UNITED
NATIONS

E

Economic and Social
Council

Distr.
RESTRICTED

TRADE/WP.4/R.1284/Rev.1
30 June 1997

ENGLISH ONLY

ECONOMIC COMMISSION FOR EUROPE

COMMITTEE FOR TRADE, INDUSTRY AND ENTERPRISE DEVELOPMENT

Centre for the Facilitation of Procedures
and Practices for Administration,
Commerce and Transport
(Item 3 of the provisional agenda of the
Meeting of Experts on Data Elements
and Automatic Data Interchange (GE.1)
(fifty-sixth session, 17-18 September 1997)

ELECTRONIC DATA INTERCHANGE FOR ADMINISTRATION, COMMERCE AND TRANSPORT (EDIFACT) - APPLICATION LEVEL SYNTAX RULES

Part 10
Security rules for interactive EDI

* * *

Submitted by the Syntax Development Group *

As requested during the March 1997 sessions of GE.1 and CEFACT, this document presents Part 10 of the revised EDIFACT syntax as prepared by the Syntax Development Group and will, upon approval, be submitted to ISO. It takes into account comments received on the previous version.

The Group of Experts is invited to: approve this document.

The present document is reproduced in the form in which it was received by the secretariat.

GE.97-

ISO 9735-10

Release 2
1997-06-13

Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules

Part 10:
Security rules for interactive EDI

Contents

		Page
Foreword		4
Introduction		6
1	**Scope**	**7**
3	**References**	**7**
4	**Definitions**	**7**
5	**Rules for the use of interactive EDI security**	**8**
5.1	**Security headers and trailers**	**8**
5.2	**Key/certificate management message (KEYCER)**	**11**
5.3	**Secure acknowledgement message (SECACK)**	**14**
5.4	**Security information response message (SECRSP)**	**16**
5.5	**Static security information message (SECURE)**	**18**
5.6	**Security token message (STOKEN)**	**21**
Annex A:	**Syntax service directories (segments, composite data elements and simple data elements)**	**23**
Annex B:	**Examples**	**33**
Annex C:	**Implementation guide**	**43**
Annex D:	**Syntax service code directory**	**57**

Foreword

(To be amended as necessary, according to ISO procedures)

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75% approval by the member bodies voting.

International Standard ISO 9735 Release 2 was prepared by the UN/ECE Trade Division (as UN/EDIFACT) and was adopted, under the "fast-track procedure" as an existing standard, by Technical Committee ISO TC 154, *Documents and data elements in administration; commerce and industry.*

ISO 9735 consists (currently) of the following parts, under the general title *Electronic Data Interchange For Administration, Commerce And Transport (EDIFACT) - Application level syntax rules:*

ISO 9735-1	-	*Syntax rules common to all parts and the syntax service directories*
ISO 9735-2	-	*Syntax rules specific to batch EDI*
ISO 9735-3	-	*Syntax rules specific to interactive EDI*
ISO 9735-4	-	*Syntax and service report message for batch EDI (message type - CONTRL)*
ISO 9735-5	-	*Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)*
ISO 9735-6	-	*Secure authentication and acknowledgement message (message type - AUTACK)*
ISO 9735-7	-	*Security rules for batch EDI (confidentiality)*
ISO 9735-8	-	*Associated data in EDI*
ISO 9735-9	-	*Security key and certificate management message (message type - KEYMAN)*
ISO 9735-10	-	*Security rules for interactive EDI*

Further parts may be added in the future.

In this Part 10, annex A forms an integral part of this International Standard.

Introduction

This International Standard includes the rules at the application level for the structuring of data in the interchange of electronic messages in an open environment, based on the requirements of either batch or interactive processing. These rules have been agreed by the United Nations Economic Commission for Europe (UN/ECE) as syntax rules for Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) and are part of the United Nations Trade Data Interchange Directory (UNTDID) which also includes both batch and interactive Message Design Guidelines.

These syntax rules may be used in any application, but messages using these rules may only be referred to as EDIFACT messages if they comply with other guidelines, rules and directories in the UNTDID. For UN/EDIFACT, messages shall comply with the message design rules for batch or interactive usage as applicable. These rules are maintained in the UNTDID.

Communications specifications and protocols are outside the scope of this standard.

This is a new part, which has been added to ISO 9735. It provides an optional capability of securing an interactive EDIFACT structure, i.e. message, package, or dialogue.

Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules

Part 10: Security rules for interactive EDI

1 Scope

This part of ISO 9735 specifies syntax rules for interactive EDIFACT security. This provides a method to address message/package level and interchange level security in accordance with established security mechanisms.

2 Conformance

Conformance to a standard means that all of its requirements, including all options, are supported. If all options are not supported, any claim of conformance shall include a statement which identifies those options to which conformance is claimed.

Data that is interchanged is in conformance if the structure and representation of the data conforms to the syntax rules specified in this International Standard.

Devices supporting this International Standard are in conformance when they are capable of creating and/or interpreting the data structured and represented in conformance with the standard.

Conformance to this part shall include conformance to Part 1 and Part 3 of this International Standard.

When identified in this International Standard, provisions defined in related standards shall form part of the conformance criteria.

3 Normative references

This International Standard does not refer to other standards.

4 Definitions

For the purpose of this International Standard, the definitions in Part 1 annex A and in Part 5 annex A apply.

5. Rules for the use of interactive EDI security

5.1 Security headers and trailers

Security services addressed in this section shall be provided by the inclusion of security header and trailer segments after the UIH segment and before the UIT segment, in a way which shall be applied to any existing message.

5.1.1 Security headers and trailers schematic

Figure 1 describes a dialogue containing a secured message.

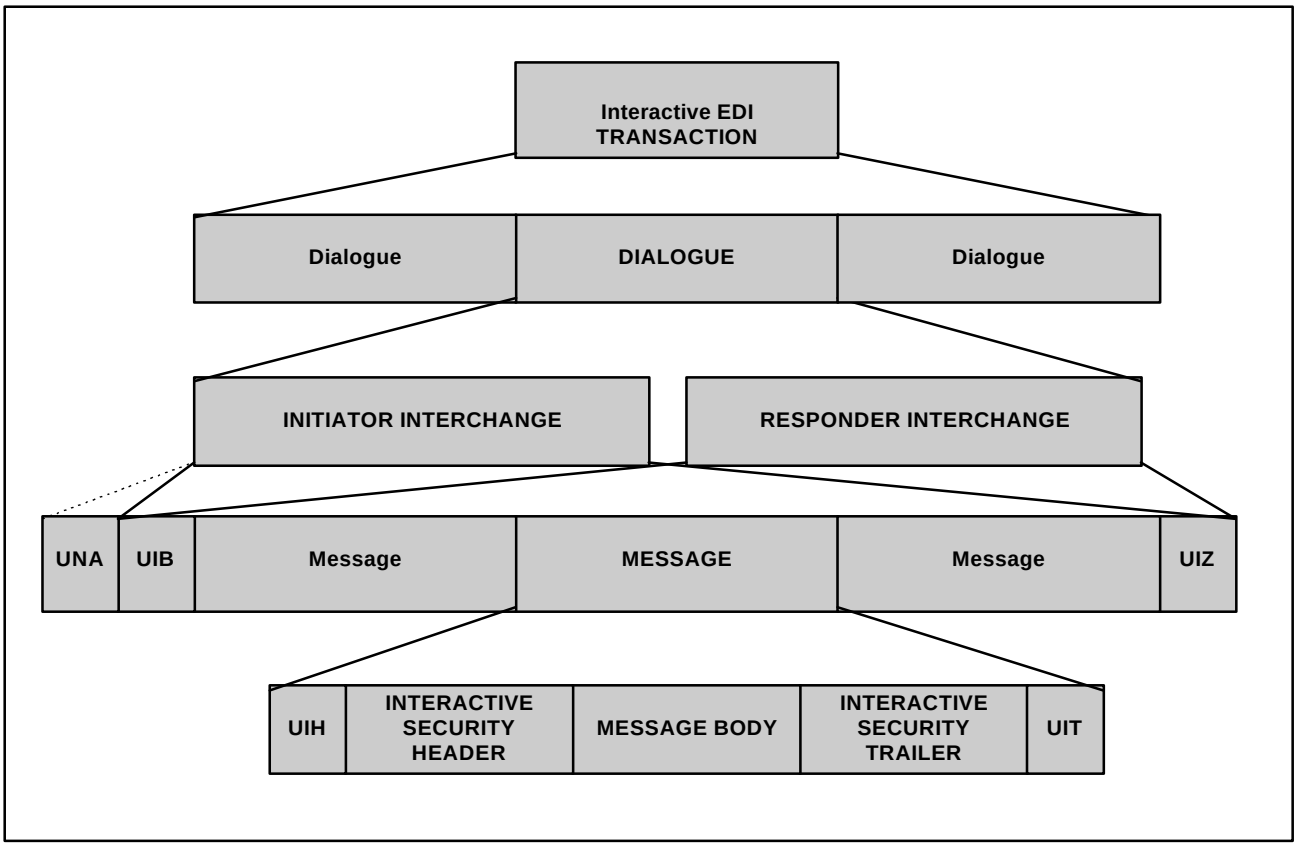

Figure 1 - Dialogue containing a secured message (schematic)

5.1.2 Security headers and trailers segment structure

TAG	Name	S	R	Notes
UIH	Interactive message header	M	1	1
UXH	Interactive security header	C	9	
	INTERACTIVE MESSAGE BODY			
UXT	Interactive security trailer	C	9	
UIT	Interactive message trailer	M	1	1

Note 1: UIH interactive message header and UIT interactive message trailer are specified in Part 1 of this international standard. They are not described further in this Part.

5.1.3 Data segment clarification

UXH, Interactive security header
This segment specifies variable security details applied to an interactive message in which the segment is included. The static security details shall be specified in the corresponding SECURE message which shall precede the use of the UXH segment.

Data element 0534 shall be used to provide a link between the related UXS segment in the corresponding SECURE message, UXH and UXT segments, and data element 0503 shall refer to the SECACK acknowledgement message. There shall always be a linked UXT segment for every UXH segment.
The algorithm parameter (S503) may hold an initialisation vector where data is to be encrypted, which shall be the only time the length of encrypted data and number of padding bytes are used.

The UXH segment may include a security sequence number, to provide sequence integrity, and the date of creation of the security elements.

The final secured message, coded, may be used to indicate whether the protected message is the final message secured by the service referred to in the corresponding SECURE message. This gives the ability to ensure that, where message sequence integrity is used, there are no missing messages before the end of the interchange.

UXT, Interactive security trailer
This segment contains the result of the security functions applied to the message as specified in the linked security header and the corresponding SECURE message. Data element 0534 shall provide a link between the related UXS segment in the corresponding SECURE message, UXH and UXT segments. There shall always be a linked UXT segment for every UXH segment. Data element 0588 shall contain a count of the related UXH segment and this UXT segment.

