INTERNATIONAL STANDARD

ISO 9735-5

Second edition
2002-07-01

Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) —

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

Échange de données informatisé pour l'administration, le commerce et le transport (EDIFACT) — Règles de syntaxe au niveau de l'application (Numéro de version de syntaxe: 4, numéro d'édition de syntaxe: 1) —

Partie 5: Règles de sécurité pour EDI par lots (authenticité, intégrité et non-répudiation de l'origine)

Reference number
ISO 9735-5:2002(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Page

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 9735 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 9735-5 was prepared by Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration* in collaboration with UN/CEFACT through the Joint Syntax Working Group (JSWG).

This second edition cancels and replaces the first edition (ISO 9735-5:1999). However ISO 9735:1988 and its Amendment 1:1992 are provisionally retained for the reasons given in clause 2.

Furthermore, for maintenance reasons the Syntax service directories have been removed from this and all other parts of the ISO 9735 series. They are now consolidated in a new part, ISO 9735-10.

At the time of publication of ISO 9735-1:1998, ISO 9735-10 had been allocated as a part for “Security rules for interactive EDI”. This was subsequently withdrawn because of lack of user support, and as a result, all relevant references to the title “Security rules for interactive EDI” were removed in this second edition of ISO 9735-5.

Definitions from all parts of the ISO 9735 series have been consolidated and included in ISO 9735-1.

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, Syntax release number: 1)*:

— *Part 1: Syntax rules common to all 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)*

— *Part 10: Syntax service directories*

Further parts may be added in the future.

Annexes A to E of this part of ISO 9735 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, Syntax release number: 1) —

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

Whereas this part shall use a version number of "4" in the mandatory data element 0002 (Syntax version number), and shall use a release number of "01" in the conditional data element 0076 (Syntax release number), each of which appear 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 and its Amendment 1:1992 — *Syntax version number: 3*

— ISO 9735:1998 — *Syntax version number: 4*

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 this part of ISO 9735 shall include conformance to parts 1, 2, 8 and 10 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 normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 9735. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 9735 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 9735-1:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 1: Syntax rules common to all parts*

ISO 9735-2:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 2: Syntax rules specific to batch EDI*

ISO 9735-6:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 6: Secure authentication and acknowledgement message (message type — AUTACK)*

ISO 9735-7:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 7: Security rules for batch EDI (confidentiality)*

ISO 9735-8:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 8: Associated data in EDI*

ISO 9735-10:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 10: Syntax service directories*

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 Terms and definitions

For the purposes of this part of ISO 9735, the terms and definitions given in ISO 9735-1 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

5.1.1 General

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

This subclause 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.

5.1.2 Security header and trailer segment groups

Figure 1 describes an interchange showing security at message level.

