 – Transparent response format 28
84 Table 41 – DTLS completed request message format 29
85 Table 42 – DTLS completed response message format 29
86 Table 43 – RCT IP parameter request message format 29
87 Table 44 – RCT IP parameter response message format 30
88 Table 45 – Message flow during the commissioning of a new SPT 32
89 Table 46 – Message flow during connection setup 36
90 Table A.1 – Result codes 37
91 Table B.1 – Protocol identifiers 38
92 Table C.1 – Character set 40

93

94

1 Scope

This European Standard 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 and EN 50136-1-5.

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

2 Normative references

The following referenced documents are indispensable for the application 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 50136-1:201X [1)], *Alarm systems – Alarm transmission systems – Part 1: General requirements for alarm transmission systems*

EN 50136-1-5, *Alarm systems – Alarm transmission systems and equipment – Part 1-5: Requirements for Packet Switched Network PSN*

NIST SP 800-38A, *Recommendation for Block Cipher Modes of Operation – Methods and Techniques.* NIST Special Publication 800-38A, December 2001.
Available from http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

RFC793, *Transmission Control Protocol*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc793.txt.pdf

RFC958, *Network Time Protocol (NTP)*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc958.txt.pdf

RFC4330, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4330.txt.pdf

SIA DC-03-1990.01 (R2003.10), *DCS SIA Format Standard*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions defined in EN 50136-1:201X apply.

[1)] At draft stage.

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 European Standard 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.

Any other alarm transmission protocol or equipment not covered by this European Standard may be used, provided that the requirements of EN 50136-1 are met.

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

5 Messaging

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 up to 10 000 SPTs according to EN 50136-1 category D3 simultaneously, 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.

5.1 Message format overview

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).

5.1.1 Identifiers

The following identifiers exist:

Table 1 – Identifiers

Description	Purpose	Present in	Encrypted	See
Connection Handle	Look up the current symmetric encryption key	All messages	No	5.1.3
Device ID	Uniquely identify the hardware	Contributing to hashes in all messages	N/A	5.1.4

The Connection Handle is unencrypted. It is a randomly chosen 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 (this is not an account code) once the connection has been established. The RCT shall use the Device ID as index to its SPT database. 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.

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

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

5.1.2 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 2 – Basic unencrypted format of messages

Byte index	Bytes	Description	See	Group
0	4	Connection Handle	5.1.3	Header
4	16	Device ID	5.1.4	
20	2	Tx Sequence number	5.1.7	
22	2	Rx Sequence number	5.1.7	
24	2	Flags	5.1.8	
26	1	Protocol version number		
27	1	Message ID	5.1.5	Message
28	2	Message Length	5.1.6	
30	n	Message Data	Clause 6	

The basic encrypted, transmitted format of all messages is as follows. 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	5.1.3	No	Header
4	2	Tx Sequence number	5.1.7	Yes	
6	2	Rx Sequence number	5.1.7	Yes	
8	2	Flags	5.1.8	Yes	
10	1	Protocol version number		Yes	
11	1	Message ID	5.1.5	Yes	Message
12	2	Message Length	5.1.6	Yes	
14	n	Message Data	Clause 6	Yes	
14 + n		Padding	5.2.1	Yes	Tail
	32 32	Hash – SHA-256, or Hash – RIPEMD-256	5.3	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.

5.1.3 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.

5.1.4 Device ID

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.

5.1.4.1 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 assured by the following principles.

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

5.1.4.2 RCT Device ID

The Device ID of the RCT is an ID that is unique within the receiver and never changed within the lifetime of a receiver. It represents a serial number of the RCT, only that it can be initially setup by the ARC.

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

5.1.5 Message ID

The Message ID’s as used are listed in the following table.

260 **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
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
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

261

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

264 **5.1.6 Message length**

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

267 – in variable length messages (see for example 6.2.1 and 6.3.15) to check for the end of data;

268 – to be able to determine the start of an embedded reverse command (see 5.7).

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

5.1.7 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 any replay attacks.

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

5.1.8 Flags

The following flags are defined:

Table 5 – Flags

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

5.2 Padding and message length

5.2.1 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 data. Random bytes are appended to the actual messages data until the total message length is one of those as specified in the next subclause.