Depending on the security mechanisms specified in the linked UXS segment, this result shall be either:
- computed directly on the message by the algorithm specified in the USA segment within the SECURE message containing the USX segment, or
- computed by signing with an asymmetric algorithm (specified in the certificate referred to by the USX segment within the SECURE message containing the USX segment) a hash result computed on the message (using the algorithm specified in the USA segment within the corresponding SECURE message).

The security trailer shall always be present even if there is no cryptographic result, for instance where the security header identifies confidentiality (encryption).

5.1.4 Scope of application

The scope of security application, coded refers only to the calculation of a result in a UXT segment (it does not apply to confidentiality, nor to entity authentication), and shall be either:

- Related SECURE message segments, related interactive security header and interactive message body

UIH	SECURE message segments 2	UIT

.....
.

UIH	Security header 3	Security header 2	Security header 1	Message body	Security trailer 1	Security trailer 2	Security trailer 3	UIT

or

- Related SECURE message segments, all other embedded interactive security headers, interactive message body and all other embedded interactive security trailers

UIH	SECURE message segments 2	UIT

.....
.

UIH	Security header 3	Security header 2	Security header 1	Message body	Security trailer 1	Security trailer 2	Security trailer 3	UIT

where the scope of application of the security service defined in the security header 2 is represented by the shaded boxes, and is the concatenation of all of the segments indicated, in each case starting with the first character of the segment(s), namely a “U” to the separator ending the segment(s), inclusive.

5.1.5 Principles of usage

The interactive security header UXH segment and trailer UXT segment surround a message body to provide the required security service. Each security header/trailer pair shall refer back to the static security information, defined in the corresponding SECURE message in the current dialogue, through the use of the security reference number. The security headers and trailers may occur up to 9 times, to allow for multiple security services, if required.

The interactive security header provides the following capabilities:

- linkage to the service initialisation message SECURE and the scope of application;
- a time-variant parameter, such as a reference number or a time-stamp;
- response type requested (e.g. whether a secure acknowledgement is required);
- a dynamic parameter (such as an initialisation value) needed for the provision of the security service;

The interactive security trailer provides the following capabilities:

- linkage to the service initialisation message SECURE
- count of the security segments of the security trailer and the related security header
- the ability to carry a security result such as a MAC or a digital signature

Confidentiality for a whole message body shall be provided for by using the Interactive message header, and specifying the Initialisation vector, if required, and the length in bytes of the encrypted message body which follows and the number of padding bytes, if any. A matching UXT segment shall also be present, with no validation result. The scope of ‘security application, coded’ has no meaning for confidentiality.

The initialisation vector for encrypted data (for confidentiality) shall be defined using composite data element S503, using an appropriate algorithm parameter qualifier. If other related algorithm parameters are required, these shall be defined in a USA segment of the related SECURE message.

The length of encrypted data (for confidentiality) shall start with the first octet following the segment terminator of the UXH segment, up to the octet before the first character, namely a “U” of the corresponding UXT segment, inclusive. The encrypted data need not be filtered, but if it is filtered for other reasons (eg transmission protocol) then the filter function may be indicated in the related SECURE message and the length of encrypted data shall contain the length after filtering. The number of padding bytes shall indicate the number of bytes in the decrypted message which had to be added to the original plain text message in order to utilise the security algorithm. Any compression function applied before encryption may also be indicated in the SECURE message.

5.2 Key/certificate management message (KEYCER)

5.2.1 Functional definition

The interactive EDI message KEYCER allows requests and deliveries of keys/certificates to be made. It has the following capabilities:

- Support of key/certificate management functions, such as certificate request and delivery, and also more advanced functions such as registration, key generation, certification, validation, revocation, discontinuation, and status notification;
- Transfer of keys, certificates, and/or credentials with status information;
- Conveying up to 9 certificates (which can be independent or linked in a certificate chain).

5.2.2 Field of application

The key/certificate management message (KEYCER) may be used for both national and international trade. It is based on universal practice related to administration, commerce and transport, and is not dependent on the type of business or industry.

5.2.3 Principles

The message may be used to request or deliver security keys, certificates, or certification paths (this includes requesting other key and certificate management actions, for example renewing, replacing or revoking certificates, and delivering other information, such as certificate status. This message may be secured like any other, using the security headers and trailers. Key messages shall use only the first USA segment, while certificate messages shall use the USC group with at least the USC segment present.

This message may be secured like any other, using the security headers and trailers. Key messages shall use only the first USA segment, while certificate messages shall use the USC group with at least the USC segment present.

5.2.4 Message definition

5.2.4.1 Data segment clarification

0010 **UIH, Interactive message header**
This is a service segment starting and uniquely identifying a message.
The message type code for the key/certificate management message is KEYCER.

Note: messages conforming to this document must contain the following data in segment UIH, composite S306:

Data element	0065	KEYCER
	0052	4
	0054	1
	0051	UN

0020 **Segment group 1: UXF-USA- SG2**

A group of segments containing a single key, single certificate, or group of certificates forming a certification path.

0030 **UXF, Key management function**

A segment identifying the function of the group it triggers, either a request or a delivery. It shall only repeat to indicate elements of the certification paths, in which case the certificate sequence number shall indicate the position of the following certificate within the certification path.

The UXF segment may also specify the filter function used for the binary fields of the USA segment immediately following this segment.

0040 **USA, Security algorithm**

A segment identifying a security algorithm, the technical usage made of it, and containing the technical parameters required (as defined in Part 5). This segment shall be used for symmetric key requests, discontinuation or delivery. It may also be used for an asymmetric key pair request.

0050 **Segment group 2: USC-USA-USR**

A group of segments containing the data necessary to validate the security methods applied to the message/package, when asymmetric algorithms are used (as defined in Part 5). This group shall be used in the request or delivery of keys and certificates.

0060 **USC, Certificate**

This segment contains the credentials of the certificate owner and identifying the certification authority which has generated the certificate (as defined in Part 5). This segment shall be used for certificate requests such as renewal, or asymmetric key requests such as discontinuation, and for certificate deliveries.

Where it is desired to refer to a non-EDIFACT certificate (such as X.509), the certificate reference in the USC segment (0536) shall contain the reference identification number (0802) from the UNO segment of the package containing the non-EDIFACT certificate, and no other data elements (in order to distinguish it from an EDIFACT certificate reference).

0070 **USA, Security algorithm**

A segment identifying a security algorithm, the technical usage made of it, and containing the technical parameters required (as defined in Part 5). This segment shall be used for certificate requests such as credentials registration, and for certificate deliveries.

0080 **USR, Security result**

This segment contains the result of the security functions applied to the certificate by the certification authority (as defined in Part 5). This segment shall be used for certificate validation or certificate deliveries.

0090 **UIT, Interactive message trailer**
This is a service segment ending a message, giving the total number of segments and the control reference number of the message.

5.2.4.2 Message structure

5.2.4.2.1 Segment table

```
POS        TAG      Name                            S   R     Notes

0010       UIH      Interactive message header      M   1

0020       -----    Segment group 1 -------------   M   9     -------+
                                                                      |
0030       UXF      Interactive key/certificate     M   1             |
                    management function                               |
0040       USA      Security algorithm              C   1             |
                                                                      |
0050       -----    Segment group 2 -------------   C   1     ----+   |
                                                                  |   |
0060       USC      Certificate                     M   1         |   |
0070       USA      Security algorithm              C   3         |   |
0080       USR      Security result                 C   1     -------+

0090       UIT      Interactive message trailer     M   1
```

5.3 Secure acknowledgement message (SECACK)

5.3.1 Functional definition

The message SECACK serves two purposes:

1. secure acknowledgement of one or more previous message(s) in the current dialogue (or possibly a previous dialogue belonging to the same transaction);
2. notification to the other party of error conditions in the security processing.

This message can provide:

- a reference to dialogue, interchange sender, interchange recipient, and message;
- a timestamp;
- a validation result (such as a hash value or signature) regarding the referred message;
- an error code.

SECACK may refer to one or more previous message(s) in the current (or possibly a previous) dialogue. SECACK may also be used to indicate an error condition during entity authentication. This message shall be secured like any other, using the security headers and trailers.

5.3.2 Field of application

The secure acknowledgement message (SECACK) may be used for both national and international trade. It is based on universal practice related to administration, commerce and transport, and is not dependent on the type of business or industry.

5.3.3 Principles

The applied security procedures shall be agreed to by trading partners and specified in an interchange agreement.

The secure acknowledgement message (SECACK) provides secure acknowledgement to secured EDIFACT structures.

The security services are provided by cryptographic mechanisms applied to the content of the original EDIFACT structures. The results of these mechanisms form the body of the SECACK message, supplemented by relevant data such as references of the cryptographic methods used, reference numbers and the date and time of the original EDIFACT structures.

The SECACK message shall use the standard interactive security header and trailers.

The SECACK message may apply to messages from any dialogue belonging to the same transaction.

Note :
Secure acknowledgement is only meaningful for secured EDIFACT structures. To prevent endless loops, a SECACK message shall not be sent in response to a SECACK message.

5.3.4 Message definition

5.3.4.1 Data segment clarification

0010 **UIH, Interactive message header**
This is a service segment starting and uniquely identifying a message.
The message type code for the secure acknowledgement message is SECACK.

Note: messages conforming to this document must contain the following data in segment UIH, composite data element S306:

Data element	0065	SECACK
	0052	4
	0054	1
	0051	UN

0020 **Segment group 1: UXX-USY**
This segment group shall be used to identify a party in the security process and to give security information on the referenced EDIFACT structure.

0030 **UXX, Security references**
This segment shall contain references to the party involved in the security process.
The composite data element “security date and time” may contain the original generation date and time of the referenced EDIFACT structure.

0040 **USY, Security on references**
The USY segment contains a link to a security header and corresponding SECURE message, and the result of the security services applied to the referenced EDIFACT structure as specified in this linked security header and corresponding SECURE message.
In a USY segment the value of data element 0534 shall be identical to the value in data element 0534 in the corresponding UXH segment of the referenced EDIFACT structure.

0050 **UIT, Interactive message trailer**
This is a service segment ending a message, giving the total number of segments and the control reference number of the message.

5.3.4.2 Message structure

5.3.4.2.1 Segment table

```
POS      TAG       Name                            S  R     Notes

0010     UIH       Interactive message header      M  1

0020     -----     Segment group 1 -------------   M  9999  -------+
0030     UXX       Interactive security references M  1            |
0040     USY       Security on references          M  9     -------+

0050     UIT       Interactive message trailer     M  1
```

5.4 Security information response message (SECRSP)

5.4.1 Functional definition

The security information response permits the recipient of a SECURE message to indicate whether the proposed security is supported. A rejection allows further security proposals to be made and to be rejected or accepted. This provides a simple means of negotiation (e.g. implicit negotiation). This message can provide:

- a reference to the corresponding service initialisation message SECURE;
- an indication whether the proposed security mechanisms are supported or not.

5.4.2 Field of application

The security information response message (SECRSP) may be used for both national and international trade. It is based on universal practice related to administration, commerce and transport, and is not dependent on the type of business or industry.

