INTERNATIONAL STANDARD

ISO 11166-1

First edition
1994-11-15

Banking — Key management by means of asymmetric algorithms —

Part 1:
Principles, procedures and formats

Banque — Gestion des clés au moyen d'algorithmes asymétriques —
Partie 1: Principes, procédures et formats

Reference number
ISO 11166-1:1994(E)

Contents

International Organization for Standardization
Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Annexes

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.

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

International Standard ISO 11166-1 was prepared by Technical Committee ISO/TC 68, *Banking and relating financial services*, Subcommittee 2, *Operations and procedures*.

This part of ISO 11166 is related to ISO 8732, *Banking — Key management (wholesale)* from which it was developed.

ISO 11166 consists of the following parts, under the general title *Banking — Key management by means of asymmetric algorithms:*

— Part 1: Principles, procedures and formats

— Part 2: Approved algorithms using the RSA cryptosystem.

Annexes A to F of this part of ISO 11166 are for information only.

Introduction

This part of ISO 11166 describes procedures for the secure management of keying material used to protect messages in a wholesale banking environment.

Key management is the process whereby cryptographic keys and initialisation vectors (keying material) are provided for use by two parties and continue to be subject to secure handling procedures until they have been destroyed. The security of the data enciphered or authenticated by means of keying material is dependent upon the prevention of unauthorised disclosure, modification, substitution, insertion or deletion of keys or initialisation vectors (IVs). If they are compromised, the security of the related data can no longer be ensured. Thus, key management is concerned with the generation, distribution, storage, custody, monitoring, destruction, and back-up procedures for keying material. Also, by the formalisation of such procedures provision is made for audit trails to be established.

Automated key distribution is the means by which two communicating parties are provided with identical symmetric keying material (cryptographic keys and, where needed, IVs) by using a defined procedure via a communication channel. Automated key distribution in this standard employs two types of keys :

1) Keys for asymmetric algorithms. These are used to encipher and decipher data keys and also to produce and verify digital signatures for the authentication of data keys or messages containing these keys.

2) Keys for symmetric algorithms. These are the data keys which are the subject of key management. They are also used to provide integrity check values (MACs) on some messages exchanged for key management.

The level of security to be achieved needs to be related to a number of factors, including the sensitivity of the data concerned and the likelihood that it will be intercepted, the practicality of any envisaged encipherment process, and the cost of providing, and breaking, a particular means of providing security. It is therefore necessary for each communicating pair to agree about the extent and detail of security and key management procedures. Absolute security is not practically achievable, so key management procedures need to aim to reduce the opportunity for a breach of security and also to aim for a "high" probability of detection of any illicit access or change to keying material that may occur despite any preventative measures. This applies at all stages of the generation, and use of keying material, including those processes that occur in cryptographic equipment and those related to communication of cryptographic keys and initialisation vectors between communicating pairs or key centres. Thus, while wherever possible this International Standard has specified requirements in absolute terms, in some instances a level of judgement cannot be practically avoided. For instance, defining the frequency of key change is beyond the scope of this standard, and will be dependent upon the degree of risk associated with the factors listed above.

Part 1 of this International Standard has been divided into sections, as follows :

1. General
2. Key certification
3. Automatic distribution of keying material
4. Cryptographic Service Messages.

The final details of the key management procedures need to be agreed between the communicating pair(s) concerned and will thus remain the responsibility of the communicating pair(s). An aspect of the details to be agreed upon will be the identity and duties of particular individuals. This International Standard does not concern itself with allocation of individual responsibilities as this needs to be considered uniquely for each key management implementation.

Annex A gives an overview of the principles employed.

Banking — Key management by means of asymmetric algorithms —

Part 1:
Principles, procedures and formats

Section 1: General

1.1 Scope

This part of ISO 11166 specifies methods for the management of keying material used for the encipherment, decipherment and authentication of messages exchanged in the course of wholesale financial transactions only. It specifies requirements for:

i) the control of keying material during its life to prevent unauthorised disclosure, modification, substitution, and replay;

ii) the manual or automatic distribution of keying material to permit interoperability between cryptographic equipment or facilities using the same algorithms;

iii) ensuring the integrity of keying material during all phases of its life, including its generation, distribution, storage, entry, use, archival and destruction;

iv) recovery in the event of failure of the key distribution process.

This part of ISO 11166 provides a means whereby an audit log can be identified for all keying material.

This International Standard is designed for the management of keying material for use with symmetric algorithms. The key distribution is performed using an asymmetric algorithm or algorithms. It is designed for messages formatted and transmitted in coded character sets.

This International Standard defines two alternative methods for two communicating parties to obtain identical symmetric keying material which they can then use to protect communications.

One method is *key transport*, in which one party generates keying material and sends it to the other party, suitably protected by using asymmetric algorithms.

The second method is *key exchange*, in which each party of the communicating pair sends the other a block of data and using the received block each party derives keying material in such a way that the sets of keying material thus derived by the two parties are identical.

NOTE — A well known example of key exchange is the Diffie-Hellman key exchange method.

In this International Standard, the term *key distribution* will encompass both key transport and key exchange. This part of ISO 11166 includes key transport but does not preclude key exchange. Specific key exchange procedures may be the subject of subsequent parts of this International Standard.

The procedures specified are appropriate for use by financial institutions and by their corporate and government customers, and in other relationships where the interchange of information requires confidentiality, protection and authentication.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 11166. 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 11166 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/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange.*

ISO 8730:1990, *Banking — Requirements for message authentication (wholesale).*

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

ISO 8731-2:1992, *Banking — Approved algorithms for message authentication: — part 2: Message authenticator algorithm.*

1.3 Definitions

For the purposes of this part of ISO 11166, the following definitions apply.

1.3.1 asymmetric algorithm: A cryptographic algorithm employing a *public key* and a *secret key*. Together these form an asymmetric key set.

1.3.2 audit log: See *security audit log*.

1.3.3 authentication: A process used, between a sender and a receiver, to ensure *data integrity* and to provide *data origin authentication*.

1.3.4 certificate: The public key of a user, together with some other information, rendered unforgeable by a signature with the secret key of the certification authority which issued it.

1.3.5 certification authority: An authority trusted by all users to create and sign certificates.

1.3.6 ciphertext: Enciphered information.

1.3.7 code: A symbol representing data, typically to facilitate automated processing.

1.3.8 communicating pair: Two *logical parties* who have previously agreed to exchange data.

NOTE — A party and a Key Certification Centre do not constitute a communicating pair.

1.3.9 Co-ordinated Universal Time: The time scale maintained by the Bureau International de l'Heure (International Time Bureau) that forms the basis of a co-ordinated dissemination of standard frequencies and time signals.

NOTE — May alternatively be described as Greenwich Mean Time (GMT).

1.3.10 counter: An incrementing count used between two parties to control their key distributions.

1.3.11 cryptographic equipment: Equipment in which cryptographic functions (e.g. *encipherment, authentication, key generation*) are performed.

1.3.12 cryptographic key: A parameter used in conjunction with an algorithm for the purpose of *validation, authentication, encipherment, decipherment, signature construction or deconstruction.*

1.3.13 cryptographic keying material: See *keying material*.

1.3.14 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 unauthorised use.

NOTE — Cryptography determines the methods used in encipherment and decipherment. An attack on a cryptographic principle, means or method is cryptanalysis.

1.3.15 cryptoperiod: A defined period of time during which a specific *cryptographic key* is authorised for use, or during which time the *cryptographic keys* for a given system may remain in effect.

1.3.16 data integrity: The property that data has not been altered or destroyed in an unauthorised manner.

1.3.17 data key: A *cryptographic key* used for the *encipherment, decipherment* or *authentication* of data.

1.3.18 data origin authentication: The corroboration that the source of data received is as claimed.

1.3.19 decipherment: The reversal of a corresponding reversible *encipherment*.

1.3.20 decryption: See *decipherment*.

1.3.21 digital signature: A value used to validate the source and content of part or all of the text of a message. For the purposes of this part of ISO 11166, it is generated and verified using an asymmetric algorithm. In some applications, it can also provide a non-repudiation service.

1.3.22 dual control: A process of utilising two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information whereby no single entity is able to access or utilise the materials, e.g. *cryptographic key*.

1.3.23 encipherment: The cryptographic transformation of data (see *cryptography*) to produce *ciphertext*.

1.3.24 encipherment public key: The public key which may be used to encipher data keys.

1.3.25 encryption: See *encipherment*.

1.3.26 exclusive-or: See *modulo-2 addition*.

1.3.27 field tag: A unique string of characters used in formatted messages that identifies the meaning and location of the associated data field.

1.3.28 hexadecimal digit: A single character in the range 0-9, A-F (upper case), representing a four bit string.

1.3.29 initialisation vector (IV): A number used as a starting point for *encipherment* of a data sequence. It increases security, by introducing additional cryptographic variance, and also facilitates the synchronisation of *cryptographic equipment*.

1.3.30 interoperability: The ability to exchange *cryptographic keys*, whether manually or in an automated environment, with any other party.

1.3.31 key: See *cryptographic key*.

1.3.32 key component: One of at least two parameters having the format of a *cryptographic key* that is combined with one or more like parameters by means of *modulo-2 addition* to form a *cryptographic key*.

1.3.33 Key Certification Centre: A facility operated by the certification authority which generates and returns certificates.

1.3.34 key generator: A type of *cryptographic equipment* used for generating *cryptographic keys* and, where needed, *initialisation vectors*.

1.3.35 key loader: An electronic, self-contained unit which is capable of storing at least one *cryptographic key* and transferring that *cryptographic key*, upon request, into *cryptographic equipment*.

1.3.36 key management facility: A protected enclosure (e.g. room or *cryptographic equipment*) and its contents where cryptographic elements reside.

1.3.37 keying material; cryptographic keying material: The data (e.g. keys and IVs) necessary to establish and maintain a *keying relationship*.

1.3.38 keying relationship: The state existing between a *communicating pair* during which time they share at least one *data key*.

1.3.39 logical party: One or more physical parties forming one member of a *communicating pair*.

1.3.40 Message Authentication Code (MAC): A field in a message between a sender and a receiver used to validate the source and content of part or all of the text of a message. A MAC is generated and verified using a symmetric algorithm.

1.3.41 modulo-2 addition; exclusive-or: A binary addition with no carry, giving the following values:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$

1.3.42 originator: The party (logical or other) that is responsible for originating a Cryptographic Service Message.

1.3.43 plaintext: Unenciphered information.

1.3.44 public key: That part of an asymmetric key set which is known to other parties than the originator of the key set.

1.3.45 recipient: The party (logical or other) that is responsible for receiving a Cryptographic Service Message.

1.3.46 secret key: (in an asymmetric or public key cryptosystem) That part of an asymmetric key set which is kept secret by the originator of the key set.

1.3.47 security audit: An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures and to recommend any indicated changes in control, policy and procedures.

1.3.48 security audit log: Log containing data collected and potentially used to facilitate a *security audit.*

1.3.49 signature construction: The cryptographic transformation of data (See *cryptography*) to ensure data origin authentication.

1.3.50 signature deconstruction: The reversal of *signature construction* to produce data from which it is possible to determine whether the signature is valid or not.

1.3.51 signature public key: A public key which may be used to deconstruct signatures.

1.3.52 symmetric algorithm: a cryptographic algorithm employing the same value of key for both enciphering and deciphering or for both authentication and validation.

1.3.53 validation: The process of checking the *data integrity* of a message, or selected parts of a message.

1.3.54 zeroisation: A method of erasing or overwriting electronically stored data.

1.4 Abbreviations and notation

1.4.1 Abbreviations

The following abbreviations are used in this part of ISO 11166.

Abbreviation	Meaning	Description (see also table 2)
BC	Block	The elementary unit of data that an asymmetric algorithm can encipher.
BE	Enciphered Block	A field that holds keys enciphered by an asymmetric master key.
BES	Enciphered and Signed Block	A field that holds keys enciphered by an asymmetric master key and signed by another asymmetric master key.
CKC	Key Certification Centre	A facility operated by the certification authority which generates and returns certificates.
CSM	Cryptographic Service Message	A message for the establishment of keys or for carrying related information used to control a keying relationship.
CTP	Counter P	Counter used in a keying relationship.
CTR	Counter R	The received value of the counter found to be in error.
CV	Certificate Value	The value of a certificate (certified public key) exchanged between two users.

CVO	Certificate Value Only	Indicates that only a Certificate Value is requested by a Cryptographic Service Message.
DEA	Data Encryption Algorithm	The symmetric block encipher algorithm specified in ANSI X3.92.
DSM	Disconnect Service Message	A message type used to discontinue one or more keys or to terminate a keying relationship.
ECB	Electronic Code Book	A mode of implementing an encipherment algorithm.
EDC	Error Detection Code	A field in a Cryptographic Service Message used to validate the data integrity of the message.
EDCV	Effective Data of a Certificate Value	Date and Coordinated Universal Time before which a certificate is not valid.
EDK	Effective Date of Key(s)	Date and Co-ordinated Universal Time on which the data key(s) are activated.
ERF	Error Field	The identification of error conditions detected in a prior Cryptographic Service Message.
ESM	Error Service Message	A message type used to give a negative acknowledgement on receipt of any Cryptographic Service Message other than an ESM, and to give the recipient data with which to recover.
EXCV	Expiration date of a Certificate Value	Date and Co-ordinated Universal Time after which a certificate is not valid.
FFC	Function and Format Code	A code word in a certificate obtained by transforming text. It defines both the function of the certified public key and the content and format of the rest of the certificate text.
FKD	Function of Data Key	Indicates the functions for which a data key may be used.
FKP	Function of Public Key	Indicates whether the KAM to which a public key belongs can be used for encipherment, signature verification or both.
IABE	Identifier of block encipherment algorithm	-
IABS	Identifier of block signature algorithm	-
IACS	Identifier of certificate signature algorithm	-
IAPK	Identifier of algorithm used with the certified key	-
IDA	Identifier of Authentication Key	Identifies the data key used to authenticate a Cryptographic Service Message.
IDCC	Identifier of Key Certification Centre	-
IDCK	Identifier of certification centre key	Identifier (name) of the key used by the Key Certification Centre to produce a certificate.
IDCV	Identifier of the Public Key certified	Identifier (name) of a Public Key certificate.
IDD	Identifier of Key to be Discontinued	-

IDEK	Identifier of Enciphering Key Certificate	Identifier (name) of the certificate containing the public key used to encipher the Enciphered Block being transmitted in a Cryptographic Service Message.
IDK	Key Identifier	Identifier of the key being transmitted in a Cryptographic Service Message.
IDSK	Identifier of Signature Key Certificate	Identifier (name) of the certificate containing the public key to be used to deconstruct the signature of the Enciphered Block being transmitted in a Cryptographic Service Message or the signature of the whole message.
IV	Initialisation Vector	See 1.3.29
KAM	Asymmetric Master Key Set	Comprises a public and a secret key employed to protect the distribution of data keys by encipherment or digital signature.
KD	Data Key	A key used to operate directly on data.
KDX	Fixed Data Key	A key with fixed value used in the computation of an Error Detection Code.
KP	Public Key	The value of an asymmetric Public Key in a certificate.
KSM	Key Service Message	A message type used to transfer keys between communicating pairs. Optionally may transfer a certificate.
MAC	Message Authentication Code	See 1.3.40
MCL	Message Type	The tag for the field that defines the type of Cryptographic Service Message.
ND	Name of Data Key	Field containing the identification of a data key contained in an Enciphered Block.
ORG	Originator	Originator of CSM.
OWN	Owner	The name of the party to whom a certificate was originally issued.
P	Key Parity	Indicates that the plaintext key conforms to the specification for odd parity.
RCV	Recipient	Recipient of CSM.
RSI	Request Service Initiation Message	Used to request keys and/or a certificate from another party. Optionally may transfer a certificate.
RSM	Response Service Message	Used to provide an authenticated acknowledgement. Optionally may transfer a certificate .
SIG	Digital Signature field	Optional field of a Key Service Message containing a signature of the whole message.
SVR	Service Request	Specifies type of service requested.

