Joint ISO/TC 154 – UN/CEFACT Syntax Working Group (JSWG) publication of ISO 9735-5

equivalent to the official ISO publication: ISO 9735-5 **(First edition 1999-04-01)**

Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) —

Part 5:

Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)

Contents

Page

1 Scope 1

2 Conformance 1

3 Normative references 1

4 Definitions 2

5 Rules for the use of security header and trailer segment groups for batch EDI 2

6 Rules for the use of interchange and group security header and trailer segment groups for batch EDI 9

Annex A: Definitions 13

Annex B: Syntax service directories (segments, composite data elements and simple data elements) 14

Annex C: EDIFACT security threats and solutions 29

Annex D: How to protect an EDIFACT structure 32

Annex E: Message protection examples 35

Annex F: Filter functions for UN/EDIFACT character set repertoires A and C 42

Annex G: Service code directory 44

Annex H: Security services and algorithms 45

Annex I: Bibliography 51

Foreword

This part of ISO 9735 was prepared by the UN/ECE Trade Division (as UN/EDIFACT) and was adopted, under a special "fast-track procedure", by Technical Committee ISO/TC 154, *Documents and data elements in administration, commerce and industry.*

Whereas this part supersedes the earlier publications, and shall use a version number of “4” in the mandatory data element 0002 (Syntax version number) in the segment UNB (Interchange header), interchanges continuing to use the syntax defined in the earlier published versions shall use the following Syntax version numbers, in order to differentiate them from each other and from this part:

ISO 9735:1988 — *Syntax version number: 1*

ISO 9735:1988 (amended and reprinted in 1990) — *Syntax version number: 2*

ISO 9735:1988 (amended and reprinted in 1990) plus Amendment 1:1992 — *Syntax version number: 3*

ISO 9735 consists of the following parts, under the general title *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4)*:

— *Part 1: Syntax rules common to all parts, together with the syntax service directories for each of the parts*

— *Part 2: Syntax rules specific to batch EDI*

— *Part 3: Syntax rules specific to interactive EDI*

— *Part 4: Syntax and service report message for batch EDI (message type - CONTRL)*

— *Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)*

— *Part 6: Secure authentication and acknowledgement message (message type - AUTACK)*

— *Part 7: Security rules for batch EDI (confidentiality)*

— *Part 8: Associated data in EDI*

— *Part 9: Security key and certificate management message (message type - KEYMAN)*

Further parts may be added in the future.

Annexes A and B form an integral part of this part of 9735. Annexes C to I are for information only.

Introduction

This part of ISO 9735 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.

Communications specifications and protocols are outside the scope of this part of ISO 9735.

This is a new part, which has been added to ISO 9735. It provides an optional capability of securing batch EDIFACT structures i.e. messages, packages, groups or interchange.

Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules — (Syntax version number: 4)

Part 5:
Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)

1 Scope

This part of ISO 9735 specifies syntax rules for EDIFACT security. It provides a method to address message/package level, group level and interchange level security for authenticity, integrity and non-repudiation of origin, 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 conform to the syntax rules specified in this part of ISO 9735.

Devices supporting this part of ISO 9735 are in conformance when they are capable of creating and/or interpreting the data structured and represented in conformance with this part of ISO 9735.

Conformance to part of ISO 9735 shall include conformance to Part 1, Part 2 and Part 8 of ISO 9735.

When identified in this part of ISO 9735, provisions defined in related standards shall form part of the conformance criteria.

3 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 9735. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 9735 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security architecture.*

ISO/IEC 9594-8:1995, *Information technology — Open Systems Interconnection — The Directory: Authentication framework.*

ISO 9735-1:1998, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) — Part 1: Syntax rules common to all parts, together with syntax directories for each of the parts.*

ISO 9735-2:1998, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) — Part 2: Syntax rules specific to batch EDI.*

ISO 9735-6:1998, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) — Part 6: Secure authentication and acknowledgement message (message type — AUTACK).*

ISO 9735-7:—[1)], *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) — Part 7: Security rules for batch EDI (confidentiality).*

ISO 9735-8:1998, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) — Part 8: Associated data in EDI.*

ISO/IEC 10181-2:1996, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Authentication framework.*

ISO/IEC 10181-4:1997, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Non-repudiation framework.*

ISO/IEC 10181-6:1996, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Integrity framework.*

4 Definitions

For the purposes of this part of ISO 9735, the definitions in ISO 9735-1:1998, annex A apply.

5 Rules for the use of security header and trailer segment groups for batch EDI

5.1 Message/package level security - integrated message/package security

The security threats relevant to message/package transmission and the security services which address them are described in annexes C and D.

This section describes the structure of EDIFACT message/package level security.

Security services addressed in this part of ISO 9735 shall be provided by the inclusion of security header and trailer segment groups after the UNH and before the UNT, in a way which shall be applied to any existing message, or after the UNO and before the UNP, for any existing package.

1) To be published.

5.1.1 Security header and trailer segment groups

Figure 1 describes an interchange showing security at message level.

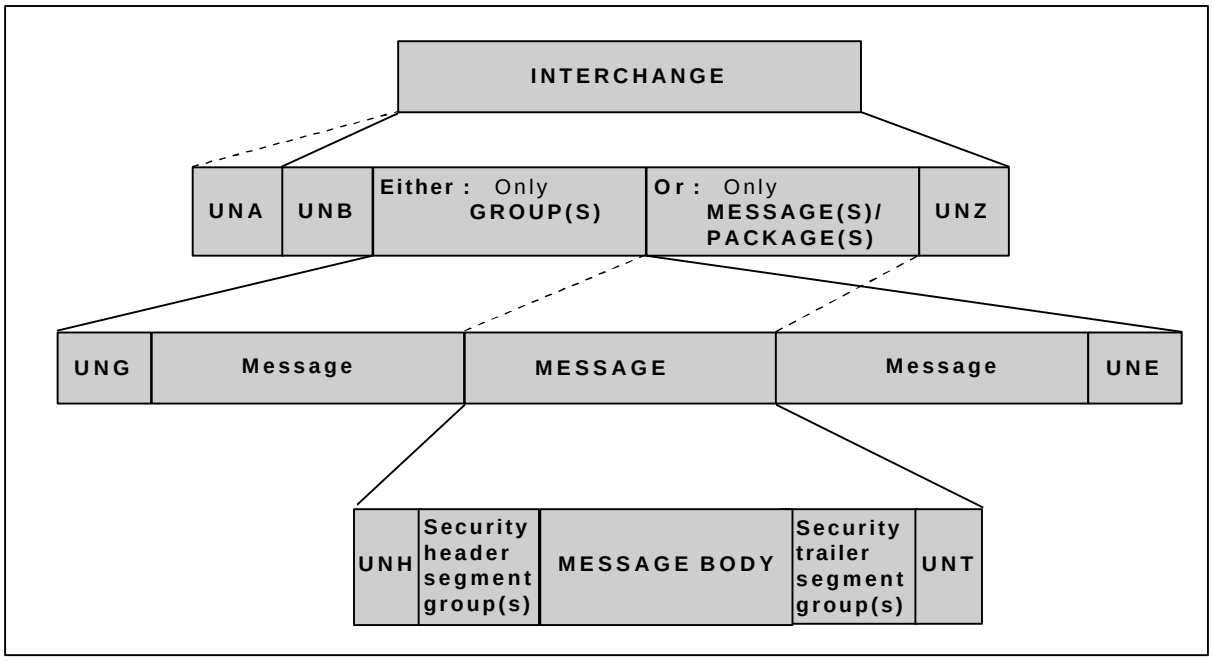


Figure 1 - Interchange showing security at message level (schematic)

Figure 2 describes an interchange showing security at package level.

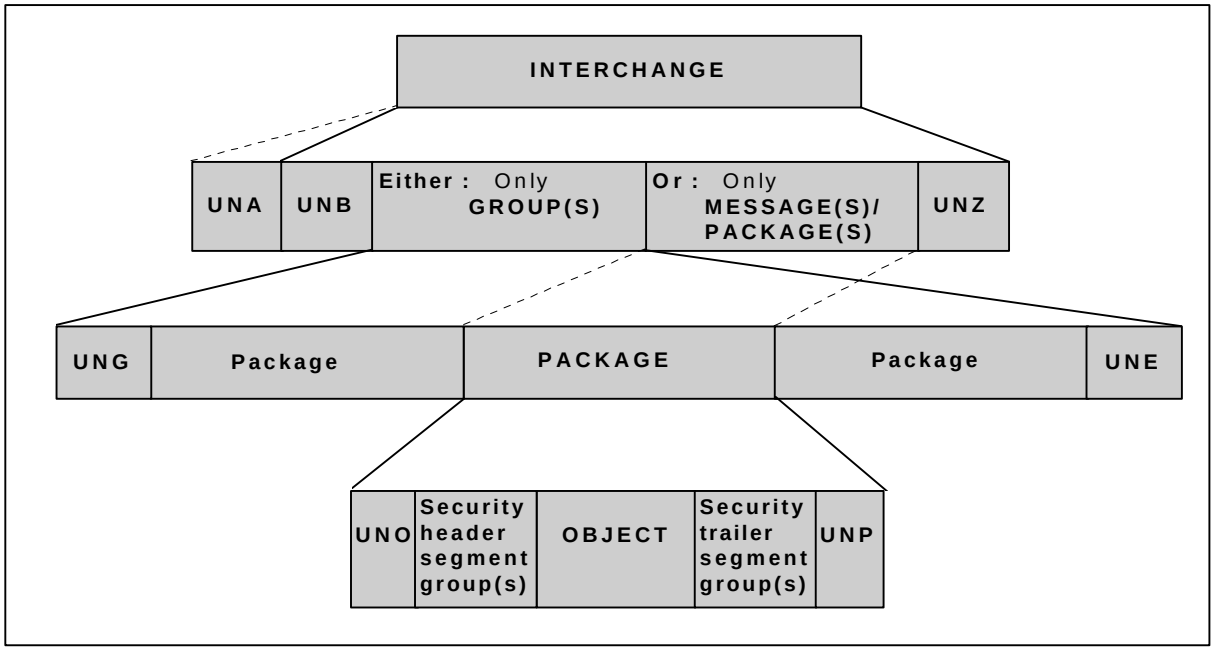
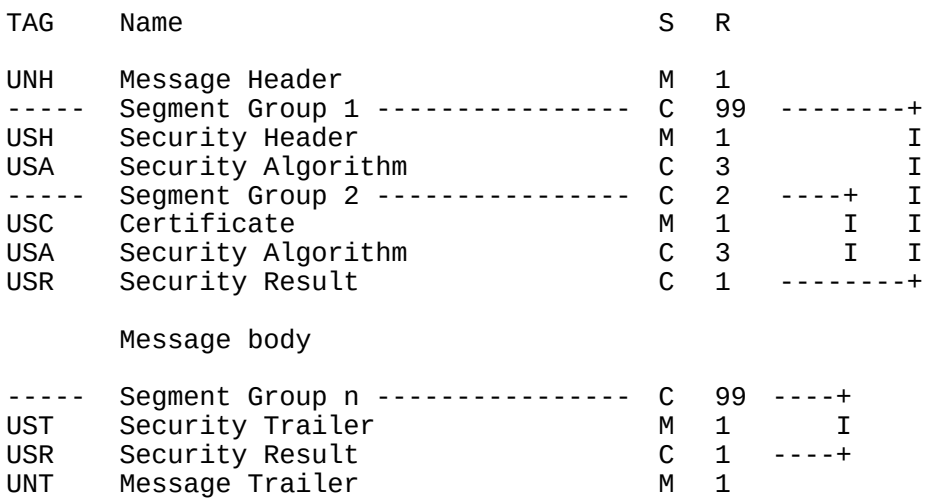