5.2.2 Message length

The message lengths as used fulfil the requirements as mentioned in 5.2.1 (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.

5.3 Hashing

The following methods of message validation are supported:

Table 6 – Hashing ID's

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

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

SPTs must 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 6.3.9 and 6.3.10.

The hashing method to be used is negotiated during session initialization, using the messages as defined in 6.3.9 and 6.3.10.

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.

5.4 Encryption

5.4.1 Encryption method

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:

Table 7 – Encryption ID's

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 must at least implement the default method, but can implement all methods. The default method is 1 (AES-128) until explicitly updated using the messages as defined in 6.3.5 and 6.3.6.

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).

5.4.2 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 ongoing session key exchange action, the RCT must maintain the old session key until the first successful transmission of a packet with the new session key is acknowledged.

5.5 Timeouts and retries

The timeouts (after which a message will be retried) will increase with each retry as defined in RFC793.

In addition to RFC793, the resulting time-out value is upper-bound by an absolute maximum value of 100 s plus/minus an evenly randomly distributed time offset of 10 %.

NOTE RFC793 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 100 s upper bound is changed by an evenly distributed random component.

The random component must be based on a 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.

5.6 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.

5.7 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 5.1.8).

The Message ID and the Message Data will be added to the message as follows:

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 shall within the Response message 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.

5.8 Initial values

The following values 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	Not set yet
Hash	0	SHA-256
Encryption ID	1	AES-128
Heartbeat interval time	0	No Polling
TX sequence counter	random	
RX sequence counter	0	No packet received yet

6 Message types

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.

6.1 Path supervision

This subclause 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 6.3.11. 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.

6.1.1 Poll message

The Poll message has the following format:

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 (6.3.11/6.3.12) during connection setup.

6.1.2 Poll response

The Poll response message has the following format:

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

6.2 Event reporting

6.2.1 Event message format

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 following format:

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 following fields 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.

6.2.1.1 Event field

SPT: Mandatory
RCT: Mandatory

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>

6.2.1.2 Time event field

SPT: Optional
RCT: Mandatory

Table 15 – Time event field

Relative Byte Index	Bytes	Description
0	8	Time format according to RFC958 (NTP) / RFC4330 (SNTP V4)

This field holds the timestamp on which the event occurred.

Time format is a 64 bit integer as described in RFC958 (NTP) / RFC4330 (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, etc.) 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.

6.2.1.3 Time message field

SPT: Optional
RCT: Mandatory

Table 16 – Time message field

Relative Byte Index	Bytes	Description
0	8	Time format according to RFC958 (NTP) / RFC4330 (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.

6.2.1.4 Link field – IP Address

SPT: Optional
RCT: Optional

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.

6.2.1.5 Link field – IP Port number

SPT: Optional
RCT: Optional

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.

6.2.1.6 Link field – URL

SPT: Optional
RCT: Optional

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.

6.2.1.7 Link field – Filename

SPT: Optional
RCT: Optional

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.

6.2.2 Event response format

The Event response message has the following format:

RCT → SPT

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.

6.3 Configuration messages

This subclause 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.

6.3.1 Connection handle request

The Connection handle request message has the following format:

SPT → RCT

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.

6.3.2 Connection handle response

The Connection handle response message has the following format:

RCT → SPT

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.

6.3.3 Device ID request

The Device ID request message has the following format:

SPT → RCT

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, which is a random number. The Device ID 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 Device ID.

The following applies to allow for 2nd channel commissioning:

– when the ‘Master Device ID request’ flag is set, the SPT requests a new Device ID only once, and stores this new Device ID as received in the reply message in NVM. Also the RCT stores this Device ID;

– when the ‘Master Device ID request’ flag is cleared and the ‘Device ID’ field is not equal to 0, the SPT already has a ‘Master Device ID’ and informs the RCT about this. The RCT will return the same Device ID in its response.

Otherwise the ‘Master Device ID request’ flag is cleared and the ‘Device ID’ field is equal to 0.