1.4.2 Notation

1.4.2.1 *Operators* are represented by the following lower case letters :

a	for authentication
e	for encipherment
d	for decipherment
s	for signature construction by transforming the text
v	for signature deconstruction
σ	for signature construction by means of a separate signature

"[ ]" denotes representation of field contents

"{ }" denotes the subject of an operator. If the operator involves a key, the type of key is inserted between the operator and its subject.

For an asymmetric algorithm, e and v are operations performed with public keys and d and s are the operations performed with secret keys.

1.4.2.2 *Concatenation* is indicated by "||"

1.4.2.3 *Modulo-2* addition is indicated by "+"

1.4.2.4 *Fields* are separated by "b̲", space

1.4.2.5 *Sub-fields* within a field are separated by ".", full stop

1.4.2.6 *The types* of key are :

KAM	asymmetric master key set
KD	data key
KDX	data key, plaintext, for computation of the Error Detection Code, hexadecimal e.g. 0123456789ABCDEF (see 3.3.1.1.)

1.5 Key management facility

1.5.1 General

A key management facility shall provide a means of access control whereby its contents are protected from unauthorised disclosure, modification, substitution, replay, insertion or deletion.

NOTE — To achieve such control, action needs to be taken to either preclude access or to ensure that attempts to gain access have a high probability of being detected and reported.

1.5.2 Contents of key management facility

All cryptographic equipment, including key generation equipment, shall be located within a key management facility.

NOTE — Cryptographic equipment may itself act as the key management facility, and so provide all the required functions.

1.6 Key generation, storage, back-up and destruction

1.6.1 Generation of keys and initialisation vectors

1.6.1.1 General

The output from a key generator shall be automatically checked for generation failure (e.g. the repeated output of the same key). Operation of the key generator shall stop immediately if any failure is detected.

The design of this generation process shall be such that no cryptographic advantage is gained by attacking the key generation process rather than the encipherment process.

Secret keys shall not be available in plaintext form from cryptographic equipment, even upon failure of the equipment, other than for backup purposes under dual control.

A means shall be provided for the manual zeroisation of plaintext keys.

1.6.1.2 Generation of symmetric keys and initialisation vectors

A symmetric key set shall be generated only in a key management facility and under management control.

The generation of symmetric keys and initialisation vectors shall be by means of a process that ensures that all keys and initialisation vectors are random or pseudo-random.

1.6.1.3 Generation of asymmetric key sets

An asymmetric key set shall be generated only in a key management facility and under management control.

The generation process shall conform to a standard specified as part of an asymmetric algorithm approved for use with this standard.

1.6.2 Key storage

Secret key values in plaintext form shall be stored only as components for backup purposes (see 1.6.3.1) or within a key management facility.

Security features shall be built into the software and hardware for the purpose of preventing unauthorised access. Any attempt which threatens to gain unauthorised access to plaintext secret key values in the protected memory shall result in the stored plaintext secret key being erased or otherwise rendered unintelligible.

Where plaintext public keys are stored and are not in the form of a certificate, or where their certificate has been checked and they will be used without re-checking the certificate, access which can change the public key value shall be prevented by security features built into thc software and hardware.

A short term power failure shall not result in the loss of a key.

Where keys are associated with counters (see 3.3.2), the counters shall be protected against erasure, loss or lowering of a counter by security features built into the software and hardware.

Custody of keys, whether in the form of key components, secure storage devices or backup key management facilities with keys loaded, shall be recorded on the security audit log. Lists of staff designated to hold or access keys shall be kept.

1.6.3 Backup of secret key values and recovery

1.6.3.1 Backup of secret keys

Obtaining of plaintext values of secret keys for backup purposes shall be under dual control and recorded in the security audit log.

Backup values may be held in a secure storage device such as a key loader or in a backup key management facility.

All backup copies of keys shall be subject to the same or a greater level of security control as keys in current use.

If readable backup copies are made, these shall be in the form of at least two key components, requiring all the components to be known in order to recover the key values. Each component shall include a sum check of sufficient size that the probability of undetected error is less than one in 1000. Key components shall be in the custody of a different individual for each component. The identity of the person receiving the key component shall be recorded in the security log.

If backup secret key values are held in a secure storage device, access to the stored values shall be controlled by positive user authentication, e.g. access identifier and password or other methods, to prevent unauthorised access.

If secret key values are transferred to a backup key management facility, this shall be under dual control and be recorded in the security audit log.

1.6.3.2 Recovery

Recovery of key values from backup or the placement into operation of a backup key management facility shall be under dual control and be recorded in the security audit log.

Where backup keys are held as key components in individual custody, a means shall be provided for manual entry of key components. Components shall be entered by each individual with no others present. A means of correcting errors or re-entering the entire key component shall be provided. The sum check shall be performed at entry and an indication of error given to the operator. A set of key components with sum check error shall not be accepted by the system. If a plaintext key component is displayed during entry, it shall be visible only to authorised persons, and shall be cleared immediately after the key entry process is completed.

Each component entry attempt and its outcome shall be recorded in the security audit log, with the identity of the operator.

1.6.4 Key destruction

Plaintext secret key values or the components thereof shall be zeroised at the end of their life.

Any intermediate storage of plaintext keys shall be zeroised once the transfer of the key to another location is complete.

Key destruction that is triggered by an attempt at unauthorised access shall be by zeroisation. Key destruction shall be effective in the presence of power failure.

All copies of keys that are no longer required shall be destroyed under dual control.

A detailed record of withdrawal from service and destruction shall be retained for audit trail purposes.

1.6.5 Security audit log

A security audit log shall be required both for operation and to support the audit function. The security audit log shall provide sufficient detail to reconstruct events, provide "due care," meet legal requirements and allocate liability. The contents of a security audit log shall be date and time stamped.

The security audit log shall be maintained in a form which prevents unauthorised modification or destruction of the record.

The security audit log shall record all key management operations, such as key generation, backup, recovery and destruction, together with the identity of the person authorising the operation and persons handling any key material (such as key components or keys stored in portable devices or media). All actions resulting from the suspected compromise of any keying material shall be recorded in the security audit log.

Custody of keys and devices or media holding keys shall be recorded in the security audit log.

The security audit log shall not contain any record of plaintext key values. It may hold check values derived from keys by means of a one-way function, as a means of identifying keys and verifying their correctness.

The use to be made of the security audit log is not specified in this part of ISO 11166.

1.7 Keys

1.7.1 Custody of secret keys

Secret keys stored outside the key management facility shall be enciphered or otherwise not be capable of being disclosed.

Lists of the staff authorised to manage or access keys shall be kept. These lists shall not contain any plaintext key values.

1.7.2 Validity of keys

Keys should normally be allocated a unique identifier and an effective date. The communicating pair shall agree upon the cryptoperiod for each key.

Data keys may be distributed on the basis that they are for immediate or for future use (see 1.7.4). The sender of a key shall not employ that key until an authenticated acknowledgement has been received from the receiving party, checked and found to be valid. Where a key has not been specifically identified (e.g. by number or effective date), it shall be the only such key and shall be put into service by the receiver after the acknowledgement has been sent and by the sender after the acknowledgement has been received, checked and found to be valid.

Where it is suspected or known that a key has been compromised, it shall no longer be considered to be valid and shall be withdrawn from current use.

1.7.3 Key changes

Possession of another party's certified public key potentially enables a data key to be automatically established with that party. Management controls shall be applied such that any new keying relationship shall be positively authorised before any data key established by such automatic means becomes valid.

Keys shall be changed :

a) at the end of the cryptoperiod; or

b) at the request of either member of the communicating pair; or

c) immediately after it is known or suspected that a key has been compromised.

All key changes shall be acknowledged. Where the cryptoperiods of an existing and a new key overlap, the effective date of the replacement key (or other implicit time reference) shall be specified whereupon the old key is no longer active. During this changeover period, both keys shall be held under the same level of security.

Keys withdrawn from use shall not be knowingly or intentionally re-used, except for the purpose of reconstructing a key/message pair (see 1.7.5).

1.7.4 Reserve keys

Where keys are stored in reserve, to facilitate planned or unexpected key changes, they shall be subject to the same level of security control as keys in current use.

1.7.5 Archiving of keys

Where the continued storage (archiving) of a key after the expiration of its cryptoperiod, or compromise, is required each such key shall be uniquely identified, or converted into a different form or format so that there is no ambiguity that it is archived and obsolete. All archived keys shall be enciphered under a key designated for that purpose. It shall not be possible to use archived keying material other than for the reconstruction of a key/message pair.

All restrictions and security measures that apply to operational keys also apply to archived keys.

NOTE — The detailed procedures for the archiving of keys are application dependent and are not defined in this part of ISO 11166.

Section 2: Key certification

2.1 Minimum requirements for secure operation of a Key Certification Centre

2.1.1 General

The authentication of public keys is an essential security requirement. This is ensured by exchanging them in the form of certificates. A certificate contains the public key and its identifying data, with a digital signature provided by the Key Certification Centre (CKC).

Public keys contained in a certificate from the CKC may be exchanged by automated or other means. In no instance shall a public key before its certification be automatically handled by a process defined in section 3 of this part of ISO 11166. A recipient of a certificate shall verify the signature of the certificate before using the public key contained in the certificate. After checking the certificate, the public key may be used, provided it has been stored in a manner which prevents modification or substitution of the stored value or of its identification data and that it has not been cancelled (see 2.1.5).

The signature of the certificate shall be verified with the aid of the public key of the CKC.

The Key Certification Centre is "trusted" by it subscribers. Such trust is based on cryptographic equipment designed to meet requirements for financial institution use, sound management and control practices and is confirmed by an independent audit function (internal, external or both) which shall report audit results to the subscribers.

The security audit log shall be maintained as described in 1.6.5.

The CKC security audit log shall be protected from modification or substitution, for example by the use of a digital signature. In this case, the public/secret key pair used for signing the security audit log shall not be used for any other purpose.

An example of the contents of a typical CKC security audit log contents is contained in annex E.

2.1.2 Generation and storage of the asymmetric key set of the CKC

The asymmetric key set of the CKC shall be generated in a key management facility under dual control, in addition to the criteria specified in 1.6.1.3. for key generation.

The secret key of the CKC shall be stored in the key management facility in which it is used. Backup of the CKC secret key shall be provided as specified in 1.6.3.1 with appropriate means of recovery as specified in 1.6.3.2.

2.1.3 Distribution of the CKC public key

The CKC public key shall be entered into each key management facility. The authenticity of this key is essential. It is the recipient's responsibility to ensure that the CKC public key is authentic. Methods of accomplishing this are described below.

Initial loading of the CKC public key may be performed in a secure environment under the control of the CKC before the key management facilities are distributed.

Other circumstances may require the distribution of a CKC public key to key management facilities that are already in use. If a trustworthy keying relationship exists between the CKC and each party, an existing shared key may be used to protect the integrity of the transmitted CKC public key and to provide authentication.

If no such keying relationship exists, the distribution of the CKC public key shall be protected by manual methods, such as distribution by a number of different and independent channels, with cross-checking of the received values (examples are : letter, fax, E-mail, recorded messages accessed by phone, newspaper).

Because the CKC public key is a large number, manual distribution and cross-checking is inconvenient. The key may be transmitted to the key management facility automatically and without protection, provided that its authenticity is then verified by means of a cryptographic check value which is authenticated. The check value shall be formed by a public one-way function giving at least a 32 bit result.

The check value for a planned CKC public key distribution may be distributed well in advance to allow adequate verification of its authenticity, without exposing the CKC public key during the period before it comes into use.

NOTE — If an authentic CKC public key is loaded by a number of parties, their exchange of certificates with other parties and the verification of the signatures will reveal the existence of false CKC public key values if such exist.

2.1.4 Submission of a user public key for certification

To preserve the authenticity of public keys, the CKC must ensure that public keys for which it provides certificates are genuine.

The method used to submit a public key and receive a certificate in return shall meet these criteria :

a) The CKC shall verify the authentic origin of the submitted key

b) The CKC shall authenticate the submitted public key value.

The trust in the Certification Authority is in part based on the methods employed to identify the authenticity of its members prior to the certification of their public keys. The method used shall be defined and enforced by the centre for its membership.

This part of ISO 11166 does not specify the methods to be used.

NOTE — Authentication of a submitted public key by the CKC is the first step in providing secure automatic key distribution. This first step cannot be entirely automatic but it can be based on authentication data which were exchanged when a business relationship was established (examples of such data are the manual signature of an authorised signatory or a cryptographic key held in a smart card). The nature of this pre-existing data varies among applications of this part of ISO 11166.

Examples are given in annex C.

2.1.5 Action on compromise of a user's secret key

If it becomes evident that a user's secret key has been compromised, the key shall immediately be replaced and the compromised key cancelled. All keying material ever sent and protected using that compromised key (without regard to type) shall be replaced at the earliest opportunity.

The owner of the key shall take immediate action to inform all parties in the system that all certificates of the public key relating to the compromised key have been cancelled. This may take the form of a message authenticated by the CKC and sent to all parties, or the maintenance of an on-line list of revoked certificates (Hot List) by the CKC.

The party with the compromised key shall take immediate action to generate a new asymmetric key set and obtain a certificate for the public key. By sending this certificate to other communicating parties, they can be informed of the change of key.

2.1.6 Lifetime of a certified public key

A certificate shall contain an expiry date in the identifying data. A new certificate shall be generated and distributed before that date.