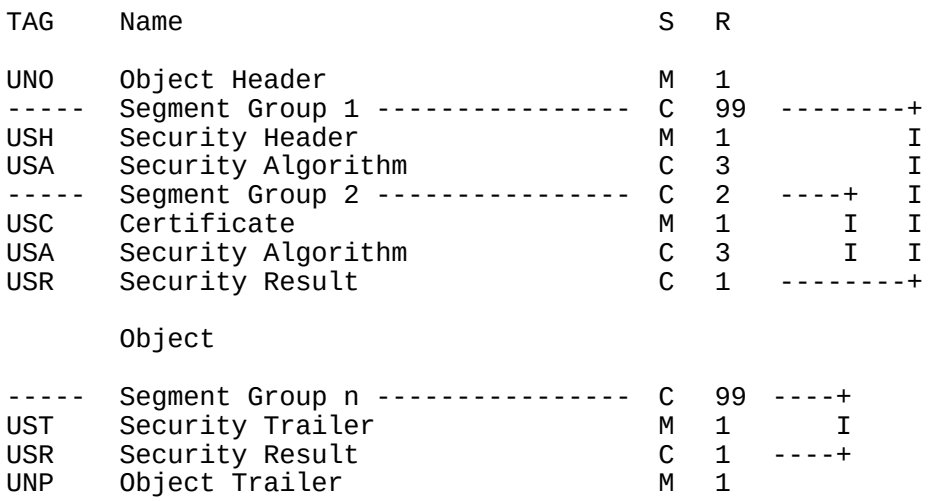
Figure 2 - Interchange showing security at package level (schematic)

5.1.2 Security header and trailer segment groups structure



TAG	Name	S	R	
UNH	Message Header	M	1	
-----	Segment Group 1 ----------	C	99	--------+
USH	Security Header	M	1	I
USA	Security Algorithm	C	3	I
-----	Segment Group 2 ----------	C	2	----+ I
USC	Certificate	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	--------+
	Message body			
-----	Segment Group n ----------	C	99	----+
UST	Security Trailer	M	1	I
USR	Security Result	C	1	----+
UNT	Message Trailer	M	1	

Table 1 - Security header and security trailer segment groups segment table (message level security)



TAG	Name	S	R	
UNO	Object Header	M	1	
-----	Segment Group 1 ----------	C	99	--------+
USH	Security Header	M	1	I
USA	Security Algorithm	C	3	I
-----	Segment Group 2 ----------	C	2	----+ I
USC	Certificate	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	--------+
	Object			
-----	Segment Group n ----------	C	99	----+
UST	Security Trailer	M	1	I
USR	Security Result	C	1	----+
UNP	Object Trailer	M	1	

Table 2 - Security header and security trailer segment groups segment table (package level security)

Note: UNH message header, UNT message trailer, UNO object header and UNP object trailer are specified in Part 1 of ISO 9735. They are not described further in this Part.

The complete directory specification of the segments and data elements may be found in annex B.

5.1.3 Data segment clarification

Segment Group 1: USH-USA-SG2 (security header group)
A group of segments identifying the security service and security mechanisms applied and containing the data necessary to carry out the validation calculations.
There may be several different security header segment groups within the same message/package, if different security services are applied to the message/package (e. g. integrity and non-repudiation of origin) or if the same security service is applied by several parties.

USH, Security header
A segment specifying a security service applied to the message/package in which the segment is included.

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 (USC segment) when asymmetric algorithms are used.
Security identification details composite data element (S500) shall be used in USH 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 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 S500 present in the USC segment in segment group 2, 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 of segment group 1.

S500/0538 in USH segment may be used if there is no need to convey a USA segment in segment group 1 (because the cryptographic mechanisms have been agreed previously between the partners).

Nevertheless, it is strongly recommended to use either S500/0538 in the USH segment, or S503/0554 with the appropriate qualifier in the USA segment, but not both of them, within the same security header group.

USH segment may specify the filter function used for the binary fields of USA segment within segment group 1 and of the USR segment of the corresponding security trailer group.

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

USA, Security algorithm
A segment identifying a security algorithm, the technical usage made of it, and containing the technical parameters required. This shall be the algorithm applied directly on the message/package. This algorithm may be either symmetric, a hash function or a compression algorithm. For example, for a digital signature, it indicates the message-dependent hash function to be used.
Asymmetric algorithms shall not be referred to directly in this USA segment within segment group 1 but may appear only within segment group 2, triggered by a USC segment.
Three occurrences of the USA segment are allowed. One occurrence shall be used for the symmetric algorithm or the hash function required to provide the security service specified in the USH segment. The other two occurrences are described in Part 7 of ISO 9735.
Indication of padding mechanism may be used when appropriate.

Segment Group 2: USC-USA-USR (certificate group)
A group of segments containing the data necessary to validate the security methods applied to the message/package, when asymmetric algorithms are used. Certificate segment group shall be used when asymmetric algorithms are used to identify the asymmetric key pair used, even if certificates are not used.

Either the full certificate segment group (including the USR segment), or the only data elements necessary to identify unambiguously the asymmetric key pair used, shall be present in the USC segment. The presence of a full certificate may be avoided if the certificate has already been exchanged by the two parties, or if it may be retrieved from a database.

Where it is decided to refer to a non-EDIFACT certificate (such as X.509), the certificate syntax and version shall be identified in data element 0545 of the USC segment. Such certificates may be conveyed in an EDIFACT package.

Two occurrences of this segment group are allowed, one being the message/package sender certificate (that the message/package receiver will use to verify the sender's signature), the other being the message/package receiver certificate (only referred to by certificate reference) in the case where the receiver public key is used by the sender for confidentiality of symmetric keys.

If both are present within one security header segment group, the security identification details composite data element (S500) together with the certificate reference data element (0536) allow them to be differentiated.

This segment group shall be omitted if no asymmetric algorithm is used.

USC, Certificate
A segment containing the credentials of the certificate owner and identifying the certification authority which has generated the certificate. The data element filter function, coded (0505) shall identify the filter function used for the binary fields of USA segments and USR segment within segment group 2.
USC certificate may contain two occurrences of 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 or CA).

USA, Security algorithm
A segment identifying a security algorithm, the technical usage made of it, and containing the technical parameters required. The three different occurrences of this USA segment in segment group 2 are identifying:

1 the algorithm used by the certificate issuer to compute the hash value of the certificate (hashing function)
2 the algorithm used by the certificate issuer to generate the certificate (i.e. to sign the result of the hash function computed on the certificate content) (asymmetric algorithm)
3a - either the algorithm used by the sender to sign the message/package (i.e. to sign the result of the hash function described in the USH segment, computed on the message/package content) (asymmetric algorithm),
3b - or the receiver's asymmetric algorithm used by the sender to encrypt the key required by a symmetric algorithm applied to the message/package content and referred to by the segment group 1 triggered by the USH segment (asymmetric algorithm)

Indication of padding mechanism may be used when appropriate.

USR, Security result
A segment containing the result of the security functions applied to the certificate by the certification authority. This result shall be the signature of the certificate computed by the certification authority by signing the hash result computed on the data of the credentials.
For the certificate, the signature computation starts with the first character of the USC segment (namely the "U") and ends with the last character of the last USA segment (including the separator following this USA segment).

Segment Group n: UST-USR (security trailer group)
A group of segments containing a link with security header segment group and the result of the security functions applied to the message/package.

UST, Security trailer
A segment establishing a link between security header and security trailer segment groups, and stating the number of security segments containedin these groups.

USR, Security result
A segment containing the result of the security functions applied to the message/package as specified in the linked security header group. Depending on the security mechanisms specified in the linked security header group, this result shall be either:
- computed directly on the message/package by the algorithm specified in the USA segment within segment group 1 of the security header group, or
- computed by signing with an asymmetric algorithm specified in USA segment within segment group 2 of the security header segment group a hash result computed on the message/package by the algorithm specified in the USA segment within segment group 1 of the security header segment group

5.1.4 Scope of security application

There are two possibilities for the scope of security application:

1. The computation of each of the integrity and authentication values and of the digital signatures starts with and includes the current security header segment group and the message body, or object, itself. In this case no other security header or security trailer segment groups shall be encompassed within this scope.

   The security header segment group shall be counted from the first character, namely a "U", to the separator ending this security header segment group, both included, and the message body, or object, from the first character following the separator ending the last security header segment group to the separator preceding the first character of the first security trailer segment group, both included.

   Thus the order in which security services integrated in this manner are performed, is not prescribed. They are completely independent of each other.

   Figure 3 illustrates this case (the scope of application of the security service defined in the security header 2 is represented by shaded boxes):

UNH/ *UNO*	Security header segment group 3	Security header segment group 2	Security header segment group 1	MESSAGE BODY/ *OBJECT*	Security trailer segment group 1	Security trailer segment group 2	Security trailer segment group 3	UNT/ *UNP*

Figure 3 - Scope of application: security header segment group and message body/object only (schematic)

2. The computation starts with and includes the current security header segment group to the associated security trailer segment group. In this case the current security header segment group, the message body, or object, and all the other embedded security header and trailer segment groups shall be encompassed within this scope.

   The scope shall include every character from the first character, namely a "U", of the current security header segment group, to the separator preceding the first character of the associated security trailer segment group, both included.

   Figure 4 illustrates this case (the scope of application of the security service defined in the security header 2 is represented by shaded boxes):

UNH/ *UNO*	Security header segment group 3	Security header segment group 2	Security header segment group 1	MESSAGE BODY/ *OBJECT*	Security trailer segment group 1	Security trailer segment group 2	Security trailer segment group 3	UNT/ *UNP*

Figure 4 - Scope of application: from security header segment group to security trailer segment group (schematic)

For each added security service, either of the two approaches may be chosen.

In both cases, the relation between the security header segment group and associated security trailer segment group shall be provided by the data elements security reference number of the USH and of the UST segments.

5.2 Principles of usage

5.2.1 Choice of service

The security header segment group may include the following general information:

- Security service applied
- Identification of the parties involved
- Security mechanism used
- "Unique" value (sequence number and/or timestamp)
- Non-repudiation of receipt request

If more than one security service is required for the same EDIFACT structure, then the security header segment group may be present several times. This shall be the case when several pairs of parties are involved. However, if several services are required between the same two parties they may be included in a single pair of security header and trailer segment groups, as certain services include others implicitly.

5.2.2 Authenticity

If origin authentication of a EDIFACT structure is required, it shall be provided in accordance to the principles defined in ISO 10181-2, using an appropriate pair of security header and security trailer segment groups.

The security service of origin authentication shall be specified in the USH segment and the algorithm shall be identified in the USA segment in segment group 1. It shall be a symmetric algorithm.
The party acting as security originator shall compute an authenticity value that shall be conveyed in the USR segment of the security trailer segment group. The party acting as security recipient shall check the authenticity value.

This service may include integrity service and may be obtained as a sub-product of non-repudiation of origin service.

If an appropriate implementation of this "origin authentication" service, based on tamper resistant hardware or trusted third parties, is used, it may be considered as an instance of "non repudiation of origin" service. Such a practice shall be defined in the interchange agreement.

5.2.3 Integrity

If content integrity of a EDIFACT structure is required, it shall be provided in accordance to the principles defined in ISO 10181-6, using an appropriate pair of security header and security trailer segment groups.

The security service of integrity shall be specified in the USH segment and the algorithm shall be identified in the USA segment in segment group 1. It shall be hash function or a symmetric algorithm.

The party acting as security originator shall compute an integrity value that shall be conveyed in the USR segment of the security trailer segment group. The party acting as security recipient shall check the integrity value.

This service may be obtained as a sub-product of origin authentication service or of non-repudiation of origin service.

If sequence integrity is required, either a security sequence number or a security timestamp, or both, shall be contained by the security header segment group and either content integrity service or origin authentication service or non-repudiation of origin service shall be used.

5.2.4 Non-repudiation of origin

If non-repudiation of origin of a EDIFACT structure is required, it shall be provided in accordance to the principles defined in ISO 10181-4, using an appropriate pair of security header and security trailer segment groups.

The security service of non-repudiation of origin shall be specified in the USH segment and the hashing algorithm shall be identified in the USA segment in segment group 1, and the asymmetric algorithm used for signature in the USA segments of segment group 2, if certificates are used.

If the certificate is not conveyed in the message/package, the asymmetric algorithm shall be implicitly known by the receiving party. In this case the asymmetric algorithm shall be defined in the interchange agreement.

The party acting as security originator shall compute a digital signature that shall be conveyed in the USR segment of the security trailer segment group. The party acting as security recipient shall verify the digital signature value.

This service provides also content integrity and origin authentication services.