559 Flags:

560 **Table 25 – 'Master Device ID request' flag**

Bit	Description
0	Master Device ID request
1	0: SPT Device ID 1: RCT Device ID
2…7	Unused

561

562 ### 6.3.4 Device ID response

563 The Device ID response message has the following format:

564 RCT → SPT

565 **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

566

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

568 The Device ID in the header of this message itself and previous messages are:

569 – 0 in case of DTLS;

570 – the ID of the 'one-time-pad'/2nd channel setup.

571 The Flags field holds the value 0.

572 ### 6.3.5 Encryption selection request

573 The Encryption selection request message has the following format:

574 SPT → RCT

575 **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

576

577 This message is issued during commissioning by the SPT to indicate the encryption methods it
578 supports. See 5.4 for possible encryption methods.

579 Flags:

580 **Table 28 – 'Master Encryption Selection request' flag**

Bit	Description
0	Master Encryption Selection request
1…7	Unused

581

582 **6.3.6 Encryption selection response**

583 The Encryption selection response message has the following format:

584 RCT → SPT

585 **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

586

587 The Flags field holds the value 0.

588 **6.3.7 Encryption key exchange request**

589 The Encryption key exchange request message has the following format:

590 SPT ← → RCT

591 **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 256 bits -> 32 bytes)
HL + 1 + L		Padding
		Hash

592

593 Flags:

594 **Table 31 – 'Master Key request' flag**

Bit	Description
0	Key Request (SPT) / Key Push (RCT)
1	Master Key Request
2…7	Unused

595

596 This message is issued to request an encryption key update. Both SPT and RCT can request an
597 encryption key update, in which case the 'Master Key Request' flag is set. When this flag is set the
598 Encryption key holds the key currently in use.

599 The SPT requests a new key from the RCT. The new key shall be created by the RCT instead of the
600 SPT, as it shall be generated using a cryptographically strong random number generator, and the
601 random generator of the RCT is usually of much better quality than the one of the SPTs.

602 The RCT can push a new session key to the SPT by setting the 'Key Request' flag. The new key is in
603 the 'Encryption key' field. The SPT will then acknowledge by replying back this key in the Encryption
604 key exchange response message.

605 ### 6.3.8 Encryption key exchange response

606 The Encryption key exchange response message has the following format:

607 SPT ← → RCT

608 **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 256 bits -> 32 bytes)
HL + 2 + L		Padding
		Hash

609

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

6.3.9 Hash selection request

The Hash selection request message has the following format:

SPT → RCT

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 Hashes it supports. See 5.3 for possible hash functions.

6.3.10 Hash selection response

The Hash selection response message has the following format:

RCT → SPT

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 (after reboot) the Internet Checksum (value 1) is used as Hash.

6.3.11 Path supervision request

The Path supervision request message has the following format:

SPT → RCT

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.

6.3.12 Path supervision response

The Path supervision response message has the following format:

RCT → SPT

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

6.3.13 Set time command

The Set time command message has the following format:

RCT → SPT

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 RFC958 (NTP) / RFC4330 (SNTP V4)
HL + 8		Padding
		Hash

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

6.3.14 Set time response

The Set time response message has the following format:

SPT → RCT

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

6.3.15 Transparent message

The Transparent message has the following format:

Table 39 – 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.

6.3.16 Transparent response

The Transparent response has the following format:

Table 40 – 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

6.3.17 DTLS completed request

The DTLS completed request message has the following format:

SPT → RCT

Table 41 – 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.

6.3.18 DTLS completed response

The DTLS completed response message has the following format:

RCT → SPT

Table 42 – 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.

6.3.19 RCT IP parameter request

The RCT parameter request message has the following format:

SPT → RCT

Table 43 – 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 standard.

690 Implementation of this message is optional for the SPT.

691 **6.3.20 RCT IP parameter response**

692 The RCT IP parameter response message has the following format:

693 RCT → SPT

694 **Table 44 – 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 6.2.1.4
HL + 3	2	Field Length (L1)
HL + 5	L1	Field Data
HL + 5 + L1	1	Field Identifier – RCT 1 Port number – see 6.2.1.5
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 6.2.1.4
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

695

696 # 7 Commissioning and connection setup

697 ## 7.1 Commissioning

698 The objective of the commissioning procedure is to enable the Supervised Premises Transceiver and
699 the Receiving Centre Transceiver to mutually authenticate each other.

