SIA Digital Communication Standard – Internet Protocol Event Reporting

ANSI/SIA DC-09-2013

ANSI Logo only applies after approval

Sponsor
The Security Industry Association

Publication Order Number: xxxxx

Printed in the United States of America
Published by
The Security Industry Association
8405 Colesville Road, Ste 500, Silver Spring, MD 20910

Contents

Foreword

This standards document is published by the Security Industry Association (SIA) and was developed and adopted by a consensus of industry volunteers in accordance with SIA's standards development policies and procedures. It is intended to facilitate product compatibility and interchangeability, to reduce misunderstandings between manufacturers and purchasers, and to assist purchasers in obtaining the proper products to fulfill their particular needs.

The existence of this or any SIA standards document shall not prevent any SIA member or non-member from manufacturing, selling, or using products not conforming to this or any SIA standard. SIA standards are voluntary. SIA encourages the use of this document but will not take any action to ensure compliance with this or any other SIA Standard. SIA assumes no responsibility for the use, application or misapplication of this document. Industry members using this document, particularly those having participated in its development and adoption, are considered by SIA to have waived any right they might otherwise have had to assert claims against SIA regarding the development process of this standard.

Although some SIA standards establish minimum performance requirements, they are intended neither to preclude additional product features or functions nor to act as a maximum performance limit. Any product the specifications of which meet the minimum requirements of a SIA standard shall be considered in compliance with that standard. Any product the specifications of which exceed the minimum requirements of a SIA standard shall also be considered in compliance with the standard, provided that such product specifications do not exceed any maximum requirements set by the standard. SIA standards are not intended to supersede any recommended procedures set by a manufacturer for its products.

SIA reserves the right to revise this document at any time. Because SIA policy requires that every standard be reviewed periodically and be revised, reaffirmed, or withdrawn, users of this document are cautioned to obtain and use the most recent edition of this standard. Current information regarding the revision level or status of this or any other SIA standard may be obtained by contacting SIA.

Requests to modify this document are welcome at any time from any party, regardless of membership affiliation with SIA. Such requests, which must be in writing and sent to the address set forth below, must clearly identify the document and text subject to the proposed modification and should include a draft of proposed changes with supporting comments. Such requests will be considered in accordance with SIA's standards development policies and procedures.

Written requests for interpretations of a SIA standard will be considered in accordance with SIA's standards development policies and procedures. While it is the practice of SIA staff to process an interpretation request quickly, immediate responses may not be possible since it is often necessary for the appropriate standards subcommittee to review the request and develop an appropriate interpretation.

Requests to modify a standard, requests for interpretations of a standard, or any other comments are welcome and may be sent to:

The Security Industry Association
8405 Colesville Road, Ste. 500
Silver Spring, MD 20910
(703) 683-2075
www.siaonline.org
standards@siaonline.org

Revisions - 2013 Version

Paragraph	*Revision*
contents	expanded to 4 levels
Foreword	updated SIA address
Acknowledgements	added 2013 section
3.4	redefined encryption
4.1.1	added section regarding source port number
4.2	added requirement to mark equipment with list of supported tokens
5.3	close socket is optional
5.4.7	revised for clarity
5.5.1.1	added "0x0A", sent as binary value
5.5.1.2	four ASCII characters
5.5.1.4	ASCII, must support SIA-DCS or ADM-CID
5.5.1.5	revised for clarity, four ASCII characters
5.5.1.7	revised for clarity
5.5.1.8	added several data identifiers, including "A" for authentication hash
5.5.1.9	ASCII
5.5.1.10	added "0x0D"
5.5.3.1	revised for clarity
5.6.2.2	**shall** discard
5.6.2.4	added paragraph, no response to decryption errors
Annex B	revised examples
Annex B	corrected title of example B6
Annex E, F, G, H, I	added annex

Acknowledgements - 2013

Working Group Technical Editor:
North Latitude Technology, ... Ted Nesse

Working Group Participants:
ADT ..... Tony Mucci
Amstein & Walthert, DCC Consulting ..... Stephane Jacquet
Bay Alarm ..... Shane Clary
Bosch Security ..... Denis Caler
Bosch Security ..... Michael Reimer
Bosch Security ..... Kevin Ritchie
Bosch Security ..... Stephan Kovaciss
DSC ..... Dan Nita
Honeywell ..... Rich Hinkson
Honeywell ..... John Slogick
LRD Consulting ..... Larry Dischert
UL ..... Bryon Monte
UTC ..... Kevin Flanders

Acknowledgements - 2007

This standard was approved as an American National Standard by the Security Industry Standards Council (SISC). The voting members are listed below.

Acquient Security Consulting ..... Jack Sigler
Acquient Security Consulting ..... Terry France
ACTA ..... Dr. Jimmy Salinas
ACTA ..... Tim Jeffries
ADT ..... Larry Dischert
American Protective Services ..... Dan Jacquish
ASIS ..... Susan Melnicove
Bay Alarm Co. ..... Shane Clary
Bosch Security Systems ..... Stephen Kovaciss
Brinks ..... Rick Sheets
Corbin Russwin ..... Tom Harris
Corbin Russwin ..... Richard McKeown
CSAA ..... Lou Fiore
CSAA ..... Ralph Sevinor
Emergency24 ..... Pat Devereaux
Emergency24 ..... Kevin McCarthy
Fargo Electronics ..... Gary Klinefelter
GE ..... Frank Clark
HES / Folger Adam ..... Chuck Christiansen
HES / Folger Adam ..... Brian Moses
HID ..... Bill DeVoe
HID ..... Mark Peterson
Hirsch Electronics ..... Rob Zivney
Honeywell ..... Gordon Hope
Honeywell ..... Isaac Papier
IDS Research and Development, Inc. ..... Jeffrey Zwirn
Integrated Command Software ..... R. Hunter Knight
Integrated Command Sofware ..... Sherrie Knight
Mastec ..... George Bish
NBFAA ..... Dale Eller
NBFAA ..... Rick Simpson
Pelco ..... Dave Smith

Salient Systems Corp.	Per Hanssen
Sandia National Laboratories	George Wagner
SecuraKey	Bill Newill
Securitron	Scott Baker
Securitron	Larry Kern
SEIWG	Chris Hernandez
Security Access Design, LLC	Richard Kelley
Sequel Technologies	Ted Nesse
Siemens Building Technologies	Noelle Britton
Siemens Building Technologies	Bill Gorski
SIA	Hunter Knight
SIA	Mark Visbal
State Farm Insurance	Joe Miskulin
UL	Lou Chavez
UL	Derek Mathews
U.S. Department of Commerce / NIST	Ron Martin
U.S. Department of Commerce / NIST	Jim Dray
Vector Security	Pam Petrow
Vector Security	John Murphy
Walker Engineering	Matthew Kenjura
Walker Engineering	Dale Berti
Yale Commercial Locks and Hardware	Reid Wilson

This standard was developed in SIA Standards; specifically within the IP Working Group of the SIA Security Communications Subcommittee. The following are additional acknowledgements to be made:

Chairman of the SIA Standards Committee:
Integrated Command Software ..... R. Hunter Knight

Chairman of the SIA Standards Security Communications Subcommittee:
Deister Electronics USA, Inc. ..... William Nuffer

Chairman of the IP Working Group:
Bosch Security Systems ..... Stephen Kovaciss

Working Group Technical Editor:
Sequel Technologies, LLC ..... Theodore Nesse

Participants:

ADT	Bernie Worst
ADT	Warren Burgess
AT&T	Jimmy Salinas
Bosch	Stephen Kovacsiss
Bosch	Rich Ader
ChannelSmarts	Drew Chernoy
DSC	Dan Nita
DSC	Sascha Kylau
DSC	Stephan Frenette
Elk Products	Kelly Carter
Fire Monitoring of Canada Inc.	Holly Barkwell-Holland
GE	Frank Clark
GE	Jack Fewx
HID	Gary Withrow
HID	Bill DeVoe
HID	Mike Davis
Honeywell	Rich Hinkson
Honeywell	Steve Yawney

Honeywell ..... Bill Blum
Honeywell ..... Bob Orlando
Honeywell ..... Gordon Hope
Honeywell ..... Rich Hinkson
Honeywell ..... Scott Simon
Integrated Command Software ..... R. Hunter Knight
Napco ..... Al DePierro
Salient Systems Corporation ..... Per Hanssen
Security Sales Integration Magazine ..... Robert Dolph
The Command Center ..... Morgan Hertel
Tyco ..... Jay Hauh
SIA gratefully acknowledges the efforts of the many volunteers from the security industry that helped the Subcommittee to develop this standard.

SIA Digital Communication Standard DC-09-2013
Internet Protocol Event Reporting

1. Scope

This standard details the protocol and related details to report events from premises equipment to a central station using Internet protocol (IP) to carry the event content. It is important to distinguish that, while this reporting method uses the SIA Receiver-to-Computer Interface Protocol as a foundation, it is intended for event transport from protected premises to a central station - possibly using the public Internet.

This standard is intended for use by manufacturers of control panels and central station receivers to ensure equipment compatibility, as well as all affected parties. Compliance with this standard is voluntary.

2. References

2.1. Normative Reference

The following document provides a normative reference for this standard:

- DC-07: SIA Digital Communications Standards – Receiver-to-Computer Interface Protocol (Type 2).

2.2. Informational References

Additional guidance on areas relating to this standard, as noted and otherwise, can be obtained from the sources below.