It shall be possible for the CKC or the party identified in the certificate to withdraw a certificate prior to its expiry date.

After the expiry date, any message requiring the use of the expired certificate shall be rejected.

When a certificate expires, keying material sent and protected by that certificate (without regard to type) should be replaced as soon as it is possible without disrupting service.

2.1.7 Compromise of the CKC secret key

If the CKC secret key is suspected of being compromised, the CKC shall take immediate action to generate a new key set and distribute the new CKC public key.

The most cautious reaction of users in this event is to distrust all keying material, since the start of the compromise is not known. At some risk, users may choose to trust public keys of communicating parties for which the certificates were received well before the known compromise, and data keys subsequently exchanged using these public keys.

Design of the physical and operational security of the CKC (such as software integrity) shall minimise the possibility of the compromise of its secret key and maximise the probability that a compromise will be detected.

NOTE — Optionally, more than one CKC may be provided, each CKC operating independently of the others. A user's public key may then be contained in certificates from more than one CKC. The compromise of the secret key of one CKC does not invalidate certificates produced by other CKCs.

This optional method of operation increases immunity to the compromise of the secret key of one CKC. A user's public key may continue to be used, employing a certificate which is still valid.

Annex D describes a different method of operating using two CKCs.

Section 3: Automatic distribution of keying material

3.1 Requirements for the automated key management architecture

This part of ISO 11166 is designed to meet the following requirements for automated key management. It is assumed that:

1) the number of parties in the network is expandable;

2) each party of a communicating pair has an asymmetric key set with a certificate of the public key produced by a certification authority;

3) each party has an authentic and up-to-date copy of the certification authority's public key.

3.1.1 The architecture shall support the ability to have at least one data key between communicating pairs.

3.1.2 The architecture shall support the ability to change data keys automatically between communicating pairs.

3.1.3 A particular data key shall be used for either encipherment/decipherment or for authentication, but not for both, except when authenticating a Cryptographic Service Message.

3.1.4 A data key shared between a communicating pair shall not be disclosed to a third party.

3.1.5 An asymmetric secret key shall not be disclosed to any other party.

3.1.6 The same data key shall not be knowingly or intentionally used by more than one communicating pair.

3.1.7 The compromise of any key shared between any communicating pair shall not compromise any third party.

3.1.8 The architecture shall support any party initiating a secure connection with any other party, subject to management controls.

3.2 Automated key management architecture

3.2.1 General

The architecture shall consist of two layers of keys (see figure 1).

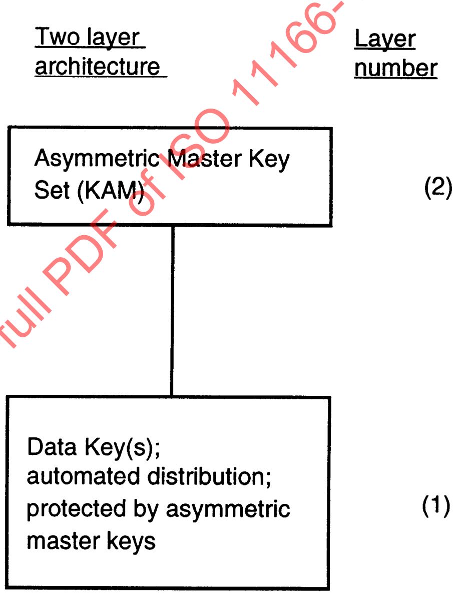
Two layer architecture
Layer number
Asymmetric Master Key Set (KAM)
(2)
Data Key(s); automated distribution; protected by asymmetric master keys
(1)

Figure 1 — Key distribution architecture

The upper layer shall comprise one or two asymmetric master key sets (KAM). The lower layer shall comprise a data key or keys (KD).

3.2.1.1 Protection of keys for key transport

The public key of a KAM shall be used to encipher the data key(s) such that it/they can only be deciphered by the intended recipient.

The secret key of a KAM shall be used to authenticate the source of the enciphered data key and, optionally, to provide a digital signature on the message transporting it.

Depending on the properties of the algorithm(s) used, these processes may share a KAM or alternatively separate encipherment and signature KAMs may be employed. The corresponding public keys of these KAMs are called *encipherment public key* and *signature public key* respectively. When a single KAM is used for both purposes, the qualifiers *encipherment* and *signature* may be ignored.

A KAM shall comprise an asymmetric pair of keys. The KAM shall be generated in a key management facility. If the secret key is released from a facility for purposes of backup and recovery, this shall be performed under management control, according to the principles described in 1.6.3.1. The secret key shall be used for constructing the digital signature of originated messages and/or decipherment of received blocks of enciphered data. The public key(s) shall be certified by the key certification centre according to the principles described in clause 2.1. Only when a certificate has been received and verified shall the public key be used for encipherment of originated messages and/or the deconstruction of digital signatures in received messages.

3.2.2 Distribution of keying material

Public keys shall be distributed in the form of a certificate created by the key certification centre. The requirements for the operation of the key certification centre are given in clause 2.1.

For the transport of data keys, they shall be enciphered prior to distribution, using the public key of the KAM of the recipient, provided that the certificate of the public key has been checked and its signature verified.

NOTE — Two data keys may be sent, enciphered, in a single Cryptographic Service Message (see clause 4.1).

The encipherment of data or initialisation vectors (IV) prior to distribution shall be carried out using data keys. Alternatively, the IV may be included in the block carrying enciphered data keys.

When a new key is received, all stored keys with the same identifier (name) shall be replaced at the effective date of the new key. In addition, when a key is discontinued, all keys of the same identifier (name) shall be discontinued.

The recipient of cryptographically protected data from which keying material is to be extracted shall be able to identify the relevant KAM(s).

3.2.3 Environments

Key management by means of asymmetric algorithms employs only one environment. In the terminology of ISO 8732, it is a point-to-point environment in which the communicating pair share (in the form of certificates) the public key parts of their KAMs.

In this part of ISO 11166, the environment is implicit and it will not be referred to elsewhere in the standard.

Each member of a communicating pair has its own KAM(s). In order to send data keys to another party, the sending party requires the receiving party's encipherment public key. The receiving party requires the sending party's signature public key in order to verify the received data keys.

All public keys shall be certified by a common key certification centre. A method of operating with dual key certification centres is illustrated in annex D. The resulting certificates may be exchanged in advance of the communication of data keys and then stored by the parties. Alternatively, they can be exchanged as part of the sequence of messages used to communicate data keys in which case, parties do not need to maintain a store of their correspondent's certificates.

Each member of the communicating pair shall verify the validity of the certificate of the other member before using it.

3.3 Primitive processes

NOTE — The notation used in this clause is described in clause 1.4 .

3.3.1 Encipherment, decipherment, authentication, error detection, signature construction and signature deconstruction

This subclause specifies primitive processes for use in key distribution by the method of key transport. Primitive processes for use in the key exchange method may be the subject of subsequent parts of this International Standard.

3.3.1.1 Approved algorithms and notations for encipherment, decipherment, authentication and error detection

Approved algorithms for asymmetric encipherment will be given in subsequent parts of this International Standard and the approved algorithms for authentication (also used for error detection) are given in ISO 8731. The notations for encipherment, decipherment, authentication and error detection, respectively, are as follows (both symmetric and asymmetric encipherment use these notations):

enciphered text = eK{plaintext}

plaintext = dK{enciphered text}

authentication code = MAC = aK{data}

error detection code = EDC = aKDX{data}

where key KDX, if of 64 bit length, shall be :

KDX = 0123456789ABCDEF

Where the key of the authentication algorithm is not of 64 bit length, the value of KDX shall be the quoted value appropriately reiterated or truncated.

The processes for asymmetric encipherment and decipherment as applied in this standard are specified in 3.3.1.5. and 3.3.1.6. and in figures 2 and 3.

3.3.1.2 Approved algorithms and notations for digital signature construction and deconstruction

Two forms of signature are employed in this part of ISO 11166:

a) Signature construction by transforming the text.

Only the transformed text need be sent in the message but, optionally, the untransformed text may also be sent.

Approved processes for digital signature by transforming the text as applied in this part of ISO 11166 to the first sub-field of the field BES, will be specified in subsequent parts of this International Standard. This method of digital signature is also known as "digital signature with message/text recovery".

The notations for signature construction and deconstruction are as follows :

signed text = sKS{text}

text = vKP{signed text}

where KS and KP are the respective secret and public keys of an appropriate asymmetric key set.

Signature construction operates directly on the text to be protected.

Signature deconstruction recovers the original text. Verification of the correctness of signature shall employ redundancy. In this part of ISO 11166 redundancy is provided in the form of a MAC and a counter (CTP).

The processes for signature construction and deconstruction of type a), as applied to an enciphered block, are specified in 3.3.1.7 and 3.3.1.8 and in figures 4 and 5.

This form of digital signature may optionally be used in the formation of a certificate (see 3.3.4.1) and in this case, the redundancy shall be contained within the signed text.

b) Signature construction by means of a separate signature.

Both the signature and the text are sent in the message.

When Cryptographic Service Messages are to be signed, the signature shall be computed using an algorithm approved for use with this part of ISO 11166. The signature is a function of the whole text to be signed and a secret key. The notation used is:

SIG = σKS{text}

where KS is the appropriate secret key of an asymmetric key set.

Signature deconstruction consists of extracting from the message the text which has been signed and verifying the value of SIG in relation to this text. The verification process is specified in the part of this International Standard which contains the applicable digital signature algorithm.

This form of digital signature may optionally be used in the formation of a certificate (see 3.3.4.3).

3.3.1.3 Encipherment and decipherment of initialisation vectors

When IVs are enciphered and deciphered with a symmetric algorithm, a data key (KD) shall be employed with the following notations :

enciphered IV = eKD {IV}

IV = dKD {enciphered IV}

When DEA is used, it shall be in ECB mode and IVs shall have a maximum length of 64 bits.

See 3.3.3 for the encipherment of IVs with asymmetric algorithms.

3.3.1.4 Encipherment and decipherment of a block using an asymmetric algorithm — General

The block to be enciphered is denoted BC and is formed as specified in 3.3.3. BC is not a field of any cryptographic service message. When BC is enciphered it becomes the first sub-field of BE, which is a field of the cryptographic service message of type KSM. When BC is both enciphered and signed it becomes the first sub-field of BES, which is a field of the cryptographic service message of type KSM.

Optional extra sub-fields of BE and BES identify the asymmetric key sets (KAMs) employed for encipherment and signature construction. In the case of BE, the key set employed for signature of the whole message in the SIG field of the KSM may be identified in the third of its optional sub-fields (see 4.1.5.1).

3.3.1.5 Encipherment of a block by asymmetric keys

A block BC shall be enciphered with the asymmetric cryptographic algorithm using the following formula :

enciphered BC = eKP2{BC}

where :

BC is a block as described in 3.3.3.

KP2 is the (encipherment) public key of the receiver.

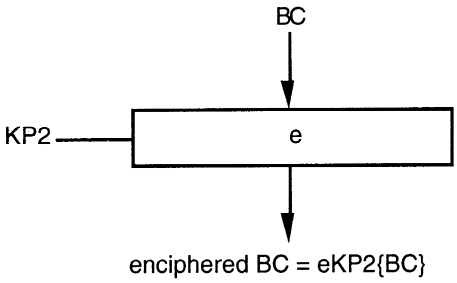

Figure 2 — Encipherment

3.3.1.6 Decipherment of an enciphered block by asymmetric keys

An enciphered block BC shall be deciphered using the following formula :

BC = dKS2{enciphered BC}

where :

enciphered BC is a block as described in 3.3.1.5.

KS2 is the (encipherment) secret key of the receiver.

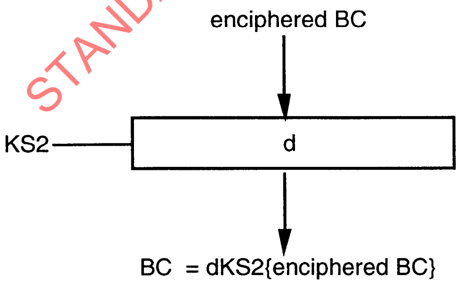

Figure 3 — Decipherment

3.3.1.7 Signature construction of an enciphered block by asymmetric keys

An enciphered block shall be signed by transforming the text with the asymmetric cryptographic algorithm using the following formula :

Signed enciphered BC = sKS1{enciphered BC}

where :

enciphered BC is specified in 3.3.1.5.

KS1 is the (signature) secret key of the sender

The combined processes of encipherment and signature are :

signed enciphered BC = sKS1{eKP2{BC}}

This combined process shall be performed as part of a single call to the key management facility.

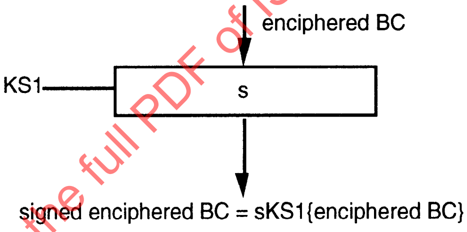

Figure 4 — Signature construction, as applied to enciphered BC

3.3.1.8 Signature deconstruction and decipherment of a signed enciphered block by asymmetric keys

The signature of a signed and enciphered block BC shall be deconstructed using the following formula :

enciphered BC = vKP1{signed enciphered BC}

where :

enciphered BC is specified in 3.3.1.5.

KP1 is the (signature) public key of the sender

The verification of the correctness of the signature shall employ redundancy in the form of a MAC and a counter (CTP) as specified in 4.3.4 (table 11, entries for BES and MAC).

The combined processes of signature deconstruction and decipherment are :

BC = dKS2{vKP1 {signed enciphered BC}}

This combined process shall be performed as part of a single call to the key management facility.

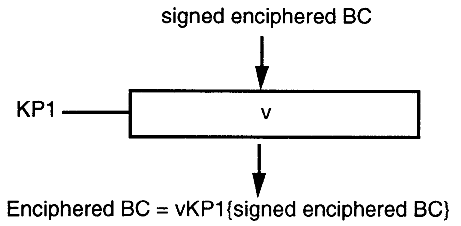

Figure 5 — Signature deconstruction, to recover enciphered BC

3.3.1.9 Authentication of Cryptographic Service Messages

For the authentication of the cryptographic service messages (of type KSM) which hold enciphered data keys, there are two options :

a) The data keys are enciphered by an asymmetric algorithm and signed by an asymmetric algorithm, producing the first sub-field of a field called BES (Block Enciphered and Signed). The whole message is authenticated using a MAC generated using the data key or keys contained therein.

It is allowable as an option to include a BE field as well as the BES field.

Verification of signature is not completed until the MAC and counter values have been checked.