5.4.3 Principles

A SECRSP message shall always be issued on receipt of a SECURE message. It shall not be secured by security headers and trailers. A SECRSP message shall be used only to indicate whether the proposed security details are supported or not. All other errors shall be handled by the appropriate mechanism - for example SECACK for security errors or UIR for syntax and other errors.

5.4.4 Message definition

5.4.4.1 Data segment clarification

0010 **UIH, Interactive message header**
This is a service segment starting and uniquely identifying a message.
The message type code for the **security information response** message is SECRSP.

Note: messages conforming to this document must contain the following data in segment UIH, composite S306:

Data element	0065	SECRSP
	0052	4
	0054	1
	0051	UN

0020 **UXR, Security Response**
This is a service segment responding to a security mechanism specified in a SECURE message for (an) interactive EDIFACT message(s).

0030 **UIT, Interactive message trailer**
This is a service segment ending a message, giving the total number of segments and the control reference number of the message.

5.4.4.2 Message structure

5.4.4.2.1 Segment table

POS	TAG	Name	S	R	Notes
0010	UIH	Interactive message header	M	1	
0020	UXR	Security response	M	1	
0030	UIT	Interactive message trailer	M	1	

5.5 Static security information message (SECURE)

5.5.1 Functional definition

The static security information pertaining to a security service is contained in the SECURE message, transmitted once before the security service is applied (possibly many times). This message can provide:

- a specification of details such as linkage to the secured message(s), security service, initiator and responder identification, filter, character set;
- a specification of the security protocol mechanisms used for entity authentication and key establishment;
- a specification of up to 4 security techniques (e.g. a hash function and a signature algorithm need to be specified when non-repudiation of origin is used);
- references for up to 2 certificates (e.g. the initiator and the responder certificates).

5.5.2 Field of application

The static security information message (SECURE) may be used for both national and international trade. It is based on universal practice related to administration, commerce and transport, and is not dependent on the type of business or industry.

5.5.3 Principles

The SECURE message shall be transmitted once before each new security technique is applied (possibly many times). The SECURE message may be protected by the security headers and trailers applied in the related messages where services other than confidentiality or entity authentication of any form are specified - that is the security result in a subsequent related message trailer can encompass the SECURE message segments. It shall not be secured by interactive security headers and trailers on the SECURE message itself - that is no UXH or UXT segments shall appear between the UIH and UIT of the SECURE message. The SECURE message shall only apply to related messages within the current dialogue.

5.5.4 Message definition

5.5.4.1 Data segment clarification

0010 **UIH, Interactive message header**
This is a service segment starting and uniquely identifying a message.
The message type code for the static security information message is SECURE.

Note: messages conforming to this document must contain the following data in segment UIH, composite S306:

Data element	0065	SECURE
	0052	4
	0054	1
	0051	UN

0020 **UXS, Security information**
This is a service segment specifying a security mechanism applied to (an) EDIFACT structure(s). The parties involved in the security service (security elements originator and security elements recipient), may be identified in this segment, unless they are unambiguously identified by means of certificates (referred to by the UXC segment) when asymmetric algorithms are used.

Security identification details composite data element (S500) shall be used in the UXS segment either:
- if symmetric algorithms are used, or
- if asymmetric algorithms are used and when two certificates are present, in order to distinguish between the originator and the recipient certificates

In this latter case, the identification of the party in composite data element S500 (any of the data elements S500/0511, S500/0513, S500/0515, S500/0586) shall be the same as the identification of the party, qualified as "certificate owner" in one of the composite data elements S500 present in the UXC segment, and data element S500/0577 shall identify the function (originator or recipient) of the party involved.

Data element key name in security identification details composite data element (S500/0538) may be used to establish the key relationship between the sending and receiving parties.
This key relationship may also be established by using the data element identification of the key of the algorithm parameter composite data element(S503/0554) in the USA segment.

S500/0538 in the UXS segment may be used if there is no need to convey a USA segment (because the cryptographic mechanisms have been agreed previously between the partners).

Nevertheless, it is strongly recommended to use either data element S500/0538 in the UXS segment, or data element S503/0554 with the appropriate qualifier in the USA segment, but not both of them.

The UXS segment may specify the filter function used for the binary fields of the USA segment and of the S508 composite data element in the corresponding UXE or UXT segments.

0030 **UXI, Security token initialisation**
This is a service segment defining the token mechanism, initiator and method of calculation.
The security party qualifier may identify the party making the first pass in entity authentication. When “message sender” is used, then the sender of this SECURE message shall make the first pass, whilst when “message recipient” is used, the recipient of this SECURE messages shall make the first pass.

0040 **USA, Security algorithm**
A segment identifying a security algorithm, the technical usage made of it, and containing the technical parameters required. These occurrences of the USA segment outside the certificate may be used to specify asymmetric techniques, in the absence of an explicit certificate reference, as well as symmetric techniques and compression algorithms. For asymmetric techniques, two occurrences may contain the hashing algorithm and the asymmetric algorithm. The other two occurrences are described in Part 7 of this International Standard. For entity authentication, up to four occurrences of the USA segment may be required.

0050 **UXC, Certificate identification**
This is a service segment identifying certificate information and used to locate the full certification the user’s system or at a certification authority. The UXC certificate may contain two occurrences of composite data element S500: one for the certificate owner (identifying the party which signs with the private key associated to the public key contained in this certificate), one for the certificate issuer (certification authority).
This segment shall only be used to provide a link between the certificate owner(s) identified by composite data element S500 in the UXS segment, via an identical composite data element S500 in the UXC segment, to the authenticating party identified by another composite data element S500 in the UXC segment, in the case that certificates can only be uniquely identified by the certification authority details.

0060 **UIT, Interactive message trailer**
This is a service segment ending a message, giving the total number of segments and the control reference number of the message.

5.5.4.2 Message structure

5.5.4.2.1 Segment table

POS	TAG	Name	S	R	Notes
0010	UIH	Interactive message header	M	1	
0020	UXS	Security information	M	1	
0030	UXI	Token initialisation	C	1	
0040	USA	Security algorithm	C	4	
0050	UXC	Certificate	C	2	
0060	UIT	Interactive message trailer	M	1	

5.6 Security token message (STOKEN)

5.6.1 Functional definition

The message STOKEN allows the dialogue initiator and responder to exchange entity authentication and key establishment information, often at the beginning of a dialogue, but also at any time during the dialogue. This message can provide:

- a linkage to the corresponding service initialisation message SECURE, pass number, initiator and responder identification;
- a time-variant parameter, such as a reference number or a time-stamp;
- up to four qualified security result values (allowing to transfer authentication values, key agreement values, etc.)

5.6.2 Field of application

The security token message (STOKEN) may be used for both national and international trade. It is based on universal practice related to administration, commerce and transport, and is not dependent on the type of business or industry.

5.6.3 Principles

Usage of this message shall only be permitted if an entity authentication (and key establishment) service has been initialised through the appropriate SECURE message. The security token message is self-contained, and shall not be protected by security headers and trailers.

5.6.4 Message definition

5.6.4.1 Data segment clarification

0010 **UIH, Interactive message header**
This is a service segment starting and uniquely identifying a message.
The message type code for the security token message is STOKEN.

Note: messages conforming to this document must contain the following data in segment UIH, composite S306:

Data element	0065	STOKEN
	0052	4
	0054	1
	0051	UN

0020 **UXE, Entity authentication and key establishment**
A segment allowing the dialogue initiator and responder to exchange security authentication and key establishment information.
The security reference number shall identify the corresponding SECURE message which initialised this entity authentication.
The security party qualifier may identify the party (A or B) of the entity authentication mechanism. When the initiator sends an STOKEN message and the security party qualifier is “message sender”, then party A shall be implied, whilst if the security party qualifier is “message recipient” then party B shall be implied. When the responder sends an STOKEN message, and the security party qualifier is “message sender”, then party B shall be implied, whilst if the security party qualifier is “message recipient” then party A shall be implied.
Where certificates are needed as part of the authentication process, these may be transmitted using the KEYCER message. The algorithms used, filtering techniques, etc may be identified in the preceding SECURE message.
The validation value qualifier shall define the contents of the validation value.

The pass number may identify the number of the pass in the token mechanism specified in the associated SECURE message.

0030 **UIT, Interactive message trailer**
This is a service segment ending a message, giving the total number of segments and the control reference number of the message.

5.6.4.2 Message structure

5.6.4.2.1 Segment table

POS	TAG	Name	S	R	Notes
0010	UIH	Interactive message header	M	1	
0020	UXE	Entity authentication and key establishment	M	1	
0030	UIT	Interactive message trailer	M	1	

Annex A
(normative)
Addendum - to be added to Part 1 annex C when approved

Syntax service directories
(segments, composite data elements and simple data elements)

A.1. Segment directory

A.1.1 Legend

Function — The function of the segment

POS — The sequential position number of the segment or stand-alone data element or composite data element in the segment table

TAG — The tags of all service segments contained in the segment directory start with the letter “U”. The tags of all service composite data elements start with the letter "S", and the tags of all service simple data elements start with the figure "0"

Name — Name of a SEGMENT in capital letters
Name of a COMPOSITE DATA ELEMENT in capital letters
Name of a STAND-ALONE DATA ELEMENT in capital letters
Name of a component data element in small letters

S — The status of the segment in the structure or of the stand-alone data element or composite data element in the segment, or of the components in the composite (where M = Mandatory, C = Conditional)

R — The maximum number of occurrences of the segment in the structure or of the standalone data element or composite data element in the segment

Repr. — Data value representation of the stand-alone data element or component data element in the composite

a	alphabetic characters
n	numeric characters
an	alphanumeric characters
a3	3 alphabetic characters, fixed length
n3	3 numeric characters, fixed length
an3	3 alphanumeric characters, fixed length
a..3	up to 3 alphabetic characters
n..3	up to 3 numeric characters
an..3	up to 3 alphanumeric characters

A.1.2 Dependency note identifiers

Code	Name
D1	One and only one
D2	All or none
D3	One or more
D4	One or none
D5	If first, then all
D6	If first, then at least one more
D7	If first, then none of the others

See clause 11.5 in Part 1 for the definition of the dependency note identifiers.

A.1.3 Index of segments by tag

TAG	Name
UXC	Certificate identification
UXE	Entity authentication
UXF	Key/certificate management function
UXH	Interactive security header
UXI	Token initialisation
UXR	Security response
UXS	Security information
UXT	Interactive security trailer
UXX	Interactive security references

A.1.4 Index of segments by name

TAG	Name
UXC	Certificate identification
UXE	Entity authentication
UXH	Interactive security header
UXX	Interactive security references
UXT	Interactive security trailer
UXF	Key/certificate management function
UXS	Security information
UXR	Security response
UXI	Token initialisation

A.1.5 Segment specifications

Note:

Only segments not defined in other parts of this International standard are included here.

```
-----------------------------------------------------------------------------

      UXC  CERTIFICATE IDENTIFICATION

      Function: To identify certificate information.

POS   TAG   Name                                        S R   Repr.    Notes

010   0536  CERTIFICATE REFERENCE                       C 1   an..35

020   S500  SECURITY IDENTIFICATION DETAILS             C 2
      0577   Security party qualifier                   M        an..3
      0538   Key name                                   C        an..35
      0511   Security party identification              C        an..17
      0513   Security party code list qualifier         C        an..3
      0515   Security party code list responsible       C        an..3
             agency, coded
      0586   Security party name                        C        an..35
      0586   Security party name                        C        an..35
      0586   Security party name                        C        an..35

-----------------------------------------------------------------------------
```

UXE ENTITY AUTHENTICATION AND KEY ESTABLISHMENT

Function: To allow dialogue initiator and responder to exchange security authentication and key establishment information.