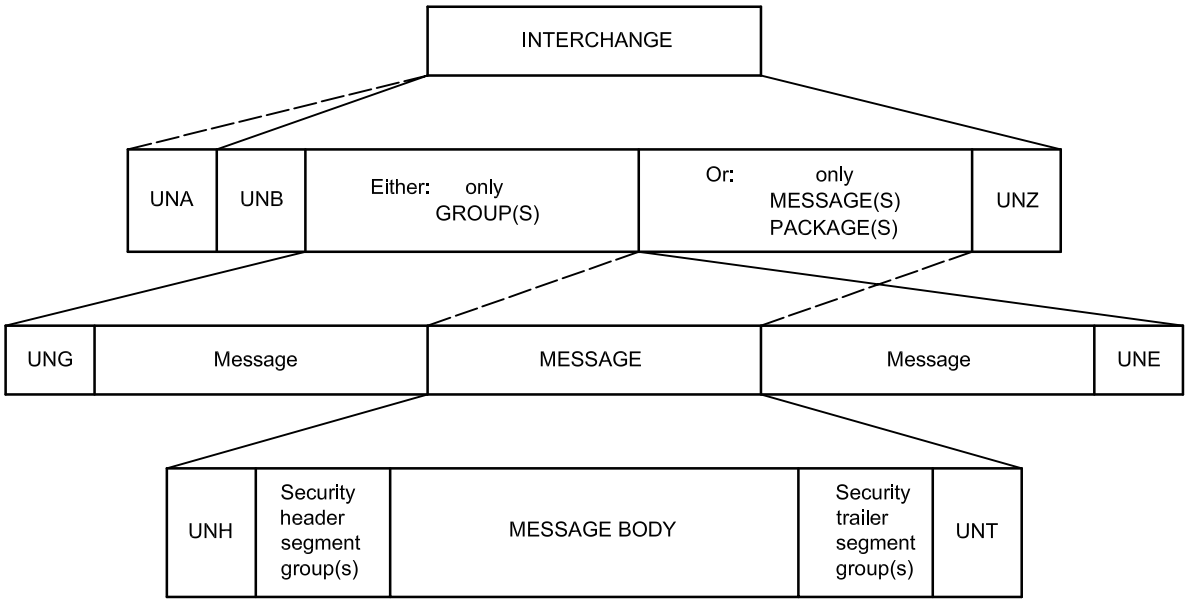

Figure 1 — Interchange showing security at message level (schematic)

Figure 2 describes an interchange showing security at package level.

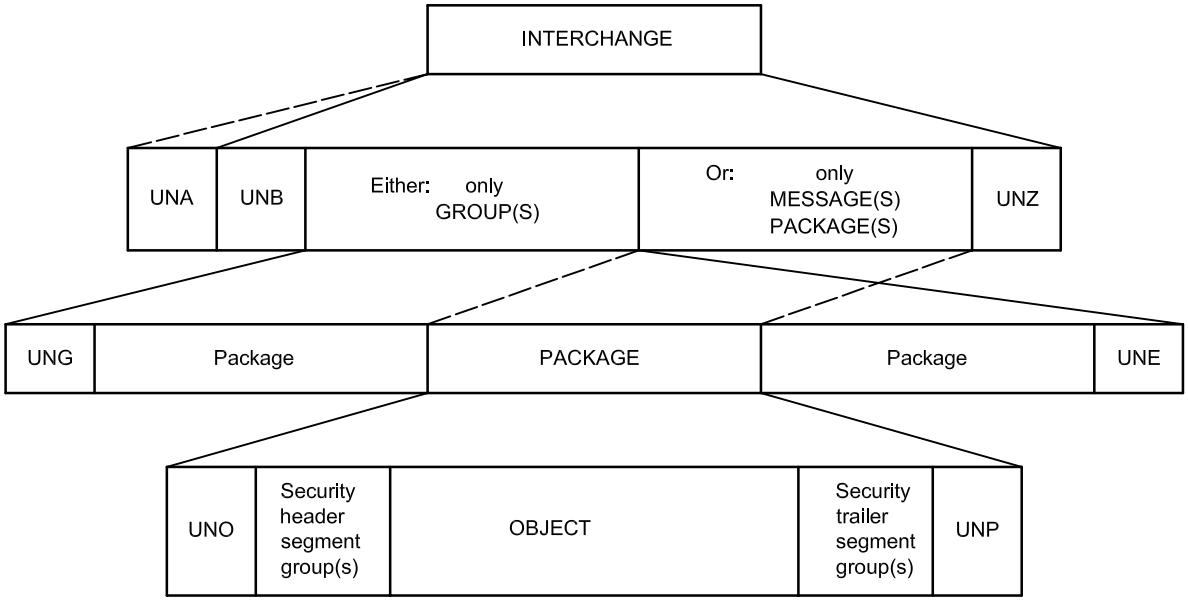

Figure 2 — Interchange showing security at package level (schematic)

5.1.3 Security header and trailer segment groups structure

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

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 2 — Security header and security trailer segment groups segment table (package 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	

NOTE The complete directory specification of the segments and data elements, including segments UNH message header, UNT message trailer, UNO object header and UNP object trailer are specified in ISO 9735-10. They are not described further in this part of ISO 9735.

