SLS 1055 : 1995
(ISO 8732 : 1988)

Sri Lanka Standard
BANKING - KEY MANAGEMENT (WHOLESALE)

Gr. X

SRI LANKA STANDARDS INSTITUTION

SLS 1055 : 1995
ISO 8732 : 1988

Sri Lanka Standard
BANKING - KEY MANAGEMENT (WHOLESALE)

NATIONAL FOREWORD

This standard was finalized by th Sectoral Committee on Information Technology and was authorized for adoption and publication as a Sri Lanka Standard by the Council of the Sri Lanka Standards Institution on 1995-05-25.

This Sri Lanka Standard is identical with ISO 8732 : 1988 Banking -Key Management - (Wholesale) published by the International Organization for Standardization. (ISO).

Terminology and conventions

The text of the International standard has been accepted as suitable for publication, without deviation, as a Sri Lanka Standard. Howevers, certain terminology and conventions are not identical with those used in Sri Lanka Standards, attention is therefore drawn to the following:

a) Wherever the words 'International Standard/publication' appear, referring to this standard they should be interpreted as "Sri Lanka Standard".

Wherever page numbers are quoted, they are ISO/IEC page numbers.

CROSS - REFERENCES

International Standard	Corresponding Sri Lanka Standards
ISO 7982 - 1 : 1987, Bank Telecommunications -Fund -transfer messages - part 1 -Part 1 : Vocabulary and data elements	SLS 1045 : 1995 Bank telecommunications -Fund transfer messages - Part 1 : Vocabulary and data elements
ISO 8730 : 1990, Banking - requirements for messages authentication (wholesale)	SLS 1053 1995, Banking requirements for messages authentication (wholesale)
ISO 8731- 1 1987, Banking - -Approved algorithms for message authentication - -Part 1 : DEA	SLS 1054 :Part 1 1995, Banking approved algorithms for message authentication - Part 1 DEA.
ISO 8731-2:1992, Banking - Approved algorithms for message authentication - Part 2 : Message authentication algorithm	SLS 1054 : Part 2 : 1995, Banking -Approved algorithms for message authentication _Part 2 : Message authenticator algorithm.

-/ltf.

INTERNATIONAL STANDARD

ISO
8732
First edition
1988-11-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Banking — Key management (wholesale)

Banque — Gestion de clés

Reference number
ISO 8732 : 1988 (E)

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 8732 was prepared by Technical Committee ISO/TC 68, *Banking and related financial services.*

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Printed in Switzerland

Contents

Figures

Tables

Introduction

This International Standard describes procedures for the secure management of the secret cryptographic keys used to protect messages in a wholesale banking environment, for instance messages between banks, or between a bank and a corporate customer, or a bank and a government.

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 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 these 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 electronic transmission of cryptographic keys (and, where needed, IVs) via a communication channel. Automated key distribution utilises two types of keys:

1) Key Enciphering Keys: used to encipher and decipher other keys.

2) Data keys: used to encipher and decipher initialisation vectors (IVs), to authenticate Cryptographic Service Messages, and to encipher/decipher or authenticate data.

Since key management facility(s) can be designed to replace electronically distributed Key Enciphering Keys and data keys automatically, manual intervention is kept to a minimum. Key Enciphering Keys generally have longer cryptoperiods than data keys.

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 the extent and detail of security and key management procedures. Absolute security is not practically achievable so key management procedures need not only to aim to reduce the opportunity for a breach of security but 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, exchange 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, whilst wherever possible this International Standard has specified requirements in absolute terms, in some instances a level of subjectivity 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.

This International Standard has been divided into sections, as follows:

One: General
Two: Manual distribution of keying material
Three: Automatic distribution of keying material

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 detail to be agreed 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 example of the implementation of the requirements for manual distribution of keying material.

Banking — Key management (wholesale)

Section 1 : General

1 Scope and field of application

This International Standard 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. It specifies requirements for

i) the control during its life of keying material 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 algorithm;

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 management process or when the integrity of the keying material is questioned.

It thus provides a means whereby an audit trail can be identified for all keying material.

This International Standard is designed for the use of symmetric algorithms for key distribution, where originator and recipient use the same key. It is designed for messages formatted and transmitted in coded character sets. It is intended that provision will, in due course, be made to cover the use of asymmetric algorithms for key distribution.

This standard does not provide a means to distinguish cryptographically between two physical parties when they share a common key.

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.

2 References

ISO 646, *Information processing — ISO 7-bit coded character set for information processing interchange.*

ISO 7982-1, *Bank telecommunications — Funds transfer messages — Part 1: Vocabulary and data elements.*

ISO 8372, *Information processing — Modes of operation for a 64-bit block cipher algorithm.*

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

ISO 8731, *Banking — Approved algorithms for message authentication.*

ANSI X3.92, *1981 Data Encryption Algorithm.*

3 Definitions

For the purpose of this International Standard the following definitions apply.

3.1 audit trail: see *security audit trail.*

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

3.3 bias: The condition where, during the generation of random or pseudo-random numbers, the occurrence of some numbers is more likely than others.

3.4 ciphertext: Enciphered information.

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

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

NOTE — A party and a Key Distribution Centre or Key Translation Centre exchanging Cryptographic Service Messages do not constitute a communicating pair.

3.7 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).

3.8 counter: An incrementing count used between two parties to control successive key distributions under a particular *Key Enciphering Key.*

3.9 cryptographic equipment: Equipment in which cryptographic functions (eg *encipherment, authentication,* key generation) are performed.

3.10 cryptographic key; key: A parameter used in conjunction with an algorithm for the purpose of *validation, authentication, encipherment* or *decipherment.*

3.11 cryptographic keying material: see *keying material.*

3.12 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.

3.13 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.

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

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

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

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

3.18 decryption: see *decipherment.*

3.19 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, eg *cryptographic key.*

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

3.21 encryption: see *encipherment.*

3.22 exclusive-or: see *modulo-2 addition.*

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

3.24 financial message: A message containing information which has financial implications.

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

3.26 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.*

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

3.28 key: see *cryptographic key.*

3.29 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.*

3.30 Key Distribution Centre: A facility which generates and returns *cryptographic keys* for distribution.

3.31 Key Enciphering Key: A *cryptographic key* used for the *encipherment* and *decipherment* of *cryptographic keys.*

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

3.33 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.*

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

3.35 key offset; offset: The result of adding a *counter* to a *cryptographic key* using *modulo-2 addition.*

3.36 Key Translation Centre: A facility which transforms and returns *cryptographic keys* for distribution.

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

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

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

3.40 Message Authentication Code (MAC): A *code* in a message between a sender and a receiver used to validate the source and part or all of the text of a message. The *code* is the result of an agreed calculation.

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

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 0$$

3.42 notarisation: A method of modifying a *Key Enciphering Key* in order to authenticate the identities of the *originator* and the ultimate *recipient.*

3.43 notarising key: A *cryptographic key* used for *notarisation.*

3.44 notary seal: A value created from the identities of the *logical parties* of a *communicating pair*, and used in the creation of a *notarising key* (pair).

3.45 offset: See *key offset.*

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

3.47 plaintext: Unenciphered information.

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

3.49 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.

3.50 security audit trail: Data collected and potentially used to facilitate a *security audit.*

3.51 security life: The time span over which cryptographically protected data has value.

3.52 split knowledge: A condition under which two or more parties separately and confidentially have custody of the constituent parts of a single key that, individually, convey no knowledge of the resultant *cryptographic key.*

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

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

4 Abbreviations

The following abbreviations are used in this International Standard:

The notation used in clauses 12 to 15 is described in annex B.

Abbreviation	Meaning	Description (see also table 2)
CKD	Key Distribution Centre	A facility which generates and returns cryptographic keys for distribution.
CKT	Key Translation Centre	A facility which transforms and returns keys for distribution.
CSM	Cryptographic Service Message	A message for transporting keys or related information used to control a keying relationship.
CTA	Counter A	Counter used between a CKD or CKT and party "A".
CTB	Counter B	Counter used between a CKD or CKT and party "B".
CTP	Counter P	Counter used in a Point-to-Point keying relationship.
CTR	Counter R	The value of the counter found to be in error.
DEA	Data Encryption Algorithm	—
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 the encipherment algorithm.
EDC	Error Detection Code	A code in a Cryptographic Service Message used to validate the data integrity of the message.
EDK	Effective Date of Key	Date and Co-ordinated Universal Time on which the data key is activated.
ERF	Error Field	The identification of error conditions detected in a prior Cryptographic Service Message.
ERS	Error Recovery Service	A message type used to recover from count or other errors in a Key Distribution Centre or Key Translation Centre environment.
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.
IDA	Identifier of Authentication Key	Identifies the key to be used to authenticate a Disconnect Service Message.The identified key is discontinued.
IDC	Identifier of Key Distribution Centre or Key Translation Centre	—
IDD	Identifier of Key to be Discontinued	—
IDK1	Key Identifier	Identifier of the key being transmitted in a Cryptographic Service Message.

Abbreviation	Meaning	Description
IDK2	Key Enciphering Key Identifier	Identifier (name) of the Key Enciphering Key or key pair used to encipher the key being transmitted in a Cryptographic Service Message.
IDU	Identity of Ultimate Recipient	The identity of the intended final recipient of a Cryptographic Service Message sent within a Key Distribution Centre or a Key Translation Centre environment.
IV	Initialisation Vector	—
KD	Data Key	A key used to encipher/decipher, or authenticate data.
KDU	Notarised Data Key	A data key enciphered under a notarising key (pair).
KDX	Fixed Data Key	A data key with fixed value used in the computation of an Error Detection Code.
KK	Key Enciphering Key	A cryptographic key used for the encipherment and decipherment of cryptographic keys.
*KK[1)]	Key Enciphering Key Pair	A pair of keys used for the encipherment and decipherment of keys.
KKM	Master Key Enciphering Key	The highest level Key Enciphering Key in a multi-layer key management architecture.
KKU	Notarised Key Enciphering Key	A Key Enciphering Key enciphered under a notarising key.
*KKU[1)]	Notarised Key Enciphering Key Pair	A Key Enciphering Key Pair enciphered under a notarising key pair.
KN	Notarising Key	A cryptographic key used for notarisation.
KSM	Key Service Message	A message type used to transfer keys between communicating pairs.
MAC	Message Authentication Code	—
MCL	Message Type	The tag for the field that defines the type of Cryptographic Service Message.
NOS	Notarisation Indicator	A tag that, when present, indicates that notarisation was used.
NS	Notary Seal	A value used for notarisation purposes.
ORG	Originator	Originator of CSM.
P	Key Parity	Indicates that the plaintext key conforms to the specification for odd parity.
RCV	Recipient	Recipient of CSM.
RFS	Request For Service Message	Used to request translation of keys by a Key Translation Centre for retransmission to another party.
RSI	Request Service Initiation Message	Used to request keys from another party.
RSM	Response Service Message	Used to provide an authenticated acknowledgement.
RTR	Response To Request Service Message	Used to send keys from a Key Distribution Centre or from a Key Translation Centre.
SVR	Service Request	Specifies type of service requested.