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

701 – Master Device ID's of SPT and RCT;

702 – Master Encryption Key;

703 – Master Encryption Selection;

704 – (optional) RCT IP Address(es) and Port(s) with which the SPT should communicate (this allows
705 for a separate 'commissioning server' to handle the 'initial contact' for multiple receivers. In this
706 situation the commissioning server will have to securely transfer the session parameters to the
707 appropriate RCT. The mechanism for doing this is outside the scope of this protocol).

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

7.1.1 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 7.1.4).

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.

7.1.2 Commissioning message sequence

The ‘Master Set’ is exchanged used the message flow as described below. The messages are the same, irrespective 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 follows:

726 **Table 45 – Message flow during the commissioning of a new SPT**

SPT	Direction	RCT	Remarks
			The hash to start with is the Internet Checksum
CONN_HANDLE_REQ	→		
	←	CONN_HANDLE_RESP	
DEVICE_ID_REQ	→		Master Device ID flag set SPT Device ID flag set
	←	DEVICE_ID_RESP	
DEVICE_ID_REQ	→		Master Device ID flag set RCT Device ID flag set
	←	DEVICE_ID_RESP	
SESSION_KEY_REQ	→		Master Key request flag set
	←	SESSION_KEY_RESP	
ENCRYPT_SELECT_REQ	→		Master Encryption Selection request flag set
	←	ENCRYPT_SELECT_RESP	
			Key update complete, proceed using new encryption key and method
DTLS_COMPLETE_REQ	→		Only when using X.509/DTLS
	←	DTLS_COMPLETE_RESP	

727

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

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

730 **7.1.3 Commissioning using Shared Secret**

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

733 For this procedure, the RCT will generate a SECRET which consists of 2 ‘tokens’/‘text messages’ (see
734 7.1.3.1 for requirements on the SECRETS). These represent the Device ID and the Encryption key.

735 The ‘tokens’/‘text messages’ will be used by both the RCT and SPT to generate the following
736 deliverables:

737 – the 16-byte Device ID’s (SPT and RCT);

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

739 These are generated by taking the SHA-256 hash of the tokens (Device ID: SPT highest 16 bytes,
740 RCT lowest 16 bytes).

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 sent from the RCT to SPT, the session will be deleted and never re-used.

For Shared Secret commissioning, the 'one-time-pad' Device ID, Encryption key and Encryption selection are used to establish the initial connection. After commissioning is completed, the token is not accepted by the RCT.

Next, these parameters are renewed (SPT uses the Master Device ID and Master Key request flags), and stored into non-volatile memory as the new 'Master set'. This new 'Master set' will be used to reconnect after disconnects or power failures.

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

– Physical and logical means of 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 SECRET. The following methods may be considered appropriate depending on the security level of protected premises:

  - ARC operator dictates the SECRET to the field technician over the phone;
  - the SECRET is transmitted using SMS;
  - the SECRET is sent in an encrypted and signed e-mail;
  - the SECRET is printed at the Management centre / ARC and the field technician brings it to the protected premises himself;
  - the SECRET is programmed into SPT at the Management centre / ARC and then transported to the protected premises;
  - the 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 SECRET vs. the security level of the protected premises.

– The SECRET shall not be sent over a channel which is used for communication between the SPT and RCT for alarm reporting and monitoring.

– The SECRET shall be generated using cryptographically strong random number generator [2)].

– The SECRET shall be represented as text composed of printable characters from the character set described in Clause C.1.

[2)] See RFC4086.

– To cope with potential typos and other human typical transmission errors, the text representation of the SECRET is extended by a 16 bit checksum, calculated as weighted CRC as described in Clause C.2, directly appended to the SECRET string and encoded in the same way as the SECRET string.

– The strength of the SECRET must be equivalent to 128 bit. As each of the allowed encoding characters represents a chunk of 4 bits, and the SECRET is protected by an additional 16 bit CRC, the whole SECRET is encoded by (128 + 16) / 4 = 36 characters. An example is given in Clause C.3.