5.1.4 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 ISO 9735-7.

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 contained in 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.

5.1.5 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).

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).

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/IEC 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 subproduct 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/IEC 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 subproduct 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/IEC 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 A and B, are also valid at group and interchange level.

The techniques described in the previous clause 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.

6.2 Security header and trailer segment groups

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

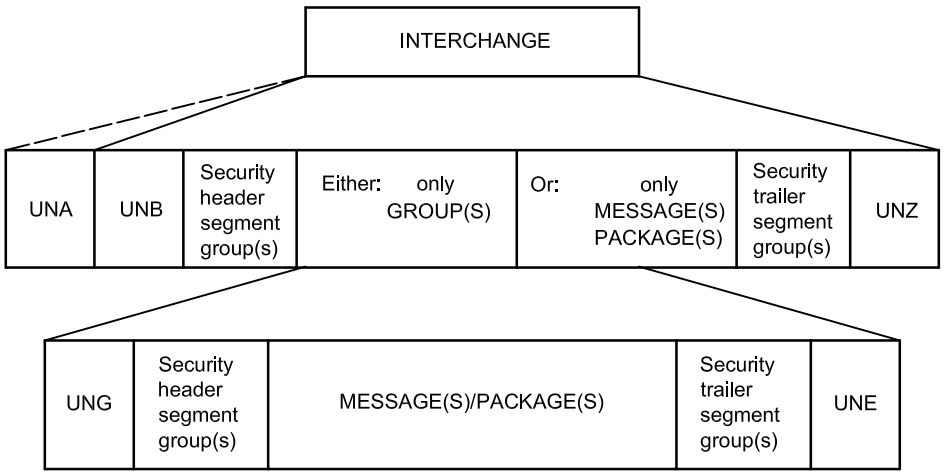

Figure 5 — Interchange showing security at both interchange and group levels (schematic)

6.3 Security header and trailer segment groups structure

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

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 4 — Security header and security trailer segment groups segment table (group 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	

NOTE The complete directory specification of the segments and data elements, including segments UNB interchange header, UNZ interchange trailer, UNG group header and UNE group trailer are specified in ISO 9735-10. They are not described further in this part of ISO 9735.

6.4 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 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).

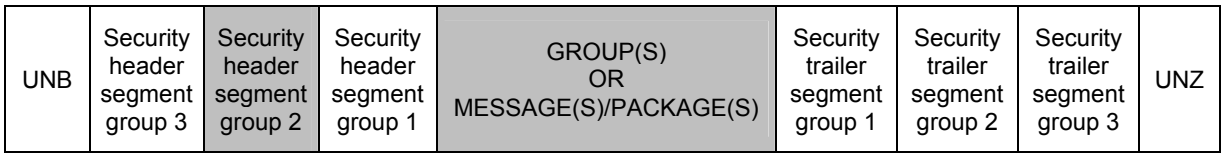

Figure 6 — Scope of application: security header segment group and group(s) or message(s)/package(s) only (schematic)

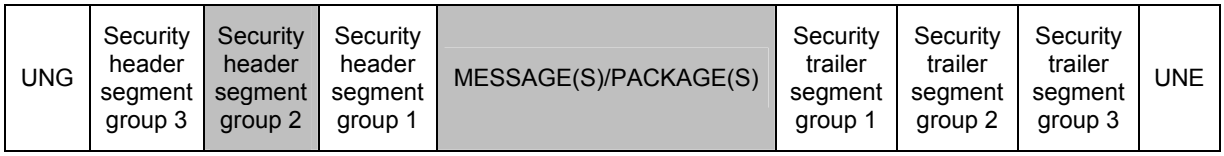

Figure 7 — Scope of application: security header segment group and message(s)/package(s) only (schematic)

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).