1) The asterisk* indicates that a pair of keys is involved. Where the use of a pair of keys is an option, in the main text the asterisk is enclosed in parentheses.

5 Key management facility

5.1 General

A key management facility shall provide 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.

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.

6 Requirements of cryptographic equipment

6.1 Generation of keys and initialisation vectors (IVs)

6.1.1 General

Key and IV generation procedures shall be under dual control.

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

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

Keys shall not be available in plaintext form from cryptographic equipment, even upon failure of the equipment, other than at the time of initial generation of a key.

A means shall be provided for the manual zeroisation of plaintext keys (see annex F).

6.1.2 Keys and IVs for manual distribution

All key generation, distribution and storage resources (eg copies, ribbons, etc) shall be protected from unauthorised use, alteration, replacement, destruction or exposure. Waste products shall be destroyed under dual control. The key generation process shall take place in an area where unauthorised viewing is prevented.

Where keys or IVs are printed, provision shall be made to protect them from unauthorised disclosure or replacement.

NOTE – Such protection may include uniquely identified key books with numbered pages protected by tamper resistant packaging so that page substitution is not possible.

Where a specially designed device containing electronically protected memory is used, safety devices shall be built into the software procedures or hardware to prevent unauthorised access. Any attempts to gain unauthorised access into the protected memory shall result in the stored plaintext key being automatically erased, or otherwise rendered unintelligible. There shall be no external display, control or means of extracting the stored key without the linking or insertion of the device containing electronically protected memory into a secure receiver.

Where distribution of a key involves split knowledge, to ensure security, each key component shall be produced on a separate printed form or storage medium.

6.1.3 Keys and IVs for automated distribution

Where cryptographic equipment is used to generate keys and IVs automatically it shall be physically protected to prevent:

1) the disclosure, modification and replacement of the keys

2) the modification or replacement of the IVs

3) the modification or replacement of the key generation algorithm, or device.

6.2 Entry of keys

6.2.1 General

Cryptographic equipment shall permit, at either the system level or the device level, the entry of keys having a format complying with this standard. Access to key entry controls or systems shall be limited by physical or logical means, or both.

6.2.2 Manual entry of keys

A means shall be provided for the manual entry of keys or key components. A means of correcting individual errors or of re-entering the entire key shall be provided. If any plaintext key component is displayed it shall be visible only to authorised personnel and shall be cleared immediately after the key entry process is completed.

NOTE – Re-entry of an entire key may also be used as a means of verifying a previously entered key.

6.2.3 Automated entry of keys

Where a means is provided for automated entry of keys there shall be no display of the key during key entry. Keys retained on special devices such as key loaders shall be entered under dual control.

6.2.4 Parity checking

Where parity checking is available, the parity of plaintext keys or key components shall be verified during entry in order to preclude unintentional single bit modification of the key.

6.2.5 Storage of plaintext keys

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

6.2.6 Retention of electronically stored keys

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

6.2.7 Electromagnetic interference

Protection shall be provided against compromise of keys as a result of radiation or conduction of electromagnetic interference from cryptographic equipment or key loaders.

6.2.8 Functional test

Immediately prior to manual key entry and system initialisation, the cryptographic equipment shall be subject to a test to check that it is operating correctly. This test shall include the operation of all control functions.

6.2.9 Operational error or failure

A means shall be provided to indicate the failure or incorrect operation of the cryptographic equipment (see also 6.1.1). A manual or automatic process shall be provided for the reporting and documentation of all such errors or failures.

6.3 Counter checking

Where keys are associated with counters (see 12.2) the cryptographic equipment shall provide a means for detecting and reporting the erasure, loss or lowering of a counter.

7 Keying material

7.1 Transportation and storage of keying material

Keying material shall be transported and stored in such a manner as to protect it against modification or substitution, and to prevent disclosure of plaintext keys before, during or after the period in which the keys are active.

Access to storage, including the movement of any keying material to or from storage, shall be under dual control. When keying material is entered or removed, the physical access shall be specifically authorised, physically or logically constrained, and fully documented.

7.2 Keys

7.2.1 Custody of keys

Dual control shall be maintained over keys at all times. Keys stored on a computer shall be enciphered or otherwise not be capable of being disclosed.

Lists of staff designated to hold or access keys shall be kept. These lists shall not contain any details of the content of keys.

7.2.2 Validity of keys

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

Data keys may be exchanged on the basis that they are for immediate or for future use (see 7.2.4). No key shall be operational until an authenticated acknowledgement has been received from the recipient. Where a key has not been specifically identified (eg by number or effective date) it shall be the only such key and shall be put into service by the communicating pair immediately after the recipient's acknowledgement is received by the originator.

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.

7.2.3 Key changes

Keys shall be changed:

a) at the end of the cryptoperiod; or

b) with the agreement of both members 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, an explicit date (or other implicit time reference) shall be specified whereupon the old key is no longer current. 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 7.2.5).

7.2.4 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.

7.2.5 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.

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

7.2.6 Back-up of keys

When a printed key is exchanged, the original printed form shall be retained for back-up. Where keys are exchanged automatically a protected copy shall be kept in storage. All back-up copies of active keys shall be subject to the same level of security control as keys in current use.

7.2.7 Destruction of keys (see also annex F)

All copies of keys that are no longer required shall be destroyed under dual control. Printed keys shall be destroyed by means of incineration, cross-cut shredding, or pulping, or other secure method.

Keys stored on magnetic media shall either be zeroised, under password control, or the magnetic media shall be destroyed as for printed keys.

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

Section 2 : Manual distribution of keying material

An example of manual key distribution and control procedures appears in annex A.

8 Despatch of manually distributed keying material

All documents accompanying manually distributed keying material shall be prepared prior to the generation of the keying material. This documentation shall include:

a) A receipt for the keying material for signature by the recipient.

b) Details of the recipient.

c) Details of any passwords required for access to material distributed on magnetic storage media or other secure storage devices (eg key loaders).

d) Where a courier service is used, a receipt for signature by the courier.

e) Details of the date of generation of keying material, together with details of the issuer and the issue date.

All such documentation shall be signed by authorised signatories.

Once keying material has been generated (see 6.1), access to key components shall be controlled by the processes of dual control and split knowledge. Each key component shall be placed in a separate envelope which is sealed in such a manner that any subsequent unauthorised interference can be detected. Each envelope shall be marked to indicate its contents and the address of the appropriate function, and then placed in a second, separate, envelope that is sealed, and addressed to the recipient. The second envelope shall give no indication of its contents.

NOTE — Each package thus consists of an outer envelope with a single inner envelope containing a single key component.

The individual components of a key shall be despatched, together with a receipt using a method to ensure separate despatch, for example, on different days. Any passwords required for access to magnetic storage media or other storage devices, eg key loaders, shall be despatched separately from the medium or device.

Where keying material is transported by mail then a secure method shall be used. Where delivery is by means of courier a receipt shall be obtained from the courier by the sender. The courier shall not be aware of the nature of the contents of an envelope.

9 Receipt of manually distributed keying material

Upon receipt of a package containing a key component the recipient shall examine the innermost envelope in order to check, so far as is possible, that access to its contents has not been attempted or achieved. If it is suspected that the security of the inner envelope has been compromised the sender shall be advised immediately. The signatures on the accompanying documentation shall be checked by the recipient for authenticity. The identity of the key components, eg sequence number or effective date, shall be recorded. When the recipient of the key component is satisfied with the authenticity of the key component the receipt that accompanied it shall be signed and returned (see also 7.2.2). Keys shall be placed in secure storage immediately upon receipt. Inner envelopes (ie those containing the key components) shall be retained under appropriate control (see clause 7).

Section 3 : Automatic distribution of keying material

10 Requirements for the automated key management architecture

This International Standard is designed to meet the following requirements for automated key management. It is assumed that:

1) the data network is expandable.

2) either a communicating pair has a Key Enciphering Key in common or each has a Key Enciphering Key Pair in common with a Key Distribution Centre or a Key Translation Centre.

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

10.2 Any communicating pair may share more than one Key Enciphering Key.

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

10.4 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.

10.5 A data key or Key Enciphering Key shared between a communicating pair shall not be disclosed to a third party (except for a Key Translation Centre (CKT) or a Key Distribution Centre (CKD)).

10.6 A key used between any communicating pair shall not intentionally be used between any other communicating pair.

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

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

10.9 The architecture shall support communicating parties that do not have a key generation capability.

10.10 The architecture shall support any party initiating a secure connection with any other party.

10.11 In a three layer architecture (see 11.1) the ability to exchange Key Enciphering Keys automatically between a communicating pair shall be provided.

11 Automated key management architecture

11.1 General

The architecture shall consist of either two or three layers of keys (see figure 1). All implementations shall have the capability of functioning in a two layer architecture.

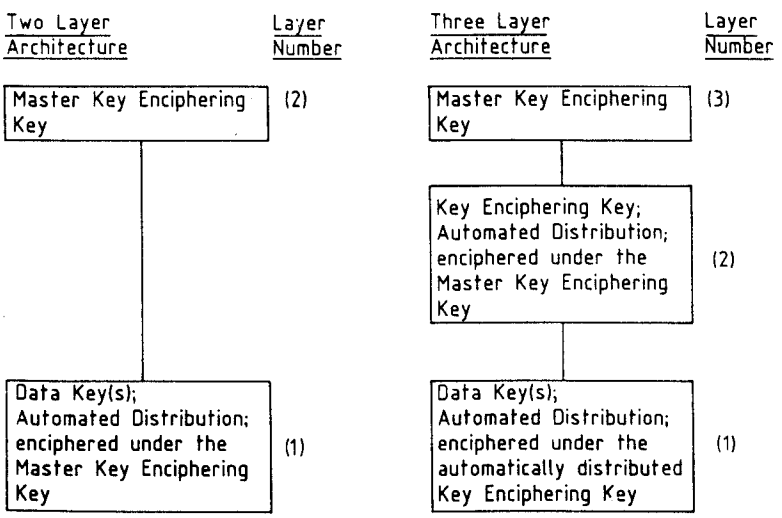


Figure 1 — Key distribution architecture

In a two layer architecture the upper layer shall comprise a Master Key Enciphering Key or keys (KKM). The lower layer shall comprise a data key or keys (KD). These keys are enciphered using the KKM (see 11.2).