National Fire Protection Association
- NFPA 72, National Fire Alarm Code

Underwriters Laboratories, Inc.
- UL 609, Local Burglar-Alarm Units and Systems
- UL 611, Central-Station Burglar-Alarm Systems
- UL 681, Installation and Classification of Mercantile and Bank Burglar-Alarm Systems
- UL 864, Control Units for Fire-Protective Signaling Systems
- UL 985, Household Fire Warning System Units
- UL 1023, Household Burglar-Alarm System Units
- UL 1076, Proprietary Burglar Alarm Units and Systems
- UL 1610, Central Station Burglar-Alarm Units
- UL 1635, Digital Burglar Alarm Communicator System Units
- UL 1641, Installation and Classification of Residential Burglar Alarm Systems

Security Industry Association
- DC-02: Digital Communications -Generic Protocols Overview Technical Report
- DC-03: Digital Communication Standard - “SIA Format” Protocol for Alarm System Communications.
- DC-04: SIA 2000 Protocol for Alarm System Communications
- DC-05: Digital Communication Standard - “Contact ID” Protocol for Alarm System Communications.

National Institute of Standards and Technology
- Federal Information Processing Standards Publication 197 (AES)
- NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation

3. Conventions and Definitions

3.1. Typographic Conventions

Elements of messages are enclosed in <braces>, and may consist of one or more bytes. The notational braces do not appear in the transmitted messages.

Complete messages are enclosed in a frame.

Italic font is used for variable elements in a message.

3.2. UML Notation

UML sequence diagrams are used to document requirements of the standard that involve a sequence of messages.

3.3. Binding Language

This standard uses the term “shall” to convey binding requirements.

The term “may” is used to convey features that are allowed but not required.

Terms such as “is”, “are”, “will”, and others are used to convey statements of fact for advisory purposes only. The annotation “NOTE:” also precedes advisory information.

Where this standard is silent on a feature, the feature is permitted so long as it is not in conflict with the requirements contained herein.

3.4. Glossary

ACK – Acknowledgment
A return message indicating correct receipt of a transmitted message.

Authentication
A process to assure that a received message is not a counterfeit sent by an unauthorized sender.

Central Station Receiver – “CSR”
A central station receiver accepts connections from premises equipment, for the purpose of transmitting event information to the central station.

Encryption
The process of obscuring the content of a transmitted data message so it can not be read or replicated by unauthorized persons or equipment.

Encryption Key
A data word used to encrypt and decrypt a message.

Frame
The elements that make up a complete message for this protocol.

IP Address
The unique identifier number assigned to a device on an IP network.

NAK – Negative Acknowledgement
A return message indicating rejection of a transmitted message.

Premises Equipment – “PE”

"Premises equipment" is used to describe a general class of electronic systems that are field-installed for the purpose of reporting event data to a central station. Security systems, fire alarm control panels and access control systems are examples of premises equipment.

4. Compatibility

4.1. User Datagram Protocol (UDP) and Transmission Control Protocol (TCP)

Premises Equipment (PE) and Central Station Receivers (CSRs) shall support either UDP or TCP for event message transmission, and may support both UDP and TCP. When CSRs support both protocols, the CSR shall automatically use the appropriate protocol for incoming messages without requiring a configuration setting. When PE support both protocols, the protocol use may be manually specified, or the manufacturer may implement a method to attempt message delivery with one protocol, and switch to the other as necessary.

When PE or CSRs support only one protocol, UDP is the preferred implementation but TCP may be used.

4.1.1. UDP Source Port Number

When UDP is used, the transmitter may set the source port number in the UDP header to the desired port at which replies from the central station receiver will be accepted. This transmitter behavior is recommended.

4.2. Marking

Equipment capable of using UDP/IP for communication shall be marked as compatible with "SIA IP Reporting (UDP-2013)". Equipment capable of using TCP/IP for communication shall be marked as compatible with "SIA IP Reporting (TCP-2013)". Equipment capable of using UDP/IP or TCP/IP for communication shall be marked as compatible with "SIA IP Reporting (UDP/TCP-2013)".

Additionally, equipment shall be marked to show the list of DC-07 tokens (protocols) that are supported.

When PE and CSRs implementing this standard share at least one protocol (UDP or TCP), they are intended to be interoperable.

5. Requirements

5.1. Media

This standard can be implemented on any media that carries Internet protocol (IP), including but not limited to Ethernet, 802.11x, CDMA 1x or GPRS. The ability of the network to perform media conversion and provide limited protection for message integrity is assumed.

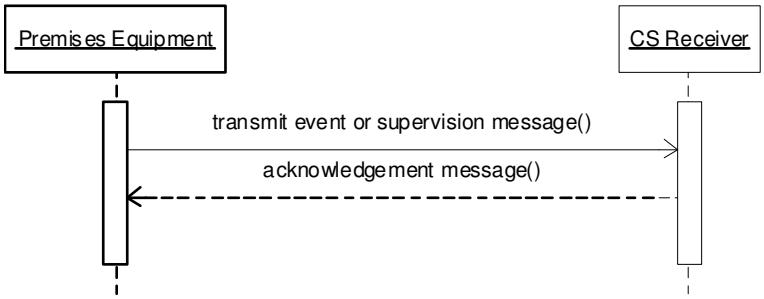
5.2. IP Addresses

The central station receiver (CSR) shall be hosted on a static IP address. The premises equipment (PE) may be hosted on a dynamic or static IP address and shall be capable of being programmed with the IP address to which events are to be sent.

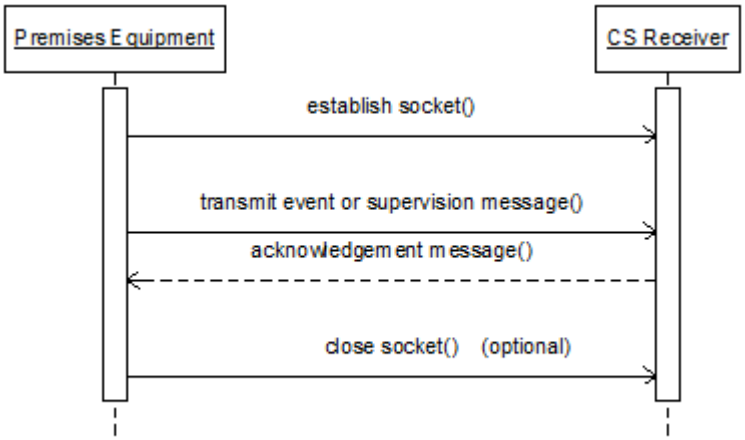
The PE may have an option to use DNS to obtain the address of the CSR, however the installation is not compliant with this standard when the DNS option is enabled.

5.3. Communication Sequence

When UDP is used, a simple transmit/acknowledge sequence is used to transmit messages:



When TCP is used, the process is very similar:



5.4. Encryption

PE may indicate use of AES encryption to transmit events. Encryption support is optional for PE. Encryption support is mandatory for CSRs.

5.4.1. Encryption Standard

For the AES requirements, refer to Federal Information Processing Standards Publication 197 that is available from the National Institute of Standards and Technology.

Additionally, encrypted messages shall use Cipher Block Chaining, as described in section 6.2 of NIST Special Publication 800-38A (2001 Edition).

5.4.2. Identifying Encrypted Messages

Encrypted messages shall be identified with an asterisk as described in section 0 of this standard.

5.4.3. Encrypted Elements

When encryption is used, only the data, timestamp and padding content of a message are encrypted. Encryption begins on the byte after the opening bracket "[" on the data element, and ends just before the terminating <CR>.

The encrypted elements are shaded in the frame, below:

<LF><*crc*><0*LLL*> <"*id*"><*seq*><R*rcvr*><L*pref*><#*acct*>[<*pad*>|...*data*...][x…data…]<*timestamp*> <CR>

5.4.4. Padding

Only when encryption is used, messages shall be padded with pseudo-random data (pad) so that the byte count of the encrypted region is an even multiple of 16.

5.4.4.1. Padded Region

The characters counted for padding and encryption begin after the opening bracket "[" on the data element, and include the pad field (pad), data, and timestamp field. The final <CR> is not included in the count for padding and is not encrypted.

5.4.4.2. Pad Data

Pad data shall be pseudo-random bytes which vary from one message to the next. This data will consist of binary values 0-255, except that it shall not contain the ASCII values for the character "|" (124, x7C), "[" (91, x5B) or "]" (93, x5D).

5.4.4.3. pad (Pad Data Field)

When a message is encrypted, padding is inserted between the open bracket character "[" and the pad termination character "|". The number of characters in the pad field shall be such that the total number of encrypted characters (from the first pad character up to and including the last timestamp character is an even multiple of 16.

When a message is already an even multiple of 16 bytes, 16 pad bytes shall be added to the message.

The "|" character immediately following the pad field shall appear in all encrypted messages. This pad termination "|" character shall not appear in unencrypted messages, though the "|" character that typically separates the account number from the rest of the data may appear.

5.4.5. Encryption Key

When encryption is selected, the PE may use a key length of 128, 192 or 256 bits. A matching key value (and therefore matching key length) must be programmed at the PE and the CSR.

5.4.5.1. Central Station Receiver Requirements

A CSR shall support all three key lengths, as well as messages using no encryption. Other key lengths may be incorporated into the standard in the future, but are currently non-compliant.