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE NUMBER	M	1	an..14	
020	0584	PASS NUMBER	C	1	n1	
030	S501	SECURITY DATE AND TIME	C	1		
	0517	Date and time qualifier	M		an..3	
	0338	Date, extended	C		n..8	
	0314	Event time	C		an..15	
	0336	Time offset	C		n4	
040	0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
050	S500	SECURITY IDENTIFICATION DETAILS	C	2		
	0577	Security party qualifier	M		an..3	
	0538	Key name	C		an..35	
	0511	Security party identification	C		an..17	
	0513	Security party code list qualifier	C		an..3	
	0515	Security party code list responsible agency, coded	C		an..3	
	0586	Security party name	C		an..35	
	0586	Security party name	C		an..35	
	0586	Security party name	C		an..35	
060	0594	SECURITY TEXT	C	1	an..256	
070	S508	VALIDATION RESULT	C	4		
	0563	Validation value qualifier	M		an..3	
	0560	Validation value	M		an..512	

UXF KEY/CERTIFICATE MANAGEMENT FUNCTION

Function: to specify a key/certificate request or response

POS	TAG	Name	S	R	Repr.	Notes
010	0579	KEY MANAGEMENT FUNCTION QUALIFIER	M	1	an..3	
020	0567	SECURITY STATUS, CODED	C	1	an..3	
030	0572	CERTIFICATE SEQUENCE NUMBER	C	1	n..4	
040	0505	FILTER FUNCTION, CODED	C	1	an..3	

UXH INTERACTIVE SECURITY HEADER

Function: To specify security details applied to an interactive message

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE	M	1	an..14	
020	0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
030	S501	SECURITY DATE AND TIME	C	1		
	0517	Date and time qualifier	M		an..3	
	0338	Date, extended	C		n..8	
	0314	Event time	C		an..15	
	0336	Time offset	C		n4	
040	0541	SCOPE OF SECURITY APPLICATION, CODED	C	1	an..3	
050	0503	RESPONSE TYPE, CODED	C	1	an..3	
060	0553	FINAL SECURED MESSAGE, CODED	C	1	an..3	
070	S503	ALGORITHM PARAMETER	C	1		
	0531	Algorithm parameter qualifier	C		an..3	
	0554	Algorithm parameter value	C		an..512	
080	0556	LENGTH OF DATA IN OCTETS OF BITS	C	1	n..18	
090	0582	NUMBER OF PADDING BYTES	C	1	n..2	

UXI SECURITY TOKEN INITIALISATION

Function: To define the token mechanism, initiator, and method of calculation.

POS	TAG	Name	S	R	Repr.	Notes
010	0597	TOKEN MECHANISM, CODED	M	1	an..3	
020	0599	TOKEN MECHANISM CODE LIST IDENTIFIER	C	1	an..3	
030	0577	SECURITY PARTY QUALIFIER	C	1	an..3	
040	0581	METHOD OF TOKEN CALCULATION, CODED	C	1	an..3	

UXR SECURITY RESPONSE

Function: To respond to a security mechanism specified for (an) interactive EDIFACT message(s).

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE NUMBER	M	1	an..14	
020	0593	SECURITY RESPONSE, CODED	M	1	an..3	

UXS SECURITY INFORMATION

Function: To specify a security mechanism applied to (an) interactive EDIFACT message(s).

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE NUMBER	M	1	an..14	
020	0501	SECURITY SERVICE, CODED	M	1	an..3	
030	S500	SECURITY IDENTIFICATION DETAILS	C	2		
	0577	Security party qualifier	M		an..3	
	0538	Key name	C		an..35	
	0511	Security party identification	C		an..17	
	0513	Security party code list qualifier	C		an..3	
	0515	Security party code list responsible agency, coded	C		an..3	
	0586	Security party name	C		an..35	
	0586	Security party name	C		an..35	
	0586	Security party name	C		an..35	
040	0505	FILTER FUNCTION, CODED	C	1	an..3	
050	0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	
060	0509	ROLE OF SECURITY PROVIDER, CODED	C	1	an..3	

UXT INTERACTIVE SECURITY TRAILER

Function: To contain the result of the security mechanisms.

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE NUMBER	M	1	an..14	
020	0588	NUMBER OF SECURITY SEGMENTS	M	1	n..10	
030	S508	VALIDATION RESULT	C	2		
	0563	Validation value qualifier	M		an..3	
	0560	Validation value	C		an..512	

UXX INTERACTIVE SECURITY REFERENCES

Function: To refer to the EDIFACT structure to be secured and its creation date and time.

POS	TAG	Name	S	R	Repr.	Notes
010	0340	INTERACTIVE MESSAGE REFERENCE NUMBER	M	1	an..35	
020	S302	DIALOGUE REFERENCE	C	1		
	0300	Initiator control reference	M		an..35	
	0303	Initiator reference identification	C		an..35	
	0051	Controlling agency, coded	C		an..3	
	0304	Responder control reference	C		an..35	
030	S002	INTERCHANGE SENDER	C	1		
	0004	Interchange sender identification	M		an..35	
	0007	Identification code qualifier	C		an..3	
	0008	Interchange sender internal identification	C		an..35	
	0042	Interchange sender internal sub-identification	C		an..35	
040	S003	INTERCHANGE RECIPIENT	C	1		
	0010	Interchange recipient identification	M		an..35	
	0007	Identification code qualifier	C		an..3	
	0014	Interchange recipient internal identification	C		an..35	
	0046	Interchange recipient internal sub-identification	C		an..35	
050	S501	SECURITY DATE AND TIME	C	1		
	0517	Date and time qualifier	M		an..3	
	0338	Date, extended	C		n..8	
	0314	Event time	C		an..15	
	0336	Time offset	C		n4	
060	S509	SECURITY REFERENCE IDENTIFICATION	C	1		
	0591	Security reference qualifier	M		an..3	
	0588	Security reference	C		an..256	

A.3 Simple data element directory

A.3.1 Legend

TAG The tags of all service simple data elements contained in the simple data element directory start with the figure “0”.

Name Name of a simple data element

Desc. Description of the simple data element

Repr. Data value representation of the simple data element :

a	alphabetic characters
n	numeric characters
an	alphanumeric characters
a3	3 alphabetic characters, fixed length
n3	3 numeric characters, fixed length
an3	3 alphanumeric characters, fixed length
a..3	up to 3 alphabetic characters
n..3	up to 3 numeric characters
an..3	up to 3 alphanumeric characters

Notes Simple data element note number(s)

A.3.2 Index of simple data elements by tag

TAG	Name
0553	Final secured message, coded
0581	Method of token calculation, coded
0584	Pass number
0593	Security response, coded
0594	Security text
0597	Token mechanism, coded
0599	Token mechanism code list identifier

A.3.3 Index of simple data elements by name

TAG	Name
0553	Final secured message, coded
0581	Method of token calculation, coded
0584	Pass number
0593	Security response, coded
0594	Security text
0597	Token mechanism, coded
0599	Token mechanism code list identifier

A.3.4 Simple data element specifications

Note:

Only simple data elements not defined in other parts of this International standard are included here.

```
-----------------------------------------------------------------------------

    0553  FINAL SECURED MESSAGE, CODED

    Desc: Identification of the final secured message in a dialogue.

    Repr: an..3

-----------------------------------------------------------------------------

    0581  METHOD OF TOKEN CALCULATION, CODED

    Desc: Identification of the data representation used in calculating a
          security token.

    Repr: an..3

-----------------------------------------------------------------------------

    0584  PASS NUMBER

    Desc: Number indicating the pass number of a protocol.

    Repr: n1

-----------------------------------------------------------------------------

    0593  SECURITY RESPONSE, CODED

    Desc: Response to the proposal of a security mechanism.

    Repr: an..3

-----------------------------------------------------------------------------

    0594  SECURITY TEXT

    Desc: Text related to the security mechanism.

    Repr: an..256

-----------------------------------------------------------------------------

    0597 TOKEN MECHANISM, CODED

    Desc: Identification of authentication and key establishment standards and
          types.

    Repr: an..3

-----------------------------------------------------------------------------
```

0599 TOKEN MECHANISM CODE LIST IDENTIFIER

Desc: Identification of the code list used to identify the token mechanism.

Repr: an..3

Annex B
(informative)

Examples

Throughout these examples “I” is used as an abbreviation for the initiator, and “R” as an abbreviation for the responder.

B.1 Non-repudiation of origin

This section gives an example of the interactive EDI implementation of non-repudiation of origin, for one message in the dialogue, using security headers and trailers.

B.1.1 Dialogue example

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	Comment
Interchange Header (UIB)	→		
	←	Interchange Header (UIB)	
UIH UXS[1] USA UXC UIT	→		SECURE
	←	UIH UXR[1] UIT	SECRSP
Messages (UIH UXH[1]..UXT[1] UIT)	→		secured messages
	←	Messages (UIH..UIT)	
Interchange Trailer (UIZ)	→		
	←	Interchange Trailer (UIZ)	

B.1.2 Interactive EDI implementation

UXS[1] - SECURITY INFORMATION

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ 1st (only) UXS
0501	SECURITY SERVICE, CODED	M	1	an..3	‘1’ refers to NRO
S500	SECURITY IDENTIFICATION DETAILS	C	2		I’s details
0577	Security party qualifier	M		an..3	‘1’ message sender
0538	Key name	C		an..35	Key name
0511	Security party identification	C		an..17	
0513	Security party code list qualifier	C		an..3	
0515	Security party code list responsible agency, coded	C		an..3	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0505	FILTER FUNCTION, CODED	C	1	an..3	‘2’ HEX
0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	‘1’ ASCII
0509	ROLE OF SECURITY PROVIDER, CODED	C	1	an..3	‘1’ issuer
0519	COMPRESSION FUNCTION, CODED	C	1	an..3	

Details of the algorithm in use may be included in the SECURE USA segments. Details of the public key certificates in use may be held in the SECURE UXC segments. If it is required to transmit complete certificates, this shall be achieved using the KEYCER message.