In a three layer architecture the uppermost layer shall comprise a Master Key Enciphering Key or keys (KKM). The second layer shall comprise one or more Key Enciphering Keys (KK) or key pairs (*KK). (These keys are enciphered using the KKM (see 11.2) prior to being automatically distributed). The lowest layer shall comprise a data key or keys (KD) enciphered using the KK or *KK (see 11.2). One or two data keys shall be transmitted with the automatically distributed Key Enciphering Key. These data keys shall be enciphered under the automatically distributed Key Enciphering Key. Subsequent key distribution messages need not include a Key Enciphering Key. When no Key Enciphering Key is transmitted, one or two data keys enciphered under a previously exchanged and automatically distributed Key Enciphering Key shall be sent.

11.2 Distribution of keying material

Master Key Enciphering Keys shall be manually distributed using the procedure specified in section two of this standard, or other secure means by mutual agreement. In no instance shall a master key be automatically distributed by a process defined in section three of this standard. Other keying material shall be distributed using Cryptographic Service Messages (see clause 13).

Automatically distributed Key Enciphering Keys or key pairs shall be enciphered prior to distribution using a Master Key Enciphering Key.

Data keys shall be enciphered prior to distribution using a Key Encipherment Key or key pair.

NOTE — Two data keys with the same identifier may be sent, enciphered, in a single Cryptographic Service Message (see clause 13).

The encipherment of data or initialisation vectors (IV) prior to distribution shall be carried out using data keys. Key Enciphering Keys shall only be used to encipher/decipher Key Enciphering Keys or data keys, and for no other purpose.

When a new key (keys) is received, all stored keys of the same type (Key Enciphering Keys or data keys) with the same name shall be replaced. In addition, when a key is discontinued, all keys of the same name (without regard to type) shall be discontinued.

Where keys are sent along with other protected data to the intended recipient, the recipient shall be able to recognise the key so that it can be deciphered and loaded before any cryptographic process can begin. If the recipient has multiple Key Enciphering Keys, information shall be sent identifying the key to be used.

11.3 Environments

11.3.1 Classification of key distribution environments

This sub-clause specifies requirements for the following key distribution environments:

a) Point-to-Point;

b) Key Distribution Centre (CKD);

c) Key Translation Centre (CKT).

A Key Distribution Centre or Key Translation Centre may be used to reduce the number of Master Key Enciphering Keys in large networks. A mutually trusted party is designated as the centre. A *KK shared between any party and a Key Distribution Centre or Key Translation Centre permits secure communications to be established between that party and any other party that has a *KK relationship with the centre.

A Key Distribution Centre has the capability to generate and send data keys for distribution, and may send keys unsolicited or upon request.

A Key Translation Centre has the capability to transform and return keys for distribution by the originating party (eg, a Key Translation Centre does not require key generation capability; parties in the network have a peer relationship).

For compliance with this standard all key management systems providing automated distribution of keys shall provide a Point-to-Point implementation.

NOTE — The accommodation of a Key Distribution Centre or Key Translation Centre key management system is not a requirement of this standard.

11.3.2 Point-to-Point

A Point-to-Point environment exists when a communicating pair shares a Key Enciphering Key, either a single key or a key pair, so that further keys (whether Key Enciphering Keys or data keys) may be exchanged. At least one member of the communicating pair shall have the capability of generating or otherwise acquiring keys.

11.3.3 Key Distribution Centre

A Key Distribution Centre exists for the purpose of distributing generated or acquired data keys to a communicating pair, each member of which:

a) wishes to communicate with the other but does not currently share keys; and

b) shares a Master Key Enciphering Key Pair with the same Key Distribution Centre; but

c) may not have the ability to generate keys

One member of the communicating pair (the originator) requests data key(s) from the Key Distribution Centre for later communication to the other member (the ultimate recipient).

The Key Distribution Centre shall generate or acquire data key(s) and shall send two sets to the originator using the Master Key Enciphering Key Pair shared with the originator. One set shall be notarised using the Key Enciphering Key shared between Party A and the centre; the other set shall be notarised using the Key Enciphering Key shared between Party B and the centre.

The originator shall then send the second set to the ultimate recipient.

11.3.4 Key Translation Centre

A Key Translation Centre is used to translate keys for future communication between a communicating pair, each member of which:

a) wishes to communicate with the other but does not currently share keys;

b) shares a Master Key Enciphering Key Pair with the same Key Translation Centre; and

c) has the ability (through the originator) to generate or otherwise acquire keys

Key Enciphering Keys and data keys may be translated and exchanged, though only one of the two types (Key Enciphering Key or data key) shall be processed by the centre at one time.

A Key Enciphering Key or data key for use in future communication with the other member of the communicating pair (the ultimate recipient) shall be sent by the originator to the Key Translation Centre, enciphered under the offset Key Enciphering Key Pair shared between the originator and the Key Translation Centre.

The Key Translation Centre shall decipher this key, and then re-encipher it using notarisation and the Master Key Enciphering Key Pair shared with the ultimate recipient. The re-enciphered key shall be returned to the originator.

The originator shall redirect the re-enciphered key on to the ultimate recipient.

Dual Key Translation Centres may be implemented by the method described in annex E.

12 Encipherment and decipherment of keys and initialisation vectors

NOTE — The notation used in this clause is described in clause 4 and annex B.

12.1 Encipherment, decipherment, authentication and error detection

12.1.1 General

a) Master Key Enciphering Keys shared with a Key Distribution Centre (CKD) or a Key Translation Centre (CKT) shall comprise Key Enciphering Key Pairs (see 11.3.3 and 11.3.4).

b) Other communicating pairs shall share either single Master Key Enciphering Keys or Master Key Enciphering Key Pairs.

NOTE — Key pairs may be used where additional security is needed (eg, the Key Enciphering Key has a long security life).

c) Master Key Enciphering Keys shall not be superseded except by other Master Key Enciphering Keys.

d) Other Key Enciphering Keys may be single keys or key pairs. A Key Enciphering Key Pair shall not be enciphered/deciphered using a single key.

e) Data keys shall be single keys.

f) Key pairs shall not be used to encipher data.

The general formats for encipherment, decipherment, authentication and error detection, respectively are as follows:

(enciphered quantity) = eK(plaintext quantity)

(deciphered quantity) = dK(enciphered quantity)

(authentication code) = MAC = aKD(data)

(Error Detection Code) = EDC = aKDX(data); where a 64 bit key is used KDX shall be

KDX = 0123456789ABCDEF

NOTE — Where the key is not 64 bits in value, the value of KDX shall be this value appropriately reiterated or truncated.

The processes for encipherment and decipherment are specified in 12.1.2 to 12.1.6 and in figures 2-4. Authentication and error detection are described in 12.1.7 and 12.1.8.

Where DEA (ISO 8731-1) is used for encipherment/decipherment, it shall be in ECB mode.

12.1.2 Encipherment and decipherment of a single key by a single key

Where a single Key Enciphering Key is used, a single KK or KD shall be enciphered and deciphered with the cryptographic algorithm using the following formulae:

K( )Z encipherment = eKKY(K( )Z)

enciphered K( )Z decipherment = dKKY(enciphered K( )Z)

where:

K( )Z is a single key and may be a Key Enciphering Key, KKZ, or a data key, KDZ

KKY is a single Key Enciphering Key

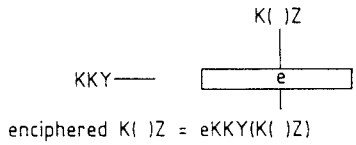


Figure 2(a) — encipherment

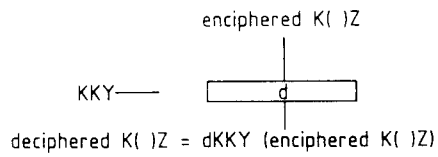


Figure 2(b) — decipherment

Figure 2 — Encipherment and decipherment of a single key by a single key

12.1.3 Encipherment and decipherment of a single key by a key pair

Where a Key Enciphering Key Pair is used, single KK or KD shall be enciphered and deciphered with the cryptographic algorithm using the following formulae:

K( )Z encipherment = ede*KKY(K( )Z)
= eKKlY(dKKrY(eKKlY(K( )Z)))

K( )Z decipherment = ded*KKY(K( )Z)
= dKKlY(eKKrY(dKKlY(K( )Z)))

where:

K( )Z is a single key and may be a Key Enciphering Key, KKZ, or a data key, KDZ

*KKY is a Key Enciphering Key Pair; and

KKlY is the left member of the key pair,
KKrY is the right member of the key pair.

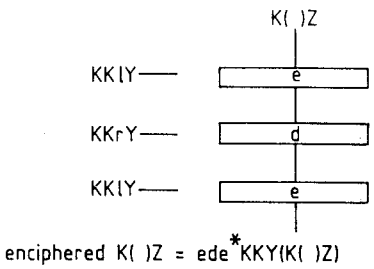


Figure 3(a) — encipherment

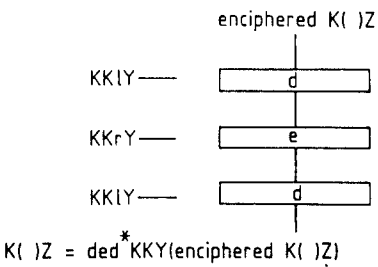


Figure 3(b) — decipherment

Figure 3 — Encipherment and decipherment of a single key by a key pair

12.1.4 Encipherment and decipherment of a key pair by a key pair

*KK pairs shall be enciphered and deciphered using a Key Enciphering Key Pair with the cryptographic algorithm using the following formulae:

*KKZ encipherment = ede*KKY(*KKZ)
= eKKlY(dKKrY(eKKlY(KKlZ)))
|| eKKlY(dKKrY(eKKlY(KKrZ)))

*KKZ decipherment = ded*KKY(*KKZ)
= dKKlY(eKKrY(dKKlY(KKlZ)))
|| dKKlY(eKKrY(dKKlY(KKrZ)))

where:

*KKZ is the Key Enciphering Key Pair being enciphered; and

KKlZ is one member of the key pair,
KKrZ is the other member of the key pair

*KKY is a Key Enciphering Key Pair; and

KKlY is one member of the key pair,
KKrY is the other member of the key pair

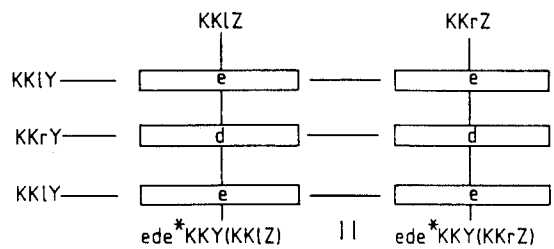


Figure 4(a) — encipherment

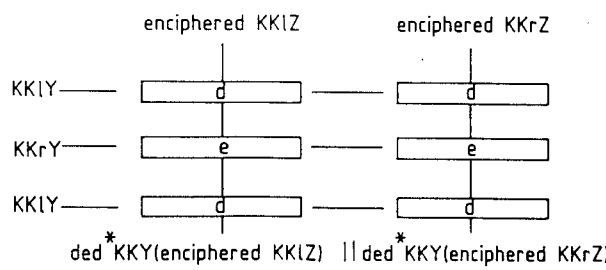


Figure 4(b) — decipherment

Figure 4 — Encipherment and decipherment of a key pair by a key pair

12.1.5 Encipherment and decipherment of a single key in systems using Key Enciphering Key Pairs

Facilities designed for use with key pairs can be made downward compatible to those using single keys by setting KKlY = KKrY = KKY.