5.3 Internal representation and filters for compliance with EDIFACT syntax

The use of mathematical algorithms to compute integrity values and digital signatures introduces two problems.

The first problem is that the result of the calculation depends on the internal representation of the character set. Thus the computation of the digital signature by the sender and its verification by the recipient shall be executed using the same character set encoding. Therefore the sender may indicate the representation used to produce the original security validation result.

The second problem is that the result of the calculation is a seemingly random bit pattern. This may cause problems during transmission and with interpretation software. To avoid these problems the bit pattern may be reversibly mapped on to a particular representation of the character set used by means of a filtering function. For simplicity, only one filtering function shall be used for each security service. Any appearance of an anomalous terminator in the output of this mapping is dealt with by including an escape sequence.

6 Rules for the use of interchange and group security header and trailer segment groups for batch EDI

6.1 Group and interchange level security - integrated message security

The security threats relevant to message/package transmission and the security services which address them, as described in annexes C and D, are also valid at group and interchange level.

The techniques described in the previous section for applying security to messages/packages may also be applied to interchanges and groups.

For group and interchange level security, the same header and trailer segment groups as those described at message/package level, shall be used, and header-trailer cross referencing shall always apply at the same level, even when security is applied separately at more than one level.

When security is applied at message/package level, the protected structure is the message body or object. At group level it is the set of messages/packages in the group including all message/package headers and trailers. At interchange level, it is the set of messages/packages or groups in the interchange, including all message/package or group headers and trailers.

6.1.1 Security header and trailer segment groups

Figure 5 describes an interchange showing security at both interchange and group levels.

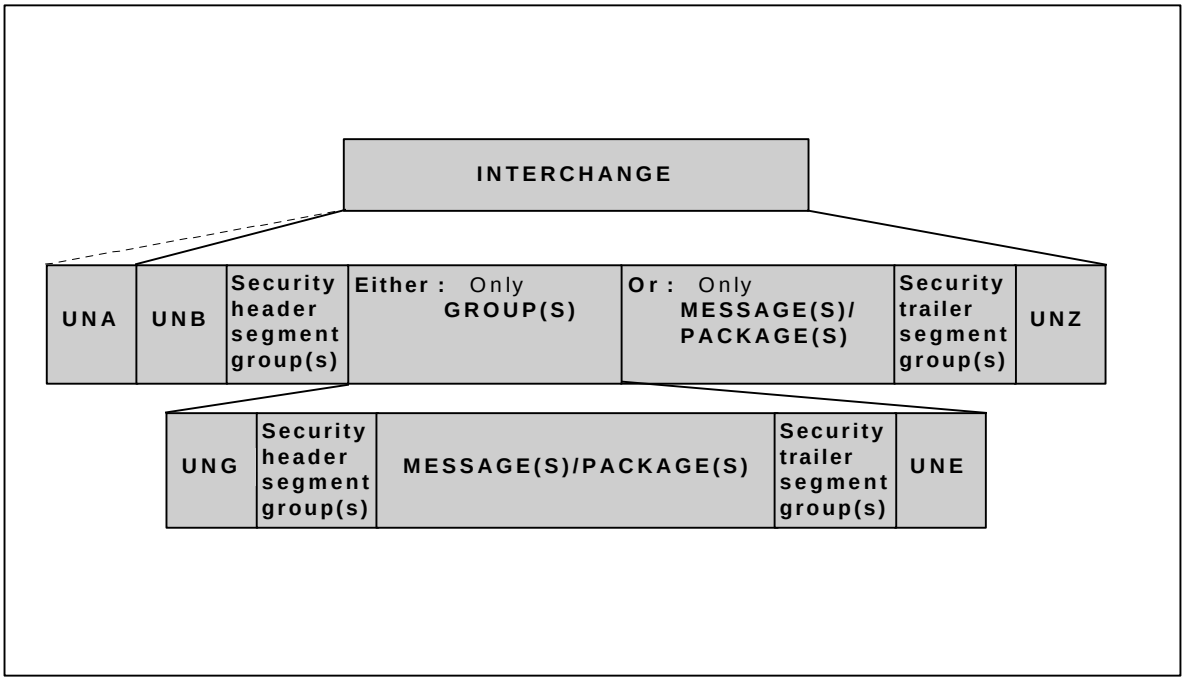


Figure 5 - Interchange showing security at both interchange and group levels (schematic)

6.1.2 Security header and trailer segment groups structure

TAG	Name	S	R	
UNB	Interchange Header	M	1	
-----	Segment Group 1 ----------	C	99	--------+
USH	Security Header	M	1	I
USA	Security Algorithm	C	3	I
-----	Segment Group 2 ----------	C	2	----+ I
USC	Certificate	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	--------+
	Group(s) or Message(s)/Package(s)			
-----	Segment Group n ----------	C	99	----+
UST	Security Trailer	M	1	I
USR	Security Result	C	1	----+
UNZ	Interchange Trailer	M	1	

Table 3 - Security header and security trailer segment groups segment table (interchange level security only)

TAG	Name	S	R	
UNG	Group Header	M	1	
-----	Segment Group 1 ----------	C	99	--------+
USH	Security Header	M	1	I
USA	Security Algorithm	C	3	I
-----	Segment Group 2 ----------	C	2	----+ I
USC	Certificate	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	--------+
	Message(s)/Package(s)			

```
-----  Segment Group n ---------------  C   99  ----+
UST    Security Trailer                 M   1       I
USR    Security Result                  C   1   ----+
UNE    Group Trailer                    M   1
```

Table 4 - Security header and security trailer segment groups segment table (group level security only)

Note: UNB interchange header, UNZ interchange trailer, UNG group header and UNE group trailer are specified in Part 1 of ISO 9735. They are not described further in this Part.

The complete directory specification of the segments and data elements may be found in annex B.

6.1.3 Scope of security application

There are two possibilities for the scope of security application:

1. The computation of each of the integrity and authentication values and of the digital signatures starts with and includes the current security header segment group and the group(s) or message(s)/package(s), themselves. In this case no other security header or security trailer segment groups shall be encompassed within this scope.

   The security header segment group shall be counted from the first character, namely a "U", to the separator ending this security header segment group, both included, and the group(s) or message(s)/package(s), from the first character following the separator ending the last security header segment group to the separator preceding the first character of the first security trailer segment group, both included.

   Thus the order in which security services integrated in this manner are performed, is not prescribed. They are completely independent of each other.

   Figures 6 and 7 illustrate this case (the scope of application of the security service defined in the security header 2 is represented by shaded boxes):

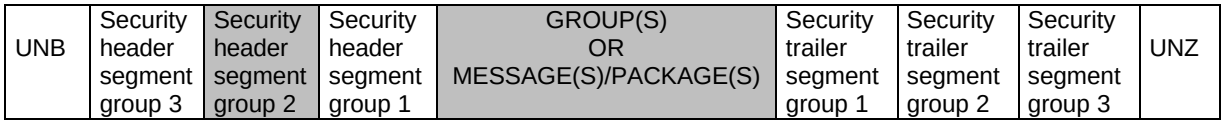


Figure 6 - Scope of application: security header segment group and group(s) or message(s)/package(s) only (schematic)

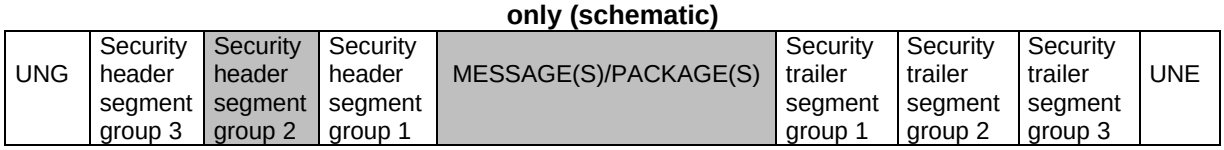


Figure 7 - Scope of application: security header segment group and message(s)/package(s) only (schematic)

2. The computation starts with and includes the current security header segment group to the associated security trailer segment group. In this case the current security header segment group, the group(s) or message(s)/package(s), and all the other embedded security header and trailer segment groups shall be encompassed within this scope.

   The scope shall include every character from the first character, namely a "U", of the current security header segment group, to the separator preceding the first character of the associated security trailer segment group, both included.

   Figures 8 and 9 illustrate this case (the scope of application of the security service defined in the security header 2 is represented by shaded boxes):

UNB	Security header segment group 3	Security header segment group 2	Security header segment group 1	GROUP(S) OR MESSAGE(S)/PACKAGE(S)	Security trailer segment group 1	Security trailer segment group 2	Security trailer segment group 3	UNZ

Figure 8 - Scope of application: from security header segment group to security trailer segment group (schematic)

UNG	Security header segment group 3	Security header segment group 2	Security header segment group 1	MESSAGE(S)/PACKAGE(S)	Security trailer segment group 1	Security trailer segment group 2	Security trailer segment group 3	UNE

Figure 9 - Scope of application: from security header segment group to security trailer segment group (schematic)

For each added security service, either of the two approaches may be chosen.

In both cases, the relation between the security header segment group and associated security trailer segment group shall be provided by the data elements security reference number of the USH and of the UST segments.

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

Definitions

A.1 **asymmetric algorithm**: A cryptographic algorithm employing a public key and a private key. Together these form an asymmetric key set. [1]

A.2 **authentication**: See ‘data origin authentication’. [2]

A.3 **certificate**: The public key of a user, together with some other information, rendered unforgeable by a signature with the private key of the certification authority which issued it. (ISO/IEC 9594-8) [3]

A.4 **certification authority**: An authority trusted by one or more users to create and assign certificates. (ISO/IEC 9594-8) [4]

A.5 **confidentiality**: The property that information is not made available or disclosed to unauthorized individuals, entities or processes. (ISO 7498-2) [5]

A.6 **credential**: Data that serves to establish the claimed identity of an entity. (ISO 7498-2) [6]

A.7 **cryptography**: The discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorized use. (ISO 7498-2) [7]

A.8 **data integrity**: The property that data has not been altered or destroyed in an unauthorised manner. (ISO 7498-2) [8]

A.9 **data origin authentication**: The corroboration that the source of data received is as claimed. (ISO 7498-2) [9]

A.10 **decryption**: See decipherment. (ISO 7498-2) [10]

A.11 **decipherment**: The reversal of a corresponding reversible encipherment. (ISO 7498-2) [11]

A.12 **digital signature**: Data appended to, or a cryptographic transformation (see cryptography) of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery e.g. by the recipient. (ISO 7498-2) [12]

A.13 **encipherment**: The cryptographic transformation of data (see cryptography) to produce ciphertext. (ISO 7498-2) [13]

A.14 **encryption**: See encipherment. (ISO 7498-2) [14]

A.15 **filtering**: The process by which octets containing arbitrary bit patterns are converted to octets belonging to the character set which the underlying syntax is capable of supporting. [15]

A.16 **hash function**: A (mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A 'good' hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range. (ISO/IEC 9594-8) [16]

A.17 **integrity**: See ‘data integrity’. [17]

A.18 **key**: A sequence of symbols that controls the operations of encipherment and decipherment. (ISO 7498-2) [18]

A.19 **non-repudiation**: Denial by one of the entities involved in a communication of having participated in all or part of the communication. (ISO 7498-2) [19]

A.20 **private key**: (In a public key cryptosystem) that key of a user's key pair which is known only by that user. (ISO/IEC 9594-8) [20]

A.21 **public key**: (In a public key cryptosystem) that key of a user's key pair which is publicly known. (ISO/IEC 9594-8) [21]

A.22 **secret key**: a key used with symmetric cryptographic techniques and usable only by a set of specified entities. (ISO/IEC 11770-1) [22]

A.23 **symmetric algorithm**: A cryptographic algorithm employing the same value of key for both enciphering and deciphering or for both authentication and validation.