Each port in the CSR that receives events shall be configurable with at least one encryption key, to be used for all PE reporting to that IP address/port. Optionally, each port may be configurable with multiple encryption keys, to be used with individual accounts or groups of accounts.

5.4.5.2. Premises Equipment Requirements

When the PE supports encryption, it shall be able to store a private key of 128, 192 or 256 bits in length, at the option of the manufacturer

5.4.5.3. Key Contents

When private or session keys are created, a pseudo-random process shall be used. Each bit shall have an equal probability of being 0 or 1 as the key is created. The use of a binary-encoded ASCII phrase as the key is specifically discouraged.

5.4.6. Cipher Block Chaining

The encrypted blocks of a message implement cipher block chaining as described in section 6.2 of NIST Special Publication 800-38A (2001 Edition). For this protocol, the initialization vector (IV) will be all zeros, as the padding characters provide the variability needed for message confidentiality.

Refer to Annex A for a copy of the cipher block chaining method to be used.

5.4.7. Encoding

In the encrypted region of the message, each byte shall be encoded for transmission as two ASCII characters (0-9, A-F) representing the hexadecimal value of the encrypted byte.
For example, the message:

<LF>B3680040"ADM-CID"0001L000000#1234[#1234|1140 00 007]_22:49:34,01-22-2012<CRC>

When encrypted, might (depending on the encryption key used) be transmitted using the following ASCII characters:

<LF>4B89007B"*ADM-CID"0001L000000#1234 [371baac130fe81508f556e6fd2ccfd8826e9ba186f0fb67 4bb87c079484e546dff35532aa285936a00c27b6feb053f68 <CR>

5.5. Messages

PE may send two types of messages: events and link supervision.

CSRs send only one type of message, acknowledgment, which may have three types: ACK, NAK, or DUH.

5.5.1. Event Messages (PE)

The format used for the events is based on SIA protocol (DC-07-2001.04) outlined in SIA Digital Communications Standards – Receiver-to-Computer Interface Protocol (Type 2).

The template for each event:

<LF>*<crc>*<0*LLL*> <"*id*">*<seq>*<R*rcvr*><L*pref*>*<#acct>*[*<pad>*|...*data*...][*x*…*data*…]*<timestamp>* <CR>

5.5.1.1. LF

This is the ASCII linefeed character, transmitted as a binary value 0x0A.

5.5.1.2. crc

The portion of the message starting with the first quote character of the ID and ending with the character before the terminating CR, are included in the CRC calculation. This is the middle line in the frame shown above. Refer to DC-07-2001.04 for the detailed CRC implementation. When messages are encrypted, the CRC shall be applied after the encryption is applied.

The CRC shall be transmitted as four ASCII characters.

5.5.1.3. 0LLL

This length element consists of the character "0" (ASCII zero) followed by 3 hex digits (in ASCII) giving the length of the message. The characters counted are the same as are included in the CRC calculation, as described in section 5.5.1.2.

5.5.1.4. "id" (ID Token)

The <"id"> field contains an ASCII token to indicate the format used in the data field of the message, and whether or not encryption is used. The quote characters are included in the message.

A CSR compliant with this standard shall support at least the SIA-DCS and ADM-CID tokens (protocols) shown in Annex H, based on the token definition in DC-07-2001.04.

PE shall support at least one of the tokens SIA-DCS and ADM-CID, and may support any others shown in Annex H.

5.5.1.4.1. Encryption Flag

When the data and timestamp of a message are encrypted, the ID Token is modified to insert an ASCII "*" after the quotation character and before the first character of the token. For example, an unencrypted SIA DCS packet would use the token "SIA-DCS" and an encrypted SIA DCS packet would use the token "*SIA-DCS".

5.5.1.5. seq

The PE applies a sequence number to each message as it is queued. The CSR shall echo the sequence number of the message to which it is replying in its acknowledgement messages.

The PE shall not increment the sequence number when repeating a message due to a communication failure or no response from a CSR.

The PE shall increment the sequence number to be used as each new message is queued. When the sequence number is 9999, the next sequence number is 0001. Refer to section 7.1.5 of DC-07-2001.04 for additional information.

The sequence number shall be transmitted as four ASCII characters.

Segment numbers, as described in DC-07, are not used in this protocol.

5.5.1.6. Account Identification (Rrcvr, Lpref, #acct)

Each set of PE may be provided with up to three complementary identifying tokens.

5.5.1.6.1. #acct (Account Number)

The account number is the most specific token, and is always programmed into the premises equipment to identify it. The account token appears both in the header of the message (which is never encrypted) and in the data of the message (which may be encrypted).

This element consists of an ASCII "#", followed by 3-16 ASCII characters representing hexadecimal digits for the account number. There is no corresponding element in the DC-07 protocol.

In certain special applications, the information provided in the **#acct** element may not match the account number contained within the message data (see paragraph 5.5.1.7). For example, a manufacturer may choose to transmit a MAC address as an identifier.

5.5.1.6.2. Lpref (Account Prefix)

The account prefix can be programmed into the PE to extend the identification provided by the account number.

This element is required, and consists of an ASCII "L", followed by 1-6 HEX ASCII digits for the account prefix. When the PE does not need to transmit an account prefix, "L0" shall be transmitted for this element.

This element corresponds with the receiver line number element in the DC-07 protocol.

5.5.1.6.3. Rrcvr (Receiver Number)

In some cases, PE may be programmed to further extend the identification provided by the account number and account prefix by providing a receiver number.

This element is optional, and consists of an ASCII "R", followed by 1-6 HEX ASCII digits for the receiver number. When the PE does not need to transmit a receiver number, nothing shall be transmitted for this element (i.e. "R" or "R0" are not to be transmitted in this case).

This element corresponds with the receiver number element in the DC-07 protocol.

5.5.1.7. [Data] or [<pad>|Data] (Message Data)

All data is in ASCII characters and the bracket characters "[" and "]" are included in the transmitted message. The data field format is dependent upon the ID token of the message.

Where an account number is associated with a message (most message types), the account number data appears at the start of the data, preceded by the "#" character and followed by the field separator "|". The account number is 3-16 ASCII characters representing hexadecimal digits.

Refer to appendix A in DC-07-2001.04 for a definition of the data packets for the various message types.

Note that the data element and timestamp in a message may be encrypted, as described in section 5.4.

Refer to section 5.4.4.3 for a description of the <pad> field that appears within the data field of encrypted messages.

5.5.1.8. [x…data…] (Optional Extended Data)

This field allows the PE to attach additional information to the message, by including one or more optional extended data fields.

Use of this field is optional for the PE.

The receiver shall be able to receive and properly process a message containing the optional extended data, however use of the information contained within the extended data field is optional.

The start field is delimited with the ASCII character "[", followed by a single ASCII character (data identifier) that identifies the content of the data field. The data identifier may be any upper case ASCII character in the range "G" to "Z" The field is terminated with the ASCII character "]".

Following the identifier, extended data may contain characters encoded according to the legacy Windows 1252 character set, as shown in annex E.

The following data idenitfiers are defined:

Name	Identifier	Description	Data ("[",	Example

			"]" or "|" disallowed)	
Authentication Hash	"A"	A hash of the message that allows the message to be authenticated	12 ASCII characters (0-F, A-F or a-f)	A6F2348C99335DF38 (Note 4)
Time of Occurrence	"H"	Time that event occurred (may be different than message time stamp)	ASCII	H13:59:58,12-31-2012
MAC Address	"M"	MAC address of the PE	12 ASCII characters (0-F, A-F or a-f)	M0026B9E4268B
Verification	"V"	information about audio or video information that may be associated with the event report	Windows 1252 characters	Vhttp:\\verify.com/34AC4DE3446 (Note 2)
Programming Data	"P"	contains a message used to support programming or other interactive operations with the receiver	Windows 1252 characters	(future use) (Note 2)
Alarm Text	"I" (capital letter I)	alarm text which may be a description of the event or a comment regarding the event	Windows 1252 characters	I2nd Floor West PIR (Note 2)
Site Name	"S"	site name descrbing the premises	Windows 1252 characters	S123Main St., 55123 (Note 2)
Building Name	"O" (capital letter O)	building name	Windows 1252 characters	OIDS Center (Note 2)
Location	"L"	location of event on site	Windows 1252 characters	L3rd Floor Hallway (Note 2)
Room	"R"	room of event (Note 1)	Windows 1252 characters	RL2322 (Note 2)
Alarm Trigger	"T"	trigger for event (Note 1)	ASCII	Tx (Note 3)
Longitude	"X"	location of event, longitude (Note 1)	ASCII	X093W23.456 (Note 2)
Latitude	"Y"	location of event, latitude (Note 1)	ASCII	Y45N23.456 (Note 2)
Altitude	"Z"	altitude of event (Note 1)	ASCII	Z123.2M (Note 2)

Table 1: Extended Data Identifiers

Note 1: The content of these fields may be assigned by a local authority having jurisdiction.

Note 2: The format of these variable length fields are free form, and are not specified here.
Note 3: The following trigger identifers are defined:

Tag	*Assignment*	*Type*
F	Fire / Smoke Detector	Automatic
G	Gas Detector	Automatic
W	Water / flooding detector	Automatic
S	Sensor (temp / humid. / pressure)	Semi-automatic
C	Contact (intrusion or other)	Semi-automatic
M	Manual Trigger (eg switch)	Manual

Table 2: Trigger Identifiers

Note 4: The length, format and algorithm used for the data of this field is determined by the manufacturer, except that binary-formatted data is disallowed. When the extended data field is used to transfer a hash code, this adds an integrity control function to the end-to-end transmission (from premises to central station). In some countries this may be required by the authorities having jurisdiction.