When Cryptographic Service Messages are to be authenticated by a MAC, the MAC shall be computed using a data key and the technique defined in ISO 8730 using the entire message text with no editing, and an authentication algorithm from ISO 8731. The following formulae shall be used :

MAC = aKD{data}

A Cryptographic Service Message containing a single KD shall be authenticated using that KD. If two KDs are sent in a Cryptographic Service Message, the key used for authentication shall be the result of calculating the modulo-2 sum of the two data keys sent in the message.

b) The data keys are enciphered by an asymmetric algorithm, producing the first sub-field of a field called BE (Block Enciphered). The whole message is authenticated using a digital signature (SIG) generated using the secret key of the originator.

The primitive process for computing a digital signature of a Cryptographic Service Message is defined in 3.3.1.2 (b).

3.3.1.10 Error detection

To detect transmission or process errors in certain types of CSM, the authentication process specified in 3.3.1.9 shall be used, as in the following formula, to derive an Error Detection Code (EDC) :

EDC = aKDX{data}

where KDX is a key with a fixed value (see 3.3.1.1).

3.3.2 Counters

3.3.2.1 Purpose

Counters are used to :

a) detect duplicate Cryptographic Service Messages of type KSM

b) detect a message received out of sequence

c) resynchronise a service when counters indicate a loss of synchronisation.

Counters shall only increment, and never decrement, except when set to "1" as specified in 3.3.2.4.

Counters for key management purposes (as defined in this part of ISO 11166) shall be independent of the means used to differentiate between individual data messages (e.g. sequence numbers).

Counters shall be transmitted as fields in Cryptographic Service Messages of type KSM, and shall be authenticated by an appended MAC or SIG field as specified in 4.1.6.2 (table 3).

3.3.2.2 Management of counters

The contents of a counter shall be defined and manipulated as a binary number.

Each member of a communicating pair shall maintain a counter for originated messages to the other party and a counter for received messages from the other party.

The counter CTP used in a block (see 3.3.3) shall be that counter associated with the originator of that block.

Any difference between the counter received by the recipient in a KSM and its expected value shall be detected and action taken as defined below to resolve the problem.

In the event that the two parties are unable to agree on a common counter value, the procedure of 3.3.2.4 shall be applied.

Duplicate messages shall be rejected and an error shall be reported to the originator. The recipient prepares an Error Service Message (see 4.2.3) in which the nature of the error being reported (duplicate message), the value of the reception counter (the recipient's expected count) and the value of the origination count as received in the related message, are included.

3.3.2.3 Processing of counters (see table 1)

NOTE — In this subclause and table 1 only, the symbol + means arithmetic addition.

This subclause specifies the processing of counters to manage the unidirectional flow of key service messages from one party of a communicating pair to another. Each such flow shall employ this procedure.

For the purpose of subclauses 3.3.2.3 and 3.3.2.4, the following notation will be used :

The member of a communicating pair which sends a KSM is called A, and the member which receives the KSM is called B.

The counter for originated messages at A which are destined for B, known as a sending counter, contains the number CS.

The counter maintained by B for messages received from A contains the number CR.

CT denotes the number extracted from the CTP field of a KSM received at B from A. It is also used to denote the number extracted from the CTP field of an ESM received at A from B in response to a KSM.

When A originates a KSM destined for B, the CTP field of that message shall be set to CS, and the value of counter CS shall be incremented by 1.

When B receives a KSM from A, the value CT extracted from the CTP field shall be processed as follows:

a) If CT=CR, the KSM shall be accepted and CR incremented by 1. If there are no other errors in the KSM, B shall respond to the KSM by sending an RSM to A.

b) If CT>CR, this error shall be logged, the KSM shall be accepted and CR shall be set to CT+1. If there are no other errors in the KSM, B shall respond to the KSM by sending an RSM to A.

c) If CT<CR, this error shall be logged and the KSM shall not be accepted. An ESM shall be generated as specified in 4.2.3 and sent to A. The error code shall have the value P, the CTR field shall contain the value CT, and the CTP field shall contain the value CR.

If the KSM which has not been accepted was a response to an RSI from B, a new RSI may be sent by B to A in order to restart the procedure. In all other cases, A has the responsibility for restarting the procedure.

NOTE — The purpose of the CTR field, which reports back the received CTP, is to assist party A in identifying the KSM to which this ESM is a response.

When A receives an ESM from B with error code P, the value CT is extracted from the CTP field and compared with the value CS. Processing shall proceed as follows :

a) If CT>CS, set CS to the value of CT.

b) If CT=CS or CT<CS, the value of CS is not changed.

c) The receipt of the ESM and the action taken shall be logged.

d) A may restart the procedure unless the original KSM was a response to an RSI from B.

NOTE — CS has been incremented since the related KSM was sent.

3.3.2.4 Reset of counters

In order to prevent the replay of a KSM, the values CS and CR shall not be decremented or reset except as specified in this paragraph. In the event that A and B are unable to agree on a common counter value, then A shall change its KAM, withdraw any certificates associated with the replaced KAM and shall set CS=1 for the new KAM. Before accepting further key exchanges from A, B shall also change its KAM and set CR=1. Similar action shall be taken by both parties whenever a counter value reaches its maximum value.

NOTE — A change of KAM does not require the reset of counters, except as required by 3.3.2.4.

Table 1 — Processing counters

Action on receiving a KSM			Action on receiving an ESM	
CT=CR	CT>CR	CT<CR	CT>CS	CT<CS or CT=CS
Accept message	Accept message	Reject message	Set CS=CT	Counter value CS is not changed
	Log error	Log error		

Send RSM	Send RSM	Send ESM with :
		CTR=CT
		CTP=CR
			Log receipt of ESM and action taken.	

Increment CR by 1	Set CR=CT+1	Send new RSI (1)	Restart procedure or wait for RSI (1)	

(1) If the KSM was a response to an RSI.

3.3.3 Block construction

The block (BC) contains binary data (not character coded). The size of the block is a characteristic of the algorithm used.

Parsing of the block depends on fixed length fields. The number of data keys in a block is indicated by the number of ND fields in the CSM. The presence of an IV in the block is signalled by the first character of an IV field. The length of data keys and the IV shall be determined either by prior agreement or by the corresponding data key identity in field ND.

3.3.3.1 Block construction with one KD

In this case, the block is :

> [(padding value) || (IV) || (ORG) || (KD) || (FKD) || (CTP)]

The padding value shall be random or pseudorandom.

In the event that subsequent (algorithm dependent) processing of the block results in an overflow, the block shall be reformed with a new padding value and if necessary IV and the process repeated as described in the part of ISO 11166 which relates to the algorithms employed.

The presence of an IV in the block is optional and its presence is indicated by the field IV/B in the CSM.

When a MAC field is not present in the KSM, the ORG value in the block is mandatory and its value shall equal that in the ORG field of the KSM.

When a MAC field is present in the KSM, the ORG value shall not be present in the block.

If FKD takes a value other than 0, it shall equal the value of the subfield FKD of a corresponding ND field which is present in the CSM. In this block, FKD shall be represented as 8 bits (0, b7 ... b1) as defined in ISO 646.

CTP in this block shall be a 56 bit number and shall equal the CTP field in the CSM. Leading zeros shall not be suppressed when representing CTP in block BC.

3.3.3.2 Block construction with two KDs

In this case, the block is :

> [(padding value) || (IV) || (ORG) || (first KD) || (first FKD) || (second KD) || (second FKD) || (CTP)]

The padding value shall be random or pseudorandom.

In the event that subsequent (algorithm dependent) processing of the block results in an overflow, the block shall be reformed with a new padding value and if necessary IV and the process repeated as described in the part of ISO 11166 which relates to the algorithms employed.

The presence of an IV in the block is optional and its presence is indicated by the field IV/B in the CSM.

When a MAC field is not present in the KSM, the ORG value in the block is mandatory and its value shall equal that in the ORG field of the KSM.

When a MAC field is present in the KSM, the ORG value shall not be present in the block.

Each FKD specifies the function of the corresponding data key KD.

If FKD takes a value other than 0, it shall equal the value of the subfield FKD of a corresponding ND field which is present in the CSM. In this block, FKD shall be represented as 8 bits (0, b7 ... b1) as defined in ISO 646.

CTP in this block shall be a 56 bit number and shall equal the CTP field in the CSM, which may be shorter because leading zeros are suppressed.

3.3.4 Certificate signature, construction, deconstruction and verification

The certificate value shall be constructed from the data specified in 4.1.5.4. These data shall be signed by the Certification Authority using its secret key using one of the forms of signature specified in 3.3.1.2. Certificate values with signature computed by transforming the text shall be computed as described in 3.3.4.1 and shall be verified as described in 3.3.4.2. Certificate values with a separate signature shall be computed as described in 3.3.4.3 and shall be verified as described in 3.3.4.4. Finally all the checks listed in 3.3.4.5. shall be performed prior to the certified public key being used.

3.3.4.1 Construction of a certificate signature by transforming the text.

The amount of text for a certificate obtained by transformation of text is algorithm dependent and is defined in the corresponding subsequent part of this standard. Certain minimum requirements are specified hereafter.

The first octet of the text to be signed shall be a Format and Function Code (FFC). It shall indicate the content of the certificate, its representation, and also the function of the Public Key being certified. Values of the FFC shall be assigned as follows :

00 to 3C by TC68/SC2
40 to 5C by National Bodies
60 to 7C for private use

All assigned values shall be a multiple of four. The FFC in the certificate text shall be obtained by adding the assigned value and the numeric value of the FKP which indicates the function of the certified Public Key (see table 2). Thus if the assigned value of FFC is 04 and the key being certified is for signature deconstruction (FKP = 2), the first octet takes the value 06 hexadecimal.

The certificate text shall be formed from the FFC and other data as required by the FFC. As a minimum this shall comprise data corresponding to each of the sub-fields EXCV, OWN and KP as specified in 4.1.5.4 together with redundancy as specified in the part of this standard which relates to the algorithm employed.

The subfields mentioned above need not be present in the CV field.

The certificate text shall be signed with the (identified) asymmetric algorithm as follows :

Certificate signature = sKSC{certificate text}

where KSC is the (identified) secret key of the CKC

The resulting value, expressed in hexadecimal notation, becomes the first sub-field of a CV field. Other sub-fields may optionally be appended whether or not they were included in the certificate text.

3.3.4.2 Deconstruction of a certificate signature by transforming the text

The CSM is parsed to extract the CV field and its sub-fields. The certificate text is recovered by appiying the formula :

Certificate text = vKPC{certificate signature}

where certificate signature is specified in 3.3.4.1. and KPC is the public key of the CKC.

If the KPC and the corresponding algorithm are not implicitly identified, they shall be identified by subsequent sub-fields of the CV field.

The verification of the correctness of the certificate text shall be performed by verifying that the text (including redundancy) complies with the formatting rules defined by the Function and Format Code (FFC) recovered from the first octet of the certificate text.

NOTE — Comparison of text recovered in this process with the text of corresponding sub-fields verifies the correctness of the sub-fields. It cannot be relied upon to verify the correctness of the certificate text itself. This can only be achieved by the checks of the format and redundancy of the recovered certificate text itself.

3.3.4.3 Generation of a certificate with a separate signature at a CKC

The second and all subsequent sub-fields of a CV field shall be assembled as defined in 4.1.5.4. using the character set and representation defined in 4.1.3. The resulting text shall be signed as specified in 3.3.1.2.(b) using the (identified) secret key of the CKC. The resulting signature shall be the first sub-field of a CV field.

3.3.4.4 Verification of a certificate with a separate signature

The CSM is parsed to extract the CV field. The recipient shall verify that the signature in the first sub-field is consistent with the text in the second and subsequent sub-fields using the (identified) Public Key of the CKC and the procedure specified in the part of ISO 11166 which relates to the algorithm employed.

3.3.4.5 Further checks on certificate validity

Regardless of the method of certificate construction, the recipient of another party's certificate shall also verify that :

a) The identity of the certificate owner (extracted from the CV field) is consistent with the party named in the ORG field of the CSM carrying the certificate.

b) The certificate has not passed the expiry date included within it.

c) The effective date of the certificate (if any) has been reached.

d) The certificate has not been withdrawn by the CKC or the party named in the certificate.

Section 4: Cryptographic Service Messages

4.1 Message formats and flows

4.1.1 General

Cryptographic Service Messages (CSM) shall be used for the automatic distribution and control of symmetric cryptographic keys and, where required, IVs and certificates between a communicating pair.

NOTE — For audit and control purposes, Cryptographic Service Messages may be journalised.

4.1.2 Message types

The following Cryptographic Service Message types are described in this part of ISO 11166:

NOTE — The formats of these messages are specified in 4.1.4. The content is specified in clause 4.2.

1. Request Service Initiation message (RSI) (optional)

   A message that initiates an exchange of certificates and/or requests (a) new data key(s).

2. Key Service Message (KSM) (mandatory for key transport)

   A message that transfers Data Key(s) from an originator to a recipient and may also transmit a certificate value.

3. Response Service Message (RSM) (mandatory)

   A message that provides an acknowledgement to an RSI, KSM or DSM and, in response to an RSI, may transmit a certificate value.

4. Error Service Message (ESM) (mandatory)

   A message that reports an error in a previous Cryptographic Service Message.

5. Disconnect Service Message (DSM) (optional)

   A message that is used to discontinue one or more keys.

4.1.3 Character set and representation

The character set for Cryptographic Service Messages shall comprise the following:

> 0-9, A-Z (capital letter), comma (,), full stop (.), space (b̲), solidus (/), hyphen (-), asterisk (*), left parenthesis ((), right parenthesis ()), carriage return and line feed.

The characters "." and "b̲" shall not be used in a sub-field; "b̲" shall not be used within a field (except for the MAC field). The character "." shall only be used to separate sub-fields within a field.

All characters shall be represented as eight-bit characters (0, b7, ... b1), where (b7, b6, ... b1) are defined in ISO 646. Where this necessitates a code translation, the translation shall be for internal processing and computational purposes only.

Hexadecimal characters shall be represented by the characters 0-9 and A-F.

4.1.4 Message formats

a) The presence of a Cryptographic Service Message shall be denoted by "CSM".

b) A Cryptographic Service Message shall begin with a left parenthesis "(", and end with a right parenthesis ")".

c) Field tags (see table 2) shall be separated from field contents by a solidus "/".

d) Fields shall be separated by a space "b̲" and, if desired for readability, a carriage return and line feed.

e) Sub-fields within a field shall be separated by a full stop (or period) "." and, if desired for readability, a carriage return and line feed.

f) If required for readability, long fields may be divided by inserting either of these sequences :

space "b̲", comma ",", and
solidus "/"
or
space "b̲", carriage return, line
feed, comma ",", and solidus "/".

4.1.5 Fields and sub-fields

Cryptographic Service Messages shall comprise a tag (specifying that the message is a Cryptographic Service Message) and a sequence of fields, sub-fields and associated parameters.