7.1.4 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 Device ID 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 European Standard;

– 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 [3)]. 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.

[3)] The length of the supplier identifier and supplier specific identifier to be agreed and specified.

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:

– Connection handle;

– Device ID SPT (Authentication);

– Device ID RCT (Authentication);

– Session key;

– Encryption selection;

– Hash;

– Path supervision;

– Protocol version level mutually agreed by SPT and RCT.

NOTE 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 initialization phase (it should remain constant throughout the exchange even if the secured premises have the most restrictive stateful firewall).

The message flow during connection setup (to request the session parameters) is as follows:

840 **Table 46 – Message flow during connection setup**

SPT	Direction	RCT	Remarks
			The hash to start with is the Internet Checksum
CONN_HANDLE_REQ	◊		Omitted if this follows directly after commissioning, in which case the previously established Connection Handle stays active
	⇓	CONN_HANDLE_RESP	
DEVICE_ID_REQ	◊		Master Device ID flag cleared SPT Device ID flag set
	⇓	DEVICE_ID_RESP	
DEVICE_ID_REQ	◊		Master Device ID flag cleared RCT Device ID flag set
	⇓	DEVICE_ID_RESP	
SESSION_KEY_REQ	◊		Master Key request flag cleared
	⇓	SESSION_KEY_RESP	
ENCRYPT_SELECT_REQ	◊		Master Encryption Selection request flag cleared
	⇓	ENCRYPT_SELECT_RESP	
			Key update complete, proceed using new key
HASH_SELECT_REQ	◊		
	⇓	HASH_SELECT_RESP	
PATH_SUPERVISION_REQ	◊		
	⇓	PATH_SUPERVISION_RESP	
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.
			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	

841

842 The SPT starts with the ‘Master set’ which it holds in NVM. The Master Device ID, Master Encryption
843 Selection request and Master Key request flags are cleared. The ‘Master set’ is thus used only to
844 obtain new session parameters. Typically only the session key changes. The Device ID identifies SPT
845 and RCT and does not change.

Annex A
(normative)

Result codes

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 summarizes the possible Protocol Identifiers for application layer protocol carried by the protocol defined in this European Standard.

Each compatible implementation of this protocol must 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;

– Sia DC-03 message structures 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 datafield

A manufacturer wishing to send messages that don't 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 European Standard.

Annex C
(normative)

Shared secret

C.1 Character set for secret encoding, formatting and decoding

When encoding and formatting the master key secret into a string format, readable for human beings, characters may only be used from the following character set and as described.

– **Encoding** of the key value (and its appended checksum) itself must use the characters {'0',…,'9'} + {'A',…,'F'} + {'G', 'H', 'K', 'M', 'N', 'Q',…,'U', 'W', 'X',…,'Z'} + {'*', '#'} only. Each of these characters thereby represents a four-bit value as indicated in the character set table (see Table C.1).

– **Formatting** of the key value to improve the readability for humans must use one of the explicitly named separator symbols {'-'} or {space}. 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.

– **All other characters** are not to be used explicitly for key encoding. If they occur during decoding, then one of two cases is to be considered:

  - if the character has been assigned a four-bit value in the character set table, then an ambiguity error is assumed and the assigned value is silently used;
  - if the character has not been assigned a four-bit value in the character set table, then the end of the key string is assumed and internal key processing may start.

Character mapping is defined in the following table.

889 **Table C.1 – Character set**

Character	Character-Code (hex)	Represented value (binary)	Comment
0	0x30	0000	
1	0x31	0001	
2	0x32	0010	
3	0x33	0011	
4	0x34	0100	
5	0x35	0101	
6	0x36	0110	
7	0x37	0111	
8	0x38	1000	
9	0x39	1001	
A	0x41	1010	
B	0x42	1011	
C	0x43	1100	
D	0x44	1101	
E	0x45	1110	
F	0x46	1111	
G	0x47	0000	
H	0x48	0001	
K	0x4B	0010	
M	0x4D	0011	
N	0x4E	0100	
P	0x50	0101	
R	0x52	0110	
S	0x53	0111	
T	0x54	1000	
U	0x55	1001	
W	0x57	1010	
X	0x58	1011	
Y	0x59	1100	
Z	0x5A	1101	
*	0x2A	1110	
#	0x23	1111	
Space	0x20		Separator
!	0x21	0001	Mapped to "1"
"	0x22		
$	0x24		