12.1.6 Encipherment and decipherment of initialisation vectors

Where IVs are to be enciphered and deciphered, they shall be represented as hexadecimal characters. When DEA is used, IVs shall have a maximum length of 64 bits (represented as 16 hexadecimal characters).

When IVs are enciphered and deciphered, they shall be enciphered and deciphered using a data key (KD) and with the cryptographic algorithm using the following formulae:

enciphered IV = eKD(IV)

IV = dKD(enciphered IV)

12.1.7 Authentication of Cryptographic Service Messages

When Cryptographic Service Messages are to be authenticated, 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 formula 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.

12.1.8 Error detection

When it is desired to detect transmission or process errors and other means are not available, the authentication process specified in 12.1.7 shall be used, as in the following formula, to derive an Error Detection Code (EDC):

EDC = aKDX(data)

where KDX is a key with the fixed value 0123456789ABCDEF (see NOTE in 12.1.1).

12.2 Counters

12.2.1 Purpose

Counters are used to:

a) detect duplicate Cryptographic Service Messages.

b) detect a message received out of sequence.

c) indicate the number of messages enciphered under a particular KK or *KK.

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

Counters shall only increment, and never decrement, and shall never repeat during the use of the associated (*)KK.

Counters for key management purposes (as defined in this standard) shall be independent of the means used to differentiate between individual data messages. (eg. sequence numbers).

Counters shall be transmitted as fields in selected Cryptographic Service Messages and shall be authenticated by an appended MAC. Error status messages are an exception in that they are transmitted without a MAC.

NOTE – Counter integrity may be provided by an appended Error Detection Code (see 12.1.8).

Messages which use a Master Key Enciphering Key to encipher either KD (two layer architecture) or automatically distributed (*)KK (three layer architecture), shall use a counter which reflects the number of messages which have been transmitted using the particular Master Key Enciphering Key. In a three layer architecture where only KD(s) enciphered under an automatically distributed (*)KK appear in a Cryptographic Service Message, the counter shall reflect the number of messages which have been transmitted using the particular (*)KK for encipherment.

NOTE – A (*)KK may be key offset with a counter; the resulting key is used to encipher another key for transmission (see 12.3 and 12.4).

12.2.2 Management of counters

The contents of a counter shall be defined and manipulated as a binary number. Counters shall always be set to one upon successfully loading the associated (*)KK, except in the case where a (*)KK is being sent in a Request Service Message to a Key Translation Centre. In that case alone, the counter associated with the (*)KK being sent to the centre for translation shall be set to zero.

In a Point-to-Point and Key Translation Centre environment separate counters shall be maintained for each (*)KK shared between communicating pairs, ie an origination count and a reception count.

In a Key Distribution Centre (CKD) environment (where keys are never sent to a CKD; only from it) only one counter shall be maintained for each *KK shared between a party and a CKD, ie the CKD maintains an origination count and each communicating pair maintains its own reception count.

Any difference between the originator's counter and the recipient's counter shall be detected and, unless a 'window' has been agreed (see other considerations in 12.2.4 and annex D), the difference shall be immediately reported and the reason for its occurrence ascertained.

Duplicate messages shall be rejected and an error shall be reported to the originator. The recipient prepares an Error Service Message (see clause 14.4) 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.

Where a counter resets to zero, is lost or lowered, the situation shall be interpreted as a catastrophic error and shall invalidate the associated (*)KK, which shall be immediately withdrawn from use. Detection of such a condition requires the enabling of an associated alarm. Recovery from such an event requires the use of a new (*)KK (either from storage or newly distributed), with the associated counters being reset to one.

12.2.3 Management of Cryptographic Service Messages (see table 1)

When a recipient receives a Cryptographic Service Message whose count equals the expected (stored) count, the message shall be accepted. Both the originator's counter and recipient's reception counter shall be incremented by one prior to the next message.

Where the counts do not match a log record shall be made.

When the recipient receives a Cryptographic Service Message whose count is greater than the expected (stored) count, the message shall be accepted. The recipient's reception count shall be set to the received count plus one. This will be the new expected count.

When a recipient receives a Cryptographic Service Message whose count is less than the expected (stored) count, the message shall be rejected and an error shall be reported to the originator. The recipient shall return an Error Service Message in which the nature of the error being reported (count error), the value of the reception counter (the recipient's expected (stored) count) and the value of the count as received in the related Cryptographic Service Message, shall be included.

NOTE — The count as received may be used to identify the Cryptographic Service Message which contained the counter in error.

The originator, upon receipt of the Error Service Message shall either:

a) adjust his origination counter up to the expected count value returned in the Error Service Message, or

b) establish a new (*)KK with the recipient and also reset the value of the associated counters to one.

Table 1 — Processing counters (message authenticated)

Received count (CTA, CTB or CTP) equal to expected (stored) count	Received count (CTA, CTB, or CTP) greater than expected count	Received count (CTA, CTB, or CTP) less than expected (stored) count	*Action to be taken on receipt of an ESM or ERS*	
			Received count (CTA, CTB, or CTP) greater than origination count	Received count (CTA, CTB, or CTP) less than or equal to origination (stored) count
o Accept message o Generate RSM o Log (optional) o Increment expected count (+1)	o Accept message o Generate RSM o Log (mandatory) o Set expected count equal to received count +1	o Reject message o Send ESM with expected count o Log (mandatory)	o Set origination (stored) count equal to expected count (CTA, CTB, or CTP) OR o Change (*)KK associated with the CTA, CTB, or CTP o Log (mandatory) o Send new KSM or RTR with corrected count OR o Wait for new key request (RSI or RFS)	o Log (mandatory) o Send new KSM or RTR with current origination (stored) count OR o Wait for new key request (RSI or RFS)

12.2.4 Other considerations

In a CKD or CKT environment, it is possible for a recipient to receive Cryptographic Service Messages whose counts are out of sequence, yet the Cryptographic Service Messages are authenticated. In these two environments a party assuming the role of recipient may establish a window, representing a range of reception counter values such that the corresponding Cryptographic Service Messages may arrive out of sequence, and are accepted without declaring an error.

NOTE — A method of defining and managing a window is described in annex D.

12.3 Key offsetting

Key offsetting shall be used to transform a (*)KK *prior* to encipherment of a key by that (*)KK.

The modulo-2 sum of KB and CB shall be derived as follows:

$$KB + CB = (k1 + c1, k2 + c2, \ldots k7 + c7, p)$$

where:

KB is an eight bit key byte in which individual bits are designated as k1, k2 . . . k7 and p (see below).

CB is an eight bit counter byte in which individual bits are designated as c1, c2 . . . c7, 0.

p may be set to give the key byte, KB, odd parity.

The modulo-2 sum of a 64 bit key, KK, with a 56-bit counter is derived by modulo-2 adding the first byte of KK with the counter byte formed from the first seven high order bits of the counter (with a zero bit added) to derive the first byte of the result. Then the second byte of KK is modulo-2 added with the counter byte formed from the second seven bits of the counter (with a zero bit added). The process continues until the eighth byte of KK is modulo-2 added with the counter byte formed from the last seven bits of the counter (with a zero bit added).

The operation of key offsetting a single key is indicated by the expression:

$$KKo = (KK + CT)$$

where:

KKo is the offset Key Enciphering Key
KK is the original Key Enciphering Key
CT is the counter

Where a key pair is to be key offset, then the following equation is used:

$$*KKo = (KKl + CT) \,||\, (KKr + CT)$$

where:

*KKo is the offset Key Enciphering Key Pair

KKl and KKr are the two members of the original key pair.

12.4 Notarisation of keys

Notarisation is a method for sealing keys with the identities of the communicating pair. It is achieved by creating a notary seal (NS). This sub-clause describes the derivation of the notary seal and how it is used. Once notarised, using the method described, keys can only be recovered with knowledge of the key used to perform notarisation and the identities of the communicating pair. A KD or a KK may be notarised before transmission by encipherment using a notarising key, (*)KN. (*)KN is formed by taking the modulo-2 sum of (*)KK with the notary seal (NS).

When notarisation is used the notarisation field "NOS" shall be present in the Cryptographic Service Message, unless the use of notarisation of a field is made explicit by the field tag (eg, KKU, KDU).

Where notarisation is to be achieved by combination with an ISO 646 character the following formula shall be used:

$$KB + AC = (k1 + b7, k2 + b6, \ldots k7 + b1, p)$$

where:

KB is an eight bit key byte in which individual bits are designated as k1, k2 . . . k7 and p (see below)

AC is a eight bit character byte in which individual bits are designated as b1, b2 . . . b7, p (see below).

p may be set to give the byte odd parity.

Where notarisation is to be achieved by combination with an output byte of an encipherment algorithm the following formula shall be used:

$$KB + OB = (k1 + 01, k2 + 02, \ldots k7 + 07, p)$$

where:

OB is any eight bit output byte of an encipherment algorithm, in which individual bits are designated 01, 02 . . . 08.

p may be set to give the byte odd parity.

The complete notarised key is formed by taking the modulo-2 sum of the eight pairs of bytes of a key and eight characters, or output bytes of an algorithm.

EXAMPLE

Suppose that Party A wishes to send a key to Party B. Let FM1 be the first eight characters of Party A's identity and let FM2 be the second eight characters of Party A's identity. Similarly let T01 and T02 represent the first and second halves of Party B's identity. If necessary, the identities are replicated to form sixteen character identifiers. For example, if Party A's identity is 'CITYB' and Party B's identity is 'MANHAN', then FM1 is 'CITYBCIT'; FM2 is 'YBCITYBC'; T01 is 'MANHANMA'; and T02 is 'NHANMANH'.

Case 1: Computing a notarising key using a KK

Let KK be the key which is to be used to compute the notarising key. Then:

$$KKR = KK + FM1$$

$$KKL = KK + T01$$

$$NSl = eKKR(T02)$$

$$NSr = eKKL(FM2)$$

$$NS = [(\text{leftmost 32 bits of NSl}) \,||\, (\text{rightmost 32 bits of NSr})] + CT( )$$

$$KN = KK + NS$$

KN is then used to notarise (by encipherment) either a KD or a KK.