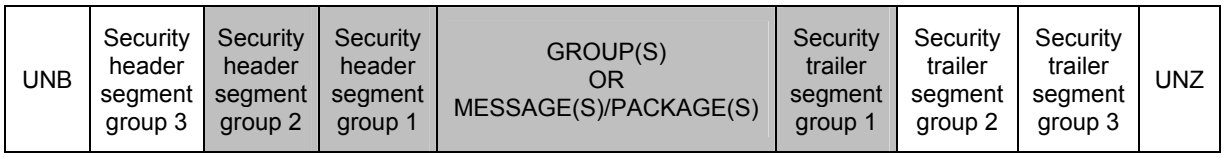

Figure 8 — Scope of application: from security header segment group to security trailer segment group (schematic)

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
(informative)

EDIFACT security threats and solutions

A.1 Introduction

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.

A.2 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.

A.3 Security solutions — Basic services and principles of usage

A.3.1 General

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.

A.3.2 Sequence integrity

Sequence integrity protects against the duplication, addition, deletion, loss or replay of an 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.

A.3.3 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 subproduct of origin authentication or non-repudiation of origin.

A.3.4 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 subproduct of non-repudiation of origin.

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

A.3.5 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.

A.3.6 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.

A.3.7 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 ISO 9735-7.

A.3.8 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.

Table A.1 summarizes these interrelations.

Table A.1 — Interrelation table

Use of	Also implies		
	Content integrity	**Origin authentication**	**Non-repudiation of origin**
Content integrity	yes	—	—
Origin authentication	yes	yes	—
Non-repudiation of origin	yes	yes	yes

Annex B
(informative)

How to protect an EDIFACT structure

B.1 General

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 A and to 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 Table A.1.

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.

B.2 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.

B.3 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.

B.4 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.

B.5 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.

B.6 Separated message security at message/package level

B.6.1 Business requirements

There are two business requirements for this feature, namely

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

b) 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 ISO 9735-6.

B.6.2 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.

B.6.3 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 ISO 9735-6.

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.

B.7 Separated message security at group or interchange levels

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

The two business requirements for this feature, are:

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

b) 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 ISO 9735-6.

Annex C
(informative)

Message protection examples

C.1 Introduction

Three examples are provided in this annex 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.

C.2 Example 1: message origin authentication

C.2.1 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.

C.2.2 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)

C.3 Example 2: non-repudiation of origin, first technique

C.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.

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.

C.3.2 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 10118-2:2000; transformation u and u' as specified in annex A of ISO/IEC 10118-2:2000. DES block cipher algorithm is used.
CERTIFICATE	Certificate of Mr. SMITH
CERTIFICATE REFERENCE	This certificate is referenced, by AUTHORITY: 00000001.
SECURITY IDENTIFICATION DETAILS Security party qualifier	 Certificate owner (Mr. SMITH of Company A)
SECURITY IDENTIFICATION DETAILS Security party qualifier Key name	 Authenticating party (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
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	An issuer hashing algorithm is used.
Cryptographic mode of operation	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 10118-2:2000; transformation u and u' as specified in annex A of ISO/IEC 10118-2:2000.
Algorithm	DES block cipher algorithm is used.
SECURITY ALGORITHM	
Use of algorithm	An issuer signing algorithm is used.
Cryptographic mode of operation	No mode of operation is relevant here.
Algorithm	RSA is the asymmetric algorithm.
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a public exponent for signature verification.
Algorithm parameter value	AUTHORITY's public key
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a Modulus for signature verification.
Algorithm parameter value	AUTHORITY's modulus
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as the length of AUTHORITY's modulus (in bits).
Algorithm parameter value	AUTHORITY's modulus is 512 bits long.
SECURITY RESULT	Digital signature of the certificate
VALIDATION RESULT	
Validation value qualifier	Unique validation value is 1.
Validation value	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	Unique validation value is 1.
Validation value	512 Bit digital signature

C.4 Example 3: non-repudiation of origin, second technique

C.4.1 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.