890

891 **Table C.1 – Character set** *(continued)*

Character	Character-Code (hex)	Represented value (binary)	Comment
%	0x25		
&	0x26		
'	0x27		
(	0x28		
)	0x29		
+	0x2B		
,	0x2C		
-	0x2D		Separator
.	0x2E		
/	0x2F		
:	0x3A		
;	0x3B		
<	0x3C		
=	0x3D		
>	0x3E		
?	0x3F		
I	0x49	0001	Mapped to "1"
J	0x4A	0001	Mapped to "1"
L	0x4C	0001	Mapped to "1"
O	0x4F	0000	Mapped to "0"
Q	0x51	1001	Mapped to "9"
V	0x56	1001	Mapped to "U"
a	0x61	1010	
b	0x62	1011	
c	0x63	1100	
d	0x64	1101	
e	0x65	1110	
f	0x66	1111	
g	0x67	0000	
h	0x68	0001	
i	0x69	0001	Mapped to "1"
j	0x6A	0001	Mapped to "1"
k	0x6B	0010	
l	0x6C	0001	Mapped to "1"
m	0x6D	0011	
n	0x6E	0100	
o	0x6F	0000	Mapped to "0"

892

893 **Table C.1 – Character set** *(continued)*

Character	Character-Code (hex)	Represented value (binary)	Comment
p	0x70	0101	
q	0x71	1001	Mapped to "9"
r	0x72	0110	
s	0x73	0111	
t	0x74	1000	
u	0x75	1001	
v	0x76	1001	Mapped to "U"
w	0x77	1010	
x	0x78	1011	
y	0x79	1100	
z	0x7A	1101	

NOTE 1 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 2 All characters which could lead to ambiguities (either in lower, upper or even mixed case scenarios) have been re-mapped to represent the same binary value. E.g., “l” (lower case “L”), “i” (lower case “I”) and “j” (lower case “J”) have all been mapped to their graphically similar character “1” (digit one).

NOTE 3 Each of the four-bit binary values has two allowed characters for encoding (e.g.: value “1010” can be encoded either by “A” or by “W”). Both representations are equivalent and allow to improve readability. Note that the values have been chosen to allow thereby a representation consisting completely by digits and letters only, but also support an alternative encoding, consisting only out of symbols which can be transmitted by DTMF tones {‘0’,…,’9’, ‘*’, ‘#’} as well.

NOTE 4 Character encoding with characters {‘0’,…,’9’, ‘A’,…’F’} has been chosen such that the resulting encoding string is identical to the SECRET key’s hexadecimal representation in Network byte order (see example in Clause C.3).

NOTE 5 All character codes have been chosen to be a common denominator of well-known ASCII-character set [4] and GSM-7bit-alphabet [5].

894

895 C.2 Checksum for shared secret encoding, formatting and decoding

896 CRC-16-CCITT Checksums are used to detect possible errors in shared secrets before they are used.
897 This clause provides examples of the cheksuming procedure.

898 redundant.

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

900 – Polynomial: 0x1021

901 – Initial crc value: 0xffff

[4] American Standard Code for Information Interchange (ASCII), defined by ANSI X3.4-1968.

[5] Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information (GSM 03.38 version 7.2.0 Release 1998)”, chapter 6.2.1, “Default alphabet”, ETSI TS 100 900 V7.2.0 (1999-07).