UXR[1] - SECURITY RESPONSE

TAG	Name	S	R	Repr.	Notes
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ refers to UXS[1]
0593	SECURITY RESPONSE, CODED	M	1	an..3	‘1’ security supported

UXH[1] - INTERACTIVE SECURITY HEADER

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ refers to UXS[1]
0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
S501	SECURITY DATE AND TIME	C	1		time stamp
0517	Date and time qualifier	M		an..3	
0338	Date, extended	C		n..8	
0314	Event time	C		an..15	
0336	Time offset	C		n4	
0541	SCOPE OF SECURITY APPLICATION, CODED	C	1	an..3	‘4’ SECURE segments, UXH & body only
0503	RESPONSE TYPE, CODED	C	1	an..3	
0553	FINAL SECURED MESSAGE, CODED	C	1	an..3	
S503	ALGORITHM PARAMETER	C	1		
0531	Algorithm parameter qualifier	C		an..3	
0534	Algorithm parameter value	C		an..512	
0556	LENGTH OF ENCRYPTED OBJECT	C	1	n..9	
0582	NUMBER OF PADDING BYTES	C	1	n..2	

UXT[1] - INTERACTIVE SECURITY TRAILER

TAG	Name	S	R	Repr.	Notes
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ refers to UXS[1]
0588	NUMBER OF SECURITY SEGMENTS	M	1	n..10	‘2’ segments - UXH/UXT
S508	VALIDATION RESULT	C	1		Signature
0563	Validation value qualifier	M		an..3	‘1’ unique value
0560	Validation value	C		an..512	

The data in UXS, USA and UXC are protected by the signature which includes these segments, as well as UXH and the message body.

B.2 Non-repudiation of receipt

This section gives an example of the interactive EDI implementation of non-repudiation of receipt, for one message in the dialogue, using security SECACK. The Initiator’s segment details (and SECRSP for the responder) are as described in section B.1.

B.2.1 Dialogue example

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	Comment
Interchange Header (UIB)	→		
	←	Interchange Header (UIB)	
UIH UXS[1] USA UXC UIT	→		I's SECURE
	←F	UIH UXR[1] UIT	R's SECRSP
	←L	UIH UXS[2] USA UXC UIT	R's SECURE
UIH UXR[2] UIT	F→		I's SECRSP
UIH UXH[1] message[1] UXT[1] UIT	L→		secured message
	←	UIH UXH[2] UXX[1] USY[1] UXT[2] UIT	SECACK message
Interchange Trailer (UIZ)	→		
	←	Interchange Trailer (UIZ)	

The notation F, L *in the second column indicates the use of the UIH transfer position (0323) to define the first, intermediate and last messages in a transfer.*

B.2.2 Interactive EDI implementation

UXS[2] - SECURITY INFORMATION

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	'2' 2nd UXS
0501	SECURITY SERVICE, CODED	M	1	an..3	'5' refers to NRR
S500	SECURITY IDENTIFICATION DETAILS	C	2		R's details
0577	Security party qualifier	M		an..3	'1' message sender
0538	Key name	C		an..35	Key name
0511	Security party identification	C		an..17	
0513	Security party code list qualifier	C		an..3	
0515	Security party code list responsible agency, coded	C		an..3	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0505	FILTER FUNCTION, CODED	C	1	an..3	'2' HEX
0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	'1' ASCII
0509	ROLE OF SECURITY PROVIDER, CODED	C	1	an..3	'1' issuer
0519	COMPRESSION FUNCTION, CODED	C	1	an..3	

Details of the algorithm in use may be included in the SECURE USA segments. Details of the public key certificates in use may be held in the SECURE USC segments. If it is required to transmit complete certificates, this shall be achieved using the KEYCER message. In practice, the responder's UXS may be

labelled ‘1’, since it may be possible to distinguish the initiator’s and responder’s security reference number by usage. In this example they are labelled differently for clarity.

UXR[2] - SECURITY RESPONSE

TAG	Name	S	R	Repr.	Notes
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘2’ refers to UXS[2]
0593	SECURITY RESPONSE, CODED	M	1	an..3	‘1’ security supported

UXH[2] - INTERACTIVE SECURITY HEADER

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘2’ refers to UXS[2]
0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
S501	SECURITY DATE AND TIME	C	1		time stamp
0517	Date and time qualifier	M		an..3	
0338	Date, extended	C		n..8	
0314	Event time	C		an..15	
0336	Time offset	C		n4	
0541	SCOPE OF SECURITY APPLICATION, CODED	C	1	an..3	‘4’ SECURE segments, UXH & body only
0503	RESPONSE TYPE, CODED	C	1	an..3	
0553	FINAL SECURED MESSAGE, CODED	C	1	an..3	
S503	ALGORITHM PARAMETER	C	1		
0531	Algorithm parameter qualifier	C		an..3	
0534	Algorithm parameter value	C		an..512	
0556	LENGTH OF ENCRYPTED OBJECT	C	1	n..9	
0582	NUMBER OF PADDING BYTES	C	1	n..2	

UXX[1] - INTERACTIVE SECURITY REFERENCES

TAG	Name	S	R	Repr.	Example
0340	INTERACTIVE MESSAGE REFERENCE NUMBER	M	1	an..35	Message to be acknowledged
S302	DIALOGUE REFERENCE	C	1		Current dialogue
0300	Initiator control reference	M		an..35	
0303	Initiator reference identification	C		an..35	
0051	Controlling agency, coded	C		an..3	
0304	Responder control reference	C		an..35	
S002	INTERCHANGE SENDER	C	1		
0004	Sender identification - level one	M		an..35	
0007	Level one identification code qualifier	C		an..35	
0008	Sender identification - level two	C		an..3	
0040	Sender identification - level three	C		an..35	
S003	INTERCHANGE RECIPIENT	C	1		
0010	Recipient identification - level one	M		an..35	
0007	Level one identification code qualifier	C		an..35	
0014	Recipient identification - level two	C		an..3	
0044	Recipient identification - level three	C		an..35	
S501	SECURITY DATE AND TIME	C	1		
0517	Date and time qualifier	M		an..3	
0338	Date, extended	C		n..8	
0314	Event time	C		an..15	
0336	Time offset	C		n4	
S509	SECURITY REFERENCE IDENTIFICATION	C	1		Not used
0591	Security reference qualifier	M		an..3	
0588	Security reference	C		an..35	

USY[1] - SECURITY ON REFERENCES

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ for original UXH[1]
S508	VALIDATION RESULT	M	1		signature in original UXT[1]
0563	Validation value qualifier	M		an..3	‘1’ unique value
0560	Validation value	C		an..512	
0571	SECURITY ERROR, CODED	C	1	an..3	

UXT[2] - INTERACTIVE SECURITY TRAILER

TAG	Name	S	R	Repr.	Notes
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘2’ refers to UXS[2]
0588	NUMBER OF SECURITY SEGMENTS	M	1	n..10	‘2’ segments - UXH/UXT
S508	VALIDATION RESULT	C	1		Signature
0563	Validation value qualifier	M		an..3	‘1’ unique value
0560	Validation value	C		an..512	

B.3 Entity authentication

This section gives an example of the interactive EDI implementation of entity authentication, for the one-pass mechanism of ISO/IEC 9798-3, using the new message STOKEN.

B.3.1 Dialogue example

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	Comment
Interchange Header (UIB)	→		
	←	Interchange Header (UIB)	
UIH UXS[1] UXI USA UXC UIT	→		I’s SECURE is used to initialise entity authentication.
	←	UIH UXR[1] UIT	The responder accepts the requested entity authentication, using the SECRSP message.
UIH UXE[1] UIT	→		The STOKEN message conveys the authentication information.
	←	Messages(s) (UIH ... UIT)	After validation of STOKEN the responder begins the actual dialogue.
Interchange Trailer (UIZ)	→		
	←	Interchange Trailer (UIZ)	

B.3.2 Interactive EDI implementation

UXS[1] - SECURITY INFORMATION

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ 1st (only) UXS
0501	SECURITY SERVICE, CODED	M	1	an..3	‘11’ refers to entity authentication
S500	SECURITY IDENTIFICATION DETAILS	C	2		I’s details
0577	Security party qualifier	M		an..3	‘1’ message sender
0538	Key name	C		an..35	Key name
0511	Security party identification	C		an..17	I's identification
0513	Security party code list qualifier	C		an..3	
0515	Security party code list responsible agency, coded	C		an..3	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
S500	SECURITY IDENTIFICATION DETAILS	C	2		
0577	Security party qualifier	M		an..3	
0538	Key name	C		an..35	
0511	Security party identification	C		an..17	
0513	Security party code list qualifier	C		an..3	
0515	Security party code list responsible agency, coded	C		an..3	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0505	FILTER FUNCTION, CODED	C	1	an..3	‘2’ HEX
0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	‘1’ ASCII
0509	ROLE OF SECURITY PROVIDER, CODED	C	1	an..3	‘1’ issuer
0519	COMPRESSION FUNCTION, CODED	C	1	an..3	

ISO 9735-10:1997

UXI - SECURITY TOKEN INITIALISATION

TAG	Name	S	R	Repr.	Notes
0597	TOKEN MECHANISM, CODED	M	1	an..3	‘107’ for ISO 9798-3 public key algorithm unilateral one pass
0599	TOKEN MECHANISM CODE LIST IDENTIFIER	C	1	an..3	‘1’ for UN
0577	SECURITY PARTY QUALIFIER	C	1	an..3	‘1’ for initiator
0581	METHOD OF TOKEN CALCULATION, CODED	C	1	an..3	‘1’ for EDIFACT representation

Details of the algorithms in use may be included in the SECURE USA segments (digital signature, hash function). Details of the public key certificates in use may be held in the SECURE UXC segments. If it is required to transmit complete certificates, this shall be achieved using the KEYCER message.

UXR[1] - SECURITY RESPONSE

TAG	Name	S	R	Repr.	Notes
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ refers to UXS[1]
0593	SECURITY RESPONSE, CODED	M	1	an..3	‘1’ for security details supported.

UXE[1] - ENTITY AUTHENTICATION

TAG	Name	S	R	Repr.	Example
0534	SECURITY REFERENCE NUMBER	M	1	an..14	‘1’ refers to UXS[1]
0584	PASS NUMBER	C	1	n1	1
S501	SECURITY DATE AND TIME	C	1		time stamp
0517	Date and time qualifier	M		an..3	
0338	Date, extended	C		n..8	
0314	Event time	C		an..15	
0336	Time offset	C		n4	
0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
S500	SECURITY IDENTIFICATION DETAILS	C	2		R’s details
0577	Security party qualifier	M		an..3	‘2’ for message receiver
0538	Key name	C		an..35	
0511	Security party identification	C		an..17	R’s identification
0513	Security party code list qualifier	C		an..3	
0515	Security party code list responsible agency, coded	C		an..3	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0586	Security party name	C		an..35	
0594	SECURITY TEXT	C	1	an..256	
S508	VALIDATION RESULT	M	1		
0563	Validation value qualifier	M		an..3	‘1’ unique value
0560	Validation value	C		an..512	I’s signature

Annex C
(informative)

Implementation guide

C.1 Structure of a secure dialogue

The purpose of this section is to survey the security-related phases of a dialogue and to discuss their implementation in interactive EDI.

Interactive EDI uses a dynamic dialogue between the two partners, where two interleaved interchanges are present, and the minimum requirement is to have one request message and one response message. There are no message groups to be provided for in interactive EDI. The following table illustrates the basic structure of an (unprotected) interactive EDI dialogue.