5.5.1.9. Timestamp

The timestamp shall be included in encrypted messages, and may be included on messages that are not encrypted. This field is used to provide protection against message playback. The timestamp indicates when the event was queued for transmission to the central station, and may indicate a time significantly different than the time of event occurrence. The timestamp is always transmitted with a reference of GMT.

The format of the timestamp is: <_HH:MM:SS,MM-DD-YYYY>. The braces are not part of the transmitted message, but the underscore, colon, comma and hyphen characters are included. Integers shall be zero filled, and the year shall be four digits so that the length of the timestamp field is exactly 20 characters. All elements of the timestamp are transmitted as ASCII characters.

The CSR shall validate the timestamp against its own GMT reference. Encrypted messages with a GMT difference from the CSR greater than +20/-40 seconds shall be rejected with a NAK packet that contains the current GMT time known to the CSR in the data field. The allowed time difference may be a configurable parameter for the CSR.

The PE shall re-encrypt the message with an updated timestamp when retrying a message in response to a NAK containing a corrected GMT time.

5.5.1.10. CR

This is the ASCII carriage return character, transmitted as a binary value 0x0D~~13~~.

5.5.2. Supervision Message

Optionally, the PE and CSR may be configured to supervise the connection. When supported and enabled, the PE shall periodically send the Null Message to the CSR. The CSR shall respond with an ACK message.

On the CSR, if this function is supported, the supervision interval shall be configurable over a range of 10 seconds to 3600 seconds, and 1 hour to 1080 hours (45 days). If no message of any type is received from a supervised account during this interval, a communication failure shall be declared and reported into the central station.

It shall be possible to disable the supervision on the PE, if it is supported. When enabled, the PE shall transmit the Null Message each time the supervision interval elapses. It is permissible to

restart the supervision interval timer when the PE transmits any other message. The supervision interval may be configurable in a range compatible with the CSR.

5.5.2.1. Null Message (Link Test)

The PE may send an encrypted or an unencrypted Null Message to permit supervision of the link between the premises and the central station. The CSR shall acknowledge the message.

5.5.2.1.1. Unencrypted Null Message

<LF><CRC><0LLL><"NULL"><0000><Rrcvr><Lpref><#acct>[]<timestamp><CR>

The timestamp is optional for unencrypted messages

5.5.2.1.2. Encrypted Null Message

<LF><CRC><0*LLL*><"*NULL"><0000><Rrcvr><Lpref><#acct>[<pad>]<timestamp><CR>

Note that the ID token is "*NULL" for an encrypted null message, and that the timestamp and padding fields are required.

5.5.3. Acknowledgement Messages (CSR)

When the CSR receives an event from the PE, it shall respond with an acknowledgement message.

5.5.3.1. ACK - Positive Acknowledgement

Messages received without errors shall cause the CSR to send a positive acknowledgement packet:

<LF><CRC><0LLL><"ACK"><seq><Rrcvr><Lpref><#acct>[]<CR>

When the CSR is responding to an encrypted message, the ACK response will be encrypted:

<LF><CRC><0LLL><"*ACK"><seq><Rrcvr><Lpref><#acct>[<pad>]<timestamp><CR>

Note the required timestamp, required padding, and the modified ID token "*ACK" in the encrypted ACK message.

5.5.3.2. NAK - Negative Acknowledgement

When an encrypted message fails the timestamp test described in section 5.5.1.8, the CSR shall send a negative acknowledgement packet containing a timestamp:

<LF><CRC><0LLL><"NAK"><0000><R0><L0><A0>[]<timestamp><CR>

The NAK message is never encrypted.

5.5.3.3. DUH - Unable Acknowledgement

When the CSR is unable to process an otherwise correctly received message, it shall respond with a DUH acknowledgement packet:

<LF><CRC><0LLL><"DUH"><seq><Rrcvr><Lpref><#acct>[]<CR>

The DUH message is never encrypted.

5.5.3.4. RSP - Message Response (future use)

This message may be implemented in future versions of this standard to define a response to a message containing certain types of extended data. Neither receivers nor PE are required to support this message at this time.

<LF><CRC><0LLL><"RSP"><seq><Rrcvr><Lpref><#acct>[… response data …]<CR>

5.6. Error Handling

Various errors may occur during PE-CSR communications. Error handling is defined for these error conditions:

5.6.1. Errors Observed by Premises Equipment

5.6.1.1. No Response

The PE shall time a retry timeout period while it waits for a response to a transmitted message. If the response is not received during the retry timeout period, it shall retransmit the message. The recommended retry timeout period is 20 seconds, and the recommended number of message attempts is 3. The retry timeout period may be configurable in the range of 5-60 seconds. The message attempt count may also be configurable.

5.6.1.2. NAK Response

Upon receiving a NAK response, the PE shall immediately repeat the message a limited number of times, using an updated timestamp. The recommended number of message attempts is 3, and this count may be configurable.

5.6.1.3. DUH Response

Upon receiving a DUH response, the PE may optionally repeat the message a limited number of times, or immediately declare a failure.

5.6.2. Errors Observed by Central Station Receiver

5.6.2.1. Timestamp Mismatch (CSR)

When the timestamp on a message from the PE fails the test described in section 5.5.1.8, the CSR shall respond with the NAK containing a timestamp described in section 5.5.3.2.

5.6.2.2. Checksum Failure

The CSR shall discard messages exhibiting a checksum error. No response shall be sent to the PE.

5.6.2.3. Unsupported Message Elements

When the CSR receives an otherwise correctly formatted message message that contains unsupported message elements (e.g. an unknown "id" field), it shall return a "DUH" response.

5.6.2.4. Decryption Error

The CSR shall discard messages exhibiting an encryption error. No response shall be sent to the PE.

Annex A: Cipher Block Chaining (Informative Annex)

(Reproduced from NIST Special Publication 800-38A - 2001 Edition)

The Cipher Block Chaining (CBC) mode is a confidentiality mode whose encryption process features the combining ("chaining") of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an IV [initialization vector] to combine with the first plaintext block. The IV need not be secret, but it must be unpredictable; the generation of such IVs is discussed in Appendix C [of NIST 800-38A]. Also, the integrity of the IV should be protected, as discussed in Appendix D [of NIST 800-38A]. The CBC mode is defined as follows:

CBC Encryption:

$$C_1 = CIPH_K(P_1 \oplus IV);$$
$$C_j = CIPH_K(P_j \oplus C_{j-1}) \quad \text{for } j = 2 \ldots n.$$

CBC Decryption:

$$P_1 = CIPH^{-1}_K(C_1) \oplus IV;$$
$$P_j = CIPH^{-1}_K(C_j) \oplus C_{j-1} \quad \text{for } j = 2 \ldots n.$$

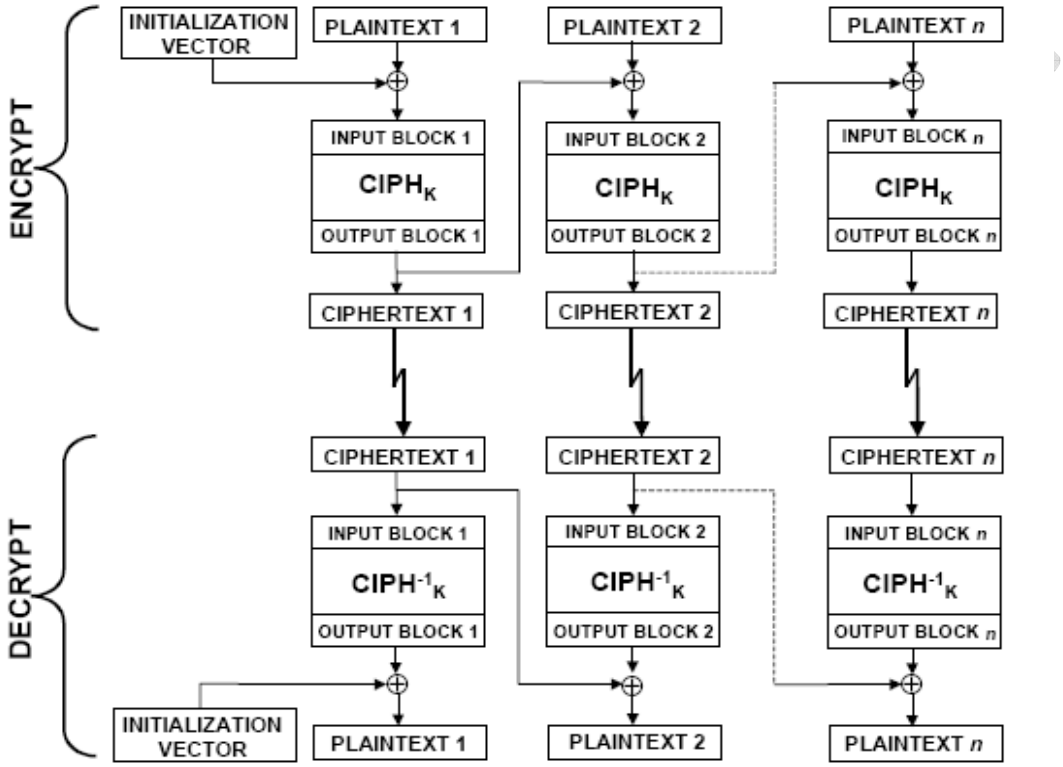


Figure 2: The CBC Mode

In CBC encryption, the first input block is formed by exclusive-ORing the first block of the plaintext with the IV. The forward cipher function is applied to the first input block, and the resulting output block is the first block of the ciphertext. This output block is also exclusive-ORed with the second plaintext data block to produce the second input block, and the forward cipher function is applied to produce the second output block. This output block, which is the second ciphertext block, is exclusive-ORed with the next

plaintext block to form the next input block. Each successive plaintext block is exclusive-ORed with the previous output/ciphertext block to produce the new input block. The forward cipher function is applied to each input block to produce the ciphertext block.