4.1.5.1 Enciphered Block — BE

A field containing an Enciphered BC may consist of up to five sub-fields. Each sub-field (whether present or not) shall be terminated by a full stop unless no subsequent sub-field is present (see the examples in the note). The ordering and content of the sub-field shall be as follows :

a) the Enciphered BC shall always be in the first sub-field, and is the only sub-field required.

Enciphered BC is specified in 3.3.1.5. The result of encipherment shall be encoded in hex characters, most significant hex character first.

b) the second sub-field IDEK shall, if present, contain the identity of the certificate containing the public key used to encipher the block.

c) the third sub-field IDSK shall, if present, contain the identity of the certificate containing the public key associated with the secret key used to construct the signature (SIG) of the KSM.

d) The fourth sub-field IABE shall, if present, contain the identity of the algorithm used for block encipherment.

e) The fifth sub-field IABS shall, if present, contain the identity of the algorithm used to construct the signature of the KSM.

NOTE — Examples of BE field format

1. BE/Enciphered BC.IDEK.IDSK.IABE.IABS is a field that contains a block, enciphered under the key whose certificate identity is IDEK, which will be used with algorithm IABE. The block is part of a message which is signed using the secret key associated with the public key whose certificate identity is IDSK, which will be used with algorithm IABS.

2. BE/Enciphered BC is a field with only one sub-field that contains data. In this case, the identities of the keys and algorithms used to encipher the block and sign the CSM are implicitly defined in the relationship.

4.1.5.2 Signed Enciphered Block — BES

Fields containing a Signed Enciphered Block (BES) may consist of up to five sub-fields. Each sub-field (whether present or not) shall be terminated by a full stop unless no subsequent sub-field is present (see the examples in the note). The ordering and content of the sub-fields shall be as follows :

a) The Signed Enciphered BC shall always be in the first sub-field, and is the only sub-field required.

Signed Enciphered BC is specified in 3.3.1.7. The result of the signature process shall be encoded in hex characters, most significant hex character first.

b) The second sub-field IDEK shall, if present, contain the identity of the certificate containing the public key used to encipher the block.

c) The third sub-field IDSK shall, if present, contain the identity of the certificate containing the public key associated with the secret key used to construct the signature of the block.

d) The fourth sub-field IABE shall, if present, contain the identity of the algorithm used for block encipherment.

e) The fifth sub-field IABS shall, if present, contain the identity of the algorithm used to sign the enciphered block.

NOTE — Examples of BES field format

1. BES/Signed Enciphered BC.IDEK.IDSK.IABE. IABS is a field that contains a block, enciphered under the key whose certificate identity is IDEK and for which the signature construction was done using the secret key associated with the public key whose certificate identity is IDSK. The encipherment algorithm is IABE and the signature algorithm is IABS.

2. BES/Signed Enciphered BC is a field with only one sub-field that contains data. In this case, the identity of the keys and algorithms used to encipher the block and to construct the signature are implicitly defined in the relationship.

4.1.5.3 Name of data key — ND

The description of the data key(s) protected in a Signed Enciphered Block (BES) or an Enciphered Block (BE) is provided by separate ND field(s). If two ND fields are present, the first one refers to the first KD included in the

Block, and the second refers to the second KD of the Block.

When contained in an RSI message, the ND field(s) describe the data keys requested by that RSI.

These ND fields may consist of up to three sub-fields. Each sub-field (whether present or not) shall be terminated by a full stop, unless no subsequent sub-field is present (see the example in the note). The ordering and content of the sub-fields shall be as follows:

a) The first sub-field shall, if present, be a "P" to indicate that the plaintext key conforms to the specification for odd parity.

b) The second sub-field shall, if present, be the sub-field FKD, coded as shown in table 2.

If present in a Key Service Message, the same value of FKD shall also be contained in the block BC (see 3.3.3.). If absent, the FKD character in block BC shall have the value 0.

c) The third sub-field IDK shall, if present, contain the identity of the key sent in the (signed) Enciphered Block.

NOTE — Examples of name field format

1. ND/ is a field without sub-fields. The identity of the new KD received in the (signed) Enciphered Block and its function are implicitly defined.

2. ND/..IDK is a field with one sub-field containing data. IDK is the identity assigned to the KD included in the (signed) Enciphered Block.

4.1.5.4 Certificate value — CV

A field containing a certificate value may consist of up to eleven sub-fields. Each sub-field (whether present or not) shall be terminated by a full stop, unless no subsequent sub-field is present. The ordering and content of the sub-fields shall be as follows :

a) The certificate signature computed as outlined in 3.3.4.1 or 3.3.4.3 shall always be the first sub-field and, for certificates computed by transforming the text (see 3.3.4.1), may be the only sub-field present.

b) The second sub-field shall, if present, contain the identity of the key (IDCK) used by the authority to produce this certificate.

c) The third sub-field shall, if present, contain the identity of the certification authority (IDCC).

d) The fourth sub-field shall, if present, indicate the effective date and time (EDCV) before which the certificate is not valid.

e) The fifth sub-field shall indicate the date and time (EXCV) after which the certificate is no longer valid. It shall always be included in the text signed by the certification authority, but the sub-field itself may be omitted if the certificate is computed by transforming the text (see 3.3.4.1).

f) The sixth sub-field (OWN) shall contain the name (identity) of the party to whom the certificate was originally issued. It shall always be included in the text signed by the CKC but the sub-field itself may be omitted if the certificate is computed by transforming the text (see 3.3.4.1).

g) The seventh sub-field shall, if present, contain the identity of the certificate (IDCV).

h) The eighth sub-field (KP) shall contain the Public Key being certified. It shall always be included in the text signed by the CKC but the sub-field itself may be omitted if the certificate is computed by transforming the text (see 3.3.4.1).

i) The ninth sub-field (FKP) shall, if present, identify the use of the public key being certified. If no sub-field is present, it shall be implied that the public key can be used either for enciphering data keys or for verifying signatures.

j) The tenth sub-field (IACS) shall, if present, contain the identity of the algorithm used by the authority to sign the certificate.

k) The eleventh sub-field (IAPK) shall, if present, contain the identity of the algorithm used with the public key which is being certified.

NOTE — Examples of CV field format

1. CV/CertificateSignature.IDCK.IDCC.EDCV. EXCV.OWN.IDCV.KP.FKP is a certificate with a separate signature. It was issued to OWN by IDCC using their key set IDCK. The Public Key KP is not valid before EDCV or after EXCV. It is used for function FKP. The certificate is named IDCV.

2. CV/CertificateValue is a field with only one sub-field that contains data. In this case, the identities of the public key certified, of the key used by the certification authority, and of the certification authority is implicitly defined in the relationship. The certificate is obtained by transforming the text.

3. CV/CertificateValue......IDCV is a field with two sub-fields containing data. IDCV is the identity of the certificate. The other sub-field contents are assumed (are implicitly defined in the relationship) or are recovered from the certificate value. The certificate is obtained by transforming the text.

4.1.6 Message flows

4.1.6.1 General

The fields and sub-fields shall be as defined in table 2. The fields used with each message class, and their sequence, shall be as defined in table 3 and the attached rules.

When a message is received with an optional field(s) that is not implemented by the receiver, an ESM with an "O" in the ERF field shall be returned, except that, in the case of the EDC field, an ESM with an "O" in the ERF field may be returned or the field may be disregarded and processing may continue.

NOTE — Field tags are identified by a / (field tag separator). The mnemonic representing a sub-field is prefixed with a full stop (or period).

Table 2 — Cryptographic Service Message Fields and Sub-fields

Field tag	Name	Definition/remarks	Specification
BE/	Enciphered Block	The first sub-field shall contain one or two KDs and other content as described in 3.3.3. It shall be enciphered as described in 3.3.1.5. It may be followed by further sub-fields which identify the certificates of the key used to encipher the KD(s) and/or the key used to compute the SIG field as described in 4.1.5.1 (IDEK and/or IDSK respectively). These may be followed by further sub-fields which identify the algorithms used for encipherment and digital signature as described in 4.1.5.1 (IABE and/or IABS respectively).	The first sub-field shall be hex characters and may be formatted for readability in accordance with 4.1.4.(f). The length of the field depends on the asymmetric algorithm used.
BES/	Enciphered and Signed Block	The first sub-field shall contain one or two KDs and other content as described in 3.3.3. It shall be enciphered as described in 3.3.1.5. and signed as described in 3.3.1.7. It may be followed by further sub-fields which identify the certificates of the key used to encipher the KD(s) and/or the key used to compute the signature as described in 4.1.5.2. (IDEK and/or IDSK respectively). These may be followed by further sub-fields which identify the algorithms used for encipherment and digital signature as described in 4.1.5.2 (IABE and/or IABS respectively).	The first sub-field shall be hex characters and may be formatted for readability in accordance with 4.1.4.(f). The length of the field depends on the asymmetric algorithm used.
CTP/	Counter P	An incrementing binary counter used between communicating pairs, but not between a CKC and another party.	Up to 14 hex characters; leading zeros may be suppressed for transmission.

CTR/	Counter R	The value of the counter found to be in error.	Up to 14 hex characters; leading zeros may be suppressed for transmission.
CV/	Certificate Value	The first sub-field shall contain the certificate signature as specified in 3.3.4. The signature shall be represented in hex characters. It may be followed by further sub-fields as described in 4.1.5.4.	The first sub-field may be formatted for readability in accordance with 4.1.4.(f). The length of this sub-field depends on the asymmetric algorithm used.
EDC/	Error Detection Code	The Error Detection Code when used shall be generated on all components of the associated service message using the editing, computation and formatting requirements for a MAC in ISO 8730 (see 3.3.1.1.).	9 characters (4 hex) b (4 hex).
.EDCV	Effective date of certificate	Date and Coordinated Universal Time before which a certificate is not valid.	12 numeric characters in the order YYMMDDHHMMSS
EDK/	Effective Date of Key(s)	Date and Coordinated Universal Time of KD activation.	12 numeric characters in the order YYMMDDHHMMSS
ERF/	Error Field	Error Codes are defined as : C : Cannot Process (optional. May be used as general error code where a more specific error code is not appropriate) E : Facility inoperative F : Format (syntax) error G : Reserved H : User defined I : Key Identifier not known to recipient J : Certification authority unknown K : Error detected in received key L : Certificate error M : MAC error (failure to authenticate) O : Option not implemented P : CTP error R : Duplicate message S : Signature deconstruction error W : Unavailable certificate (not previously exchanged or already destroyed by the receiver) X : EDC error (probable transmission error) Z : Algorithm not available to recipient	Up to 16 characters
.EXCV	Expiry date of certificate	Date and Coordinated Universal Time after which a certificate is not valid.	12 numeric characters in the order YYMMDDHHMMSS

.FKD	Function of Data Key	A single alphanumeric character that identifies the function(s) for which the recipient of the associated data key shall use the key after the completion of the key distribution procedure. The use of the data key in the generation or checking of a MAC as specified in clauses 4.2 and 4.3 is not limited by the FKD value of this key. The character shall be interpreted as follows : 0 : not specified 1 : authentication key (generate or check MAC) 2 : generate MAC only 3 : check MAC only 4 : cipher key (encipher or decipher) 5 : encipher only 6 : decipher only 7 through P : for assignment by ISO Technical Committee ISO/TC 68 QRSTU : for national assignment VWXYZ : for private assignment	1 character (sub-field)
.FKP	Function of Public Key	The function for which a certified public key shall be used. The following values are defined : 0 Both of the following functions 1 the encipherment of data keys 2 Deconstruction of signatures on enciphered keys or on a KSM	1 numeric character
.IABE	Identifiers of algorithms for:	Block encipherment	Up to 16 characters
.IABS		Block/KSM signature	
.IACS		Certificate signature	
.IAPK		Use of certified key	
IDA/	Identifier of Key for Authentication	Identifies the data key used to authenticate CSM.	Up to 16 characters
.IDCC	Certification Authority Identifier	Identifies (names) the certification authority which issued a Certificate Value.	Up to 16 characters
.IDCK	Certification Key Identifier	Identifies (names) the key used by the certification authority to compute a Certificate Value.	Up to 16 characters
.IDCV	Public Key Identifier	Identifies (names) a Certificate.	Up to 16 characters
IDD/	Identifier of Key to be Discontinued	Identifies symmetric key to be discontinued.	Up to 16 characters

.IDEK	Encipherment Key Identifier	Identifies (names) the certificate containing the public key used to encipher a Block.	Up to 16 characters
.IDK	Identifier of Data Key	Identifies (names) a symmetric KD. The name (if any) shall be retained by the recipient of a KSM for use in subsequent exchanges such as the IDA or IDD fields of a DSM.	Up to 16 characters (sub-field)
.IDSK	Signature Key Identifier	Identifies (names) the certificate containing the public key to be used to deconstruct the signature of a Signed Enciphered Block.	Up to 16 characters
IV/	Initialisation Vector	Starting point for encipherment/ decipherment process. The first character of the field indicates the method of transmitting the IV as follows: B: The IV is contained in the block BC which is enciphered by the asymmetric algorithm. See 3.3.3. In this case the "B" is the only character present. E: The IV is transmitted in this field and is enciphered by a symmetric algorithm. See 3.3.1.3 P: The IV is transmitted in plaintext form in this field. This character shall not form part of the IV when it is subsequently used for encipherment purposes. If the IV is enciphered by a symmetric algorithm, the IV value shall be enciphered using a data key (KD) in the same cryptographic service message. If two data keys are transmitted in the message, the second KD shall be used.	1 character ("B", "E" or "P") followed by up to 32 hex characters (16 when DEA is used). Leading zeros of an enciphered IV shall not be suppressed for transmission.
.KP	Public Key value	The value of the Public Key in a certificate.	This sub-field shall be represented in hex characters and may be formatted for readability in accordance with 4.1.4(f). The length of the sub-field depends on the asymmetric algorithm used.
MAC/	Message Authentication Code	The MAC shall be generated on all components of the associated Cryptographic Service Message using the editing, computation and format requirements of ISO 8730.	9 characters (4 hex) b (4 hex).
MCL/	Message Type	Type of Cryptographic Service Message.	3 characters: DSM, ESM, KSM, RSI, RSM.