INITIATOR INTERCHANGE		**RESPONDER INTERCHANGE**
Interchange Header (UIB)	⇒	
Message(s) (UIH..UIT)	⇒	
	⇐	Interchange Header (UIB)
	⇐	Message(s) (UIH..UIT)
Message(s) (UIH..UIT)	⇒	
	⇐	Message(s) (UIH..UIT)
Message(s) (UIH..UIT)	⇒	
Interchange Trailer (UIZ)	⇒	
	⇐	Message(s) (UIH..UIT)
	⇐	Interchange Trailer (UIZ)

To secure an interactive dialogue requires a number of additional phases or sub-protocols related to the provision of the security services. The following figure illustrates these phases in general terms. The security phases are all optional and need not necessarily occur in the order displayed below (however, the order shown is the natural one).

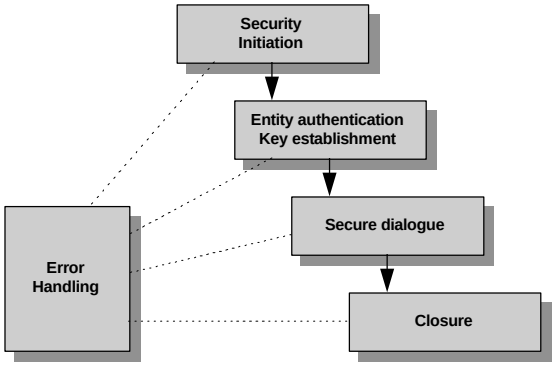

Figure C.1 General phases to provide a secure dialogue

The different phases serve specific purposes. In an implementation, these phases are typically interleaved (e.g. entity authentication and key establishment are handled in a combined mechanism).

1. Initiation: initialisation of one or more security services, including the required mechanisms, to be applied during the dialogue.
2. Entity authentication: serves to corroborate that a party is the one claimed, that is, to authenticate the dialogue partner. Key establishment is normally provided as a sub-product of entity authentication and stands for the agreement on one or more shared secret keys to be used to protect the application data during the interactive dialogue.
3. Secure dialogue: secure exchange of application data, that is, EDIFACT structures.
4. Closure: secure termination of the interactive dialogue.
5. Error handling: alerting the other party that an error condition has occurred (such as the failure to receive the expected protocol message or the failure to process a security result correctly). Error handling must be carried out in all phases of an interactive dialogue. But it is especially important during the set-up phase.

C.1.1 Initiation

In EDI in general, the partners agree beforehand on the message types, on the corresponding message implementation guides, and on the security techniques to be used. This is typically done by signing an Interchange Agreement. But the EDIFACT security syntax is open in the sense that any security technique can be supported through specifying the appropriate code in a code list.

In interactive EDI, security initiation occurs when one party sends a SECURE message to notify the other party that it wishes to apply a security service within the dialogue. Each security service can be initiated independently. SECURE holds an indicator of the security service and all the static information pertaining to the security service (such as associated algorithms and keys).

Upon reception of a security service initialisation message SECURE, the recipient must reply with a security response SECRSP notifying the originator whether the specified security service and algorithms can be supported. In case of a negative response, the originator could then try with a different set of techniques. It could be said that secure interactive EDI provides an implicit negotiation capability.

As interactive EDI is based on the concept of two interleaved interchanges (the initiator and the responder interchange), the message security services must be requested and confirmed independently in either interchange. That means, the interactive EDI security syntax allows the initiator interchange and the responder interchange to use different sets of security services and techniques. However, a minimal requirement is that both parties agree on the entity authentication and key establishment mechanism to be used.

If, during the dialogue, one of the parties wishes to change the security services and techniques, this can be done by sending a SECURE message.

C.1.2 Entity authentication and key establishment

Entity authentication is a security service not present in batch EDI, but absolutely required for interactive dialogues. See section C.2.7 for a discussion of the implementation of entity authentication in interactive EDI. In secure interactive EDI, the requesting party initiates a specific entity authentication mechanism by sending the corresponding initialisation message SECURE. The responding party accepts it by sending a positive SECRSP. The entity authentication mechanism is then started by the party indicated as the initiating party of the mechanism.

If the responding party cannot comply with the requested mechanism, an error condition must be signalled. This is done by a negative SECRSP. Those entity authentication mechanisms supported in a business context are typically prescribed in an Interchange Agreement.

Key establishment is closely related to entity authentication and normally provided together with entity authentication. The objective is to establish one or more shared secret keys for the duration of the interactive dialogue. These shared keys are typically used for the provision of dialogue-oriented security services (such as confidentiality or origin authentication). Many of the key establishment schemes (such as Diffie-Hellman) are inherently interactive, that is, they require an exchange of protocol messages. Other schemes are based

on the transport paradigm, that is, they transport the key from one party to the other, suitably encrypted under a master key (symmetric or asymmetric). ISO/IEC 11770-3 provides a number of key agreement and key transport mechanisms, typically in combination with entity authentication. In secure interactive EDI, key establishment may be accomplished as a sub-product of entity authentication. Regarding the initialisation of key establishment protocols in interactive EDI, the same comments as for entity authentication apply.

At the end of the entity authentication and the key establishment phases, the two parties involved in the dialogue must be assured that the protocols were completed successfully, e.g. that the shared secret key(s) were established correctly at either end. This is typically achieved by sending a cryptographic check value, computed using the newly established secret key, to the other party. After verification of the corresponding check values, the parties know that they are in possession of the correct key(s), and may start a secured dialogue. In interactive EDI, there is no explicit confirmation facility provided. Rather, such cryptographic check values can be sent at the end of the entity authentication and the key establishment phase, integrated into the security messages of the entity authentication and key establishment protocols.

It is also assumed that as soon as a party has successfully completed a protocol, it can directly switch to ‘secure mode’, that is, without waiting for an explicit confirmation. The other party can then follow or reply with a failure condition.

C.1.3 Secure dialogue

Once the security context between the dialogue parties has been established (i.e. the parties are authenticated, the necessary keys have been established and confirmed, and the message security services have been initialised) the parties can switch to the secure dialogue mode.

In interactive EDI, a secure dialogue means that the messages of the initiator and/or the responder interchange are protected. Interactive EDI provides the following security services for the messages within an interchange (see section C.2):

- Non-Repudiation of Origin
- Non-Repudiation of Receipt
- Confidentiality of Content
- Content Integrity
- Origin Authentication
- Sequence Integrity

In addition, it is possible to change the security services and mechanisms at any time during an interactive EDI dialogue by sending a SECURE message holding the new definitions.

Although the order of entity authentication and key establishment first, followed by a secure dialogue is the natural one, the interactive EDI security syntax allows the initialisation of entity authentication at any time during a dialogue. Thus, it is possible to request entity authentication any time a crucial data exchange is to be performed.

The following diagram shows a typical structure of a secure interactive EDI dialogue.

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	COMMENTS
Interchange Header: UIB	⇒		
UIH UXS(1) UXI USA UXC UIT	⇒		SECURE to initialise three-pass entity authentication
	⇐F	Interchange Header: UIB	
	⇐L	UIH UXR(1) UIT	SECRSP to accept three-pass entity authentication
UIH UXF USC USA USA USA USR UIT	F⇒		optional: KEYCER to transfer a public key certificate.
UIH UXE(1) UIT	L⇒		STOKEN: first pass
	⇐F	UIH UXF USC USA USA USA USR UIT	optional: KEYCER to transfer a public key certificate.
	⇐L	UIH UXE(1) UIT	STOKEN: second pass
UIH UXE(1) UIT	⇒		STOKEN: third pass
	.....		
UIH UXS(2) USA UXC UIT	⇒		SECURE to initialise non-repudiation services for the initiator.
	⇐	UIH UXR(2) UIT	SECRSP to accept non-repudiation.
UIH [UXH(2)]..[UXT(2)] UIT	⇒		Possibly signed messages.
	⇐	UIH UXS(3) USA UXC UIT	SECURE to initialise non-repudiation services for the responder.
UIH UXR(3) UIT	⇒		SECRSP to accept non-repudiation.
	⇐F	UIH UXH(3) UXX USY UXT(3) UIT	SECACK to provide non-repudiation of receipt (responder).
	⇐L	UIH [UXH(3)]..[UXT(3)] UIT	Possibly signed messages.
UIH UXH(2) UXX USY UXT(2) UIT	⇒		SECACK to provide non-repudiation of receipt (initiator).
	...		
UIH [UXH(2)]..[UXT(2)] UIT	F⇒		Possibly signed messages.
Interchange Trailer: UIZ	L⇒		
	⇐F	UIH [UXH(3)]..[UXT(3)] UIT	Possibly signed messages.
	⇐L	Interchange Trailer: UIZ	

The notation F, L *in the second column indicates the use of the UIH transfer position (0323) to define the first, intermediate and last messages in a transfer.*

Where UXS(i) UXI USA UXC is the security information in the security (SECURE) message (i = 1, 2, 3, etc), UXR(i) is the security information response (SECRSP) message body, UXH(i) the related interactive security header and UXT(i) the related interactive security trailer. UXE(i) is the new security token (STOKEN) message body corresponding to a SECURE message which defines its operation. UXX USY is the secure acknowledgement (SECACK) message body. Security information, acknowledgements, and certificates can occur at any place in the dialogue, and not all messages need be secured. The standard EDIFACT certificate is represented by USC USA USA USA USR.

Note that in the above illustration it is assumed that the security services non-repudiation of origin and non-repudiation of receipt are jointly initialised as non-repudiation services.

C.1.4 Closure

In interactive EDI, a secure closure of a dialogue can be obtained by providing message sequence integrity in combination with a flag indicating the last (secured) message of the dialogue.

C.1.5 Error handling

In any interactive dialogue, there must be extensive error handling mechanisms. Error handling applies to all phases of a dialogue. The following cases illustrate failures in the security mechanisms (syntax and other errors may be handled by UIR):

- An inappropriate message was received (such as the responder not complying with the entity authentication protocol).
- The validation of a security result (such as a MAC or a digital signature) returned a failure condition.
- The recipient is unable to support the security services and mechanisms indicated in the SECURE message.
- The validation of a certificate returned a failure condition (e.g. revoked, expired, wrong signature, unknown type).

In secure interactive EDI, when an error is detected, the detecting party sends an error message SECACK to the other party, containing a notification of the type of error encountered. Upon transmission or receipt of an unrecoverable error message, both parties must close the dialogue.

C.2 Dialogue security services

This section explains the security services that apply to interactive EDI and discusses the recommended implementation.

C.2.1 Non-Repudiation of Origin (NRO)

Non-repudiation of origin protects the receiver of a message from the sender's denial of having sent that message.

Protection is typically achieved by including a digital signature with the transmitted message. The digital signature can be verified using the public key corresponding to the secret key which was used to create the digital signature. The public verification key must be available at the verifier; this may be achieved by sending the public key/certificate within the dialogue in a message or by distributing the public key/certificate beforehand using other distribution methods.