A.24 **threat**: A potential violation of security. (ISO 7498-2) [23]

Annex B
(normative)
Addendum — to be added to Part 1 annex C when approved

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

B.1 Segment directory

B.1.1 Segment specification legend:

Function — The function of the segment

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

TAG — The tags for 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 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 stand-alone data element or composite data element in the segment, or of the components in the composite (where M = Mandatory and C = Conditional)

R — The maximum number of occurrences of a stand-alone data element or composite data element in the segment

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

B.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

B.1.3 Index of segments by tag

TAG	Name
USA	Security algorithm
USC	Certificate
USH	Security header
USR	Security result
UST	Security trailer

B.1.4 Index of segments by name

TAG	Name
USC	Certificate
USA	Security algorithm
USH	Security header
USR	Security result
UST	Security trailer

B.1.5 Segment specifications

USA SECURITY ALGORITHM

Function: To identify a security algorithm, the technical usage made of it, and to contain the technical parameters required.

POS	TAG	Name	S	R	Repr.	Notes
010	S502	SECURITY ALGORITHM	M	1		
	0523	Use of algorithm, coded	M		an..3	
	0525	Cryptographic mode of operation, coded	C		an..3	
	0533	Mode of operation code list identifier	C		an..3	
	0527	Algorithm, coded	C		an..3	
	0529	Algorithm code list identifier	C		an..3	
	0591	Padding mechanism, coded	C		an..3	
	0601	Padding mechanism code list identifier	C		an..3	
020	S503	ALGORITHM PARAMETER	C	9		1
	0531	Algorithm parameter qualifier	M		an..3	
	0554	Algorithm parameter value	M		an..512	

NOTES:

1. S503, provides space for one parameter. The number of repetitions of S503 actually used will depend on the algorithm used. The order of the parameters is arbitrary but, in each case, the actual value is preceded by a coded algorithm parameter qualifier.

USC CERTIFICATE

Function: To convey the public key and the credentials of its owner.

POS	TAG	Name	S	R	Repr.	Notes
010	0536	CERTIFICATE REFERENCE	C	1	an..35	2
020	S500	SECURITY IDENTIFICATION DETAILS	C	2		3
	0577	Security party qualifier	M		an..3	
	0538	Key name	C		an..35	
	0511	Security party identification	C		an..512	
	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	
030	0545	CERTIFICATE SYNTAX AND VERSION, CODED	C	1	an..3	2
040	0505	FILTER FUNCTION, CODED	C	1	an..3	
050	0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	4
060	0543	CERTIFICATE ORIGINAL CHARACTER SET REPERTOIRE, CODED	C	1	an..3	5
070	0546	USER AUTHORISATION LEVEL	C	1	an..35	
080	S505	SERVICE CHARACTER FOR SIGNATURE	C	5		6
	0551	Service character for signature qualifier	M		an..3	
	0548	Service character for signature	M		an..4	
090	S501	SECURITY DATE AND TIME	C	4		7
	0517	Date and time qualifier	M		an..3	
	0338	Event date	C		n..8	
	0314	Event time	C		an..15	
	0336	Time offset	C		n4	
100	0567	SECURITY STATUS, CODED	C	1	an..3	1
110	0569	REVOCATION REASON, CODED	C	1	an..3	1

DEPENDENCY NOTES:

1. D5 (110, 100) If first, then all

NOTES:

2. 0536, if a full certificate (including the USR segment) is not used, the only data elements of the certificate shall be a unique certificate reference made of: the certificate reference (0536), the S500 identifying the issuer certification authority or the S500 identifying the certificate owner, including its public key name. In the case of a non-EDIFACT certificate data element 0545 shall also be present.

3. S500/0538, identifies a public key: either of the owner of this certificate, or the public key related to the private key used by the certificate issuer (certification authority or CA) to sign this certificate.

4. 0507, the original character set encoding of the certificate when it was signed. If no value is specified, the character set encoding corresponds to that identified by the character set repertoire standard.

5. 0543, the original character set repertoire of the certificate when it was signed. If no value is specified, the default is defined in the interchange header.

6. S505, when this certificate is transferred, it will use the default service characters defined in part 1 of ISO 9735, or those defined in the service string advice, if used. This data element may specify the service characters used when the certificate was signed. If this data element is not used then they are the default service characters.

7. S501, dates and times involved in the certification process. Four occurrences of this composite data element are possible: one for the certificate generation date and time, one for the certificate start of validity period, one for the certificate end of validity period, one for revocation date and time.

USH SECURITY HEADER

Function: To specify a security mechanism applied to a EDIFACT structure (i.e.: either message/package, group or interchange).

POS	TAG	Name	S	R	Repr.	Notes
010	0501	SECURITY SERVICE, CODED	M	1	an..3	
020	0534	SECURITY REFERENCE NUMBER	M	1	an..14	
030	0541	SCOPE OF SECURITY APPLICATION, CODED	C	1	an..3	1
040	0503	RESPONSE TYPE, CODED	C	1	an..3	
050	0505	FILTER FUNCTION, CODED	C	1	an..3	
060	0507	ORIGINAL CHARACTER SET ENCODING, CODED	C	1	an..3	2
070	0509	ROLE OF SECURITY PROVIDER, CODED	C	1	an..3	
080	S500	SECURITY IDENTIFICATION DETAILS	C	2		3,4
	0577	Security party qualifier	M		an..3	
	0538	Key name	C		an..35	
	0511	Security party identification	C		an..512	
	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	
090	0520	SECURITY SEQUENCE NUMBER	C	1	an..35	
100	S501	SECURITY DATE AND TIME	C	1		5
	0517	Date and time qualifier	M		an..3	

	0338	Event date	C	n..8
	0314	Event time	C	an..15
	0336	Time offset	C	n4

NOTES:

1. 0541, if not present the default scope is the current security header segment group and the message body or object itself.

2. 0507, the original character set encoding of the EDIFACT structure when it was secured. If no value is specified, the character set encoding corresponds to that identified by the syntax identifier character repertoire in the UNB segment.

3. S500, two occurrences are possible: one for the security originator, one for the security recipient.

4. S500/0538, may be used to establish the key relationship between the sending and receiving parties.

5. S501, may be used as a security timestamp. It is security related and may differ from any dates and times that may appear elsewhere in the EDIFACT structure. It may be used to provide sequence integrity.

USR SECURITY RESULT

Function: To contain the result of the security mechanisms.

POS	TAG	Name	S	R	Repr.	Notes
010	S508	VALIDATION RESULT	M	2		1
	0563	Validation value qualifier	M		an..3	
	0560	Validation value	C		an..512	

NOTES:

1. S508, two occurrences shall be used in the case of signature algorithms requiring two parameters to express the result.
   In the case of an RSA signature, only one occurrence of S508 shall be used.
   In the case of a DSA signature two occurrences of S508 shall be used.

UST SECURITY TRAILER

Function: To establish a link between security header and security trailer segment groups.

POS	TAG	Name	S	R	Repr.	Notes
010	0534	SECURITY REFERENCE NUMBER	M	1	an..14	1
020	0588	NUMBER OF SECURITY SEGMENTS	M	1	n..10	

NOTES:

1. 0534, the value shall be identical to the value in 0534 in the corresponding USH segment.

B.2 Composite data element directory

B.2.1 Composite data element specification legend:

POS The sequential position number of the component data element in the composite data element

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

Name Name of a component data element in small letters

S The status of the component data element in the composite data element (where M = Mandatory and C = Conditional)

Repr. Data value representation of the component data elements 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

Desc. Description of the composite data element

B.2.2 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 ISO 9735-1:1998 for the definition of the dependency note identifiers

B.2.3 Index of composite data elements by tag

TAG	Name
S500	Security identification details
S501	Security date and time
S502	Security algorithm
S503	Algorithm parameter
S505	Service character for signature
S508	Validation result

B.2.4 Index of composite data elements by name

TAG	Name
S503	Algorithm parameter
S502	Security algorithm
S501	Security date and time
S500	Security identification details
S505	Service character for signature
S508	Validation result

B.2.5 Composite data element specifications

S500 SECURITY IDENTIFICATION DETAILS

Desc: Identification of parties involved in the security process.

POS	TAG	Name	S	Repr.	Notes
010	0577	Security party qualifier	M	an..3	
020	0538	Key name	C	an..35	
030	0511	Security party identification	C	an..512	1
040	0513	Security party code list qualifier	C	an..3	1
050	0515	Security party code list responsible agency, coded	C	an..3	1
060	0586	Security party name	C	an..35	
070	0586	Security party name	C	an..35	
080	0586	Security party name	C	an..35	

DEPENDENCY NOTES:

1. D2 (030, 040, 050) All or none

S501 SECURITY DATE AND TIME

Desc: Security related date and time.

POS	TAG	Name	S	Repr.	Notes
010	0517	Date and time qualifier	M	an..3	
020	0338	Event date	C	n..8	
030	0314	Event time	C	an..15	
040	0336	Time offset	C	n4	

S502 SECURITY ALGORITHM

Desc: Identification of a security algorithm.

POS	TAG	Name	S	Repr.	Notes
010	0523	Use of algorithm, coded	M	an..3	
020	0525	Cryptographic mode of operation, coded	C	an..3	1,3
030	0533	Mode of operation code list identifier	C	an..3	1,6

040	0527	Algorithm, coded	C an..3	2,3,5
050	0529	Algorithm code list identifier	C an..3	2
060	0591	Padding mechanism, coded	C an..3	4,5
070	0601	Padding mechanism code list identifier	C an..3	4

DEPENDENCY NOTES:

1. D5 (030, 020, 040) If first, then all
2. D5 (050, 040) If first, then all
3. D5 (020, 040) If first, then all
4. D5 (070, 060) If first, then all
5. D5 (060, 040) If first, then all

NOTES:

6. 0533, a mode of operation shall be chosen in relation to the chosen algorithm (data element 0527). Some combinations of mode of operation and algorithm are not appropriate.

S503 ALGORITHM PARAMETER

Desc: Parameter required by a security algorithm.

POS	TAG	Name	S	Repr.	Notes
010	0531	Algorithm parameter qualifier	M	an..3	
020	0554	Algorithm parameter value	M	an..512	

S505 SERVICE CHARACTER FOR SIGNATURE

Desc: Identification of the characters used as syntactical service characters when a signature was computed.

POS	TAG	Name	S	Repr.	Notes
010	0551	Service character for signature qualifier	M	an..3	
020	0548	Service character for signature	M	an..4	

S508 VALIDATION RESULT

Desc: Result of the application of the security mechanism.

POS	TAG	Name	S	Repr.	Notes
010	0563	Validation value qualifier	M	an..3	
020	0560	Validation value	C	an..512	1

NOTES:

1. 0560, the length of this data element shall be determined by the characteristics of the cryptographic algorithm used to compute the validation value and the filter function applied to the result.

B.3 Simple data element directory

B.3.1 Simple data element specification legend:

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

B.3.2 Index of simple data elements by tag

TAG	Name
0501	Security service, coded
0503	Response type, coded
0505	Filter function, coded
0507	Original character set encoding, coded
0509	Role of security provider, coded
0511	Security party identification
0513	Security party code list qualifier
0515	Security party code list responsible agency, coded
0517	Date and time qualifier
0520	Security sequence number
0523	Use of algorithm, coded
0525	Cryptographic mode of operation, coded
0527	Algorithm, coded
0529	Algorithm code list identifier
0531	Algorithm parameter qualifier
0533	Mode of operation code list identifier
0534	Security reference number
0536	Certificate reference
0538	Key name
0541	Scope of security application, coded
0543	Certificate original character set repertoire, coded
0545	Certificate syntax and version, coded
0546	User authorisation level
0548	Service character for signature
0551	Service character for signature qualifier
0554	Algorithm parameter value
0560	Validation value
0563	Validation value qualifier
0567	Security status, coded
0569	Revocation reason, coded
0577	Security party qualifier
0586	Security party name

0588 Number of security segments
0591 Padding mechanism, coded
0601 Padding mechanism code list identifier

B.3.3 Index of simple data elements by name

TAG	Name
0529	Algorithm code list identifier
0531	Algorithm parameter qualifier
0554	Algorithm parameter value
0527	Algorithm, coded
0543	Certificate original character set repertoire, coded
0536	Certificate reference
0545	Certificate syntax and version, coded
0525	Cryptographic mode of operation, coded
0517	Date and time qualifier
0505	Filter function, coded
0538	Key name
0533	Mode of operation code list identifier
0588	Number of security segments
0507	Original character set encoding, coded
0601	Padding mechanism code list identifier
0591	Padding mechanism, coded
0503	Response type, coded
0569	Revocation reason, coded
0509	Role of security provider, coded
0541	Scope of security application, coded
0513	Security party code list qualifier
0515	Security party code list responsible agency, coded
0511	Security party identification
0586	Security party name
0577	Security party qualifier
0534	Security reference number
0520	Security sequence number
0501	Security service, coded
0567	Security status, coded
0548	Service character for signature
0551	Service character for signature qualifier
0523	Use of algorithm, coded
0546	User authorisation level
0560	Validation value
0563	Validation value qualifier

B.3.4 Simple data element specifications

Only simple data elements not defined in other parts of ISO 9735 are included here.