Case 2: Computing a notarising key using a *KK

Let *KK be the key which is to be used to compute the notarising key. Then:

$$*KK = KKl \,||\, KKr$$

$$KKR = KKr + FM1$$

$$KKL = KKl + T01$$

Then:

$$NSl = eKKR(T02) + CT( )$$

$$NSr = eKKL(FM2) + CT( )$$

NOTE — Where CT( ) is the counter used to key offset NSl and NSr, and the process for key offsetting is defined in section 12.3, above.

$$*KN = (KKl + NSl) \,||\, (KKr + NSr)$$

*KN is then used to notarise (by encipherment) either a KD or a (*)KK.

13 Cryptographic Service Messages

13.1 General

Cryptographic Service Messages (CSM) shall be used for the automatic distribution and control of cryptographic keys and, where required, IVs, in a Point-to-Point environment and may be used to support a CKD or a CKT environment.

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

13.2 Message types

The following Cryptographic Service Message types are described in this standard:

NOTE — The formats of these messages are specified in 13.4. The content is specified in clause 14.

a) *Point-to-Point key management messages*

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

   A message that requests that a new keying relationship be initiated. The originator of an RSI might not have a key generation capability.

2. *Key Service Message (KSM)* (mandatory)

   A message that transfers a key from an originator to a recipient.

3. *Response Service Message (RSM)* (mandatory)

   A message that provides an authenticated response to a KSM.

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.

b) *Additional Cryptographic Service Message types for use with a Key Distribution Centre or Key Translation Centre*

1. *Request For Service message (RFS)* (mandatory)

   A message in a Key Translation Centre environment that sends keys to a Key Translation Centre to be translated for the ultimate recipient named in the IDU field.

2. *Response To Requestor message (RTR)* (mandatory)

   A message in a centre environment that responds to an RFS, an ERS or to an RSI to the centre. An RTR may be initiated by a Key Distribution Centre.

3. *Error Recovery Service message (ERS)* (mandatory)

   A message that reports count and key errors to the Key Distribution Centre or Key Translation Centre and requests resynchronisation of the count fields and re-initiation of service.

13.3 Character set and representation

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

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

The characters "." and "b" shall not be used in a subfield; "b" shall not be used within a field (except for the MAC field). The character "." shall only be used to separate subfields 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.

13.4 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 "b" and, if desired for readability, a carriage return and line feed.

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

13.5 Fields and subfields

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

Fields containing keys may consist of up to four subfields. Each subfield (whether present or not) shall be terminated by a full stop unless no subsequent subfield is present (see examples in NOTE). The ordering and content of the sub-field shall be as follows:

a) A key that has been enciphered or notarised shall always be in the first subfield and is the only subfield required.

b) The second subfield shall, if present, be a "P" to indicate that the plaintext key conforms to the specification for odd parity.

c) The third subfield shall, if present, contain the identity of the key sent in the first subfield.

d) The fourth subfield shall, if present, contain the identity of the key used to encipher the key sent in the first subfield.

NOTE — *Examples of Key-Field Formats*

1 KD/key . . . IDK1, is a field with two subfields containing data. IDK1 is the identity assigned to the KD. The key used to encipher the KD has not been explicitly defined. Hence its identity is assumed (is implicitly defined in the relationship).

2 KK/key.P.IDK1.IDK2, is a field that contains a key, KK, that conforms to the specification for odd parity and subsequently has been enciphered under the key whose identity is IDK2; the identity of KK is IDK1.

3 KDU/key, is a field with only one subfield that contains data. In this case, the identity of the new KD received in notarised form as KDU and the identity of the key used to decipher KDU are implicitly defined.

4 KD/key . . . IDK2, is a field with two subfields present. IDK2 is the identity of the key used to encipher the KD. No name is assigned to the KD.

Unless otherwise determined by prior agreement, if two KDs are sent, the first KD shall be used by the ultimate recipient for authentication; the second shall be used for encipherment.

3.6 Message flow

3.6.1 General

The fields and subfields shall be as defined in table 2. The fields used with each message class, and their sequence, shall be as defined in tables 3 to 5.

When a message is received with an optional field(s) that is not implemented, 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.

Table 2 — Cryptographic Service Message : Fields and subfields

Field tag	Name	Definition/remarks	Specification
CTA	Counter A	An incrementing binary counter. Associated with an *KK used to encipher either an (*)KK or KD(s) sent in a Cryptographic Service Message. Used between a CKD or a CKT and another party designated as "A". Set to one upon installation of this new *KK.	Up to 14 hex characters; leading zeros may be suppressed for transmission.
CTB	Counter B	An incrementing binary counter. Associated with an *KK used to encipher either an (*)KKU or KDU(s) sent in a Cryptographic Service Message. Used between a CKD or a CKT and another party designated as "B". Set to one upon installation of this new *KK.	Up to 14 hex characters; leading zeros may be suppressed for transmission
CTP	Counter P	An incrementing binary counter. Associated with an (*)KK or KD(s) sent in a Cryptographic Service Message. Used between communicating pairs, but not between a CKD or a CKT and another party. Set to one upon installation of this new (*)KK.	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.
EDC	Error Detection Code	The Error Detection Code when used shall be generated on all components of the associated service message (ERS for the CKD environment, ESM or RSI) using the editing, computation and formating requirements for	9 characters (4 hex) b (4 hex)

Table 2 — Cryptographic Service Message : Fields and subfields *(continued)*

Field tag	Name	Definition/remarks	Specification
EDC *(continued)*		a MAC in ISO 8730.The hexadecimal key for EDC computation shall be 0123456789ABCDEF. (See NOTE in 12.1.1)	
EDK	Effective Date of Key	Date and Coordinated Universal Time of KD activation.	12 characters YYMMDDHHMMSS
ERF	Error Field	Error Codes are defined as: Up to 16 characters A : CTA error B : CTB error C : Cannot Process (Optional. May be used as a general error code where a more specific error code is not appropriate) D : CKD unknown E : Facility inoperative F : Format (syntax) error G : Reserved H : User defined I : Key Identifier not known to recipient K : Parity error in received key M : MAC error (failure to authenticate) O : Option not implemented P : CTP error T : CKT unknown U : IDU not known to the CKT or CKD X : EDC error (probable transmission error)	
IDA	Identifier of Key for Authentication	Identifies the key to be used to authenticate DSM. This key shall be discontinued.	Up to 16 characters
IDC	Identity of CKD or CKT	Identity of CKD or CKT to be used.	4 to 16 characters
IDD	Identifier of Key to be Discontinued	Identifies key to be discontinued.	Up to 16 characters
IDK1	Key Identifier	Identifies (names) the received key.	Up to 16 characters (subfield)
IDK2	Key Enciphering Key Identifier	Identifies (names) the (*)KK used to encipher/ decipher a (*)KK or a KD.	Up to 16 characters (subfield)
IDU	Identity of Ultimate Recipient	This field is only used with a CKT or a CKD.	4 to 16 characters

Table 2 — Cryptographic Service Message : Fields and subfields *(continued)*

Field tag	Name	Definition/remarks	Specification
IV	Initialisation Vector	Starting point for encipherment/decipherment process. If enciphered, is always enciphered under a data key (KD). If only one data key is transmitted in a message, it shall be enciphered under that KD. If two KDs are transmitted, the IV shall be enciphered under the second KD. The first character shall be "E" (indicating that the IV is to be enciphered) or "P" (indicating that the IV is to be sent in plaintext form). This first character shall not form part of the IV for encipherment purposes'.	1 character followed by up to 32 characters (16 when DEA is used). Leading zeros of an enciphered IV shall not be suppressed for transmission.
KD	Data Key	Enciphered KD. May be used for encipherment or authentication. A maximum of two KD fields may be sent per message. If two KDs are sent, the first shall be used for authentication and the second shall be used for encipherment. In a Point-to-Point environment, the KD(s) sent in this field shall be notarised if and only if no KK field is present and the NOS field is present.	When DEA is used, 16 hex characters
KDU	Data Enciphering Key, Notarised	A KD enciphered under the *KN generated by the CKT or CKD for the ultimate recipient specified in the IDU field of a RTR or specified by the RCV field of a KSM that retransmits this key to the ultimate recipient. Up to two fields can be sent per message. If two KDUs are sent, the first shall be used for authentication and the second shall be used for encipherment.	When DEA is used, 16 hex characters
KK	Key Enciphering Key	Enciphered KK. In a Point-to-Point environment, the KK sent in this field shall be notarised if and only if the NOS field is present.	When DEA is used, 16 hex characters
*KK	Key Enciphering Key Pair	Enciphered KK Pair. In a Point-to-Point environment, the *KK sent in this field shall be notarised if and only if the NOS field is present.	When DEA is used, 32 hex characters

Table 2 — Cryptographic Service Message : Fields and subfields *(continued)*

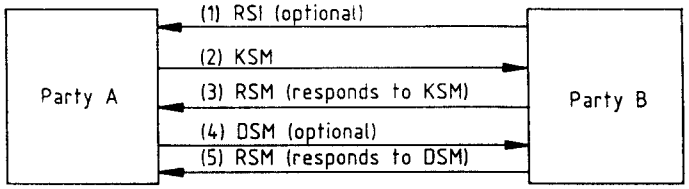
Field tag	Name	Definition/remarks	Specification
KKU	Key Enciphering Key, Notarised	A KK enciphered under the *KN generated by the CKT or CKD for the ultimate recipient, as specified in the IDU field.	When DEA is used, 16 hex characters
*KKU	Key Enciphering Key Pair, Notarised	A KK pair enciphered under the *KN generated by the CKT or CKD for the ultimate recipient, as specified in the IDU field.	When DEA is used, 32 hex characters
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, ERS, ESM, KSM, RFS, RSI, RSM or RTR.
NOS	Notarisation Indicator	Indicates use of notarisation process for (*)KK if (*)KK is present in a message. If no (*)KK is present, the KD(s) is (are) notarised.	Zero length field
ORG	Originator	Cryptographic Service Message originator.	4 to 16 characters
P	Key Parity	Used to indicate that the plaintext key conforms to the specification for odd parity.	1 character (subfield)
RCV	Recipient	Cryptographic Service Message recipient.	4 to 16 characters
SVR	Service Request	Specifies type of service requested. SVR requests one data key implicitly. KD requests two data keys. KK requests a Key Enciphering Key. *KK requests a Key Enciphering Key Pair (a KK or a *KK may be requested, but not both).	0 to 9 characters

Table 2 — Cryptographic Service Message : Fields and subfields *(concluded)*

Field tag	Name	Definition/remarks	Specification
SVR *(continued)*	Service Request	IV requests an IV. The IV shall be enciphered unless, by prior agreement, a plaintext IV is to be sent. A minimum of one and a maximum of three keys and an IV may be requested in a single ERS or RSI. Types requested shall be separated by periods.	0-9 characters

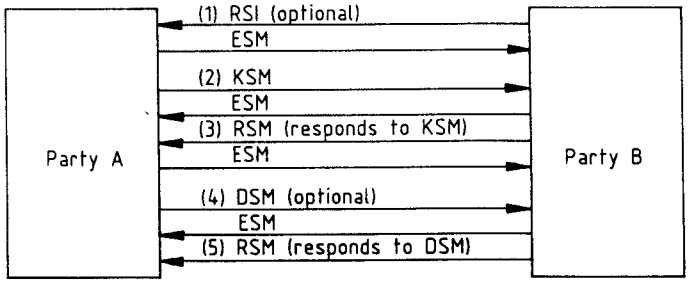
13.6.2 Point-to-Point environment

Figures 5 and 6 show the flow of Cryptographic Service Messages for the Point-to-Point environment.