The following table illustrates the implementation of NRO within the interactive EDI security syntax for the case where the initiator wishes to use the security service during the dialogue (the reverse structure applies when the responder wishes to use NRO). After the UIB segment (or at some other point during the dialogue) the initiator sends the static security information SECURE to initialise the security service non-repudiation of origin. The responder must directly acknowledge the security service using a security response message SECRSP. During the dialogue, protected and unprotected messages may be mixed arbitrarily. The security header UXH contains the link to the corresponding SECURE message. The security trailer UXT contains the digital signature.

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	COMMENTS
Interchange Header: UIB	⇒		
	⇐	Interchange Header: UIB	
Static Security information: **SECURE**	⇒		The SECURE message is used to initialise NRO.
	⇐	Security Response **SECRSP**	The SECRSP message is used to accept NRO.
	.....		
Unprotected Message(s) UIH body UIT Secured Message(s) UIH **UXH** body **UXT** UIT	⇒		Secured and unprotected messages may be mixed during the dialogue. The security trailer contains the message signature.
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

Note that the digital signature provides not only non-repudiation of origin but also message content integrity and origin authentication.

C.2.2 Non-Repudiation of Receipt (NRR)

Non-repudiation of receipt protects the sender of a message within a dialogue from the receiver's denial of having received the message.

Protection is typically achieved by the receiver sending a secure acknowledgement which includes a digital signature on the data in the received message(s). The acknowledgement takes the form of a security message SECACK from the receiver to the sender. SECACK is able to provide NRR for one or more (or even all) messages of a dialogue. The acknowledged messages need not even be contained in the same dialogue. SECACK has the possibility to refer to messages of a different dialogue.

Non-repudiation of receipt (NRR) may be requested as follows:

- The sender of an interactive message requests the recipient to acknowledge, securely, the reception of the message. This is done by setting a flag in the security header of the message.
- The business partners may have agreed beforehand that certain types of messages always require a secure acknowledgement.

The following table illustrates the implementation of NRR within the interactive EDI security syntax for the case where the initiator requests the use of the security service NRR during the dialogue (the reverse structure applies when the responder requests NRR). To be able to request NRR, the initiator must already be using a security service himself (e.g. NRO). The request for NRR can then be transmitted in the security header UXH of a protected message. Hence, the first step is an exchange of SECURE-SECRSP regarding the initiator's security service. Then, given a message for which the initiator requests NRR, the responder replies with the static security information SECURE to initialise the security service of non-repudiation of receipt. The initiator directly acknowledges NRR using SECRSP and the responder sends the secure acknowledgement SECACK back.

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	COMMENTS
Interchange Header: UIB	⇒		
	⇐	Interchange Header: UIB	
Static Security information: **SECURE**	⇒		The SECURE message is used to initialise some security service (eg NRO).
	⇐	Security Response **SECRSP**	The SECRSP message is used to accept the initiator's security service..
	.....		
Secured Message(s) UIH **UXH** body **UXT** UIT	⇒		In the security header UXH, the sender may request the service of NRR from the recipient.
	⇐	Static Security information: **SECURE**	The SECURE message is used to initialise NRR.
Security Response **SECRSP**	⇒		The SECRSP message is used to accept NRR.
	⇐	Secure Acknowledgement **SECACK**	The receiver may provide NRR through SECACK for one or more previously received message(s).
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

Non-repudiation of receipt can either be provided if it was requested in the original message (which implies that the original message was a secured message), or under the rules of an Interchange Agreement (e.g. for specific types of messages). The responder collects the hash values of all initiator messages to be acknowledged and returns a signed SECACK message containing those hash values to the sender at the end of the dialogue (or before, if appropriate).

Note that non-repudiation of origin and non-repudiation of receipt can be jointly initialised, using one SECURE message, as non-repudiation services. NRR requires in fact the application of a digital signature (i.e. NRO) to the interactive EDI message SECACK. The use of a signed SECACK provides implicitly non-repudiation of receipt.

C.2.3 Confidentiality of Content

Confidentiality of content protects the dialogue parties against the unauthorised reading, copying or disclosure of messages within a dialogue.

Protection is typically achieved by encrypting the message(s) using a symmetric algorithm with a secret key established between initiator and responder for use during the dialogue. There are various methods for arriving at a shared secret dialogue key:

1. key agreement methods based on public key techniques (such as Diffie-Hellman),
2. key transport methods based on public key techniques (such as encrypting the secret key under the public encryption key of the recipient),
3. key establishment methods based on symmetric techniques (such as encrypting the secret key under a shared master key).

Interactive key establishment methods are closely related to entity authentication and are treated separately. For the following illustration we assume that the initiator uses method 2 above and already is in possession of the responder's public key.

The following table illustrates the implementation of confidentiality of content within the interactive EDI security syntax for the case where the initiator wishes to use the security service during the dialogue (the reverse structure applies when the responder wishes to use confidentiality of content). After the UIB (or at some other point during the dialogue) the initiator sends the static security information SECURE to initialise the security service. SECURE also contains the data key encrypted under the responder's public key. The responder directly acknowledges the service initialisation using a security response message SECRSP.

During the dialogue, the initiator may mix protected and unprotected messages arbitrarily. The security header UXH contains the link to the corresponding SECURE message.

INITIATOR INTERCHANGE		**RESPONDER INTERCHANGE**	**COMMENTS**
Interchange Header: UIB	⇒		
	⇐	Interchange Header: UIB	
Static Security information: **SECURE**	⇒		The SECURE message is used to initialise the confidentiality service. It contains the encrypted key.
	⇐	Security Response **SECRSP**	The responder uses the SECRSP message to accept the initiator's confidentiality service.
	.....		
Unprotected Message(s) UIH body UIT Secured Message(s) UIH **UXH** cipher **UXT** UIT	⇒		Secured and unprotected messages may be mixed during the dialogue. The message body is encrypted.
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

Note that if the responder also requires confidentiality on his interchange, he has to initialise the confidentiality service independently, using a second SECURE.

C.2.4 Origin Authentication

Origin authentication protects the receiver against masquerade of the sender (originator) and modification of a message within a dialogue.

Protection can be achieved by including with the transmitted message an authentication value (for example, a message authentication code MAC). The authentication value depends both on the message content and on a secret key in the possession of the sender.

This function includes message content integrity and may be obtained as a sub-product of non-repudiation of origin.

The implementation of origin authentication within the interactive EDI security syntax is identical to that of NRO. After the UIB (or at some other point during the dialogue) the relevant party sends the static security information SECURE to initialise the security service origin authentication. The responder directly acknowledges the service initialisation using a security response message SECRSP. During the dialogue, protected and unprotected messages may be mixed arbitrarily. The security header UXH of the protected messages contains the link to the corresponding SECURE message. The security trailer UXT contains the authentication value.

Note that an authenticated acknowledgement can be obtained by applying origin authentication to the secure acknowledgement message SECACK.

C.2.5 Content Integrity

Content integrity protects against the modification of data. It is typically obtained as a sub-product of message origin authentication or non-repudiation of origin

Protection can be achieved by including an integrity control value with the message. This value can be computed by using an appropriate cryptographic algorithm, such as an MDC (Modification Detection Code). As this control value in itself is unprotected, additional measures, such as forwarding the control value by a separate channel or calculating a digital signature, to provide non-repudiation of origin, on the control value are necessary. Alternatively, message authentication, which is obtained using a MAC, will imply message integrity. The receiver of the message computes the integrity control value of the data received using the corresponding algorithms and parameters and compares the result with the value received.

The implementation of content integrity within the interactive EDI security syntax is identical to that of NRO and message origin authentication. Immediately after the UIB (or at some other point during the dialogue) the relevant party sends the static security information SECURE to initialise the security service content integrity. The responder directly acknowledges the service initialisation using a security response message SECRSP. During the dialogue, protected and unprotected messages may be mixed arbitrarily. The security header UXH of the protected messages contains the link to the corresponding SECURE message. The security trailer UXT contains the integrity control value. However content integrity is rarely used by itself as it requires additional security measures. In most cases it would be desirable to have origin authentication, at least.

C.2.6 Sequence Integrity

Sequence integrity protects against the duplication, addition, deletion, loss or replay of a message within a dialogue. Sequence integrity is not an independent security service; it is obtained as a sub-product of any of the integrity services.

To detect lost messages

- the sender can include and the receiver check a message sequence number (related to the message flow between the two parties concerned);
- the sender can request and check an acknowledgement.

To detect added or duplicated messages

- the sender can include and the receiver check a message sequence number.
- the sender can include and the receiver check a time stamp.

When sequence numbers are used it must be agreed how these are to be managed. The timestamp will normally be produced by the sender's system. This implies, as in the paper world, that the initial accuracy of the value of the timestamp is solely under the control of the sender. In order to give full protection, the integrity of timestamp or sequence number must be guaranteed by additional security measures such as origin authentication or non-repudiation of origin.

The implementation of sequence integrity within the interactive EDI security syntax can be obtained as a sub-product of any of the security services origin authentication, or non-repudiation of origin (or possibly content integrity). A sequence number (or time-stamp) is included in the security header UXH of the associated security service. The security result in the security trailer protects the message body and the security header including the sequence number (or time stamp).

To be able to provide dialogue completeness, that is, to detect any lost messages at the end of the dialogue, a last message flag can be included in the security header UXH of the last secured message.

C.2.7 Entity authentication

Entity authentication protects (a) one party unilaterally, or (b) both parties mutually, against false identification (i.e. masquerade) of the other party. Entity authentication may be defined as "*The corroboration that an entity is the one claimed*"(ISO/IEC 7498-2).

Protection is achieved by executing an authentication handshake (typically at the beginning of the dialogue). Distinguishing features for entity authentication protocols are:

- unilateral or mutual authentication, i.e. is only one party or are both parties authenticated during the protocol
- number of passes, i.e. how many protocol exchanges are needed until the desired entity authentication is achieved.
- symmetric or asymmetric technique, i.e. the use of secret-key or public-key cryptography.
- the time-varying parameter, such as the use of random numbers, timestamps or sequence numbers as time-varying parameter.
- the underlying mechanism, such as an encryption algorithm, a message authentication code (MAC), a signature system.
- involvement of third parties, i.e. in establishing the entity authentication
- key management, i.e. it is possible to establish a shared secret key between the parties for further use in the session.

Entity authentication requires both parties to agree on a common protocol and commits the parties to follow that protocol. The party to initialise the service is not necessarily the party to execute the first step of the handshake. Therefore, it must be possible for one party to request entity authentication initiated by the other party. In secure interactive EDI, entity authentication is accomplished as follows

1. The requesting party initiates a specific entity authentication mechanism by sending the corresponding initialisation message SECURE. The responding party accepts it by sending a positive SECRSP.
2. The entity authentication mechanism is then started by the party indicated as the initiating party of the mechanism. Note that it is not always the requesting party which is the party starting the mechanism (i.e. doing the first pass of the mechanism).

If the responding party cannot comply with the requested mechanism, an error condition must be signalled. This is done by a negative SECRSP. Those entity authentication mechanisms supported in a business context are typically prescribed in an Interchange Agreement.