In CBC decryption, the inverse cipher function is applied to the first ciphertext block, and the resulting output block is exclusive-ORed with the initialization vector to recover the first plaintext block. The inverse cipher function is also applied to the second ciphertext block, and the resulting output block is exclusive-ORed with the first ciphertext block to recover the second plaintext block. In general, to recover any plaintext block (except the first), the inverse cipher function is applied to the corresponding ciphertext block, and the resulting block is exclusive-ORed with the previous ciphertext block.

In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function, i.e., the ciphertext blocks, are immediately available, so that multiple inverse cipher operations can be performed in parallel. The CBC mode is illustrated in Figure 2 above.

Annex B: Example Message Frames (Informative Annex)

In these examples, the following parameters are held constant:

- seq: 9876
- rcvr: 579BDF
- pref: 789ABC
- acct: 12345A

For encrypted messages, the CRC is shown as "xXXXX" since the CRC is applied after encryption. Non-ASCII data is shown in brackets "<…>" and uses hexadecimal notation (e.g. "<x79BD>").

The examples are shown on two lines due to formatting, but are transmitted as a single string.

Alternating fields are highlighted to improve readability.

B.1 Fire Alarm, Zone 129, SIA DC-04 Format, Non-Encrypted, No Timestamp

<x0A>CE11**0032**"SIA-DCS"**9876**R579BDF**L789ABC**#12345A
[#12345A|NFA129]**<x0D>**

B.2 Fire Alarm, Zone 129, Contact ID DC-05 Format, Non-Encrypted, No Timestamp, Group 0

<x0A>87CD**0037**"ADM-CID"**9876**R579BDF**L789ABC#**12345A
[#12345A|1110 00 129]**<x0D>**

B.3 Fire Alarm, Zone 129, SIA DC-04 Format, Non-Encrypted, With Timestamp

<x0A>DC53**0046**"SIA-DCS"**9876**R579BDF**L789ABC**#12345A
[#12345A|NFA129]_**13:14:15,02-15-2006**<x0D>

B.4 Fire Alarm, Zone 129, SIA DC-04 Format, Encrypted

The message is first shown prior to encrypting the region from the opening "[" to the closing <x0D>. There are 13 padding bytes (ASCII "p") in this message.

<x0A>D976**0084**"*SIA-DCS"**9876**R579BDF**L789ABC**#12345A
[pppppppppppppp|12345A|NFA129]_13:14:15,02-15-2006<x0D>

Here is the encrypted message, using a 128 bit key **0123456789ABCDEF0123456789ABCDEF**:

<x0A>D976**0084**"*SIA-DCS"**9876**R579BDF**L789ABC**#12345A
[209c9d400b655df7a26aecb6a887e7ee6ed8103217079aae7cbd9dd7551e96823263460f7ef05 14864897ae9789534f1<x0D>

B.5 Fire Alarm, Zone 129, SIA DC-04 Format, Non-Encrypted, No Timestamp, MAC Address x1234567890AB

<x0A>8C86**0043**"SIA-DCS"**9876**R579BDF**L789ABC**#123456A
[#123456A|NFA129]**[M1234567890AB]**<x0D>

B.6 Intrusion Alarm, Zone 65, Contact ID DC-05 Format, Non-Encrypted, Group 2, MAC Address x1234567890AB, Validation Data (future)

<x0A>XXXX**0056**"*ADM-CID"**9876**R579BDF**L789ABC#**12345A
[4D32|#12345A|**1130 02 065]**[M1234567890AB]**[V*anydata*]**<x0D>

B.7 Open Area 2, User 3, SIA DC-04 Format, Non-Encrypted, No Timestamp

<x0A>463A**0033**"SIA-DCS"**9876**R579BDF**L789ABC#**12345A
[#12345A|Nid3OG2]**<x0D>**

Annex C: Recommended Self-Validation Procedures (Informative Annex)

A preferred approach to validation is to use a simulator to test the product that is being developed. A best practice is to have the simulator programmed by a different person/team than the product being tested, so that faulty assumptions are exposed.

C.1 Premises Equipment Testing

C.1.1 Receiver Simulator

In the case of testing premises equipment a receiver simulator tool would be used. The receiver simulator might be a PC-based application that uses the IP capabilities of the PC to support the connection with the PE. The receiver simulator could display the following message elements, parsed from the message string according to this standard:

- sequence number (seq)
- receiver number (rcvr)
- account prefix (pref)
- account number (acct)
- encryption pad characters
- data message (…data…)
- extended data (x…data…)
- timestamp

Additionally, the simulator can validate the CRC, the message length and decrypt encrypted messages.

C.1.2 Testing Process

The receiver simulator can be used to accept reports from the unit under test (UUT), which can then be manually verified. For each test, the data for each field which the PE is transmitting must be matched against the data for each field that the simulator parsed from the message. Additionally, a correctly formed CRC and message length must be observed for each transmission.

C.1.3 Test Cases

This section recommends test cases to be used for PE testing:

C.1.3.1 Marking (ref. 4.2)

- confirm that the equipment is marked with only one of the following: "SIA IP Reporting (UDP-2006)", "SIA IP Reporting (TCP-2006)" or "SIA IP Reporting (UDP/TCP-2006)"

- use the receiver simulator to confirm that the correct IP protocol (TCP, UDP or either) is actually used to transmit messages

C.1.3.2 IP Addresses (ref 5.2)

- verify that the IP address of the PE can be manually programmed

C.1.3.3 Encryption (ref 5.4) (if supported by PE)

- set the PE and receiver simulator with matching encryption keys

- for each supported DC-07 message type, transmit an encrypted message and ensure that the receiver simulator can decrypt it correctly
- transmit five messages, and observe that no order is discernable in the padding data (it may be necessary to vary the message to trigger the inclusion of a significant amount of padding)
- observe that "[". "]" and "|" do not appear in the padding data
- change the encryption key at the PE and ensure that the message is not decrypted correctly
- repeat the encryption tests for each supported encryption key length

C.1.3.4 Cipher Block Chaining (ref. 5.4.6)

- if the PE Is capable of generating a message consisting of 3 encrypted blocks (length over 32 bytes), transmit the same message to the receiver simulator three times, observing that the data of the middle block is different for each transmission

C.1.3.5 ID Token (ref. 5.5.1.4)

- confirm that the UUT generates the correct ID tokens for each DC-07 message type that it is designed to support
- confirm that the ID token is prefixed with "*" for encrypted messages (when supported by the PE)

C.1.3.6 Sequence Number (ref. 5.5.1.5)

- cause the simulator to reject the first attempt - observe the the PE uses the same sequence number on the second attempt
- send five reports - observe the sequence number increments on each report
- send reports to confirm that the sequence number properly increments through the sequence 9998, 9999, 0001, 0002

C.1.3.7 Account Identification (ref. 5.5.1.6)

Transmit reports from the PE to confirm correct reception by the receiver simulator when the following account ID parameters are set (use the PE limit when the parameters specified below fall outside the capability of the PE):

- rcvr = 0
- rcvr = 1234
- rcvr = FFFFFF
- no "R" field provided (no receiver number)
- pref = 0
- pref = 1234
- pref = FFFFFF
- acct = 0
- acct = 123456
- acct = 01234567879ABCDEF (may be truncated by some DC-04 reporting methods)
- acct = FEDCBA9876543210 (may be truncated by some DC-04 reporting methods)

C.1.3.8 Message Data (ref. 5.5.1.7)

- (using encryption, if supported) for each DC-07 message type supported by the PE, transmit each event to the receiver simulator and verify the transmitted data packet
    - observe no transmission failures which may indicate a data-sensitive encryption failure
- if the previous test was run with encryption, repeat the test on selected events (10% of total) with encrytion disabled

C.1.3.9 MAC Address Data (ref. Error! Reference source not found.)