Note – Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 5 — Point-to-Point environment (normal message flow in sequence)



Note – Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 6 — Point-to-Point environment (message flow with error messages)

Table 3 defines the fields and their order for each message in this environment. Message flow shall be as follows:

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

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

2) shares a (*)KK with Party A, and

3) does not have key generation capability or access to keys;

then

Party B sends an RSI to Party A requesting that Party A send key(s) and, optionally, an IV to Party B. If Party A receives an RSI from Party B with an error in it, an ESM shall be returned to Party B.

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

1) wishes to send key(s) to another logical party (Party B); and

2) shares a (*)KK with Party B; and

3) has key generation or acquisition capability; then

Party A generates or acquires keys and optionally an IV and sends a KSM to Party B containing the key(s) (and IV).

If a (*)KK is sent, it shall be enciphered under a (*)KK shared with Party B and the accompanying KD(s) sent in the Cryptographic Service Message shall be enciphered under the (*)KK sent in that message.

If a (*)KK is not sent, the KD(s) sent in the Cryptographic Service Message shall be enciphered under a (*)KK shared with Party B.

The IV (if enciphered) shall be enciphered under the KD (the second KD if two KDs are sent) in that Cryptographic Service Message.

The Cryptographic Service Message shall be authenticated using the KD(s) sent in that message.

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.

Party A may resend a KSM to Party B an arbitrary number of times, but Party A shall not send a new KSM (ie, utilising new keys or a new count for the (*)KK specified in a message) until the old KSM is acknowledged by an RSM or an ESM.

c) If either logical party of a communicating pair wishes to terminate a keying relationship or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with an RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party 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 field is present) shall be retained to authenticate the subsequent RSM. That key shall then be discontinued. When a (*)KK is discontinued, all keys enciphered under that (*)KK shall also be discontinued without being named in the DSM.

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

Table 3 — Fields used with each message type: Point-to-Point environment

Message type	RSI(1)	KSM	RSM	ESM	DSM(1)	RSM	ESM
Responding to	—	—	KSM	KSM	—	DSM	DSM RSI/ RSM
Reference sub-clauses	14.7 15.7	14.5 15.5	14.8 15.8	14.4 15.4	14.2 15.2	14.8 15.8	14.4 (generation) 15.4 (processing)
See Notes to Fields							
	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG
8 and 9					IDD	IDD	
1					IDA		
3		NOS					
1		(*)KK					
2		KD					
1		IV					
1		EDK					
	SVR						
		CTP		CTP			
7				CTR			
				ERF			ERF
1	EDC			EDC			EDC
		MAC	MAC		MAC	MAC	

Notes

1 Optional

2 A maximum of two such fields may be sent in a message.

3 Required when notarisation of keys is used.

4 Present if and only if there is a (*)KKU field.

5 Present if and only if there is no (*)KKU field.

6 Only one KD field shall be present if a (*)KK field is present.

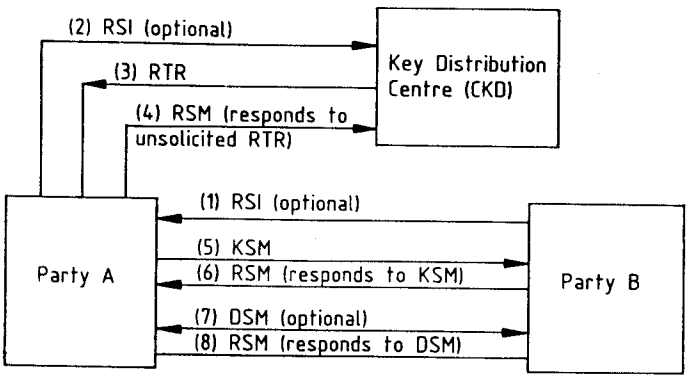
7 Required if and only if a count error occurs.

8 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.

9 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.

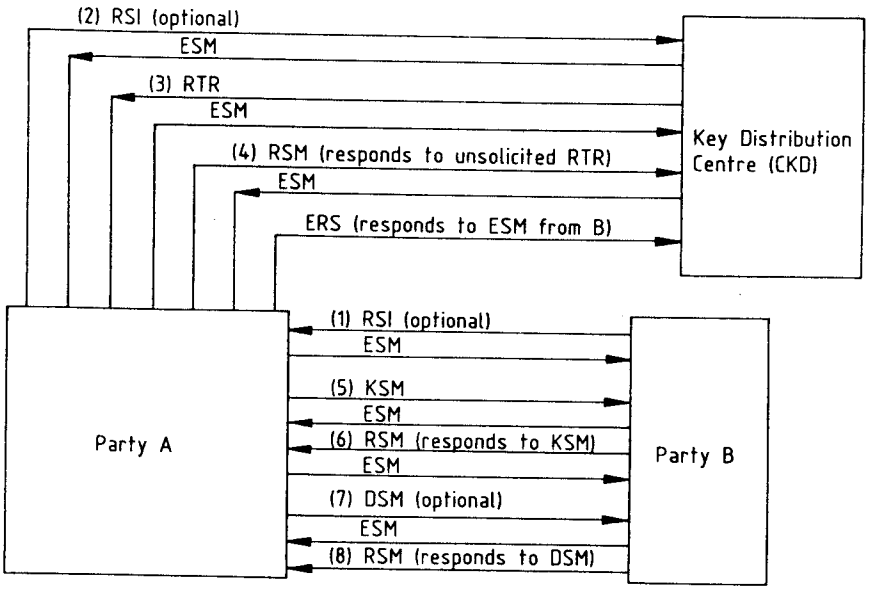
13.6.3 Key Distribution Centre (CKD) environment

Figures 7 and 8 show the flow of Cryptographic Service Messages for the Key Distribution Centre environment.



Note – Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 7 — Key Distribution Centre environment (normal message flow)



Note – Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 8 — Key Distribution Centre environment (message flow with Error Service Messages)

Table 4 defines the fields and their order for each Cryptographic Service Message in this environment. Message flow shall be as follows:

a) *Party wishing to establish cryptographic relationship has no communications link with centre.* When one logical party (Party B) wishes to communicate with another logical party (Party A), but does not share a key with Party A, then Party B may request Party A to use a Key Distribution Centre. Party B must share a *KK with a Key Distribution Centre that also has a *KK relationship with Party A. In this case Party B shall send an RSI to Party A requesting notarised KD(s) and optionally an IV (see figure 7).

If Party A receives an RSI that contains errors, an ESM shall be returned to Party B.

b) *Party establishing cryptographic relationship has communications link with centre.* When Party A wishes to send KD(s) to another party (Party B), but does not share a *KK with Party B, Party A may use a Key Distribution Centre. Party A must share a *KK with a Key Distribution Centre that also shares a *KK with Party B. In this case an RSI shall be sent to the Key Distribution Centre by Party A requesting the notarised KD(s) and optionally, an IV, for distribution to Party B, the ultimate recipient.

If the RSI received by the Key Distribution Centre contains errors, an ESM shall be returned to Party A.

13.6.3.1 When a Key Distribution Centre receives an RSI from a party (Party A), it shall generate or acquire the requested KD(s) and optionally an IV. Optionally, a Key Distribution Centre may generate an RTR without having received an RSI.

An RTR shall be sent to Party A containing two identical sets of KD(s). One set with the KD/ field tag shall be notarised using the *KK shared between Party A and the centre following the requirements of 12.4. The other set, with the KDU/ field tag shall be notarised using the *KK shared between Party B and the centre.

The IV, if enciphered, shall be enciphered under the KD (the second KD if two are present) sent in the Cryptographic Service Message. The Cryptographic Service Message shall be authenticated using the KD(s) contained in that message.

If the RTR received by Party A contains an error(s), Party A shall return an ESM to the centre (CKD). When an unsolicited RTR is received by Party A, Party A shall respond with an RSM if no errors were detected.

13.6.3.2 A KSM shall be sent by Party A to Party B containing the KDU(s) and the IV (if present) received in the RTR which caused the generation of the KSM. The Cryptographic Service Message shall be authenticated using the KD(s) received in the RTR. If Party B receives the KSM from Party A without error, then Party B shall return an RSM to Party A. If the KSM received by Party B contains errors, an ESM shall be returned to Party A. If the RSM received by Party A contains one or more errors, then Party A shall respond to that RSM by sending an ESM back to Party B. A KSM may be re-sent until acknowledged by an RSM or an ESM.

13.6.3.3 If Party A receives an ESM from Party B in response to a KSM identifying errors which can be attributed to Party B's relationship with the Key Distribution Centre, then Party A shall send an ERS to the Key Distribution Centre. These errors may be attributable to a CTB error, CKD unknown, IDK2 unknown and key parity errors in the received key. If an ERS received by the Key Distribution Centre contains errors, then the centre shall reply to Party A with an ESM.

13.6.3.4 If a Key Distribution Centre receives an ERS, then another RTR shall be generated after the identified errors have been reconciled (see 13.6.3.1 above).

13.6.3.5 If either party of a communicating pair wishes to terminate a keying relationship with the other party or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with an RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party 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 field is present) shall be retained to authenticate the subsequent RSM. That key shall then be discontinued. When a (*)KK is discontinued, all keys enciphered under that (*)KK shall also be discontinued without being named in the DSM.