ND/	Name of Data Key	Identifies the parity characteristics (sub-field P), function (sub-field FKD), and identity (sub-field IDK), of the associated KD. The formation of the field from its sub-field is described in 4.1.5.3. A maximum of two ND fields shall be sent per message.	Up to 20 characters. See the corresponding sub-fields for further details.
ORG/	Originator	Cryptographic Service Message originator.	4 to 16 characters.
.OWN	Owner	The name of the party to whom a certificate was originally issued.	4 to 16 characters
.P	Key Parity	A "P" in this sub-field indicates that the plaintext of the corresponding symmetric key confirms to the specification for odd parity.	Up to 1 character (sub-field).
RCV/	Recipient	Cryptographic Service Message recipient.	4 to 16 characters.
SIG/	Digital signature	A digital signature formed over all other fields of the Cryptographic Service Message.	Depends on the signature algorithm used; only hex characters.
SVR/	Service Request	Specifies type of service requested. SVR requests one data key implicitly unless CVO is present. In this case, CVO shall be the only request present. KD requests two data keys. IV requests an IV. The IV shall be enciphered unless, by prior agreement, a plaintext IV is to be sent. CV requests the certificate of the receiver. CVO requests only the certificate of the receiver without any keys. The types of certificate supplied are specified in 4.1.6.3. A minimum of one key or one Certificate Value, and a maximum of two keys, an IV and a Certificate Value may be requested in a single RSI. Types requested shall be separated by periods.	0 to 12 characters.

4.1.6.2 Flow of Cryptographic Service Messages

Message flow shall be as described below. Six sequences of message flow are described to meet the requirements of different applications of this standard. These are illustrated in figures 6(a) to 6(f). Error service messages are not shown. The response to any CSM may be an ESM except that an ESM cannot be sent in response to an ESM or to an RSM which is itself a response to a DSM.

The party responsible for the generation of symmetric data keys can initiate the sequence of messages to communicate such a key.

Alternatively, the other party may initiate the sequence.

The certified public key necessary to securely communicate the data keys may be exchanged each time a data key is communicated as part of the same sequence. Alternatively, they may be exchanged in advance using a separate sequence of messages.

NOTE — Certified public keys may be transmitted in messages of type RSI, RSM or KSM depending on the flow sequence employed. See the individual cases and also 4.1.6.3.

a) If one logical party of a communicating pair (Party B) :

1. wishes to communicate with another logical party (Party A), and

2. does not know the (signature) public key of Party A (KPA) (in this case Party A is assumed not to know the (encipherment) public key of Party B (KPB) either), and

3. is not responsible for the generation of symmetric keys,

then :

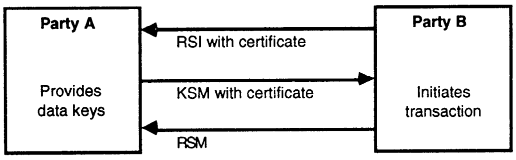

Figure 6(a)

- Party B shall send an RSI to Party A containing its own certified public key (KPB) and at least one ND field, and requesting that Party A send keys (its own certified public key KPA and at least one data key and optionally an IV) to Party B. Each ND field shall have a mandatory FDK sub-field which specifies the function(s) for which the requested data key shall be used. An ND field shall be sent for each data key requested. The ND field(s) may optionally contain other sub-fields, P or IDK, which describe attributes of the requested data key(s). If Party A receives an RSI from Party B with an error in it, an ESM shall be returned.

- Party A shall generate or acquire symmetric keys and optionally an IV. Then, it shall send a KSM to Party B, containing its certified public key, KPA, the symmetric key(s), at least one ND field and optionally the IV. The ND field(s) of the KSM shall specify the function(s) of the data key(s) supplied. The FDK sub-field(s) shall be the same as in the requesting RSI. Where the RSI specifies attributes in the P or IDK sub-fields, these shall be the same in the KSM.

- The KD(s) sent in the KSM shall be enciphered by asymmetric master keys as described in 3.3.1.5. and 3.3.3.

- The IV (if enciphered) shall be enciphered under the KD (the 2nd KD if two KDs are sent) in that KSM.

- The KSM may be authenticated using the KD(s) sent in the message. Alternatively it shall be signed with the asymmetric secret key of party A.

- Party B shall respond with an RSM if the KSM is received correctly, or with an ESM if there is an error in the received KSM. If Party A receives an RSM which contains an error(s), Party A shall return an ESM to Party B.

b) If one logical party of a communicating pair (Party A):

1. wishes to communicate with another logical party (Party B), and

2. does not know the required public key of Party B, KPB (in this case Party B is assumed not to know the required KPA),

then :

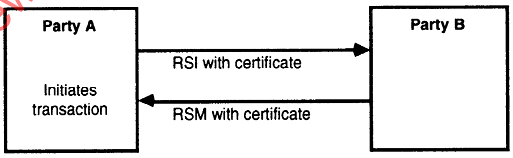

Figure 6(b)

- Party A shall send an RSI to Party B containing its own certified public key KPA and requesting that Party B send its certified public key KPB. If Party B receives an RSI from Party A with an error in it, an ESM shall be returned to Party A.

- Party B shall send an RSM to Party A containing its certified public key KPB. If Party A receives an RSM from Party B with an error in it, an ESM shall be returned to Party B.

c) If one logical party of a communicating pair (Party A):

1. wishes to send keys to another party (Party B), and

2. knows the (encipherment) public key of Party B, KPB (in this case Party B is assumed to know the (signature) public key of Party A, KPA),

3. has the responsibility for the generation of symmetric keys,

then :

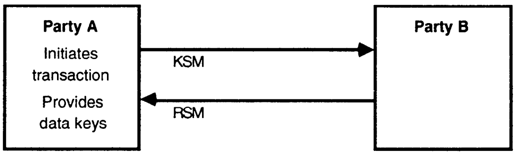

Figure 6(c)

- Party A shall send a KSM as described in a), excepted that this KSM may not contain a certificate,

- Party B shall send an RSM as described in a).

d) If one logical party of a communicating pair (Party B):

1. wishes to communicate with another logical party (Party A), and

2. knows the (signature) public key of Party A, KPA (in this case Party A is assumed to know the (encipherment) public key of Party B (KPB) as well), and

3. is not responsible for the generation of symmetric keys,

then :

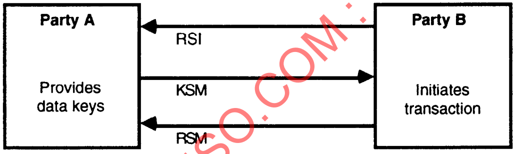

Figure 6(d)

Both parties shall proceed as described in a), excepted that the RSI and the KSM may not contain certificates or certificate requests.

e) If one logical party of a communicating pair (Party A) :

1. wishes to communicate with another logical party (Party B), and

2. does not know the (encipherment) public key of Party B, KPB (in this case Party B is assumed not to know the (signature) public key of Party A, KPA), and

3. is responsible for the generation of symmetric keys, and

4. neither party wishes to store the other's public key,

then :

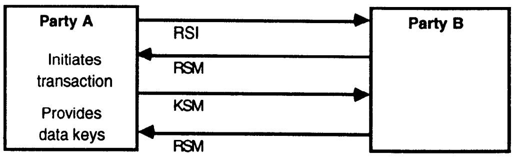

Figure 6(e)

- Party A shall send an RSI to Party B requesting only that Party B send its certified public key KPB. If Party B receives an RSI from Party A with an error in it, an ESM shall be returned to Party A.

- Party B shall send an RSM to Party A containing its certified public key KPB. If Party A receives an RSM from Party B with an error in it, an ESM shall be returned to Party B.

- Party A shall send a KSM as described in a).

- Party B shall send an RSM as described in a).

f) If either logical party of a communicating pair (Party A) wishes to terminate a keying relationship or wishes to discontinue the use of specific symmetric keys,

then :

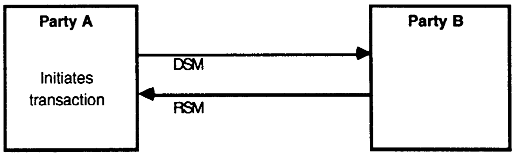

Figure 6(f)

- Party A shall send a DSM to party B.

- Party B shall respond with an RSM if the DSM is received correctly, and all information contained in the DSM was applicable. Otherwise, Party B shall respond with an ESM.

- When a DSM is sent, the key named by the IDA field (or the only data key shared between the originating and recipient parties if no IDA is present) shall be retained to authenticate the subsequent RSM. That key shall then be discontinued.

- When an RSM is received in error, no ESM shall be sent and manual recovery procedures are required.

A DSM cannot terminate or discontinue the use of any asymmetric key. This shall be handled using procedures defined by the certification authority.

Each of the above cases is illustrated by an example in annex B.

Table 3 — Fields used with each message type

Message type	RSI	RSM	KSM	RSM	ESM	DSM	RSM	ESM
Responding to	-	RSI	-	KSM	KSM	-	DSM	DSM RSI RSM
Reference subclause	4.2.5 4.3.5	4.2.6 4.3.6	4.2.4 4.3.4	4.2.6 4.3.6	4.2.3 4.3.3	4.2.2 4.3.2	4.2.6 4.3.6	4.2.3 (generation) 4.3.3 (processing)
See attached rules								
	MCL	MCL	MCL	MCL	MCL	MCL	MCL	MCL
	RCV	RCV	RCV	RCV	RCV	RCV	RCV	RCV
	ORG	ORG	ORG	ORG	ORG	ORG	ORG	ORG
6 & 7						IDD	IDD	
1 & 9					IDA	IDA		IDA
		CV						
1	CV		CV					
3			BE					
3			BES					
1, 2 & 8	ND		ND					
1			IV					
1			EDK					
	SVR							
			CTP		CTP			
5					CTR			
					ERF			ERF
1	EDC	EDC			EDC			EDC
4			MAC	MAC		MAC	MAC	
3			SIG					

NOTE — The order of the fields shall be as shown, starting with MCL.

Rules

1. Optional.
2. A maximum of two such fields may be sent in a CSM.
3. In message type KSM there are two alternative formats :
   (a) BES and MAC shall be present but not SIG. In this format, BE is optional.
   (b) BE and SIG shall be present but not BES or MAC.
4. For the use of MAC in message type KSM, see rule 3.
5. Required if and only if a count error occurs.

6. Any number of such fields may be sent in a DSM. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued.

7. The number of IDD fields in the RSM shall be equal to the number of IDD fields in the DSM to which this RSM responds. Each IDD field shall either contain the identity of a discontinued key or shall be a null field. A null field indicates that the keying relationship shall be discontinued.

8. The use of an ND field in an RSI message is specified in 4.1.6.2, case a).

9. One or two IDA fields may be present in an ESM responding to a CSM which contained a MAC field.

NOTE — Examples of fields contained in a KSM :

(1) MCL, RCV, ORG, CV, BE, ND, CTP, SIG

(2) MCL, RCV, ORG, CV, BE, BES, ND, CTP, MAC

(3) MCL, RCV, ORG, CV, BES, ND, CTP, MAC

Detailed examples of messages are shown in annex B.

4.1.6.3 Certificate selection

Where different algorithms are used for asymmetric encipherment and for digital signatures, the following rules define the type of certificate (as defined by the FKP sub-field of the certificate value) to be inserted into a cryptographic service message.

a) If the first certificate in a sequence of messages defined in 4.1.6.2 is for an encipherment public key, the response shall provide a certificate for a signature public key and if the first certificate is for a signature public key, the responding party shall provide a certificate for an encipherment public key.

b) An RSI which contains the value "CV" in the SVR field, shall always provide a certificate for an encipherment public key in its CV field (see 4.1.6.2 (a)).

c) In the event that parties are exchanging certificates for future use (see 4.1.6.2 (b)) then the RSI shall contain the value "CVO" in the SVR field and shall contain a certificate for a signature public key in the CV field. Exceptionally, if the originator of the RSI expects the other party to be responsible for the generation of symmetric keys (see 4.1.6.2 (d)), then the CV field shall contain a certificate for an encipherment public key.

d) If an RSI contains the value "CVO" in the SVR field, but does not itself offer a certificate, then the corresponding RSM shall contain a certificate for an encipherment public key (see 4.1.6.2 (e)).

4.2 Generation of Cryptographic Service Messages

4.2.1 Determination of message type

The message type of the outgoing Cryptographic Service Message is specified by the field tag which appears in the first field of that message, as shown below. (The reference is to the subclause of this part of ISO 11166 which shall be used in generating each type of Cryptographic Service Message.)

Message type	MCL field contents	Reference subclause
Disconnect Service Message	DSM	4.2.2
Error Service Message	ESM	4.2.3
Key Service Message	KSM	4.2.4
Request Service Initiation message	RSI	4.2.5
Response Service Message	RSM	4.2.6

Thus, following the rules given in sublause 4.2.4, a Disconnect Service Message commences with:

CSM(MCL/DSM…

4.2.2 Generate Disconnect Service Message (DSM)

A Disconnect Service Message (DSM) is generated in order to discontinue one or more keys or to terminate a keying relationship. It may be sent by either party of the relationship.

Disconnect Service Messages shall be generated by computing or selecting field contents in accordance with table 4.

Table 4 — Contents of fields in DSM

Field tag	Content
MCL	Insert DSM in the field. The field becomes : MCL/DSM
RCV	Insert recipient's identity in the field, e.g. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field, e.g. if OOOO is the identity, the field becomes: ORG/OOOO
IDD	Any number of such fields may be sent. Insert the identity of the symmetric key to be discontinued. Use a separate IDD field for each such key to be discontinued. If a keying relationship is to be terminated, the IDD field shall be null.
IDA	(Not required if the originating and recipient parties share one and only one data key) Insert the identity of the symmetric key to be used to authenticate this DSM. Note that this key shall also be named in an IDD field (unless the IDD field is null and the keying relationship is to be discontinued).
MAC	The MAC field contents shall be computed using the KD as follows: aKD{MCL/DSMb...bIDA/IDK1b} and using brackets to denote the representation of field contents, the field becomes: MAC/[aKD{MCL/DSMb...bIDA/IDK1b}] When the DSM has been generated, the key(s) named in the IDD field(s) may be discontinued (excluding the key used to authenticate this message which shall be retained to authenticate the RSM responding to this DSM). Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the MAC field.

4.2.3 Generate Error Service Message (ESM)

An Error Service Message (ESM) is sent in response to the detection of one or more of the error conditions in a Cryptographic Service Message (other than an ESM) as listed in table 2, ERF (see 4.1.6.1).

Error Service Messages shall be generated by computing or selecting field contents in accordance with table 5.