- transmit 5 messages with MAC address data enabled (if supported) and verify that all data fields are correctly transmitted, in addition to the MAC data

C.1.3.10 Timestamp (ref. 5.5.1.9) (when supported)

- verify timestamp is referenced to GMT and not local time
- set the time incorrectly and verify that PE will reset the timestamp to the time returned in a NAK from the receiver simulator

C.1.3.11 Supervision Message (ref. 5.5.2) (when supported)

- enable the supervision message, and verify the transmitted message format
- enable encryption (if supported) and verify the transmitted message format
- verify that the supervision interval matches the period shown in the PE documentation
- verify programmability of the supervision interval (if any) as shown in the PE documentation

C.1.3.12 ACK Message (ref. 5.5.3.1)

- confirm that the PE will not repeat a message acknowledged with an ACK
- confirm encrypted operation (when supported)

C.1.3.13 NAK Message (ref. 5.5.3.2, 5.6.1.2)

- cause the receiver simulator to return NAK messages, and observe that the PE will repeat the event report (without incrementing the sequence number)
- queue multiple event reports at the PE, and cause the receiver simulator to return only NAK messages - observe that the PE will eventually discard a NAKed message and transmit the next message in the queue

C.1.3.14 ensure that the NAK message (never encrypted) is correctly received by PE operating in encrypted mode (when supported)

- document the PE's behavior when a message transmission is abandoned due to NAK responses

C.1.3.15 DUH Message (ref. 5.5.3.3)

- cause the receiver simulator to return a DUH response, and observe that the PE does not repeat the transmitted message

- document the PE's behavior when a message transmission is abandoned due to DUH responses

C.1.3.16 No Response (ref. 5.6.1.1)

- prevent the receiver simulator from responding to transmitted messages from the PE, and observe that at least one PE retry attempt occurs within 5 - 60 seconds

C.2 Central Station Receiver Testing

C.2.1 Premises Equipment Simulator

In the case of testing a CSR, a PE simulator tool would be used. The PE simulator might be a PC-based application, that uses the IP capabilities of a PC to support the connection with the CSR. The PE simulator could allow the following message elements to be specified for a message to be sent to the CSR:

- sequence number (seq)
- receiver number (rcvr)
- account prefix (pref)
- account number (acct)
- encryption pad characters
- DC-07 event type
- data message (…data…)
- MAC address
- timestamp

Additionally, the simulator can calculate the message CRC, the message length and create encrypted messages. Capability to generate incorrect CRC is required.

C.2.2 Testing Process

The PE simulator can be used to send reports to the unit under test (UUT). For each test, the data for each field which the simulator is transmitting must be matched against the data for each field that the CSR decodes.

C.2.3 Test Cases

This section recommends test cases to be used for CSR testing:

C.2.3.1 Marking (ref. 4.2)

- confirm that the equipment is marked with only one of the following: "SIA IP Reporting (UDP-2006)", "SIA IP Reporting (TCP-2006)" or "SIA IP Reporting (UDP/TCP-2006)"

- use the PE simulator to confirm that the CSR will accept messages using the supported IP protocol (TCP and/or UDP)

C.2.3.2 IP Addresses (ref 5.2)

- verify that the IP address of the CSR can be manually programmed

C.2.3.3 Encryption (ref 5.4)

- set the PE simulator and CSR with matching encryption keys
- for each DC-07 message type, transmit an encrypted message and ensure that the receiver simulator can decrypt it correctly
- change the encryption key at the PE simulator and ensure that a message is not decrypted correctly
- repeat a subset the encryption tests (10%) for each supported encryption key length

C.2.3.4 Cipher Block Chaining (ref. 5.4.6)

- use the PE simulator to generate a message consisting of 3 encrypted blocks (length over 32 bytes) and observe that the CSR will correctly decrypt the message

C.2.3.5 CRC (ref. 5.5.1.2)

- transmit a message to the CSR with an incorrect CRC and confirm that it does not respond to the message

C.2.3.6 ID Token (ref. 5.5.1.4)

- for each DC-07 message type, transmit an unencrypted message and ensure that the receiver simulator can decode it correctly

C.2.3.7 Sequence Number (ref. 5.5.1.5)

- cause the PE simulator to transmit a 3 block message with an extra block included in a sequence like the following: 1, 2, 1, 3 - confirm that the CSR can assemble and decode the proper message
- cause the PE simulator to transmit 3 single block messages with an extra block included in the sequence like the following: 9995, 9996, 9995 (extra), 9997 - confirm that all three messages are received only once
- send reports from the PE simulator to confirm that the CSR sequence number properly increments through the sequence 9998, 9999, 0001, 0002

C.2.3.8 Account Identification (ref. 5.5.1.6)

Transmit reports from the PE simulator to confirm correct reception the the CSR when the following account ID parameters are included:

- rcvr = 0
- rcvr = 1234
- rcvr = FFFFFF
- no "R" field provided (no receiver number)
- pref = 0
- pref = 1234
- pref = FFFFFF
- acct = 0

- acct = 123456
- acct = 01234567879ABCDEF (may be truncated by some DC-04 reporting methods)
- acct = FEDCBA9876543210 (may be truncated by some DC-04 reporting methods)

C.2.3.9 Message Data (ref. 5.5.1.7)

- (using encryption, if supported) for each DC-07 message type, use the PE simulator to transmit 5 sample messages in which the event code and other parameters are varied
    - observe no transmission failures which may indicate a data-sensitive encryption failure
- if the previous test was run with encryption, repeat the test with encrytion disabled

C.2.3.10 MAC Address Data (ref. Error! Reference source not found.)

- transmit a message with MAC address data included
    - observe that all data elements of the message are correctly decoded
    - document the handling of the MAC data by the CSR

C.2.3.11 Verification Data (ref. Error! Reference source not found.)

- transmit a message to the CSR that includes arbitrary verification data and observe that all required (excludes verification data) data elements of the message are correctly decoded

C.2.3.12 Programming Data (ref. Error! Reference source not found.)

- transmit a message to the CSR that includes arbitrary programming data and observe that all required (excludes programming data) data elements of the message are correctly decoded

C.2.3.13 Timestamp (ref. 5.5.1.9)

- use the PE simulator to transmit an encrypted message with a timestamp that is 25 seconds greater than the time in the CSR - observe that the CSR rejects the message with a NAK including the CSR's time
- use the PE simulator to transmit an encrypted message with a timestamp that is 45 seconds less than the time in the CSR - observe that the CSR rejects the message with a NAK including the CSR's time

C.2.3.14 Supervision Message (ref. 5.5.2) (when supported)

- set the PE simulator to generate supervision messages every 50 seconds and the CSR to require them every minute - observe no supervision failures over 24 hours of testing
- disconnect the PE simulator and observe a supervision failure reported by the CSR within 60 seconds
- verify that the supervision interval at the CSR can be programmed in the range 10 seconds to 3600 seconds, and 1 hour to 1080 hours

- verify by code inspection or other means that the supervision interval is correctly implemented in the CSR

- verify supervision using both encrypted and unencrypted messages

C.2.3.15 DUH Message (ref. 5.5.3.3)

- cause the PE simulator to transmit a correctly formatted message with an invalid DC-07 message type - observe that the CSR returns the DUH response

C.2.3.16 No Response (ref. 5.6.1.1)

- prevent the receiver simulator from responding to transmitted messages from the PE, and observe that at least one retry attempt occurs within 5 - 60 seconds

Annex D: Encryption Keys (Informative Annex)

This standard uses fixed private keys for encryption, to avoid system overhead associated with public keys. Several recommendations for handling private keys apply:

- during system deployment, knowledge of the keys to be used for an installed group of systems should be limited to as few personnel as possible
- the design of the equipment should ensure that authorized personnel are able to store (write) keys in premises equipment or central station receivers, but it should not be possible to read back these keys
- section 5.1 of NIST Special Publication 800-38A provides a useful overview of the cipher block process used by this standard

Annex E - Windows 1252 Character Encoding (Informational Annex)

(from Microsoft Go Global Developer Center)

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152		Ž 017D	
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153		ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Annex F: Example Transmission Sequences (Informational Annex)

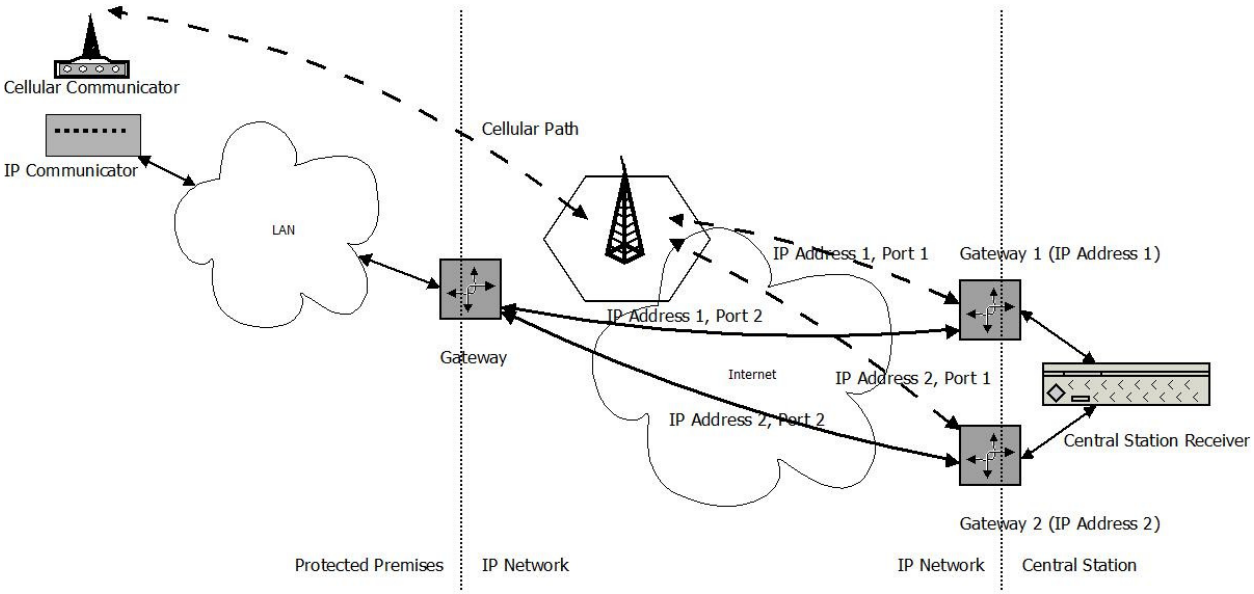


Figure: Reference System Diagram

The diagram above provides a reference example for the sequence diagrams, below, which all assume a system with redundant reporting paths. In this example, the premises equipment has 2 physical paths to communicate to the central station. Additionally, the example shows redundant central station receivers, each with a unique fixed IP address. Redundancy like this, along with scheduled test messages to supervise the various routes (4 in this example), may be required in some applications.