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

   0501  SECURITY SERVICE, CODED

   Desc: Specification of the security service applied.

   Repr: an..3

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

   0503  RESPONSE TYPE, CODED
```

Desc: Specification of the type of response expected from the recipient.

Repr: an..3

0505 FILTER FUNCTION, CODED

Desc: Identification of the filtering function used to reversibly map any bit pattern on to a restricted character set.

Repr: an..3

0507 ORIGINAL CHARACTER SET ENCODING, CODED

Desc: Identification of the character set in which the secured EDIFACT structure was encoded when security mechanisms were applied.

Repr: an..3

0509 ROLE OF SECURITY PROVIDER, CODED

Desc: Identification of the role of the security provider in relation to the secured item.

Repr: an..3

0511 SECURITY PARTY IDENTIFICATION

Desc: Identification of a party involved in the security process, according to a defined registry of security parties.

Repr: an..512

0513 SECURITY PARTY CODE LIST QUALIFIER

Desc: Identification of the type of identification used to register the security parties.

Repr: an..3

0515 SECURITY PARTY CODE LIST RESPONSIBLE AGENCY, CODED

Desc: Identification of the agency in charge of registration of the security parties.

Repr: an..3

0517 DATE AND TIME QUALIFIER

Desc: Specification of the type of date and time.

Repr: an..3

0520 SECURITY SEQUENCE NUMBER

Desc: Sequence number assigned to the EDIFACT structure to which security is applied.

Repr: an..35

Note 1: This sequence number is security related and may differ from the identification of the EDIFACT structure that may appear elsewhere. It may be used when sequence integrity is required.

0523 USE OF ALGORITHM, CODED

Desc: Specification of the usage made of the algorithm.

Repr: an..3

0525 CRYPTOGRAPHIC MODE OF OPERATION, CODED

Desc: Specification of the mode of operation used for the algorithm.

Repr: an..3

0527 ALGORITHM, CODED

Desc: Identification of the algorithm.

Repr: an..3

0529 ALGORITHM CODE LIST IDENTIFIER

Desc: Specification of the code list used to identify the algorithm.

Repr: an..3

0531 ALGORITHM PARAMETER QUALIFIER

Desc: Specification of the type of parameter value.

Repr: an..3

0533 MODE OF OPERATION CODE LIST IDENTIFIER

Desc: Specification of the code list used to identify the cryptographic mode of operation.

Repr: an..3

0534 SECURITY REFERENCE NUMBER

Desc: Unique reference number assigned by the security originator to a pair of security header and security trailer segment groups.

Repr: an..14

Note 1: The value shall be arbitrarily assigned, but the same value shall not be used more than once within the same EDIFACT structure, i.e. interchange, group, message or package.

0536 CERTIFICATE REFERENCE

Desc: Identifies one certificate for a certification authority.

Repr: an..35

0538 KEY NAME

Desc: Name used to establish a key relationship between the parties.

Repr: an..35

0541 SCOPE OF SECURITY APPLICATION, CODED

Desc: Specification of the scope of application of the security service defined in the security header.

Repr: an..3

Note 1: It defines the data that have to be taken into account by the related cryptographic process.

0543 CERTIFICATE ORIGINAL CHARACTER SET REPERTOIRE, CODED

Desc: Identification of the character set repertoire used to create the certificate it was signed.

Repr: an..3

0545 CERTIFICATE SYNTAX AND VERSION, CODED

Desc: Coded identification of the syntax and version used to create the certificate.

Repr: an..3

0546 USER AUTHORISATION LEVEL

Desc: Specification of the authorisation level associated with the owner of the certificate.

Repr: an..35

0548 SERVICE CHARACTER FOR SIGNATURE

Desc: Service character used when the signature was computed.

Repr: an..4

Note 1: In order to avoid translator problems, this service character is represented by its value in the character set identified by the original character set encoding data element (0507), hexa-filtered on, at least, two characters. For example the service character "'" is coded "27" (two characters), if ASCII 8bit code page is used.

0551 SERVICE CHARACTER FOR SIGNATURE QUALIFIER

Desc: Identification of the type of service character used when the signature was computed.

Repr: an..3

0554 ALGORITHM PARAMETER VALUE

Desc: Value of a parameter required by the algorithm.

Repr: an..512

Note 1: If necessary, this value shall be filtered by an appropriate filter function. Note that key names do not need to be filtered.

0560 VALIDATION VALUE

Desc: Security result corresponding to the security function specified.

Repr: an..512

Note 1: If necessary, this value shall be filtered by an appropriate filter function.

0563 VALIDATION VALUE QUALIFIER

Desc: Identification of the type of validation value.

Repr: an..3

0567 SECURITY STATUS, CODED

Desc: Identification of the security element (key or certificate, for instance) status.

Repr: an..3

0569 REVOCATION REASON, CODED

Desc: Identification of the reason why the certificate has been revoked.

Repr: an..3

0577 SECURITY PARTY QUALIFIER

Desc: Identification of the role of the security party.

Repr: an..3

0586 SECURITY PARTY NAME

Desc: Name of the security party.

Repr: an..35

0588 NUMBER OF SECURITY SEGMENTS

Desc: The number of security segments in a security header/trailer group pair, plus the USD and USU segments where the security header/trailer group pair is used for encryption.

Repr: n..10

Note 1: Each security header/trailer group pair shall contain ist own count of the number of security segments within that group pair.

Note 2: The count of the number of security segments includes the USR segment in the security trailer.

0591 PADDING MECHANISM, CODED

Desc: Padding mechanism or padding scheme applied.

Repr: an..3

0601 PADDING MECHANISM CODE LIST IDENTIFIER

Desc: Specification of the code list used to identify the padding mechanism or padding scheme.

Repr: an..3

Annex C
(informative)

EDIFACT security threats and solutions

This annex describes the generic security threats to message/package transmission, between the originator(s) of the message/package and the recipient(s). The general approaches to overcome these threats are also covered. These threats and solutions are relevant at any level: message/package, group or interchange.

C.1 Security threats

The storage and transfer of EDIFACT messages/packages via electronic media and means expose them to a number of threats, notably:

* the unauthorized disclosure of message/package content

* the intentional insertion of non-bonafide messages/packages

* the duplication, loss or replay of messages/packages

* the modification of message/package content

* the deletion of messages/packages

* the repudiation of message/package responsibility by its sender or its receiver

These threats may be intentionally perpetrated, as with the unauthorized manipulation of message/package content, or unintentionally perpetrated, as with a communication error resulting in the modification of message/package content.

C.2 Security solutions - basic services and principles of usage

To counter the aforementioned threats a number of security mechanisms have been identified which utilize one or more methodologies to meet their objectives.

It is important to be able to identify unambiguously the parties involved when messages/packages are secured - the security originator, henceforth called the sender for simplicity, who secures the message/package prior to transmission, and the security recipient, henceforth called the receiver, who performs checks on the received message/package. These parties may be identified in the security segments. This identification may be performed by means of so-called certificates, (in fact, either the certificate itself or a certificate reference), explained below, if asymmetric algorithms are used.

Typically, the use of a certification authority (CA) is required in an open system. This is a third party which is trusted by the involved parties to a limited degree, namely to identify and register all users with their public key. This information is conveyed to other users by means of a certificate, which is a digital signature issued by the CA on a message which consists of user identification information and the user's public key. In this situation, the trust is purely functional and does not involve secret or private keys.

Alternatively, if symmetric techniques are used the identity of the parties involved would be indicated in the security sender/recipient name fields.

A message/package may be secured by several parties (for example a message/package may have multiple digital signatures) and so the security related information may be repeated to allow the identification of several signing or authenticating parties and correspondingly to include several digital signatures or control values.

The requirements and techniques prescribed for securing EDIFACT messages/packages, groups or interchanges are presented below.

C.2.1 Sequence integrity

Sequence integrity protects against the duplication, addition, deletion, loss or replay of a EDIFACT structure (message/package, group or interchange).

To detect lost messages/packages, groups or interchanges

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

To detect added or duplicated messages/packages, groups or interchanges

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

When sequence numbers are used it shall be agreed between the parties 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 shall be guaranteed by one of the other functions mentioned below.

C.2.2 Content integrity

Content integrity protects against the modification of data.

Protection may be achieved by the sender including an integrity control value. This value may 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 actually provide non-repudiation of origin, on the control value are necessary. Alternatively, origin authentication, which is obtained using a message authentication code, will imply content integrity. The receiver computes the integrity control value of the data actually received using the corresponding algorithms and parameters and compares the result with the value received.

In conclusion, content integrity in EDI is typically obtained as a sub-product of origin authentication or non-repudiation of origin.

C.2.3 Origin authentication

Origin authentication protects the receiver against the actual sender of a message/package, group or interchange claiming to be another (authorized) party.

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

This service may include content integrity and may be obtained as a sub-product of non-repudiation of origin.

In most cases, it would be desirable to have at least origin authentication.

C.2.4 Non-repudiation of origin

Non-repudiation of origin protects the receiver of a message/package, group or interchange from the sender's denial of having sent it.

Protection may be achieved by including a digital signature (or by using an appropriate implementation of the function described under "origin authentication" based on tamper resistant hardware or trusted third parties). A digital signature is obtained by encrypting, with an asymmetric algorithm and a private key, the object or a control value derived from the data (by using a hash function, for example).

The digital signature may be verified by using the public key which corresponds to the private key used to create it. This public key may be included with the interchange agreement signed by the parties or be included in a certificate digitally signed by a certification authority. The certificate may be sent as part of the EDIFACT structure.

The digital signature provides not only non-repudiation of origin but also content integrity and origin authentication.

C.2.5 Non-repudiation of receipt

Non-repudiation of receipt protects the sender of a message/package, group or interchange from the receiver's denial of having received it.

Protection may be achieved by the receiver sending an acknowledgement which includes a digital signature based on the data in the original EDIFACT structure. The acknowledgement takes the form of a service message from the receiver to the sender.

C.2.6 Confidentiality of content

Confidentiality of content protects against the unauthorized reading, copying or disclosure of the content of a message/package, group or interchange.

Protection may be assured by encrypting the data. Encryption may be performed by using a symmetric algorithm with a secret key shared by the sender and the receiver.

However the secret key may be transmitted securely by encrypting it under the receiver's public key using an asymmetric algorithm.

Confidentiality is addressed separately in part 7 of ISO 9735.

C.2.7 Interrelation among security services

As noted already, some services by nature encompass other services, and it is thus not necessary to additionally include the services which are achieved implicitly. For example, the use of the mechanism to provide non-repudiation of origin implies content integrity.

The following table summarizes these interrelations:

also implies: Use of:	Content Integrity	Origin Authentication	Non-repudiation of origin
Content Integrity	yes		
Origin Authentication	yes	yes	
Non-repudiation of origin	yes	yes	yes

Annex D
(informative)

How to protect an EDIFACT structure

The following are some of the more fundamental steps to be taken in order to implement security for EDIFACT structures: messages/packages, groups or interchanges. For further details and explanation of principles, refer to Annex C of this part of ISO 9735, ISO 7498-2 and ISO/IEC 9594-8 / CCITT X.509.