C.3 Example of secret encoding, formatting and decoding

EXAMPLE ENCODING AND FORMATTING:

Secret key k = 721011081081110320690780321191111114108 (decimal)

First step: translate k into binary:

k = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001`
`0101 0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101`
`0111 1100` (binary, grouped into nibbles for better readability)

Second step: build Internet checksum of k:

k = `0x36 3e 2b 16 8d bb 5a 95 7d 5f 2b f4 25 a4 5d 7c` (in hex, to see byte order representation)

CRC16(k) = 0x3 + 2*0x6 + 0x3 + 2*0xe + 0x2 + 2*0xb + 0x1 +2*0x6 + 0x8 +2*0xd + 0xb +2*0xb + 0x5 +2*0xa + 0x9 +2*0x5 + 0x7 +2*0xd + 0x5 +2*0xf + 0x2 +2*0xb + 0xf +2*0x4 +0x2 +2*0x5 + 0xa +2*0x4 + 0x5 +2*0xd + 0x7 +2*0xc = 0x0689 (hexadecimal) = 0000 0110 1000 1001 (binary) = 401 (decimal) = 0x0191 (hex) = 0000 0001 1001 0001 (binary)

Third step: look up an encoding for k:

363E2B168DBB5A957D5F2BF425A45D7C

NOTE 1 Due to the chosen alphabet's encoding of {'0',…,'9', 'A',…'F'}, using these characters to encode k results in an encoding string being equivalent to k's hexadecimal notation in Network byte order, as can be seen in the example.

NOTE 2 There are multiple possible other valid encodings, e.g.: MRM*KXHRTZXXPWUPSZP#KX#NKPWNPZSY.

Fourth step: (optionally) use separators to improve readability:

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

Fifth step: look up an encoding for CRC16(k):

0191

Sixth step: Append encoding of CRC16(k) to k, optionally using separators:

363E-2B16-8DBB-5A95-7D5F-2BF4-25A4-5D7C-0191

C.4 Example decoding

received string s = "363E-2B**l**6-8DBB-5A95-7D5F-2BF4-25A4-5D7C-**o**191"

NOTE 1 The digit "1" has been replaced by a lower case letter "L" and the digit "0" has been replaced by a lower case letter "O" to simulate human typos / misreadings.

First step: Convert each character of s by the four-bit value it represents. If a character is encountered which does not have a four-bit value assigned: If it is a separator, then ignore it. Otherwise, assume end of string and continue with next step.

s = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001`
`0101 0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101`
`0111 1100 0000 0001 1001 0001` (spaces in between just for better readability of this example)

NOTE 2 The typos automatically have been re-matched to the correct nibble values.

Second step: Check proper length of s, should be 32 + 4 = 36 decoded nibbles in total. If not: raise error condition that received key string has wrong format.

Third step: Cut s into its key part k (first 32 nibbles) and its CRC part crc (last 4 nibbles):

k = `0011 0110 0011 1110 0010 1011 0001 0110 1000 1101 1011 1011 0101 1010 1001`
`0101 0111 1101 0101 1111 0010 1011 1111 0100 0010 0101 1010 0100 0101 1101`
`0111 1100` (binary, grouped into nibbles for better readability)

crc = 0000 0001 1001 0001 (binary, grouped into nibbles for better readability)

Fourth step: Convert k and crc into their numerical values:

k = 72101108108111032069078032119111114108 (decimal)

crc = 401 (decimal)

Fifth step: Calculate CRC(k) and check against crc. If equal then use k as secret key, otherwise raise error condition that received key string contains a typo.

CRC16(k) = 0x0191 (hexadecimal, calculation exactly as before in encoding example)

0x0191 (hex) = 401 (decimal) = crc , thus transmission assumed to be correct, k can be used as key.

Annex D
(informative)

Examples of application protocols

D.1 Sia

The following SIA blocks shall be present into an alarm message with the SIA protocol identifier:

– # Account block;

– N New event block; or

– O Old event block.

Additionally the message may contain the following block:

A ASCII text

If the combination: #N, #O, #NA, #OA is present in the alarm message, the message will be acknowledged by the receiver. The message header byte and the column parity will NOT be included in the SIA message format to the RCT, the message is already validated at the SPT side and the integrity of the message is guaranteed by the hash.

Other block like: & (origin), L (listen in), X (extended), @ (configuration) etc may (in addition to the above mentioned blocks) exists in the message but will not necessarily be processed by the receiver.