F.1 Primary Transmission Path Failure

Messages are transmitted first using their primary path, then in case of failure or unavailability of the latter, a secondary path.

The waiting period for an acknowledgement before retransmitting the message must be set so as to comply with the requirements of approval agencies or other authorities having jurisdiction.

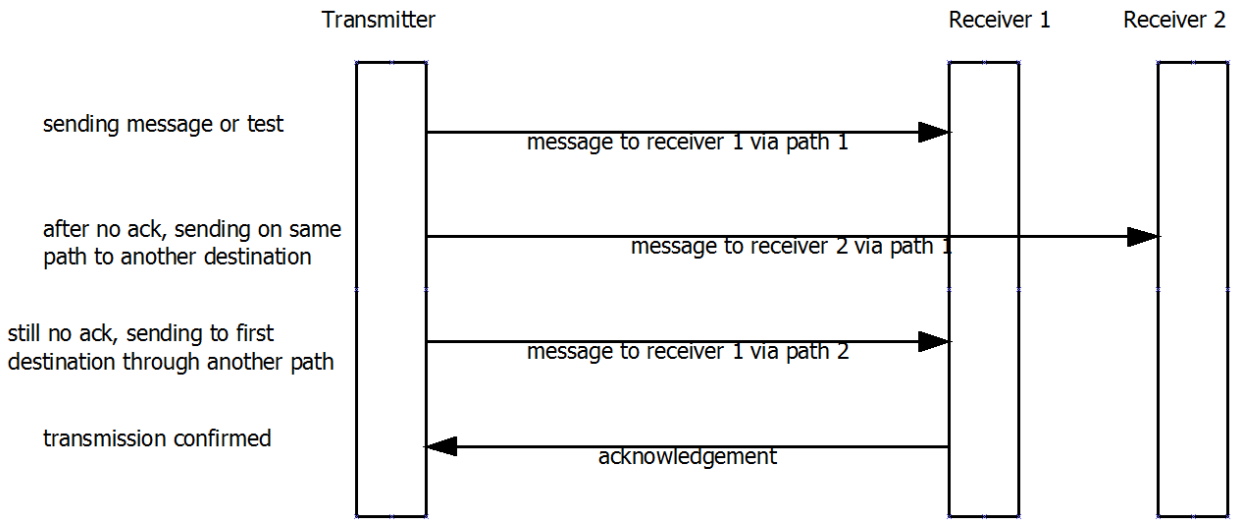


Figure F1

F.2 Successful Encrypted Transmission

An event is transmitted successfully

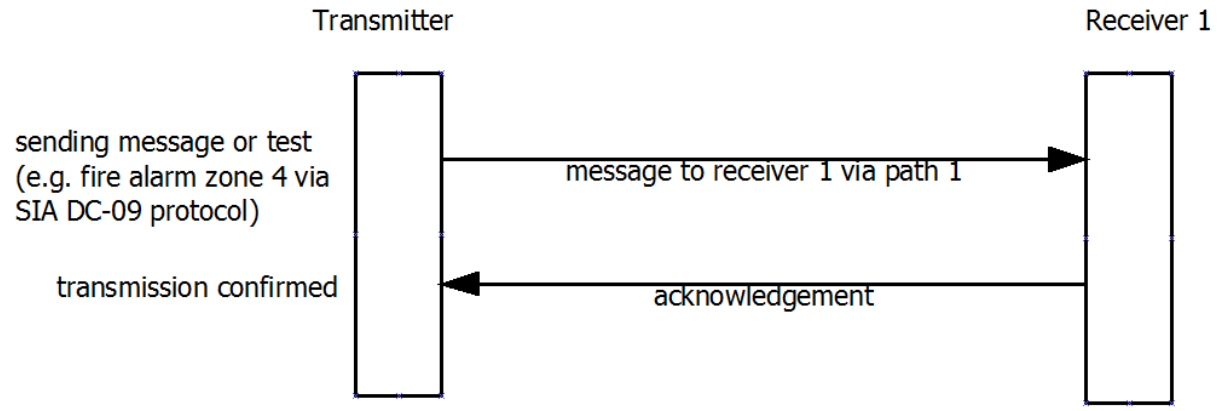


Figure F2

F.3 Primary Receiver Failure

No response to attempt, so transmitter sends to an alternate receiver.

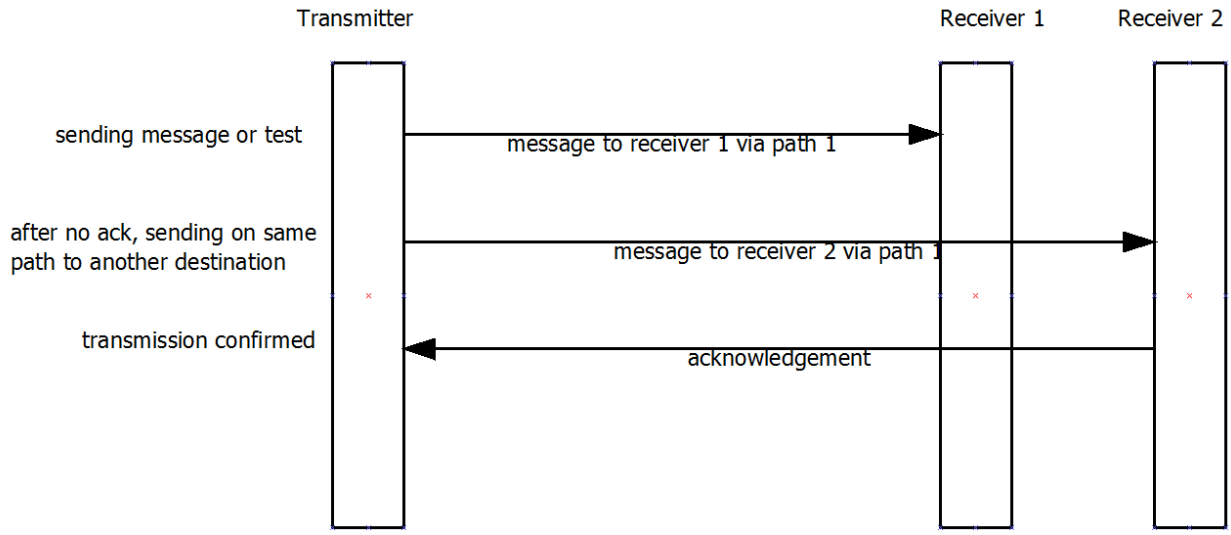


Figure F3

F.4 Unsupported Message

Receiver replies to unsupported message with DUH response.

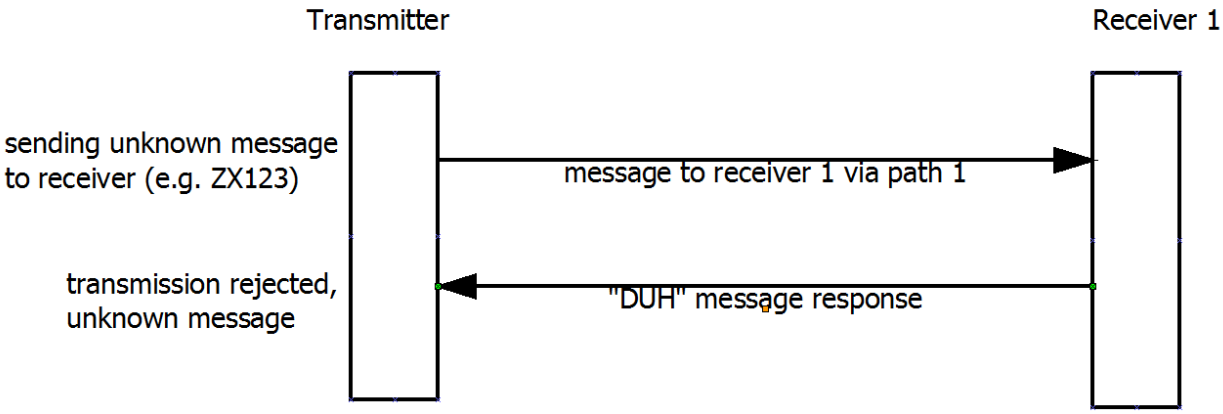


Figure F4

F.5 Expired Time Stamp

This example shows polling message that is rejected with a NAK message (which is never encrypted) with reference time.