C.4.2 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 Security party qualifier	Message sender (party securing the message: Mr. SMITH of Company A)
SECURITY IDENTIFICATION DETAILS Security party qualifier	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.
ALGORITHM PARAMETER Algorithm parameter qualifier Algorithm parameter value	Identifies the following algorithm parameter value as a clear text initialization value. Clear text initialization value (all binary 0's).
CERTIFICATE	Certificate of Mr. SMITH (message sender)
CERTIFICATE REFERENCE	This certificate is referenced: 00000001, by AUTHORITY.
SECURITY IDENTIFICATION DETAILS Security party qualifier	Certificate owner (Mr. SMITH of Company A)
SECURITY IDENTIFICATION DETAILS Security party qualifier Key name	Authenticating party (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 used when signature was computed.
Service character for signature qualifier	Service character is segment terminator.
Service character for signature	Value “ ' ” (apostrophe).
SERVICE CHARACTER FOR SIGNATURE	Service character used when signature was computed.
Service character for signature qualifier	Service character is data element separator.
Service character for signature	Value “ + ” (plus sign).
SERVICE CHARACTER FOR SIGNATURE	Service character used when signature was computed.
Service character for signature qualifier	Service character is component data element separator.
Service character for signature	Value “ : ” (colon).
SERVICE CHARACTER FOR SIGNATURE	Service character used when signature was computed.
Service character for signature qualifier	Service character is repetition separator.
Service character for signature	Value “ * ” (asterisk).
SERVICE CHARACTER FOR SIGNATURE	Service character used when signature was computed.
Service character for signature qualifier	Service character is release character.
Service character for signature	Value “ ? ” (question mark).
SECURITY DATE AND TIME	Certificate generation time
Date and time	Mr. SMITH certificate was generated on 931215 at 14:12:00.
SECURITY DATE AND TIME	Certificate start of validity period
Date and time	Validity period of Mr. SMITH's starts: 1996 01 01 000000.
SECURITY DATE AND TIME	Certificate end of validity period
Date and time	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	An owner signing algorithm is used.
Cryptographic mode of operation	No mode of operation is relevant here.
Algorithm	RSA is the asymmetric algorithm.
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a public exponent for signature verification.
Algorithm parameter value	Mr. SMITH's public key
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a modulus for signature verification.
Algorithm parameter value	Mr. SMITH's modulus
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as the length of Mr SMITH's modulus (in bits).
Algorithm parameter value	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	An issuer hashing algorithm is used.
Cryptographic mode of operation	Square-mod n hash function for RSA; Annex D, CCITT X509. ISO/IEC 9594-8
Algorithm	RSA asymmetric algorithm
SECURITY ALGORITHM	Asymmetric algorithm is used by AUTHORITY to sign.
SECURITY ALGORITHM	
Use of algorithm	An issuer signing algorithm is used.
Cryptographic mode of operation	No mode of operation is relevant here.
Algorithm	RSA is the asymmetric algorithm.
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a public exponent for signature verification.
Algorithm parameter value	AUTHORITY's public key
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as a modulus for signature verification.
Algorithm parameter value	AUTHORITY's modulus
ALGORITHM PARAMETER	
Algorithm parameter qualifier	Identifies this algorithm parameter as the length of AUTHORITY's modulus (in bits).
Algorithm parameter value	AUTHORITY's modulus is 512 bit long.
SECURITY RESULT	Digital signature of the certificate
VALIDATION RESULT	
Validation value qualifier	Unique validation value is 1.
Validation value	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	Unique validation value is 1.
Validation value	512 Bit digital signature

Annex D
(informative)

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

D.1 EDA filter

D.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.

D.1.2 UN/EDIFACT character set repertoires

The character set repertoire A possesses 44 characters whose use is unrestricted. In addition to those 44, four service characters and eight 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 three non-printable service characters.

D.1.3 two by three filtering

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

D.1.4 EDA filter specification

Having 44 allowed characters, to avoid using the space character part of those 44 and filter every pair of input characters (if odd, filter only the last character in two 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 three numbers (two 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.