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

Table 4 — Fields used with each message type: Key Distribution Centre environment

Message type	RSI to CKD	ESM	RSI from B to A	ESM	RTR	RSM	ESM	ERS	ESM	KSM	RSM	ESM	ESM	DSM	RSM	ESM
Responding to	—	RSI to CKD	—	RSI to A	—	RTR	RTR	—	ERS	—	KSM	KSM	RSM to KSM	—	DSM	DSM
Reference sub-clauses	14.7 15.7	14.4 15.4	14.7 15.7	14.4 15.4	14.9 15.9	14.8 15.8	14.4 15.4	14.3 15.3	14.4 15.4	14.5 15.5	14.8 15.8	14.4 15.4	14.4 15.4	14.2 15.2	14.8 15.8	14.4 (generation) 15.4 (processing)
See Notes to Fields																
	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG IDD IDA	MCL RCV ORG IDD	MCL RCV ORG
8 and 9 1	IDU	IDU	IDC	IDC	IDU	IDU	IDU	IDU	IDU	IDC	IDC	IDC	IDC			
2 2 1 1	SVR		SVR		KD KDU IV EDK		CTA	ERF SVR		KDU IV EDK						
7					CTB CTA		CTR	CTB CTR		CTB		CTB CTR				
1	EDC	ERF EDC	EDC	ERF EDC	MAC	MAC	ERF EDC	EDC	ERF EDC	MAC	MAC	ERF EDC	ERF EDC	MAC	MAC	ERF EDC

Notes

1 Optional

2 A maximum of two such fields may be sent in a message.

3 Required when notarisation of keys is used.

4 Present if and only if there is a (*)KKU field.

5 Present if and only if there is no (*)KKU field.

6 Only one KD field shall be present if a (*)KK field is present.

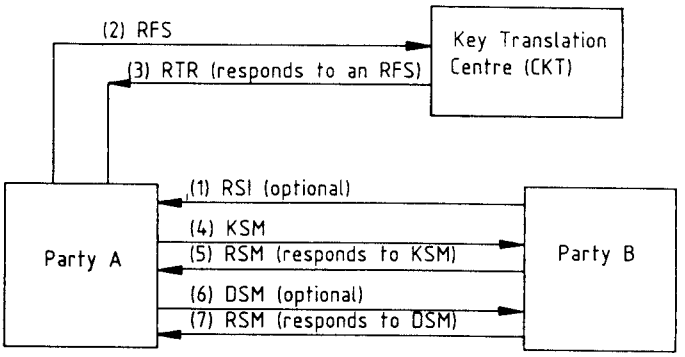
7 Required if and only if a count error occurs.

8 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.

9 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.

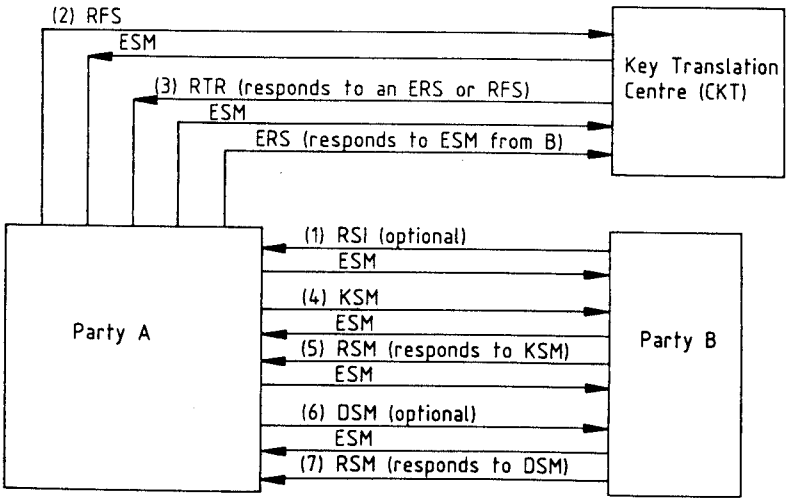
13.6.4 Key Translation Centre (CKT) environment

Figures 9 and 10 show the flow of Cryptographic Service Messages in a Key Translation Centre environment.



Note - Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 9 — Key Translation Centre environment (normal message flow)



Note - Either Party A or Party B may initiate the disconnect (DSM) process. Initiation by Party A is shown.

Figure 10 — Key Translation Centre environment (message flow with error messages)

Table 5 defines the fields and their order for each Cryptographic Service Message used in this environment.

Whenever a (*)KK relationship has been established between two parties in this environment, further exchange of keying material may be accomplished using procedures for the Point-to-Point environment. Cryptographic Service Message flow in the Key Translation Centre environment shall be as follows:

a) If one party (Party B) wishes to communicate with another party (Party A), but does not share a key with Party A, and does not have key generation capability, then Party B may use a Key Translation Centre. Party B must share a *KK with a Key Translation Centre which also has a *KK relationship with Party A.

An RSI may be sent from Party B to Party A requesting the type(s) of key(s) and optionally an IV to be provided.

If the RSI received by Party A from Party B contains an error(s), then Party A shall return an ESM to Party B.

b) If Party A wishes to send a key(s) to another party (Party B) but does not share a (*)KK with Party B, then Party A may use a Key Translation Centre. Party A must have a key generation or access capability and a *KK relationship with a Key Translation Centre that also shares a *KK with Party B.

Party A shall send an RFS to the Key Translation Centre, containing newly generated or acquired key(s) to be translated and eventually sent to Party B, the ultimate recipient.

If a (*)KK is sent, then one KD shall be sent. The (*)KK shall be enciphered under a *KK shared between Party A and the Key Translation Centre; the KD shall be enciphered under the (*)KK sent in the message. If a (*)KK is not sent, then at least one and at most two KDs shall be sent, enciphered under a *KK shared between Party A and a Key Translation Centre. The Cryptographic Service Message shall be authenticated using the KD(s) sent in the message.

The CKT shall respond with an RTR if the RFS is received correctly, or with an ESM if there is an error in the received RFS. If Party A receives an RTR which contains an error(s), Party A shall return an ESM to the CKT.

13.6.4.1 Upon receipt of an RFS by the Key Translation Centre:

1) The (*)KK, if present, shall be deciphered using the *KK shared between Party A and the Key Translation Centre. The deciphered (*)KK shall be notarised using the *KK shared between the centre and Party B, the count associated with the *KK and the identities of the two parties (A and B). It shall then be inserted as the (*)KKU field in an RTR for transmission to Party A. The KD in the RFS shall be used to authenticate both the RFS and the subsequent RTR. It shall not be translated or inserted in the RTR.

2) If a (*)KK is not present, then KD(s) shall be deciphered using the *KK shared between Party A and the centre. The deciphered KD(s) shall be notarised using the *KK shared between the centre and Party B, the count associated with the *KK and the identities of the two parties (A and B). They shall be inserted as the KDU field(s) in an RTR for transmission to Party A. The KD(s) shall be used to authenticate both the RFS and RTR messages.

If an error is detected in the RTR received by Party A, Party A shall return an ESM to the Key Translation Centre.

13.6.4.2 A KSM shall be sent by Party A to Party B containing the key(s) received in the RTR from the Key Translation Centre. If a (*)KKU is present, KD(s) (generated or acquired) shall be inserted in the KSM in the KD field(s). The KD(s) shall be enciphered under the (*)KK sent to the Key Translation Centre in the RFS that caused the RTR to be sent to Party A. KDU(s) are transferred directly from the RTR to the KSM.

If an IV (generated or acquired) is to be enciphered, it shall be enciphered under the KD (the second if two KDs are present) sent in the CSM.

The CSM shall be authenticated using the KD(s) sent in the message.

Party A may resend a KSM to Party B an arbitrary number of times, but Party A shall not send a new KSM (i.e., utilising new keys or a new count for the (*)KK specified in a message) until the old KSM is acknowledged by an RSM or an ESM.

If an error is detected by Party B in the KSM sent by Party A, then an ESM shall be returned to Party A. Otherwise, an RSM shall be returned to Party A. If Party A detects an error in the RSM, Party A shall return an ESM to Party B in response to the RSM.

13.6.4.3 If an ESM is received by Party A from Party B in response to a KSM with errors attributable to the relationship between the Key Translation Centre and Party B, then an ERS shall be sent to the Key Translation Centre. New keys shall be generated or acquired.

If an ERS received by the Key Translation Centre contains errors, then the centre shall reply to Party A with an ESM.

13.6.4.4 Upon receipt of an ERS from Party A, the Key Translation Centre shall prepare another RTR (see 14.9) after resolving any problems in the Key Translation Centre-to-Party B (ultimate recipient) relationship.

13.6.4.5 If either party of a communicating pair wishes to terminate a keying relationship with the other party or wishes to discontinue the use of a specific key(s), that party may send a DSM to the second party.

The second party shall respond with an RSM if the DSM is received correctly and all information contained in the DSM was applicable. Otherwise, the second party 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 field is present) shall be retained to authenticate the subsequent RSM. Note that that key shall then be discontinued. When a (*)KK is discontinued, all keys enciphered under that (*)KK shall also be discontinued without being named in the DSM.

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

Table 5 — Fields used with each message type: Key Translation Centre environment

Message type	RSI	ESM	RFS	ESM	ERS	ESM	RTR	ESM	KSM	RSM	ESM	ESM	DSM	RSM	ESM
Responding to	—	RSI	—	RFS or ERS	—	ERS	RFS or ERS	RTR	—	KSM	KSM	RSM to KSM	—	DSM	DSM
Reference sub-clause	14.7 15.7	14.4 15.4	14.6 15.6	14.4 15.4	14.3 15.3	14.4 15.4	14.9 15.9	14.4 15.4	14.5 15.5	14.8 15.8	14.4 15.4	14.4 15.4	14.2 15.2	14.8 15.8	14.4 (generation) 15.4 (processing)
See Notes to Fields															
	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG	MCL RCV ORG
8 and 9													IDD	IDD	
1			IDU	IDU	IDU	IDU	IDU	IDU					IDA		
	IDC	IDC							IDC	IDC	IDC	IDC			
1			(*)KK		(*)KK										
1							(*)KKU		(*)KKU						
2 and 6			KD		KD				KD						
2 and 5							KDU		KDU						
1									IV						
1									EDK						
					ERF										
	SVR														
			CTA	CTA		CTA									
					CTB		CTB		CTB		CTB				
7				CTR	CTR	CTR					CTR				
					CTA										
		ERF		ERF		ERF		ERF			ERF	ERF			ERF
1	EDC	EDC		EDC		EDC		EDC			EDC	EDC			EDC
			MAC		MAC		MAC		MAC	MAC			MAC	MAC	

Notes

1 Optional

2 A maximum of two such fields may be sent in a message.

3 Required when notarisation of keys is used.

4 Present if and only if there is a (*)KKU field.

5 Present if and only if there is no (*)KKU field.

6 Only one KD field shall be present if a (*)KK field is present.

7 Required if and only if a count error occurs.

8 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.

9 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.

14 Generation of Cryptographic Service Messages

14.1 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 clause of this standard which shall be used in generating each type of Cryptographic Service Message.)

Message type	MCL field contents	Reference
Disconnect Service Message	DSM	Clause 14.2
Error Recovery Service message	ERS	Clause 14.3
Error Service Message	ESM	Clause 14.4
Key Service Message	KSM	Clause 14.5
Request For Service message	RFS	Clause 14.6
Request Service Initiation message	RSI	Clause 14.7
Response Service Message	RSM	Clause 14.8
Response To Request message	RTR	Clause 14.9

Thus, following the rules given in 13.4, a Disconnect Service Message commences with

CSM(MCL/DSM....