The first step is to identify (in co-operation with business associates) the need for security services. The security services available in the EDIFACT world are revisited below, and it is important to establish which of these are required in the business relations to prevent the identified threats. Typically, the needs could be defined by the request for auditing, internally as well as externally. The basic security services available at the sender's end are the following:

- content integrity
- origin authentication
- non-repudiation of origin

These services are not independent, and it is thus not necessary to additionally include the services which are achieved implicitly. For example, the use of the non-repudiation of origin service implicitly achieves content integrity.

These relations are summarized in the interrelation table of Annex C, section C.2.7.

Consequently, the sender would choose at most one service of the three.

Non-repudiation of receipt is a service to be initiated by the receiver. It could either be requested explicitly by the sender or mandated in an interchange agreement. A message, AUTACK, has been developed to convey the receipt.

D.1 Bilateral agreement/third party

If security services are being integrated, additional agreements have to be set up with the business partners. There are a number of different approaches available, of which two extremes are briefly presented here.

A minimal requirement would be a bilateral agreement with each individual partner, agreeing on security services, algorithms, codes, key management methods, actions in case of misconduct, etc. A draft of such an agreement is available from the European Commission TEDIS programme. In this case, very little security-related information needs to be included in the message/package itself.

The other extreme would be to involve a third party acting as a certification authority, which registers all users and issues certificates to certify the users' public keys. In this situation, it may be adequate simply to conclude an agreement with the certification authority. The certification authority would typically be responsible for blacklisting as well. In this case, more comprehensive security-related information may need to be included.

The security services have been integrated into the EDIFACT setting in a manner that offers maximum flexibility, and caters for both extremes described above, as well as for any intermediate situation.

D.2 Practical aspects

There are, of course, a number of different aspects that need to be addressed in order to realize these security services, such as key generation, the need for a translator capable of handling security segments, internal procedures to make full use of the security services, such as storing incoming messages/packages with digital signatures, the use of multiple signatures, etc.

It shall be emphasized that integration of security services is completely transparent to, and independent of, the communication protocols used. If a system allows the transmission of an EDIFACT message/package, it will also allow the transmission of a secured EDIFACT message/package.

D.3 Procedure for constructing a secured EDIFACT structure

First, an EDIFACT structure message/package, group or interchange is created. Then the appropriate security services are determined and applied. If these are based on digital signatures, the persons possessing the private keys have to be involved, directly or indirectly. This does not have to take place immediately after the generation of the EDIFACT structure.

Likewise, on incoming EDIFACT structures, the first step would be to verify the security services , and, just as in the paper world, possibly to store the secured EDIFACT structure for later auditing and documentation.

D.4 Security services sequence of application

The order in which the security services are performed is left entirely to the users as all services may be completely independent of each other. In particular, if multiple signatures are used, without embedding of security header and security trailer segment groups, the order in which they are calculated, and verified, is of no consequence.

D.5 Separated message security at message/package level

There are two business requirements for this feature, namely

1) to provide security for one or more messages/packages in a single separate message from the sender,

2) to provide a secured acknowledgement to the sender for having received the original message/package(s), without returning them.

These requirements may be met by the secure authentication and acknowledgement message, AUTACK which is described in Part 6 of ISO 9735.

D.5.1 Separated message security used by sender

This use of the AUTACK allows the sender to provide any security service but forwarded in a separate message. Thus the security services may be communicated at a later or more appropriate stage. Additionally they may secure several original messages/packages, in contrast to direct integration, at message/package level, which secures one message/package at a time.

The principles are identical for the integrated and separated approaches, but the latter requires a unique reference to the original message/package(s) being secured.

D.5.2 Separated message security used by receiver

This use of the AUTACK addresses the requirement to provide non-repudiation of receipt. For a detailed description of the message, refer to AUTACK itself in Part 6 of ISO 9735.

The AUTACK may be used as a secured acknowledgement sent by the receiver of one or more interchanges or one or more messages/packages from one or more interchanges to their sender. The criteria and means by which an AUTACK is generated provide the sender of the original message/package(s) or interchange(s) with secured acknowledgement that it was received by the intended party.

D.6 Separated message security at group or interchange levels

The technique described as separated message/package security, in section D.5 at message/package level, may be used to secure complete groups or complete interchanges.

The two business requirements for this feature, are:

1) to provide security for one or more group or interchange in a single separate message from the sender,

2) to provide a secured acknowledgement to the sender for having received the original group(s) or interchange(s), without returning them.

These requirements may be met by the secure authentication and acknowledgement message, AUTACK described in Part 6 of ISO 9735.

Annex E
(informative)

Message protection examples

Three examples are provided herein to illustrate different application of security service segments.

These examples of message security are based on EDIFACT payment orders as described in the MIG handbook of Financial messages published by SWIFT. However, the security mechanisms described here are totally independent of the type of message and may be applied to any EDIFACT message.

"Example 1: message Origin Authentication" shows how security service segments may be used when a **symmetric algorithm** based method is applied, to provide message origin authentication. The symmetric key has been exchanged previously between the partners, and the security header segment group contains only two rather simple segments.

"Example 2: non repudiation of origin, first technique" shows how security service segments may be used when an **asymmetric algorithm** based method is applied, to provide non repudiation of origin. The algorithm applied directly to the message is a **hash-function**, which does not require any key exchange between the partners. The hash-value is signed by an asymmetric algorithm. The public key needed by the receiver to verify the signature of the message is included in a certificate segment which is conveyed in the security header segment group of the message. This certificate is signed by its issuer (the "authority") and contains the public key of the authority, in order that any partner may verify the integrity and authenticity of the certificate.

"Example 3: non repudiation of origin, second technique" shows how security service segments may be used when an **asymmetric algorithm** based method is applied, to provide non repudiation of origin. The algorithm applied directly to the message is a **symmetric algorithm**, which requires a symmetric key exchange between the partners, and provides an "integrity value". This symmetric key is exchanged within the security header segment group of the message, encrypted by means of an asymmetric algorithm, under the public key of the expected receiver.
The integrity value is signed by an asymmetric algorithm. The public key needed by the receiver to verify the signature of the message is included in the first certificate segment which is conveyed in the security header segment group of the message. This certificate is signed by its issuer (the "authority") and contains the public key of the authority, in order that any partner may verify the integrity and authenticity of the certificate.
A second certificate segment contains the reference to the public key of the expected receiver, used by the message sender to protect the symmetric key.
This technique is currently used by the French banks in the ETEBAC 5 system (secured file transfer between banks and corporate customers).

In the last two examples, any partner, trusting the authority, may verify the signature of the received message using only data contained in the message.

E.1 Example 1: message origin authentication

E.1.1 Narrative

Company A orders Bank A, sort code 603000 to debit its account number 00387806 on April 9th 1996 in the amount of 54345.10 Pounds Sterling. The amount is to be paid to Bank B, sort code 201827, in favour of account number 00663151 of Company B, West Dock, Milford Haven. The payment is in settlement of invoice 62345. The contact name at the Beneficiary is Mr. Jones in the Sales Department.

Bank A requires the payment order to be secured by the security function "message origin authentication".
This is achieved by generating a "Message Authentication Code" (MAC) with the symmetric "Data Encryption Standard" (DES) according to ISO 8731-1 at the message sender's side, which is to be validated by Bank A. It is assumed that the secret DES-key has previously been exchanged between Company A and Bank A.

Remark:

In the following, only the security relevant parts of the message will be referred to.

E.1.2 Security details

SECURITY HEADER	
SECURITY SERVICE	Message origin authentication
SECURITY REFERENCE NUMBER	The reference of this header is 1
FILTER FUNCTION	All binary values (MAC) are filtered with hexadecimal filter
ORIGINAL CHARACTER SET ENCODING	The message was coded in ASCII 8 bits when the MAC was generated.
SECURITY IDENTIFICATION DETAILS Message sender (party which generates the Message Authentication Code).	Mr. SMITH of Company A
SECURITY IDENTIFICATION DETAILS Message receiver (party which verifies the Message Authentication Code).	Bank A
SECURITY SEQUENCE NUMBER	The security sequence number of this message is 001
SECURITY DATE AND TIME	The security time stamp is : date: 1996 04 09 time: 13:59:50
SECURITY ALGORITHM	
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 A symmetric algorithm is used to achieve message origin authentication. A MAC is computed, according to ISO 8731-1. The DES algorithm is used.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies the following algorithm parameter value as the name of a previously exchanged symmetric key. The key called MAC-KEY1 is used.
SECURITY TRAILER	
SECURITY REFERENCE NUMBER	The reference of this trailer is 1
NUMBER OF SECURITY SEGMENTS	4
SECURITY RESULT	
VALIDATION RESULT Validation value qualifier Validation value	 MAC 4 Byte validation result (Message Authentication Code)

E.2 Example 2: non-repudiation of origin, first technique

E.2.1 Narrative

Bank A wants the security service of non repudiation of origin on the payment order from Company A, performed by Mr. Smith.

The interchange agreement between the parties establishes that the security service of non repudiation of origin, required by Bank A, shall be achieved for payment orders, by Mr. Smith of Company A, with the use of one digital signature.

The certificate identifying the public key of Mr. Smith is issued by an authority trusted by both parties, the certificate issuer.

E.2.2 Security details

SECURITY HEADER	
SECURITY SERVICE	Non repudiation of origin
SECURITY REFERENCE NUMBER	The reference of this header is 1
RESPONSE TYPE	No acknowledgement required
FILTER FUNCTION	All binary values (signatures) are filtered with hexadecimal filter
ORIGINAL CHARACTER SET ENCODING	The message was coded in ASCII 8 bits when its signature was generated.
SECURITY SEQUENCE NUMBER	The security sequence number of this message is 202.
SECURITY DATE AND TIME	The security time stamp is : date: 1996 01 15 time: 10:05:30
SECURITY ALGORITHM	Hash function used by Mr. SMITH for signature
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An owner hashing algorithm is used. Hash function ISO/IEC 10118-2 Hash functions using a *n*- bit block cipher algorithm applied to provide a double length hash code (128 bits); initializing values: IV = 0F 0F 0F 0F 0F 0F 0F 0F IV' = F0 F0 F0 F0 F0 F0 F0 F0; padding rules as in first variant paragraph of B.3 of ISO/IEC 10118-2:1994; transformation u and u' as specified in annex A of ISO 10118-2:1994. DES block cipher algorithm is used.
CERTIFICATE	certificate of Mr. SMITH
CERTIFICATE REFERENCE	This certificate is referenced, by AUTHORITY: 00000001.
SECURITY IDENTIFICATION DETAILS Certificate owner	 Mr. SMITH of Company A
SECURITY IDENTIFICATION DETAILS Certificate issuer Key name	 Mr. SMITH's certificate was generated by a certification Authority called: AUTHORITY. The Public Key of AUTHORITY used to generate Mr. SMITH's certificate is PK1