In the following subsections the implementation of entity authentication within the interactive EDI security syntax is discussed. For one, two, and three pass mechanisms, the message flow is explicitly illustrated. In a dialogue of two parties there is no need for more than three passes to obtain mutual authentication.

C.2.7.1 One-pass entity authentication

The following table illustrates the implementation of one-pass entity authentication within the interactive EDI security syntax for the case where the responder requests authentication of the initiator. After the initial exchange of the UIB segments the responder sends the static security information SECURE to request one-pass entity authentication. The initiator acknowledges the chosen mechanism by replying with a positive SECRSP, followed by a message STOKEN which contains the authentication information to be checked by the responder. Optionally, a KEYCER message may be used to convey the initiator's certificate directly to the responder. The actual dialogue can then begin. If the responder is unable to validate the authentication information successfully, they may then respond with a SECACK message indicating an error.

INITIATOR		RESPONDER	COMMENTS

INTERCHANGE		INTERCHANGE	
Interchange Header: UIB	⇒		
	⇐F	Interchange Header: UIB	
	⇐L	Static Security information: **SECURE**	The SECURE message is used to initialise entity authentication, requesting that the initiator is to be authenticated.
Security Response **SECRSP**	F⇒		The initiator accepts the requested entity authentication, using the SECRSP message.
Secure Token Message: **STOKEN**	I⇒		The STOKEN message conveys the authentication information.
Key Management Message: **KEYCER**	I⇒		optional: to transfer a public key certificate within the dialogue.
Messages(s) UIH body UIT	L⇒		After validation of STOKEN the initiator begins the actual dialogue.
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

The notation F, I, L *in the second column indicates the use of the UIH transfer position (0323) to define the first, intermediate and last messages in a transfer.*

Note that for clarity the above outline does not show any subsequent security services from the responder, although this would typically be the case.

C.2.7.2 Two-pass entity authentication

The following table illustrates the implementation of two-pass entity authentication within the interactive EDI security syntax. Two-pass entity authentication can either be unilateral or mutual. The initiator sends the static security information SECURE to initialise two-pass entity authentication. The responder acknowledges the chosen mechanism using the security response SECRSP. Then the initiator sends the STOKEN which contains the authentication information according to the specified two-pass mechanism. The responder processes the received STOKEN and responds with the corresponding STOKEN, as required by the entity authentication mechanism. The initiator receives the STOKEN, processes its contents and starts the dialogue upon successful validation.

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	COMMENTS
Interchange Header: UIB	F⇒		
Static Security information: **SECURE**	L⇒		The SECURE message is used to initialise entity authentication.
	⇐F	Interchange Header: UIB	
	⇐L	Security Response **SECRSP**	The SECRSP message to used to accept entity authentication.
Secure Token Message: **STOKEN**	⇒		The STOKEN message conveys the authentication information of the first pass.
	⇐	Secure Token Message: **STOKEN**	The responder replies with the second authentication message of the two-pass protocol.
Messages(s) UIH body UIT	⇒		The initiator starts the actual dialogue upon successful validation of the STOKEN.
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

The notation F, L *in the second column indicates the use of the UIH transfer position (0323) to define the first, intermediate and last messages in a transfer.*

Optionally, a KEYCER message may be used to directly convey a public key certificate to the other party.

Note that the initiator should wait for the responder to complete successfully the entity authentication protocol before entering the actual dialogue, that is, before starting to send application messages.

C.2.7.3 Three-pass entity authentication

The following table illustrates the implementation of three-pass entity authentication within the interactive EDI security syntax. Three-pass entity authentication is mutual.

The initiator sends the static security information SECURE to initialise three-pass entity authentication. The responder acknowledges the chosen mechanism using the security response SECRSP. Then the initiator sends the STOKEN which contains the authentication information to be transferred in the first pass of the specified three-pass mechanism. The responder processes the received STOKEN and responds with the corresponding STOKEN containing the authentication information to be transferred in the second pass of the mechanism. The initiator processes the contents of the second STOKEN and replies with the third STOKEN, in the form required by the entity authentication mechanism. The responder receives the STOKEN, and processes its contents. The actual dialogue can then begin. If the responder is unable to validate the authentication information successfully, they may then respond with a SECACK message indicating an error.

INITIATOR INTERCHANGE		RESPONDER INTERCHANGE	COMMENTS
Interchange Header: UIB	F⇒		
Static Security information: **SECURE**	L⇒		The SECURE message is used to initialise entity authentication.
	⇐F	Interchange Header: UIB	
	⇐L	Security Response **SECRSP**	The SECRSP message is used to accept entity authentication.
Secure Token Message: **STOKEN**	⇒		The STOKEN message conveys the authentication information.
	⇐	Secure Token Message: **STOKEN**	The responder replies with the second authentication message of the three-pass protocol.
Secure Token Message: **STOKEN**	F⇒		The initiator replies with the third authentication message of the three-pass protocol.
Messages(s) UIH body UIT	L⇒		After successful completion of the protocol the initiator begins the actual dialogue.
	.....		
Interchange Trailer: UIZ	⇒		
	⇐	Interchange Trailer: UIZ	

The notation F, L *in the second column indicates the use of the UIH transfer position (0323) to define the first, intermediate and last messages in a transfer.*

Optionally, a KEYCER message may be used to directly convey a public key certificate to the other party.

Note that A should wait for B to complete successfully the entity authentication protocol before it enters the actual dialogue, that is, before it starts sending application messages.

C.2.8 Dialogue key establishment

Establish a shared secret key for use within the dialogue. Key establishment is a supportive service and is typically integrated with the entity authentication service or with the respective message security services (confidentiality, origin authentication).

The objective is to establish one or more shared secret key(s) for the duration of the interactive dialogue. These shared keys are typically used for the provision of dialogue-oriented security services (such as confidentiality or origin authentication). There are various methods for arriving at a shared secret dialogue key:

- key agreement methods based on public key techniques (such as Diffie-Hellman),
- key transport methods based on public key techniques (such as encrypting the secret key under the public encryption key of the recipient),
- key establishment methods based on symmetric techniques (such as encrypting the secret key under a shared master key).

The interactive EDI security syntax provides the following facilities to establish a dialogue key:

1. Sending an encrypted dialogue key during the initialisation (e.g. in SECURE) of the confidentiality or the origin authentication service. This key will apply to the corresponding interchange of the initiating party.
2. Executing an integrated entity authentication and key establishment mechanism. The resulting key is available for general use during the dialogue.

Case 1 above is included in the service definition for confidentiality and origin authentication (see C.2.3 for an illustration). Case 2 above is included in the service definition for entity authentication (see C.2.7).

C.2.9 Access control

Access control protects a resource from illegitimate access.

Access control typically consists of two phases: firstly establishing the identity of the party who wants to access a resource, and secondly, authorising the access. The first phase is provided for by entity authentication. The second phase, authorisation, consists of transmitting a request which contains the access control information (privilege attributes) to the target resource. The access control information must be cryptographically protected. The recipient (i.e. the target) then validates the access control information and makes a positive (access granted) or negative (access denied) decision. Thus, the authorisation phase is conceptually very similar to one-pass entity authentication.

There are no explicit elements defined to provide access control as a security service. The reasoning being that access control is considered an application matter and outside the scope of the interactive EDI security syntax. However, as explained above, the STOKEN message could easily be used to provide for authorisation (e.g. same structure as one-pass entity authentication).

C.2.10 Key management

Key management is concerned with the management of long-term keys and certificates. It includes functions such as registration, key generation, certification, validation, revocation, discontinuation, and key status notification.

The objective is to manage long-term keys and certificates. There are various application requirements in interactive EDI:

- As a supportive service to the other security services: most typically it will be required that one party makes available their certificate (certificate chain) to the other party in an efficient way, in order to enable the other party to carry out a security service.
- As a service of its own: it will be required that a party connects to the public key infrastructure (or to a key management centre) in order to register, certify, distribute, validate and revoke keys/certificates.

The interactive EDI security syntax provides the message KEYCER to allow for key management. KEYCER may or may not be protected by a message security service. For instance, to transfer a party's certificate at the beginning of a dialogue does not require the additional protection of KEYCER.

Annex D
(informative)
Addendum - to be added to Part 1 annex D when approved

Syntax service code directory

The tags of all simple data elements contained in the simple data element directory start with the figure “0”.

```
-----------------------------------------------------------------------------

    0553  FINAL SECURED MESSAGE, CODED

    Desc: Identification of the final secured message in a dialogue.

    Repr: an..3

        1 Final secured message
             Indication of the final secured message in a dialogue (where
             known), when using message sequence integrity.

-----------------------------------------------------------------------------

    0581  METHOD OF TOKEN CALCULATION, CODED

    Desc: Identification of data representation used in calculating a security
          token.

    Repr: an..3

        1 Including separators
             Based on the EDIFACT representation of the data items, including
             separators.
        2 Excluding separators
             Based on the EDIFACT representation of the data items, excluding
             separators.
      ZZZ Mutually agreed
              Self explanatory.

-----------------------------------------------------------------------------

    0593  SECURITY RESPONSE, CODED

    Desc: Response to the proposal of a security mechanism.

    Repr: an..3

        1 Security details supported
             All of the proposed security details are supported.
        2 Security details not supported
             Some or all of the proposed security details are not supported.

-----------------------------------------------------------------------------
```

0597 TOKEN MECHANISM, CODED

Desc: Identification of authentication and key establishment standards and types.

Repr: an..3

101 ISO 9798-2 symmetric encipherment unilateral one pass
102 ISO 9798-2 symmetric encipherment unilateral two pass
103 ISO 9798-2 symmetric encipherment mutual two pass
104 ISO 9798-2 symmetric encipherment mutual three pass

121 ISO 9798-3 public key algorithm unilateral one pass
122 ISO 9798-3 public key algorithm unilateral two pass
123 ISO 9798-3 public key algorithm mutual two pass
124 ISO 9798-3 public key algorithm mutual three pass
125 ISO 9798-3 public key algorithm mutual two pass parallel

141 ISO 9798-4 cryptographic check function unilateral one pass
142 ISO 9798-4 cryptographic check function unilateral two pass
143 ISO 9798-4 cryptographic check function mutual two pass
144 ISO 9798-4 cryptographic check function mutual three pass

201 ISO 11770-3 key agreement mechanism 1
202 ISO 11770-3 key agreement mechanism 2
203 ISO 11770-3 key agreement mechanism 3
204 ISO 11770-3 key agreement mechanism 4
205 ISO 11770-3 key agreement mechanism 5
206 ISO 11770-3 key agreement mechanism 6
207 ISO 11770-3 key agreement mechanism 7
208 ISO 11770-3 key transport mechanism 1
209 ISO 11770-3 key transport mechanism 2
210 ISO 11770-3 key transport mechanism 3
211 ISO 11770-3 key transport mechanism 4
212 ISO 11770-3 key transport mechanism 5
213 ISO 11770-3 key transport mechanism 6

0599 TOKEN MECHANISM CODE LIST IDENTIFIER

Desc: Identification of the code list used to identify the token mechanism.

Repr: an..3

1 UN
United Nations.