Blocks will be separated by a '|' sign. Thus a valid message will look like

#1234|NCL001|ACenelecMember

#1234|OBA012|AFrontdoor

All modifiers and textual additions as specified in SIA DC-03-1990.01 (R2003.10) may occur in the event block.

D.2 Ademco Contact ID

The Ademco Contact ID messages, sometimes called POINT ID, have the following layout between SPT and RCT:

AAAAMTQXYZGGCCC

where

AAAA Account code [4…6] digits;

MT Message type (18 or 98);

Q Qualifier, value 1, 3 or 6;

XYZ Event code;

GG Group number;

CCC Zone number.

The RCT shall check if the length of the message is within range: [15…17] and the MT equals 18 or 98. The account code shall be 4 digits long minimum and 6 digits maximum.

Account code digits shall be in the range ['0'…'9'] (0x30…0x39). Message type and Qualifier have fixed values as defined above. All other digits shall be in the range: ['0'…'9' + 'B'…'F']

The checksum value shall NOT be present in the message.

EXAMPLE 123418113101015

Account 1234 is reporting a Perimeter Burglary Alarm on Zone 15 of Partition 1.

The length of the account code [4, 5, or 6 digits] will be determined by the total message size.

D.3 Scancom fast format

The Scancom Fast Format message can contain 8, 16 or 24 channels and also 1 up to 6 account digits. The correct format can be determined by the receiver just by checking the length of the received message size.

Layout of 8 channels scancom message:

AAAACCCCCCCCS

where

AAAA Account code;

C Status of the channel (values: 1, 2, 3, 4, 5, 6);

S System channel (values: 7, 8, 9).

The account code can vary between 1 … 6 decimal digits.

The number of channels can be: 8, 16 or 24.

The system channel is always 1 digit.

The length of an 8 channels message then can be: 10 up to 15 digits.

The length of a 16 channels message then can be: 18 up to 23 digits.

The length of a 24 channels message then can be: 26 up to 31 digits.

All bytes shall be in the range: '0' … '9'.

The receiver will acknowledge the message if the size is expected (within the above values) and all bytes have the values in the correct range: '0'…'9'.

Annex E
(informative)

Design principles

This annex is added to clarify some of the principles used to design this protocol standard.

The reader of the standard should note, that this European Standard is somewhat different from other European Standards dealing with different aspect of alarm transmission. This European Standard, unlike others, is intended to describe an exact design to achieve interoperability rather than to describe requirements for performance only.

E.1 Information security

Information security is major concern when designing alarm transmission systems and equipment. The intention of this European Standard is to achieve high level of information security while keeping the implementation and use of compatible equipment as convenient as possible. Wherever possible, known and proven algorithms and methodology was chosen over new proprietary designs.

The commissioning phase is found to be the hardest part to design in a way that is secure and still practical. An absolute requirement there is to limit the effect of compromising one site to that site only. This is not the case in many other alarm transmission protocols working over IP.

E.2 Use of UDP signalling

UDP signalling was chosen as base to this protocol because it is available for almost any platform, and it allows much better control over the transmission than TCP. In alarm transmission it is important to be able to predict the behaviour of the communication stack as precisely as possible. This is achieved with the use of UDP.

At some later date one could consider use of SCTP for a later revision of this European Standard, but as today it is not as commonly available for as many platforms as UDP.

Bibliography

CLC/TS 50136-7 , *Alarm systems – Alarm transmission systems and equipment – Part 7: Application guidelines*

ANSI X3.4:1968, *USA Standard Code for Information Interchange*
American National Standards Institute: New York (1968)

ETSI TS 100 900 V7.2.0 (1999-07), *Digital cellular telecommunications system (Phase 2+) (GSM); Alphabets and language-specific information* (GSM 03.38 version 7.2.0 Release 1998)

ITU-T Recommendation X.509, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*

RFC1071, *Computing the Internet Checksum*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc1071.txt.pdf

RFC1191, *Path MTU Discovery*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc1191.txt.pdf

RFC4086, *Randomness Requirements for Security*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4086.txt.pdf

RFC4347, *Datagram Transport Layer Security*
Available from http://www.faqs.org/ftp/rfc/pdf/rfc4347.txt.pdf