CERTIFICATE SYNTAX AND VERSION	Version of certificate of UN/EDIFACT service segment directory.
FILTER FUNCTION	All binary values (keys and digital signatures) are filtered with hexadecimal filter
ORIGINAL CHARACTER SET ENCODING	The credentials of the certificate were coded in ASCII 8 bits when the certificate was generated.
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is segment terminator. Value "'" (apostrophe).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is data element separator. Value "+" (plus sign).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is component data element separator. Value ":" (colon).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is repetition separator. Value "*" (asterisk).

SECURITY HEADER	
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is release character. Value "?" (question mark).
SECURITY DATE AND TIME Date and time	Certificate generation time Mr. SMITH certificate was generated on 931215 at 14:12:00
SECURITY DATE AND TIME Date and time	Certificate start of validity period Validity period of Mr. SMITH's starts: 1996 01 01 000000
SECURITY DATE AND TIME Date and time	Certificate end of validity period Validity period of Mr. SMITH's ends: 1996 12 31 235959
SECURITY ALGORITHM	Asymmetric algorithm used by Mr. SMITH to sign
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An owner signing algorithm is used. No mode of operation is relevant here. RSA is the asymmetric algorithm.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a Public exponent for signature verification. Mr SMITH's public key.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a modulus for signature verification. Mr SMITH's modulus.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as the length of Mr SMITH's modulus (in bits). Mr SMITH's modulus is 512 bits long.
SECURITY ALGORITHM	Hash function used by AUTHORITY to generate Mr SMITH's certificate
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An issuer hashing algorithm is used. Hash function ISO/IEC 10118-2 Hash functions using a *n*- bit block cipher algorithm applied to provide a double length hash code (128 bits); initializing values: IV = 0F 0F 0F 0F 0F 0F 0F 0F IV' = F0 F0 F0 F0 F0 F0 F0 F0; padding rules as in first variant paragraph of B.3 of ISO/IEC 10118-2:1994; transformation u and u' as specified in annex A of ISO/IEC 10118-2:1994. DES block cipher algorithm is used.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An issuer signing algorithm is used. No mode of operation is relevant here. RSA is the asymmetric algorithm.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a public exponent for signature verification. AUTHORITY's public key.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a Modulus for signature verification. AUTHORITY's modulus.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as the length of AUTHORITY's modulus (in bits). AUTHORITY's modulus is 512 bits long.
SECURITY RESULT	Digital signature of the certificate

SECURITY HEADER	
VALIDATION RESULT Validation value qualifier Validation value	 Digital signature 512 Bit digital signature
SECURITY TRAILER	
SECURITY REFERENCE NUMBER	The reference of this security trailer is 1
NUMBER OF SECURITY SEGMENTS	9
SECURITY RESULT	Digital signature of the message
VALIDATION RESULT Validation value qualifier Validation value	 Digital signature 512 Bit digital signature

E.3 Example 3: non-repudiation of origin, second technique

E.3.1 Narrative

Bank A wants the security service of non repudiation of origin on the payment order from Company A, performed by Mr. Smith. Company A requests a secured acknowledgement by Bank A (non repudiation of receipt) which will be conveyed in an AUTACK message.

The interchange agreement between the parties establishes that the security service of non repudiation of origin shall be achieved for payment orders with the use of one digital signature.
Both parties agree that this signature is computed by 512 bit RSA (asymmetric algorithm) upon a 64 bit-integrity value computed by CBC mode DES (symmetric algorithm).The certificate identifying the public key of Mr. Smith is issued by an authority trusted by both parties.

E.3.2 Security details

SECURITY HEADER	
SECURITY SERVICE	Non repudiation of origin
SECURITY REFERENCE NUMBER	The reference of this header is 1
RESPONSE TYPE	Acknowledgement required
FILTER FUNCTION	All binary values (signatures) are filtered by hexadecimal filter
ORIGINAL CHARACTER SET ENCODING	The message was coded in ASCII 8 bits when its signature was generated.
SECURITY IDENTIFICATION DETAILS Message sender (party securing the message).	 Mr. SMITH of Company A
SECURITY IDENTIFICATION DETAILS Message receiver (party verifying message security).	 Bank A
SECURITY SEQUENCE NUMBER	The security sequence number of this message is 001.
SECURITY DATE AND TIME	The security time stamp is : date: 1996 01 15 time: 10:05:30
SECURITY ALGORITHM	Symmetric algorithm used to compute an integrity value.
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An owner hashing algorithm is used. Cipher Block Chaining; ISO/IEC 10116 (*n*-bits). A 64-bit integrity value is computed; initialization value is binary zero; a DES secret-key is used. It is transmitted encrypted under Bank A public key. DES block cipher algorithm is used.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies the following algorithm parameter value as a symmetric key encrypted under a public key. Symmetric key encrypted under Bank A public key.

SECURITY HEADER	
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies the following algorithm parameter value as a clear text initialisation value. Clear text initialisation value (all binary 0's).
CERTIFICATE	Certificate of Mr. SMITH (message sender)
CERTIFICATE REFERENCE	This certificate is referenced: 00000001, by AUTHORITY.
SECURITY IDENTIFICATION DETAILS Certificate owner	 Mr. SMITH of Company A
SECURITY IDENTIFICATION DETAILS Certificate issuer Key name	 Mr. SMITH's certificate was generated by a certification authority called: AUTHORITY. The Public Key of AUTHORITY used to generate Mr. SMITH's certificate is PK1.
CERTIFICATE SYNTAX AND VERSION	Version of certificate of UN/EDIFACT service segment directory.
FILTER FUNCTION	All binary values (keys and digital signatures) are filtered with hexadecimal filter.
ORIGINAL CHARACTER SET ENCODING	The credentials of the certificate were coded in ASCII 8 bits when the certificate was generated.
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is segment terminator. Value "'" (apostrophe).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is data element separator. Value "+" (plus sign).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is component data element separator. Value ":" (colon).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is repetition separator. Value "*" (asterisk).
SERVICE CHARACTER FOR SIGNATURE Service character for signature qualifier Service character for signature	Service character used when signature was computed Service character is release character. Value "?" (question mark).
SECURITY DATE AND TIME Date and time	Certificate generation time Mr. SMITH certificate was generated on 931215 at 14:12:00
SECURITY DATE AND TIME Date and time	Certificate start of validity period Validity period of Mr. SMITH's starts: 1996 01 01 000000
SECURITY DATE AND TIME Date and time	Certificate end of validity period Validity period of Mr. SMITH's ends: 1996 12 31 235959
SECURITY ALGORITHM	Asymmetric algorithm used by Mr. SMITH to sign
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An owner signing algorithm is used. No mode of operation is relevant here. RSA is the asymmetric algorithm.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a public exponent for signature verification. Mr SMITH's public key.

SECURITY HEADER	
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a modulus for signature verification. Mr SMITH's modulus.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as the length of Mr SMITH's modulus (in bits). Mr SMITH's modulus is 512 bit long.
SECURITY ALGORITHM	Hash function used by AUTHORITY to generate Mr SMITH's certificate
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An issuer hashing algorithm is used. Square-mod n hash function for RSA. Annex D, CCITT X509. ISO 9594-8. RSA asymmetric algorithm.
SECURITY ALGORITHM	Asymmetric algorithm is used by AUTHORITY to sign
SECURITY ALGORITHM Use of algorithm Cryptographic mode of operation Algorithm	 An issuer signing algorithm is used. No mode of operation is relevant here. RSA is the asymmetric algorithm.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a public exponent for signature verification. AUTHORITY's public key.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as a modulus for signature verification. AUTHORITY's modulus.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	 Identifies this algorithm parameter as the length of AUTHORITY's modulus (in bits). AUTHORITY's modulus is 512 bit long.
SECURITY RESULT	Digital signature of the certificate
VALIDATION RESULT Validation value qualifier Validation value	 Digital signature 512 Bit digital signature
CERTIFICATE	Certificate of Bank A (message receiver)
CERTIFICATE REFERENCE	Bank A's public key related to certificate referenced 00001001 is used
SECURITY TRAILER	
SECURITY REFERENCE NUMBER	The reference of this security trailer is 1
NUMBER OF SECURITY SEGMENTS	10
SECURITY RESULT	Digital signature of the message
VALIDATION RESULT Validation value qualifier Validation value	 Digital signature 512 Bit digital signature

Annex F
(informative)

Filter functions for UN/EDIFACT character set repertoires A and C

F.1 EDA filter

F.1.1 Rationale

Hexadecimal filtering doubles the number of characters required to represent binary data. This is a waste of space. Other existing and standardized filter functions are either not adequate for UN/EDIFACT character set repertoires A and B (ISO/IEC 646) because they map to almost the full printable ISO set (94 out of the 96 printable characters), or because they are not really more space-efficient than hexadecimal filtering (the Baudot filter).
It is thus advisable to define a filter function which is sufficiently simple and which maps to (a subset of) the UN/EDIFACT level A character set repertoire, while being more efficient than the hexadecimal filter.

F.1.2 UN/EDIFACT character set repertoires

The character set repertoire A possesses 44 characters whose use is unrestricted. In addition to those 44, 4 service characters and 8 characters not allowed for TELEX transmissions are part of the set.

All those characters are also part of the UN/EDIFACT character set repertoire B, which is not intended at all for TELEX transmission, and which possesses 82 normal characters and 3 non-printable service characters.

F.1.3 3 by 2 filtering

To represent 2 binary characters by 3 filtered characters a minimum of 41 characters are required in the set: 41 ** 3 = 68 921 > 65 536 > 64 000 = 40 ** 3

F.1.4 EDA filter specification

Having 44 allowed characters, let us avoid use of the space character part of those 44 and filter every pair of input characters (if odd, filter only the last character in 2 resulting ones) by:

- considering the binary value of the unsigned integer formed by the pair of characters (this value depends naturally on the LITTLE_ENDIAN / BIG_ ENDIAN (either Least or Most Significant Byte first) nature of the computer in use. Standardize for BIG_ENDIANs: first byte most significant)
- represent the value by a succession of 3 numbers (2 for last odd byte), in the range 0 to 42, which are:
  - the result of the division by 1849 (43 squared) (absent for last odd byte)
  - the value modulo 1849 divided by 43,
  - the value modulus 43.
- to map each number in the UN/EDIFACT level A alphabet by the correspondence table:

0 to 9	are represented by	0 to 9
A to Z	are represented by	10 to 35
( ) , - . / =	are represented by	36 up to 42 in the given order.

F.1.5 Defiltering

To defilter: map each of the 43 characters back to its value between 0 and 42,
if at least 3 filtered characters remain, compute: cl * 1849 + c2 *43 + c3 = short integer
else at least 2 remain so compute: c1 * 43 + c2 = character value.

Remarks:
a. The short integer result should be < 65 536
b. The character result should be < 256
c. In a LITTLE_ENDIAN computer, switch the 2 characters of the short integer result.

F.2 EDC filter

F.2.1 Rationale

The EDA filter was developed to allow filtering into the EDIFACT level A or B repertoire. Naturally, since this repertoire is very limited in characters, the expansion rate = 3/2 is rather bad, although already much better than the one of the hexadecimal filter = 2/1.

In the repertoires C, D, E and F a much better expansion rate is easily achievable.
Indeed, in those repertoires, the only unallowable combinations include binary values 0/0 to 1/15 and values 8/0 to 9/15.
Of the 256 possible binary values 192 are thus allowed.
A level C filter, ideal as to the low expansion rate, but requiring lengthy computations, would allow to represent 18 binary bytes into 19 filtered bytes, but not 19 bytes into 20 filtered ones, because:

192**19 > 256**18 and,
192**20 < 256**19

Limiting the transformation to bit operations, the expansion rate of 8/7 is practical.

F.2.2 Filtering transformation

To transform a binary string of bytes to the level C repertoire:
- subdivide the string in 7-byte substrings, (the last substring has at most 7 bytes)
- add before each substring a control byte with starting value 64 (bit 1 = 1),
- put to 1 in the control byte every bit, with position 0 or 2 to 7, depending on if the filtering transformation is applied or not to the corresponding data byte of the substring,
- verify for every data byte in the substring if transformation is to be applied by:
  is (data byte .and. 64 == 0) ?
- If so put bit 1 to 1 in the data byte and in the position bit of the control byte.
- else keep the data byte and the control byte unchanged.

Notes:
- all filtered values are constrained to have bit 1 of every byte = 1.
- the default service characters are thus excluded from the filter target repertoire.

F.2.3 Defiltering transformation

To transform back the filtered string to the binary string:
- subdivide the string in 8-byte substrings, (the last substring has ar most 8 bytes)
- consider every start byte of each substring as a control byte, the other bytes as data bytes,
- verify bit positions 0 and 2 to 7 of the control byte,
- the corresponding byte positions are respectivily 1 to 7 of the substring,
- if bit = 0, keep the data byte of the corresponding position unchanged,
- if bit = 1, put bit 1 of the corresponding data byte to 0.

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

Service code directory

The service code directory is maintained by the UN/ECE and is part of the UN Trade Data Interchange Directory (UNTDID) and as such is not reproduced in this International Standard. The most recent version of the service code directory should be used to reference the code values for the coded data elements in the simple data element directory (see annex C within this part). The UNTDID is updated and published at regular intervals.

Annex H
(informative)
Security services and algorithms

H.1 Purpose and scope

Annex H gives examples of possible combinations of data elements and code values from the security segment groups. These examples have been chosen to illustrate some widely used security techniques, based on international standards.

The full set of possible combinations is far too large to be presented in this annex. The choices made here must not be considered as an endorsement of the algorithms or modes of operation. The user is invited to choose the techniques appropriate to the security threats he wants to be protected against.

The purpose of this annex is to provide the user, once he has chosen the security techniques, with a comprehensive starting point to work out a suitable solution for his particular application.

For easier reading and understanding the subject has been divided into two paragraphs, each of which concentrates on different basic principles for applying security.

The two sets are:

1. **Combinations using symmetric algorithms and integrated security segments**
2. **Combinations using asymmetric algorithms and integrated security segments**

List of codes used in the matrixes
(subset of the complete code list)

0501	***Security service, coded***
1	Non-repudiation of origin
2	Message origin authentication
3	Integrity

0505	***Filter function, coded***
6	UN/EDIFACT EDC filter

0527	***Algorithm, coded***
1	DES (Data Encryption Standard)
10	RSA (Rivest, Shamir, Adleman)
11	DSA (Digital Signature Algorithm)
16	SHA-1 (Secure Hashing Algorithm)

0523	***Use of algorithm, coded***
1	Owner hashing
2	Owner symmetric
3	Issuer signing (CA)
4	Issuer hashing (CA)
6	Owner signing

0525	***Cryptographic mode of operation, coded***
6	MAC (Message Authentication Code)
7	DIM1 (Data Integrity Mechanism)
9	MDC2 (Modification Detection Code)
11	HDS2 (Hash functions)
16	DSMR (Digital Signature with Message Recovery)

0531	***Algorithm parameter qualifier***
5	Symmetric key encrypted
9	Symmetric key name
10	Key encrypting key name
12	Modulus
13	Exponent
14	Modulus length
25	DSA parameter P
26	DSA parameter Q
27	DSA parameter G
28	DSA parameter Y

0563	*Validation value qualifier*
1	Unique validation value
2	DSA algorithm r parameter
3	DSA algorithm s parameter

0577	*Security party qualifier*
1	Message sender
2	Message receiver
3	Certificate owner
4	Authenticating party

Abbreviations, used

a, b, c, d, e	=	Representations of a Security Reference Number
CA	=	Certification Authority
Enc-Key	=	Encrypted Key
G	=	G public key DSA parameter
Hash	=	Hash value
KEK-N	=	Key encrypting key name
Key-N	=	Key Name
KN	=	Key Name
MAC	=	Message authenticating code
Mod	=	Modulus
Mod-L	=	Length of Modulus
P	=	P public key DSA parameter
PK/CA	=	Public Key of Certification Authority
Pub-K	=	Public Key
Q	=	Q public key DSA parameter
R	=	r parameter result of DSA signature
S	=	s parameter result of DSA signature
Sig	=	Signature
Y	=	Y public key DSA parameter

H.2 Combinations using symmetric algorithms and integrated security segments

The following matrix establishes the relationships for the specific cases of

- integrated message/package/group/interchange level security (ISO 9735-5)
- use of symmetric algorithm only
- security services provided are message origin authentication and content integrity
- message origin authentication is provided by appending a MAC (Message Authentication Code) to the message. Two examples are given, one with DES in CBC mode with a secret key which is known by the message receiver and is only referred to by a key name. This first example complies to ISO 8731-1. The second example is based on usage of DES algorithm according to the mode of operation described in ISO/IEC 9797. The secret key needed is conveyed DES encrypted under a key-encrypting key shared between sender and receiver. This key encrypting key is referred to by its name.
- content integrity is provided by a hash function based on DES algorithm used in MDC mode, according to ISO 10118-2. In this third example there is no secret key to be shared between the sender and the receiver. The hash value is conveyed unprotected, and therefore this security service may not be sufficient to secure the message.
- although sender and receiver share keys, the cryptographic mechanisms have not been completely agreed beforehand. Therefore all the algorithms and mode of operation used are explicitly named.
- only the security fields related to security techniques, algorithms and modes of operation actually used are shown.

TAG	**Name**	**S**	**R**	Message origin authenti. ISO 8731-1	Message origin authenti. ISO 9797	Content Integrity ISO 10118-2	Notes
SG 1		C	99	one per security service			1
USH	SECURITY HEADER	M	1				

0501	SECURITY SERVICE, CODED	M		2	2	3	
0534	SECURITY REFERENCE NUMBER	M		a	b	c	
0505	FILTER FUNCTION, CODED	C		6	6	6	
S500	SECURITY IDENTIFICATION DETAILS	C	2				
0577	Security party qualifier	M		1	1	1	2
0538	Key name	C		Key-N	-	-	3
S500	SECURITY IDENTIFICATION DETAILS	C	2				
0577	Security party qualifier	M		2	2	2	4
USA	SECURITY ALGORITHM	C	3				
S502	SECURITY ALGORITHM	M	1				
0523	Use of algorithm, coded	M		2	2	2	
0525	Cryptographic mode of operation, coded	C		6/*	7/*	9/*	
0527	Algorithm, coded	C		1/*	1/*	1/*	
S503	ALGORITHM PARAMETER	C	9		one for key encryp-ting key name		
0531	Algorithm parameter qualifier	M		-	10	-	5
0554	Algorithm parameter value	M		-	KEK-N	-	
S503	ALGORITHM PARAMETER	C	9		one for encrypted key		
0531	Algorithm parameter qualifier	M		-	5	-	6
0554	Algorithm parameter value	M		-	Enc-Key	-	
Data structures to be secured (user segments/object/message(s)/package(s)/group(s))							
SG n		C	99	one per security service			1
UST	SECURITY TRAILER	M	1				
0534	SECURITY REFERENCE NUMBER	M		a	b	c	
0588	NUMBER OF SECURITY SEGMENTS	M					
USR	SECURITY RESULT	C	1				
S508	VALIDATION RESULT	M	2				7
0563	Validation value qualifier	M		1	1	1	
0560	Validation value	C		MAC	MAC	Hash	8

Table H.1 - Matrix of relationship when only symmetric algorithms are used

Notes:
1. both structures must have the same occurrence number
2. message sender

3. name of the secret key shared by sender and receiver
4. message receiver
5. the key encrypting key is already shared by sender and receiver. It is referred here by its name
6. the secret key is conveyed DES encrypted with the key encrypted key
7. some signature algorithms (like DSA) require 2 result parameters
8. the result values for "integrity" are unprotected and may need to be submitted separately

* further code combinations are possible and required

H.3 Combinations using asymmetric keys and integrated security segments

The following matrix establishes the relationships for the specific cases of

- integrated message/package/group/interchange level security (ISO 9735-5)
- the security service provided is non-repudiation of origin, two methods are presented with different techniques of signature computation
- two asymmetric algorithms are presented: RSA and DSA
- two hash-functions are chosen: DES in MDC mode together with RSA, and SHA-1 together with DSA
- certificates are assumed to not have been exchanged previously
- the USC segment contains explicitly the identification of the hash function and the signature function used by the Certification Authority to sign the certificate. The public key of Certification Authority, needed to check the certificate signature is already known by the receiver. It is referred to by name in the USC segment.
- only one certificate is included, a second one would be necessary, only if a public key of the recipient were used

TAG	**Name**	**S**	**R**	Non-repudiation of origin (RSA)	Non-repudiation of origin (DSA)	Notes
SG 1		C	99	one per security service		1
USH	SECURITY HEADER	M	1			
0501	SECURITY SERVICE, CODED	M		1	1	2
0534	SECURITY REFERENCE NUMBER	M		d	e	
0505	FILTER FUNCTION, CODED	C		6	6	
S500	SECURITY IDENTIFICATION DETAILS	C	2			
0577	Security party qualifier	M		1	1	3
S500	SECURITY IDENTIFICATION DETAILS	C	2			
0577	Security party qualifier	M		2	2	4
USA	SECURITY ALGORITHM	C	3			
S502	SECURITY ALGORITHM	M	1			
0523	Use of algorithm, coded	M		1	1	5
0525	Cryptographic mode of operation, coded	C		11/*	-	
0527	Algorithm, coded	C		1/*	16	
SG 2		C	2	only one: sender certificate		
USC		M	1			
0536	CERTIFICATE REFERENCE	C	1	reference of this certificate		
S500	SECURITY IDENTIFICATION DETAILS	C	2	(certificate owner)		
0577	Security party qualifier	M		3	3	6
S500	SECURITY IDENTIFICATION DETAILS	C	2	(authenticating party)		
0577	Security party qualifier	M		4	4	7
0538	Key name	C		(PK/CA name)	(PK/CA name)	

USA	SECURITY ALGORITHM	C	3	(sender's signature function)		
S502	SECURITY ALGORITHM	M	1			
0523	Use of algorithm, coded	M		6	6	8
0525	Cryptographic mode of operation, coded	C		16	-	
0527	Algorithm, coded	C		10	11	
S503	ALGORITHM PARAMETER	C	9	(length of modulus)	DSA parameter P	
0531	Algorithm parameter qualifier	M		14	25	
0554	Algorithm parameter value	M		Mod-L	P	
S503	ALGORITHM PARAMETER	C	9	(modulus)	DSA parameter Q	
0531	Algorithm parameter qualifier	M		12	26	
0554	Algorithm parameter value	M		Mod	Q	
S503	ALGORITHM PARAMETER	C	9	(public exponent)	DSA parameter G	
0531	Algorithm parameter qualifier	M		13	27	
0554	Algorithm parameter value	M		Pub-K	G	
S503	ALGORITHM PARAMETER	C	9	-	DSA parameter Y	
0531	Algorithm parameter qualifier	M		-	28	
0554	Algorithm parameter value	M		-	Y	
USA	SECURITY ALGORITHM	C	3	(CA's hash function for certificate's signature)		
S502	SECURITY ALGORITHM	M	1			
0523	Use of algorithm, coded	M		4	4	9

Table H.2 - Matrix of relationship when asymmetric algorithms are used

TAG	**Name**	**S**	**R**	Non-repudiation of origin (RSA)	Non-repudiation of origin (DSA)	Notes
0525	Cryptographic mode of operation, coded	C		11	-	
0527	Algorithm, coded	C		1	8	
USA	SECURITY ALGORITHM	C	3	(CA's signature function for certificate's signature)		
S502	SECURITY ALGORITHM	M	1			
0523	Use of algorithm, coded	M		3	3	10
0525	Cryptographic mode of operation, coded	C		16	-	
0527	Algorithm, coded	C		10	11	
USR	SECURITY RESULT	C	1			
S508	VALIDATION RESULT	M	2			11
0563	Validation value qualifier	M		1	2	
0560	Validation value	C		Sig	R	
S508	VALIDATION RESULT	M	2			11
0563	Validation value qualifier	M		-	3	
0560	Validation value	C		-	S	
Data structures to be secured (user segments/object/message(s)/package(s)/group(s))						
SG n		C	99	one per security service		1
UST	SECURITY TRAILER	M	1			
0534	SECURITY REFERENCE NUMBER	M		d	e	
0588	NUMBER OF SECURITY SEGMENTS	M				
USR	SECURITY RESULT	C	1			

S508	VALIDATION RESULT	M	2			11
0563	Validation value qualifier	M		1	2	
0560	Validation value	C		Sig	R	
S508	VALIDATION RESULT	M	2			11
0563	Validation value qualifier	M		-	3	
0560	Validation value	C		-	S	

Table H.2 - Matrix of relationship when asymmetric algorithms are used *(concluded)*

Notes:

1. both structures must have the same occurrence number
2. Message origin authentication and Integrity are assumed to be included in the Non-repudiation of origin
3. message sender
4. message receiver
5. hash function applied by the sender on the secured structure
6. certificate owner: identification details should be the same as in USH S500 for the message sender
7. authenticating party: Certification Authority (CA)
8. sender's signature function
9. CA's hash function
10. CA's signature function
11. some signature algorithms (for instance DSA) require 2 result parameters

* further code combinations are possible and required

Annex I
(informative)

Bibliography

[1] ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange.*

[2] ISO 8601:1988, *Data elements and interchange formats — Information interchange — Representation of dates and times.*

[3] ISO 8731-1:1987, *Banking — Approved algorithms for message authentication — Part 1: DEA.*

[4] ISO/IEC 9797:1994, *Information technology — Security techniques — Data integrity mechanism using a cryptographic check function employing a block cipher algorithm.*

[5] ISO/IEC 10116:1996, *Information technology — Security techniques — Modes of operation for an* n*-bit block cipher.*

[6] ISO/IEC 10118-2:1994, *Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an* n*-bit block cipher algorithm.*

[7] ISO/IEC 10181-1:1996, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Overview.*

[8] ISO/IEC 10646-1:1993, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.*

[9] ISO/IEC 11770-1:1996, *Information technology — Security techniques — Key management — Part 1: Framework.*