D.1.5 Defiltering

To defilter: map each of the 43 characters back to its value between 0 and 42,

if at least three filtered characters remain, compute: cl * 1849 + c2 *43 + c3 = short integer

else at least two 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 two characters of the short integer result.

D.2 EDC filter

D.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.

D.2.2 Filtering transformation

To transform a binary string of bytes to the level C repertoire:

— subdivide the string in seven-byte substrings (the last substring has at most seven 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.

D.2.3 Defiltering transformation

To transform back the filtered string to the binary string:

— subdivide the string in eight-byte substrings (the last substring has at most eight 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 respectively 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 E
(informative)

Security services and algorithms

E.1 Purpose and scope

This annex 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***	***0505***	***Filter function, coded***
1	Non-repudiation of origin	6	UN/EDIFACT EDC filter
2	Message origin authentication		
3	Integrity		

0523	***Use of algorithm, coded***	***0525***	***Cryptographic mode of operation, coded***
1	Owner hashing	16	DSMR (Digital Signature with Message Recovery)
2	Owner symmetric		
3	Issuer signing (CA)		
4	Issuer hashing (CA)		
6	Owner signing		

0527	*Algorithm, coded*
1	DES (Data Encryption Standard)
8	SHA (Secure Hashing Algorithm)
10	RSA (Rivest, Shamir, Adleman)
11	DSA (Digital Signature Algorithm)
16	SHA1 (Secure Hashing Algorithm)
37	MAC (Message Authentication Code)
38	DIM1 (Data Integrity Mechanism)
40	MDC2 (Modification Detection Code)
42	HDS2 (Hash functions)

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

E.2 Combinations using symmetric algorithms and integrated security segments

The matrix given in Table E.1 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.

Table E.1 — Matrix of relationships when only symmetric algorithms are used

TAG	Name	S	R	Message origin authentication ISO 8731-1	Message origin authentication ISO 9797	Content Integrity ISO/IEC 10118-2	Notes
SG 1		C	99	one per security service			1
USH	SECURITY HEADER	M	1				
0501	SECURITY SERVICE, CODED	M	1	2	2	3	
0534	SECURITY REFERENCE NUMBER	M	1	a	b	c	
0505	FILTER FUNCTION, CODED	C	1	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		—	—	—	

TAG	Name	S	R	Message origin authentication ISO 8731-1	Message origin authentication ISO 9797	Content Integrity ISO/IEC 10118-2	Notes
0527	Algorithm, coded	C		37	38	40	
S503	ALGORITHM PARAMETER	C	9		one for key encrypting 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	1	a	b	c	
0588	NUMBER OF SECURITY SEGMENTS	M	1				
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

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 two result parameters.
8. The result values for “integrity” are unprotected and may need to be submitted separately.

E.3 Combinations using asymmetric keys and integrated security segments

The matrix given in Table E.2 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.

Table E.2 — Matrix of relationships when asymmetric algorithms are 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	1	2
0534	SECURITY REFERENCE NUMBER	M	1	d	e	
0505	FILTER FUNCTION, CODED	C	1	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		—	—	
0527	Algorithm, coded	C		42	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	

TAG	Name	S	R	Non-repudiation of origin (RSA)	Non-repudiation of origin (DSA)	Notes
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
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	1	d	e	
0588	NUMBER OF SECURITY SEGMENTS	M	1			
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

TAG	Name	S	R	Non-repudiation of origin (RSA)	Non-repudiation of origin (DSA)	Notes
0563	Validation value qualifier	M		—	3	
0560	Validation value	C		—	S	

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 two result parameters.

Bibliography

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

[2] ISO 8601:2000, *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 10116:1997, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

[6] ISO/IEC 10118-2:2000, *Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher*

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

[8] ISO/IEC 10646-1:2000, *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*

ICS 35.240.60

Price based on 38 pages