Table 5 — Contents of fields in ESM

Field tag	Content
MCL	Insert ESM in the field. The field becomes : MCL/ESM
RCV	Insert recipient's identity in the field, e.g. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field, e.g. if OOOO is the identity, the field becomes: ORG/OOOO
IDA	(Used when responding to a CSM protected by a MAC. Not required if the originating and recipient parties share one and only one data key). Insert the name of the data key used to verify the MAC of the CSM to which the ESM responds. If the ESM is reporting an error in a KSM which contained two data keys (or its corresponding RSM), then name both keys using two successive IDA fields. Using brackets to represent the contents of a field, the field becomes : IDA/[IDK]
CTP	(Used only when responding to a KSM) The count returned in this field shall contain the expected CTP (see 3.3.2). If the value is "p", the field becomes: CTP/p
CTR	(Used when a count error has been detected in a KSM) The count returned is the CTP count included in the message to which this ESM responds. The received count shall be copied from the previous message and inserted in the CTR field.
ERF	The contents of the ERF field are defined by the error conditions detected by the originator of this ESM. See the definition of ERF field contents in 4.2.6 (table 2). Multiple error conditions are indicated by returning a concatenated string of error flags, e.g. ERF/KPM
EDC	(Optional) The data key for EDC computation shall be as specified in 3.3.1.1 : The EDC is computed using: EDC = aKDX{MCL/ESMb...bERF/KPMb} and using brackets to denote the representation of field contents, the field becomes: EDC/[aKDX{MCL/ESMb...bERF/KPMb}] Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the EDC field.

4.2.4 Generate Key Service Message (KSM)

A Key Service Message (KSM) may be generated spontaneously or in response to an RSI received from another party (see 4.1.6.2).

The expected responses to a KSM are either an RSM or an ESM from the intended recipient of the KSM. If either message is not received within a predetermined period of time, a KSM containing the same data, but with an incremented value of the counter CTP, may be sent for a given number of times. Key Service Messages shall be generated by computing or selecting field contents in accordance with table 6.

For the use of BE, BES, MAC and SIG in a KSM, see the rules attached to table 3.

Table 6 — Contents of fields in KSM

Field tag	Content
MCL	Insert KSM in the field. The field becomes : MCL/KSM
RCV	Insert recipient's identity in the field, e.g. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field, e.g. if OOOO is the identity, the field becomes: ORG/OOOO
CV	(Optional)
Optional sub-fields :	
IDCK IDCC EDCV EXCV OWN IDCV KP FKP IACS IAPK	Insert the certificate value comprising the signature plus any of the listed sub-fields as received from the certification authority (CKC) and formatted according to the rules of 4.1.5. In systems which use different algorithms for asymmetric encipherment and for digital signatures, the rules of 4.1.6.3 require the certificate to contain a signature public key.
BE	May be omitted if BES is present (see rule 3 of table 3)
Optional sub-fields:	
IDEK IDSK	If it is desired to name the certificates of the key used to encipher the block and/or the key to be used to deconstruct the signature of the CSM (or both), form and insert the applicable sub-field(s) using the rules of 4.1.5.
IABE IABS	If it is desired to identify the algorithm used to encipher the block or to sign the KSM, form and insert the applicable sub-field(s), using the rules of 4.1.5.
	Construct a block BC as described in 11.3.1 for a single KD or in 3.3.3.2 for two KDs. Let BC be the result of this operation. Use the procedure of 3.3.1.5 to compute the BE.
	Using brackets to denote representation of field contents, the field becomes:
	BE/[BE(optional sub-fields)]

BES		Required if BE is not present (see rule 3 of table 3)
	Optional sub-fields:	
	IDEK IDSK	If it is desired to name the certificates of the key used to encipher the block and/or the key to be used to deconstruct the signature (or both), form and insert the applicable sub-field(s) using the rules of 4.1.5.
	IABE IABS	If it is desired to identify the algorithm used to encipher the block or to sign the enciphered block, form and insert the applicable sub-field(s), using the rules of 4.1.5.
		Construct a block BC as described in 3.3.3.1 for a single KD or in 3.3.3.2 for two KDs. Let BC be the result of this operation. Use the procedure of 3.3.1.7. to compute the BES.
		Using brackets to denote representation of field contents, the field becomes: BES/[BES(optional sub-fields)]
ND		(Optional)
	Optional sub-fields:	
	P FKD IDK	If it is desired to use odd parity feature, to specify the function of the KD and/or to name the KD, form and insert the applicable sub-fields.
		An ND field indicates the presence in the BE(S) field of a KD.
		If an FKD sub-field is present, it shall equal the FKD value for the corresponding key in the block BC. If no FKD sub-field is present, the FKD value for the corresponding key in block BC shall be 0.
		If the KSM is sent in response to an RSI requesting keys, the ND field(s) of the KSM shall contain the same FDK sub-field(s) as in the RSI field ND. IF the RSI ND specifies P or an IDK value is given, this or these shall be identical in the KSM. The attributes of the supplied key(s) shall correspond to those designated in the RSI field ND.
		The field becomes: ND/[optional sub-fields)]
IV		(Optional)
		Case 1 : Asymmetrically enciphered IV
		If the IV value is included in an asymmetrically enciphered block (BE or BES), then the field is : IV/B
		Case 2 : Symmetrically enciphered IV
		If an IV is sent enciphered by a symmetric algorithm, the IV shall be enciphered using the KD sent in the Cryptographic Service Message (the second KD if two are sent) using the equation : enciphered IV = eKD{IV}
		and the field is: IV/[E ll eKD{IV}]
		Case 3: Plaintext IV
		If an IV is sent in plaintext form, then the field is: IV/[P ll IV)]

EDK	(Optional) The EDK field shall be determined by the originating party. The contents of the EDK field shall be: YYMMDDHHMMSS and the field becomes: EDK/[YYMMDDHHMMSS]
CTP	If the value of the CTP before KSM preparation is "p", then the KSM shall contain the CTP field: CTP/p The CTP value shall equal the CTP contained in block BC.
MAC	Required if BES is present (see rule 3 of table 3) The MAC is always computed using the KDs sent in the message. If only one KD is sent, KDJ, then that key shall be used. If two KDs are sent (KDH and KDI), then the key, KDJ (used to authenticate the Cryptographic Service Message) is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ{MCL/KSMb...bCTP/pb} and the field becomes: MAC/[aKDJ{MCL/KSMb...bCTP/pb}] Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the MAC field.
SIG	Required if BES in not present (see rule 3 of table 3) The digital signature is computed using the secret signature key of the originator. The signature is computed on the character string: MCL/KSMb...bCTP/pb i.e., starting with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the SIG field. The SIG field becomes : SIG/[σKS1{MCL/KSMb...bCTP/pb}]

4.2.5 Generate Request Service Initiation message (RSI)

A Request Service Initialisation message (RSI) is generated by the originating party in order to request either that keys be sent in a subsequent KSM to establish a keying relationship or that a certificate be sent in a subsequent RSM.

Request Service Initiation messages shall be generated by computing or selecting field contents in accordance with table 7.

In systems which use separate encipherment public keys and signature public keys, see 4.1.6.3. for the rules concerning the type of certificate inserted or requested.

Table 7 — Contents of fields in RSI

Field tag	Content
MCL	Insert RSI in the field. The field becomes: MCL/RSI
RCV	Insert recipient's identity in the field, e.g. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field, e.g. if OOOO is the identity, the field becomes: ORG/OOOO
CV	(Optional)
Optional sub-fields :	
IDCK IDCC EDCV EXCV OWN IDCV KP FKP IACS IAPK	Insert the certificate value comprising the signature plus any of the listed sub-fields as received from the certification authority (CKC) and formatted according to the rules of 4.1.5.
ND	(Mandatory if the RSI requests a data key or 2 data keys, otherwise absent)
Sub-fields:	
P FKD IDK	Insert the sub-fields designating the properties of the requested key. The only sub-field required is FDK, which specifies the function(s) of the requested key(s). P and IDK sub-fields are optional.
SVR	Insert the sub-fields designating the type of service requested (see 4.1.6 and table 2, SVR). Note that a single data key is implicitly requested by the presence of an SVR field unless CVO is a sub-field, e.g. SVR/CVO to request a certificate value SVR/CV to request a certificate value and a single data key SVR/ to request one data key SVR/KD to request two data keys.
EDC	(Optional) The data key for EDC computation shall be as specified in 3.3.1.1 : The EDC is computed using: EDC = aKDX{MCL/RSIb...bSVR/KD.IVb} and using brackets to denote the representation of field contents, the field becomes: EDC/[aKDX{MCL/RSIb...bSVR/KD.IVb}] Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the EDC field.

4.2.6 Generate Response Service Message (RSM)

A Response Service Message (RSM) is generated following receipt of an acceptable DSM or KSM or RSI. Response Service Messages shall be generated by computing or selecting field contents in accordance with table 8.

Table 8 — Contents of fields in RSM

Field tag	Content
MCL	Insert RSM in the field. The field becomes : MCL/RSM
RCV	Insert recipient's identity in the field, e.g. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field, e.g. if OOOO is the identity, the field becomes: ORG/OOOO
IDD	(Only in response to a DSM) Copy the IDD field(s) from the DSM to which this RSM responds.
CV	(Only in response to an RSI) (Optional)
Optional	sub-fields :
IDCK IDCC EDCV EXCV OWN IDCV KP FKP IACS IAPK	Insert the certificate value comprising the signature plus any of the listed sub-fields as received from the certification authority (CKC) and formatted according to the rules of 4.1.5.
MAC	(Not in response to an RSI) The MAC is always computed using the KDs sent or specified in the DSM or KSM to which the RSM responds. If only one KD is sent or specified, KDJ, then that key shall be used. When responding to a DSM, the key, KDJ, shall be the key identified in the IDA field or the only data key shared between the originating and recipient parties. If two KDs are sent (KDH and KDI), then the key, KDJ (used to authenticate the Cryptographic Service Message) is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ{MCL/RSMb...bIDD/iddb} responding to a DSM aKDJ{MCL/RSMb...bORG/OOOOb} responding to a KSM and the field becomes: MAC/[aKDJ{MCL/RSMb...bIDD/iddb}] responding to a DSM MAC/[aKDJ{MCL/RSMb...bORG/OOOOb}] responding to a KSM Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the MAC field. When the RSM has been generated in response to a DSM, the key(s) identified in the IDD field(s) of the DSM (or used to compute the MAC if the IDD field is null and no IDA field is present) shall be discontinued.

EDC	(Only in response to an RSI) The data key for EDC computation shall be as specified in 3.3.1.1 : The EDC is computed using: EDC = aKDX{MCL/RSIb...bCV/cvb} and using brackets to denote the representation of field contents, the field becomes: EDC/[aKDX{MCL/RSIb...bCV/cvb}] Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the EDC field.

4.3 Processing Cryptographic Service Messages

4.3.1 Determination of message type

The message type of the Cryptographic Service Message is determined by the field tag which appears in the first field of that message, as shown below. (The reference is to the sublause of this part of ISO 11166 which shall be used in processing each type of Cryptographic Service Message.)

Message type	MCL field contents	Reference subclause
Disconnect Service Message	DSM	4.3.2
Error Service Message	ESM	4.3.3
Key Service Message	KSM	4.3.4
Request Service Initiation message	RSI	4.3.5
Response Service Message	RSM	4.3.6

Thus, following the rules given in 4.1.4, a Disconnect Service Message commences with:

CSM(MCL/DSM...

If the MCL field contains a value other than those listed above, an error condition exists, and an ESM shall be returned with

ERF/F

If the identity of a party sending a Cryptographic Service Message is not known to the recipient, an ESM may be sent or the problem may be resolved by other means. Where an ESM is sent, it shall have

ERF/C

4.3.2 Process Disconnect Service Message (DSM)

A Disconnect Service Message (DSM) notifies the recipient of the DSM that one or more keys are to be terminated. Responses to the DSM are either an RSM if the DSM is received with no errors, or an ESM if errors are detected in the DSM.

Disconnect Service Message shall be processed by computing or selecting field contents in accordance with table 9.

Table 9 — Processing of DSM

Field tag	Action
MCL	Confirm field is MCL/DSM.
RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the DSM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
IDD	If an IDD field is null and the DSM processes correctly, then the keying relationship shall be terminated. If not null, the IDD field contains the identity of a KD to be discontinued. The IDD field(s) shall be inserted into the RSM generated in response to this Cryptographic Service Message in the same sequence as in the DSM. If the IDD is not known to the recipient, this shall cause processing of the DSM to cease and the generation and transmission to the originating party of an ESM with "I" in the ERF field, i.e. ERF/I
IDA	The IDA is the identity of the KD used to compute the MAC. This same KD shall be used to authenticate the RSM generated in response to this Cryptographic Service Message. If the IDA field is not present, the data key to be used in computing the MAC shall be the only data key shared between the originating and recipient parties. The key named in the IDA field (if present) or the only data key shared by the two parties shall be discontinued after generation of the RSM that responds to this DSM, even if it is erroneously not named in the IDD field. If the key named by the IDA field is not known to the recipient, this shall cause processing of the DSM to cease and the generation and transmission to the originating party of an ESM with "I" in the ERF field, i.e. ERF/I
MAC	Compute a MAC from the message. The KD that shall be used in the MAC computation is the KD identified in the IDA field. The MAC is then: aKD{MCL/DSMb...bIDA/IDK1b} If the computed MAC does not equal the received MAC, the message fails to authenticate and either: a) an ESM shall be generated and returned to the originator. The ERF field shall include an "M", i.e. ERF/M or b) the error shall be resolved by manual means (e.g. telephone). Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the MAC field.

4.3.3 Process Error Service Message (ESM)

An Error Service Message (ESM) is received in response to a DSM, KSM, RSI, or RSM due to the detection of one or more of the error conditions of a CSM, as listed in table 2, ERF (see 4.1.6.1).

Error Service Messages shall be processed by computing or selecting field contents in accordance with table 10.

Table 10 — Processing of ESM

Field tag	Action
MCL	Confirm field is MCL/ESM.
RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the ESM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
IDA	(Used only when responding to a CSM protected by a MAC) Parse the field to identify the key(s) used by the originator of the ESM in reporting an error in a CSM. Use the result to assist in the diagnosis of the reason for the error and/or to identify the CSM to which it responds.
CTP	(Used only when responding to a KSM) Process the value of CTP field according to 3.3.2. If the field is CTP/p, the CTP value is "p".
CTR	Process the value of CTR field. If the field is CTR/r, the CTR value is "r". The value of CTR may be used in determining the Cryptographic Service Message to which this ESM responds.
ERF	Parse the field to obtain the designators for the types(s) of errors reported. These error type(s) shall be utilised in generating the Cryptographic Service Message that responds to this ESM or in the manual recovery process, if necessary. See the definition of ERF field contents in 4.1.6.1 (table 2). Multiple error conditions are indicated by a string of concatenated error flags, e.g. ERF/KPM
EDC	(When present) If this option is not implemented the field shall be disregarded and message processing may proceed. If this option is implemented, compute an EDC from the message using the data key for EDC computation (see 3.3.1.1) : EDC = aKDX{MCL/ESM$\underline{b}$...$\underline{b}$ERF/KPM$\underline{b}$} If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error). The error shall be logged and resolved by other means (e.g. telephone). Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "$\underline{b}$", preceding the EDC field.