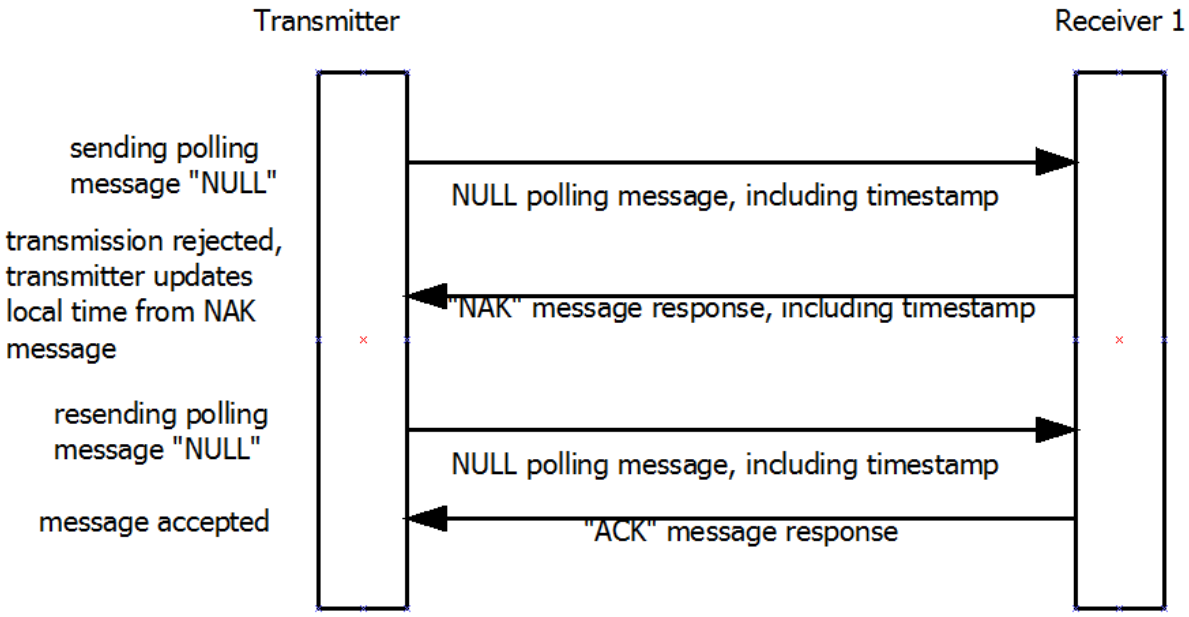


Figure F5

Annex G: Checklist of Major Requirements

This list shows major requirements for a DC-09 implementation:

Requirement	Reference
supports UDP and/or TCP transmission	4.1
marked "SIA IP Reporting (UDP/TCP-2013)"	4.2
list of supported tokens marked	4.2
static address for central station receiver	5.2
encryption, if present, is 128, 192 or 256 bit FIPS-197 (AES)	5.4.1, 5.4.5
for a receiver, encryption, if present, supports all three key lengths	5.4.5
encryption, if present, implements cipher block chaining	5.4.6
encrypted data, if supported, transmitted as ASCII encoded hex characters	5.4.7
receiver supports ACK, NAK and DUH messages	5.5
minimum message frame includes- transmitter: LF, CRC, length, id, sequence, account, account prefix, and CR receiver: LF, CRC, length, id, sequence, account, account prefix, receiver number, and CR	5.5
CRC is transmitted as four ASCII characters	5.5.1.2
sequence number applied by premises equipment	5.5.1.5
sequence number not incremented on message retry	5.5.1.5
sequence number echoed in reply from receiver	5.5.1.5
receiver can send ACK message	5.5.3.1
no response to a message with a failed CRC (receiver)	5.6.2.2
DUH response to an unsupported message (receiver)	5.6.1.3

Annex H: DC-07 Protocol Identifier Tokens (Informative Annex)

This list of tokens is drawn from SIA DC-07, which is the governing document in case of a conflict between this document and DC-07.

Token	***Definition***
SIA-PUL	Generic Pulse Codes
ACR-SF	Acron Super Fast
ADM-CID	**Ademco Contact ID**
ADM-41E	Ademco 4-1 Express
ADM-42E	Ademco 4-2 Express
ADM-HS	Ademco High Speed
DSC-43	DSC 4-3
FBI-SF	FBI Super Fast
ITI-I	ITI Standard
SCN-S8	Scancom 4-8-1, 5-8-1, 6-8-1
SCN-S16	Scancom 4-16-1, 5-16-1, 6-16-1
SCN-S24	Scancom 4-24-1, 5-24-1, 6-24-1
SCT	Scantronics Reserved
SES-SS	Sescoa Super Speed
SIA-DCS	**SIA DCS**
SIA-S2K	SIA 2000
SK-FSK1	Silent Knight FSK1
SK-FSK2	Silent Knight FSK2

Table H1: DC-07 Tokens

Annex I: Optional Remote Commands

ANSI/SIA DC-09:2007 has been successfully extended to incorporate remote commands for some applications. Examples of remote command functions include:

- Open a barrier remotely, such as for highways
- Open the door of a safe
- Activate/deactivate a remote unit securely

This annex describes this optional capability for DC-09 receivers and premises equipment.

Transmission of Commands

In response to a NULL polling message from premises equipment (PE), the central station receiver (CSR) may send a remote command instead of an ACK message. Sending remote commands in response to any other message is not recommended.

Incompatible Premises Equipment

If the PE is not able to process a transmitted remote command, it shall return the "DUH" response.

Message Format

```
<LF><CRC><0LLL>
<"RSP"><seq><Rrcvr><Lpref><#acct>[#acct|Ndata]<CR>
```

If the polling is encrypted, the RSP message will be encrypted as well:

```
<LF><CRC><0LLL>
<"*RSP"><seq><Rrcvr><Lpref><#acct>[<pad>|encrypted data]<encrypted timestamp><CR>
```

In the data field ([#acct|"data]), the character following the "|" that terminates the account number is a "manufacturer identifier".

If this manufacturer identifier character is "N", the subsequent data is in the legacy format, as implemented in some European applications.

If this manufacturer identifier character is " (quotation mark), it is to be followed by 3-7 ASCII characters (not including | o r "), and then a closing ". This field is identifying a manufacturer, similar to the manufacturer tokens defined in SIA DC-07 (see Annex H). At this time, these identifiers are selected independently by manufacturers, and are to be selected so as to clearly identify the manufacturer (e.g. "EQUIPCO").

Examples:

Example 1, "Equipco" Format

```
<LF>CCCC002A"RSP"0001L2#345[#345|"EQUIPCO"RZS22.5,XX5]<CR>
```

- CCCC: checksum, not calculated in this example
- 002A: length

- "RSP": remote command
- 0001: sequence 1
- L2: account prefix 2
- #345: account number 345
- [#345|: start of data field, repeats account number
- "EQUIPCO": manufacturer ID (fictitious in this example)
- RZS22.5,XX5: manufacturer specific message format
-]: end of data field

Example 2: Legacy Format

<LF>**CCCC**001F**"RSP"**0001**L2**#345**[#345|NZZCCnnnn**]<CR>

- CCCC: checksum, not calculated in this example
- 001F: length
- "RSP": remote command
- 0001: sequence 1
- L2: account prefix 2
- #345: account number 345
- [#345|: start of data field, repeats account number
- N: manufacturer ID "legacy"
- NZZCCnnnn: manufacturer specific message format
  - where:
  - N new message
  - ZZ code of fixed criteria, indicating a remote command has been sent from CSR to PE
  - CC code criteria for the type of action, according to a list maintained by the local authority
  - nnnn zone number (corresponds to the number of the physical contact).
-]: end of data field

When using such remote commands, the following implementation process might be used:

- The polling frequency can be increased to shorten response time
- When a remote command is to be sent in response to a NULL message, this remote command message replaces the ACK message for the NULL message. The RSP must be considered as an ACK by the PE
- The PE acknowledges the reception of the remote command message and the change of state requested by the CSR by sending a new event message
- Finally, the CSR sends an acknowledgment message (ACK) to the PE

The figure below shows an example of the sequence:

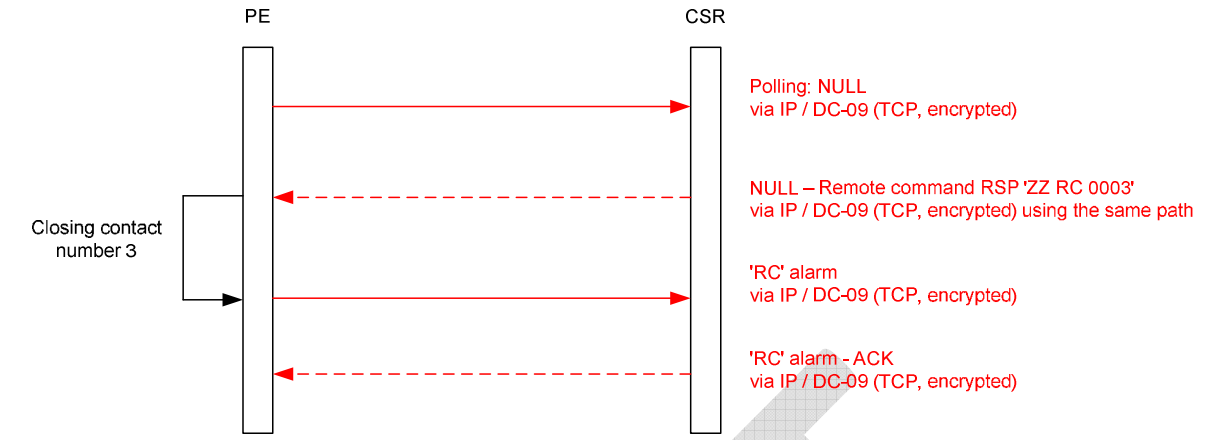


Figure: Remote Command Sequence