4.3.4 Process Key Service Message (KSM)

A Key Service Message (KSM) is received from a party in order to either :

a) establish a keying relationship and begin communications; or

b) initiate a key change in an existing relationship.

Responses to the KSM are either an RSM if the KSM is received with no errors, or an ESM if errors are detected in the KSM.

Key Service Messages shall be processed by computing or selecting field contents in accordance with table 11.

Table 11 — Processing of KSM

Field tag	Action
MCL	Confirm field is MCL/KSM.
RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the ESM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
CV	(When present) Using the rules of 4.1.5.4., parse the field to obtain the CV, and the sub-fields present.
Sub-fields (when present)	
IDCK	The IDCK sub-field, if present, defines the key to be used in the CV verification. Otherwise the public key to be used is implicitly defined.
IDCC	The IDCC sub-field, if present, defines the certification authority which produced this certificate. Otherwise, the certification authority is implicitly defined.
EDCV	The EDCV sub-field, if present, identifies the earliest date and time at which the public key contained in the certificate shall be used. Otherwise the public key is valid from the moment at which the certificate is issued. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
EXCV	The EXCV sub-field, if present, identifies the latest date and time at which the public key contained in the certificate shall be used. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
OWN	The OWN sub-field, if present, identifies the name of the party to whom the certificate was originally issued. It shall be consistent with the name of the originator in the ORG field. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
IDCV	The IDCV sub-field, if present, identifies the certificate.
KP	(present in this sub-field if the certificate value contains a separate signature) After the CV has been verified, this public key is used to deconstruct the signature in the BES or SIG field.
FKP	The FKP sub-field, if present, identifies the use of the KP in the certificate.
IACS	The IACS sub-field, if present, specifies the identity of the algorithm used by the certification authority to sign certificates.

	IAPK	The IAPK sub-field, if present, specifies the identity of the algorithm used by the public key which is being certified.
	Error processing	
		If an IACS, IAPK, IDCC or IDCK is not known to the recipient, this causes the processing of the CSM to cease, and the generation and transmission to the originating party of an ESM as follows : ERF/Z, ERF/Z, ERF/J or ERF/I respectively. If EDCV indicates that the key contained in the certificate shall not be used at the current date and time, or if the recipient has been notified that the certificate with identity IDCV has been withdrawn, processing of the CSM shall cease and an ESM shall be generated and transmitted to the originating party with an "L" in the ERF field, i.e. ERF/L This certificate value shall be checked using the rules of 11.4. If the check fails an ESM shall be generated and sent to the originator with an "L" in the ERF field, i.e. ERF/L
BE		(When present) Using the rules of 4.1.5.1., parse the field to obtain the enciphered BC, and the IDEK IDSK IABE and IABS sub-fields if present.
	IDEK	The IDEK sub-field (if present) defines the certificate of the public key used to encipher the block. Otherwise, the IDEK used is implicitly defined. If the certificate with identity IDEK is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of an ESM with an "W" in the ERF field, i.e. ERF/W
	IDSK	The IDSK sub-field (if present) defines certificate of the public key to be used to deconstruct the signature of the message. Otherwise, the IDSK to be used is implicitly defined. If the certificate with identity IDSK is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of an ESM with an "W" in the ERF field, i.e. ERF/W
	IABE IABS	If an IABE or IABS sub-field is present and the algorithm specified is not implemented by the recipient, processing of the KSM shall cease and an ESM shall be generated and transmitted to the originating party with a "Z" in the ERF field, i.e. ERF/Z
	Block processing	
		Having identified the keys and algorithms, process the block as described in 3.3.1.6.
	ORG	If the KSM is protected with a SIG field, compare the ORG value in the block with the value in the ORG field of the KSM. If they differ, an ESM shall be generated in response to the KSM with a "K" in the ERF field, i.e. ERF/K
	CTP	Compare the CTP value found in the block with the value of the field CTP. If they differ, an ESM shall be generated in response to the KSM, with a "K" in the ERF field, i.e. ERF/K

FKD	Compare the FKD value(s) found in the block with the FKD sub-fields of (corresponding) ND field(s). If they differ, an ESM shall be generated in response to the KSM, with a "K" in the ERF field, i.e. ERF/K All further processing of the message shall cease when an error is found.
BES	(When present) Using the rules of 4.1.5.2., parse the field to obtain the enciphered and signed BC, and the IDEK IDSK IABE and IABS sub-fields if present.
IDEK	The IDEK sub-field (if present) defines the certificate of the public key used to encipher the block. Otherwise, the IDEK used is implicitly defined. If the certificate with identity IDEK is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of an ESM with an "W" in the ERF field, i.e. ERF/W
IDSK	The IDSK sub-field (if present) defines the certificate of the public key to be used to deconstruct the signature of the block. Otherwise, the IDSK to be used is implicitly defined. If the certificate with identity IDSK is not known to the recipient, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of an ESM with an "W" in the ERF field, i.e. ERF/W
IABE IABS	If an IABE or IABS sub-field is present and the algorithm specified is not implemented by the recipient, processing of the KSM shall cease and an ESM shall be generated and transmitted to the originating party with a "Z" in the ERF field, i.e. ERF/Z
Block processing	
	Having identified the keys and algorithms, process the block as described in 3.3.1.6 and 3.3.1.8. If, during this process, the signature is determined as invalid, this shall cause the processing of the KSM to cease and the generation and transmission to the originating party of an ESM with an "S" in the ERF field, i.e. ERF/S
CTP	Compare the CTP value found in the block with the value of the field CTP. If they differ, an ESM shall be generated in response to the KSM, with a "K" in the ERF field, i.e. ERF/K
FKD	Compare the FKD value(s) found in the block with the FKD sub-fields of (corresponding) ND field(s). If they differ, an ESM shall be generated in response to the KSM, with a "K" in the ERF field, i.e. ERF/K All further processing of the message shall cease when an error is found.

ND	(When present) Using the rules of 4.1.5.3., parse the field to obtain the P, FKD and the IDK sub-fields if present. Optional sub-fields : P : If the "P" sub-field is present, the plaintext key shall conform to the specification for odd parity. If, on decipherment, the key does not conform to the specification for odd parity, an ESM shall be generated in response to the KSM, with a "K" in the ERF field, i.e. ERF/K All further processing of the message shall cease. If the FKD sub-field is present, the corresponding data key shall be employed only as specified in this sub-field. An exception is that keys recovered from the KSM shall be used in the computation and checking of the MACs of the KSM and the responding RSM (see 4.2.4 and 4.2.6). IDK : If present, this sub-field names the KD (see 4.1.5).
IV	(When present) If the first and only character is “B” then the IV is recovered from block BC which is obtained by processing field BE as described in 3.3.1.6 or field BES as described in 3.3.1.8. If the first character is “E”, then the IV that follows is enciphered by a KD in the message using a symmetric algorithm and shall be deciphered using the equation in 3.3.1.3 and a KD recovered from the message. If there are two KDs in the message, the second one shall be used. If the first character is “P”, then the IV that follows does not require decipherment before use. If the first character is other than “B”, “E” or “P”, or if the field contains too many characters or if the remaining characters are not members of the set (0-9), (A-F), then an error condition exists An ESM shall be generated and sent to the originator of the message with an “F” in the ERF field, i.e. ERF/F Further processing of the Cryptographic Service Message may continue prior to ESM transmission.
EDK	(When present) The EDK, if received, is the date and time at which the KDs received in the message shall be placed in use.
CTP	Process the value of the CTP field (see 3.3.2). If the field is CTP/p, the CTP value is "p". If a CTP error is detected, an ESM shall be generated to notify the originating party of the CTP error condition. The ESM shall have a "P" in the ERF field, i.e. ERF/P Further processing of the KSM may continue.

MAC	(When present) Compute a MAC from the message. The MAC is always computed using the KDs received in the message. If only one KD is received, KDJ, then that key shall be used. If two KDs are received, KDH and KDI, then the key, KDJ, used to authenticate the Cryptographic Service Message is derived from the equation: KDJ = (KDH + KDI) The MAC is then: aKDJ{MCL/KSMb...bCTP/pb} If the computed MAC does not equal the received MAC, the message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M", i.e. ERF/M Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the MAC field.
SIG	(When present) The SIG is computed over the message in the form of the character string MCL/KSMb...bCTP/pb i.e., starting with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and ending with the space,"b", preceding the SIG. The signature shall be verified using the signature public key of the originator, the certificate of which is identified by IDSK in the BE field. If the signature is not valid, an ESM shall be generated and returned to the originator. The ERF field shall include an "S", i.e. ERF/S

4.3.5 Process Request Service Initiation message (RSI)

Request Service Initiation messages (RSI) are received in order to request that keys be generated and sent to the originating party in a subsequent KSM and/or to request or transmit a certificate.

An RSI requesting a Certificate Value Only (CVO) shall result in an RSM being sent to the originating party (see 4.1.6.2 and 4.1.6.3).

If an error in the RSI is detected by the recipient, then an ESM shall be returned.

Request Service Initiation messages shall be processed by computing or selecting the field contents in accordance with table 12.

Table 12 — Processing of RSI

Field tag		Action
MCL		Confirm field is MCL/RSM.
RCV		If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the ESM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
ORG		If the field is ORG/OOOO, then OOOO is the identity of the originator.
CV		(When present) Using the rules of 4.1.5.4., parse the field to obtain the CV, and the sub-fields present.
	Sub-fields (when present)	
	IDCK	The IDCK sub-field, if present, defines the key to be used in the CV verification. Otherwise the public key to be used is implicitly defined.
	IDCC	The IDCC sub-field, if present, defines the certification authority which produced this certificate. Otherwise, the certification authority is implicitly defined.
	EDCV	The EDCV sub-field, if present, identifies the earliest date and time at which the public key contained in the certificate shall be used. Otherwise the public key is valid from the moment at which the certificate is issued. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
	EXCV	The EXCV sub-field, if present, identifies the latest date and time at which the public key contained in the certificate shall be used. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
	OWN	The OWN sub-field, if present, identifies the name of the party to whom the certificate was originally issued. It shall be consistent with the name of the originator in the ORG field. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
	IDCV	The IDCV sub-field, if present, identifies the certificate.
	KP	(present in this sub-field if the certificate value contains a separate signature) After the CV has been verified, this public key is used to deconstruct the signature in the BES or SIG field.
	FKP	The FKP sub-field, if present, identifies the use of the KP in the certificate.
	IACS	The IACS sub-field, if present, specifies the identity of the algorithm used by the certification authority to sign certificates.
	IAPK	The IAPK sub-field, if present, specifies the identity of the algorithm used by the public key which is being certified.

	Error processing If an IACS, IAPK, IDCC or IDCK is not known to the recipient, this causes the processing of the CSM to cease, and the generation and transmission to the originating party of an ESM as follows : ERF/Z, ERF/Z, ERF/J or ERF/I respectively. If EDCV indicates that the key contained in the certificate shall not be used at the current date and time, or if the recipient has been notified that the certificate with identity IDCV has been withdrawn, processing of the CSM shall cease and an ESM shall be generated and transmitted to the originating party with an "L" in the ERF field, i.e. ERF/L This certificate value shall be checked using the rules of 3.3.4. If the check fails an ESM shall be generated and sent to the originator with an "L" in the ERF field, i.e. ERF/L
ND	(When present) Using the rules of 4.1.5.3 parse the field to obtain the designators of the key(s) requested. These attributes shall be used in generating data keys for the responding KSM.
SVR	Parse the field to obtain the designators for the types of service requested. These service types shall be utilised in generating the KSM, or RSM that responds to this RSI. Note that a single data key is implicitly requested by the presence of an SVR field unless CVO is among the requests. Service requests are defined in 4.1.6.1. (table 2, SVR).
EDC	(When present) If this option is not implemented, either : a) an ESM shall be generated and returned to the originator with an "O" in the ERF field, i.e. ERF/O or b) the field shall be disregarded and message processing may proceed. If this option is implemented, compute an EDC from the message using the data key for EDC computation (see 3.3.1.1) : EDC = aKDX{MCL/RSIb...bSVR/KD.IVb} If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error). An ESM shall be generated and returned to the originator. The ERF field shall include an "X", i.e. ERF/X Input to the authentication algorithm starts with the first character following the left parenthesis,"(", of the Cryptographic Service Message, and continues through the space, "b", preceding the EDC field.

4.3.6 Process Response Service Message (RSM)

A Response Service Message (RSM) is received as an authenticated acknowledgement of a DSM or a KSM or as a non-authenticated response to an RSI. RSMs hall be processed by computing or selecting the field contents in accordance with table 13.

Table 13 — Processing of RSM

Field tag	Action
MCL	Confirm field is MCL/RSM.
RCV	If the field is RCV/RRRR, then RRRR is the identity of the recipient. If RRRR is not the identity of the party receiving the ESM for processing, then the Cryptographic Service Message has been misrouted and shall not be processed further. The routing problem shall be resolved outside of this protocol.
ORG	If the field is ORG/OOOO, then OOOO is the identity of the originator.
IDD	(When present) If no IDD field is present, this RSM is in response to a KSM or an RSI. If an IDD field is present, this RSM is in response to a DSM. If the content of the IDD field is null, this indicates that the keying relationship shall be discontinued. Otherwise, each IDD field contains the identity of a discontinued KD. If the IDD does not match one of the IDD fields sent in the DSM to which this RSM responds, this shall cause processing of the RSM to cease, and manual recovery procedures shall be used to resolve the discrepancy.
CV	(When present) Using the rules of 4.1.5.4., parse the field to obtain the CV, and the sub-fields present.
Sub-fields (when present)	
IDCK	The IDCK sub-field, if present, defines the key to be used in the CV verification. Otherwise the public key to be used is implicitly defined.
IDCC	The IDCC sub-field, if present, defines the certification authority which produced this certificate. Otherwise, the certification authority is implicitly defined.
EDCV	The EDCV sub-field, if present, identifies the earliest date and time at which the public key contained in the certificate shall be used. Otherwise the public key is valid from the moment at which the certificate is issued. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
EXCV	The EXCV sub-field, if present, identifies the latest date and time at which the public key contained in the certificate shall be used. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
OWN	The OWN sub-field, if present, identifies the name of the party to whom the certificate was originally issued. It shall be consistent with the name of the originator in the ORG field. If the CV is computed using the text recovery mode (see 3.3.4.2) the value recovered from the signature shall take precedence over this value.
IDCV	The IDCV sub-field, if present, identifies the certificate.
KP	(present in this sub-field if the certificate value contains a separate signature) After the CV has been verified, this public key is used to deconstruct the signature in the BES or SIG field.
FKP	The FKP sub-field, if present, identifies the use of the KP in the certificate.
IACS	The IACS sub-field, if present, specifies the identity of the algorithm used by the certification authority to sign certificates.