14.2 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 6.

Table 6 — Content 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. eg. if RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. eg if OOOO is the identity, the field becomes: ORG/OOOO
IDD	Any number of such fields may be sent. Insert the identity of the 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 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.

14.3 Generate Error Recovery Service message (ERS)

An Error Recovery Service message (ERS) is sent to a CKD or CKT by the originator of a KSM (ie, Party A) in response to an ESM received from the recipient of the KSM (ie, Party B) (see 13.6.3 and 13.6.4). This Cryptographic Service Message is used to

1) announce to the CKD or CKT that errors were received in the KSM (by Party B) which are attributable to a problem in the key or count shared by Party B and the CKD or CKT, and

2) request that further keys be processed (after appropriate corrections have been made) to be sent by Party A to Party B.

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

Table 7 — Content of fields in ERS

Field tag	Content
MCL	Insert ERS in the field. The field becomes: MCL/ERS
RCV	Insert identity of the CKD or CKT in the field. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDU	Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
(*)KK	(CKT environment only) (optional) In using an ERS to recover, new key(s) shall be generated or acquired.
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of 13.5.
	Let (*)KKZ be the key to be enciphered (ie, to be sent to the IDU party) and *KKY be the Key Enciphering Key shared with the CKT. Use the method of 12.3 to compute the offset, *KKoY. The enciphered (*)KK is computed using: enciphered KKZ = ede*KKoY(KKZ) or enciphered *KKZ = ede*KKoY(*KKZ) Using brackets to denote the representation of field contents, then the field becomes: KK/[ede*KKoY(KKZ) (optional subfields)] or *KK/[ede*KKoY(*KKZ) (optional subfields)]

Table 7 — Content of fields in ERS *(continued)*

Field tag	Content
KD	(CKT environment only) In using an ERS to recover, new keys shall be generated or acquired.
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message, or to specify the Key Enciphering Key to be used to decipher the KDs (or any combination thereof), form and insert the applicable subfields using the rules of Section 13.5. *Case 1:* The ERS contains a (*)KK field One and only one KD shall be sent in the ERS in the KD field. That KD shall be used to authenticate the ERS and to generate the MAC for inclusion in the RTR that responds to this ERS. Let KDI be the key to be sent in this field. Let (*)KKY be the Key Enciphering Key sent in the message. The KD is computed as follows: The KD is enciphered by a (*)KK; an offset of zero is used. a) Encipher the KDI using the equation: enciphered KDI = eKKoY(KDI) or enciphered KDI = ede*KKoY(KDI) b) If brackets are used to denote the representation of field contents, the field becomes: KD/[eKKoY(KDI).P], or KD/[ede*KKoY(KDI).P] where P is an optional subfield. *Case 2:* There is no (*)KK field in the message. At least one and at most two KDs shall be sent in a ERS as KD field(s). A KD shall be enciphered by a *KK, as follows: a) Use the procedure of 12.3 to compute the *KKoY. b) Encipher the KDI using the equation: enciphered KDI = ede*KKoY(KDI) c) and the field becomes KD/[ede*KKoY(KDI) (optional subfields)]
ERF	Copy error codes applicable to the Party B/centre relationship from the ERF field received in the ESM from which the ERS is generated to the ERF field.
SVR	(CKD environment only) Insert the subfields designating the type of service requested (see 13.6 and table 2: SVR). Note that a single data key is implicitly requested by the presence of a SVR field, and that an (*)KK shall not be requested, e.g., SVR/KD to request two data keys SVR/KD.IV to request two data keys and an enciphered IV

Table 7 — Content of Fields in ERS *(concluded)*

Field tag	Content
CTB	Let "b" be the contents of the CTB field received in the ESM from which the ERS is generated. Then the contents of the CTB field in the ERS are set equal to "b" and the field becomes: CTB/b
CTR	(used when a count error has been detected) Let "r" be the contents of the CTR field received in the ESM from which the ERS is generated. Then the contents of the CTR field in the ERS are set equal to "r" and the field becomes: CTR/r
CTA	(CKT environment only) If the value of the CTA before ERS is "a", then the ERS shall contain the CTA field: CTA/a
MAC	(CKT environment only) 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/ERSb...bCTA/ab) and the field becomes: MAC/[aKDJ(MCL/ERSb...bCTA/ab)] 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.
EDC	(CKD environment only) (optional) The data key for EDC computation shall be: KDX = 0123456789ABCDEF (but see Note in 12.1.1.) The EDC is computed using: EDC = aKDX(MCL/ERSb...b) and the field becomes: EDC/[aKDX(MCL/ERSb...b)] 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.

14.4 Generate Error Service Message (ESM)

An Error Service Message (ESM) is sent in response to the detection of one or more of the following error conditions in a Cryptographic Service Message (other than an ESM) as listed in table 2 ERF (see 13.6.1).

Error Service Messages shall be generated by computing or selecting field contents in accordance with table 8.

Table 8 — 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. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDC	Error messages responding to a KSM or an RSM in a CKD or CKT environment shall return the IDC field contents. If the IDC field contents are CCCC, then the field becomes: IDC/CCCC
IDU	All error messages responding to an ERS, RTR, or RSI in a CKD environment; or to an ERS, RFS or RTR in a CKT environment shall return the IDU field contents. If the IDU field contents are UUUU, then the field becomes: IDU/UUUU
CTA	(Used in a CKD environment when responding to an RTR, or in a CKT environment when responding to an ERS or RFS) The count returned in this field shall contain the expected CTA (see 12.2). If the value is "a", the field becomes: CTA/a
CTB	(Used in a CKD or CKT environment when responding to a KSM) The count returned in this field shall contain the expected CTB (see 12.2). If the value is "b", the field becomes: CTB/b
CTP	(Used only in a Point-to-Point environment when responding to a KSM) The count returned in this field shall contain the expected CTP (see 12.2). If the value is "p", the field becomes: CTP/p

Table 8 — Contents of fields in ESM *(concluded)*

Field tag	Content
CTR	(Used when a count error has been detected) The count returned is the count included in the message to which this ESM responds. The following table identifies the situations when the CTR field shall be used in the message along with the count. The received count shall be copied from the previous message and inserted in the CTR field. **Environment** / **Previous message** / **Count** Point-to-Point / KSM / CTP CKD / KSM / CTB CKD / RTR / CTA CKT / KSM / CTB CKT / RFS / CTA CKT / ERS / CTA
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 13.6 (table 2). Multiple error conditions are indicated by returning a concatenated string of error flags, eg: ERF/KPM
EDC	(Optional) The data key for EDC computation shall be: KDX = 0123456789ABCDEF (see NOTE in 12.1.1) 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.

14.5 Generate Key Service Message (KSM)

A Key Service Message (KSM) may be generated spontaneously or in response to an RSI received from another party in a Point-to-Point environment (see 13.6.2). In the CKD and CKT environments, however, the KSM is generated following receipt of an RTR from the CKD or CKT during a sequence of Cryptographic Service Message exchanges (see 13.6.3 and 13.6.4).

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 an identical KSM 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 9.

Table 9 — 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. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDC	(CKD or CKT environment only) Insert the identity of the CKD or CKT used in the key distribution process of which this KSM is a step. If CCCC is the identity of the centre, the field becomes: IDC/CCCC
NOS	(Point-to-Point environment only) (optional) If notarisation of the (*)KK (or KDs if no (*)KK is sent in the message) is desired, include the NOS field in the KSM as NOS/
(*)KK	(only used in a Point-to-Point environment) (optional)
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. If a (*)KK is sent, an associated count shall be established for key offsetting the (*)KK when other keys are received which are enciphered using this (*)KK. The count is initially set to one, and the value shall be used to key offset and encipher the KD which is sent in this message. The (*)KK used to encipher the (*)KK sent in the Cryptographic Service Message shall be a key currently shared with the message recipient. *Case 1:* KK to be sent enciphered by a KK, no notarisation. If a new KK is to be sent, then for a single length KK, the enciphered KK is computed using the following equations: Let KKZ be the key to be enciphered and KKY be the enciphering key. Use the procedure of 12.3 and the value of CTP to compute the KKoY.

Table 9 — Contents of fields in KSM *(continued)*

Field tag	Content
(*)KK *(continued)*	Encipher KKZ enciphered KKZ = eKKoY(KKZ) Using brackets to denote the representation of field contents, the field becomes: KK/[eKKoY(KKZ) (optional subfields)] *Case 2:* (*)KK to be sent enciphered by a *KK, no notarisation. Use the procedure of 12.3 and the value of CTP to compute the *KKoY. Encipher (*)KKZ Enciphered KKZ = ede*KKoY(KKZ) or Enciphered *KKZ = ede*KKoY(*KKZ) If brackets are used to denote field contents, the field becomes: KK/[ede*KKoY(KKZ) (optional subfields)] or *KK/[ede*KKoY(*KKZ) (optional subfields)] *Case 3:* (*)KK with notarisation. The (*)KK field contents are computed as follows: a) Compute (*)KN using the contents of the ORG, RCV, and CTP fields and the process defined in 12.4 b) Encipher the (*)KKZ to form the contents of the (*)KK field using the equations: notarised KKZ = eKN(KKZ) or notarised KKZ = ede*KN(KKZ) or notarised *KKZ = ede*KN(*KKZ) respectively. c) the field becomes: KK/[eKN(KKZ) (optional subfields)] or KK/[ede*KN(KKZ) (optional subfields)] or *KK/[ede*KN(*KKZ) (optional subfields)] respectively.
(*)KKU	(CKT environment only) (optional) Let (*)KKUC be the contents of the (*)KKU field received in the RTR message from which the KSM is generated. Then the contents of the (*)KKU field in the KSM are set equal to (*)KKUC and the field becomes: (*)KKU/[(*)KKUC (optional subfields)] If a (*)KKU is sent, an associated count shall be established for key offsetting the (*)KK when other keys are sent using this (*)KKU. The count shall be initially set to one, and the value shall be used to key offset encipher the KD which is sent in this message.

Table 9 — Contents of fields in KSM *(continued)*

Field tag	Content
KD	(Not used in a CKD environment) At least one and at most two KDs shall be sent in a KSM as KD field(s). In a CKT environment, KD field(s) shall be present if and only if a (*)KKU field is contained in this Cryptographic Service Message. If a new (*)KK is sent in the KSM, then the KDs shall be enciphered using that key.
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the KDs (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. KDs shall be enciphered or given notarisation protection using the processes that follow. *Case 1:* KD enciphered by a (*)KK; no notarisation. KDs are enciphered by a (*)KK and are not notarised (i.e., there is no NOS field in the KSM or a (*)KK or (*)KKU is sent). If a (*)KK or (*)KKU is sent in this Cryptographic Service Message, this (*)KK shall be used to encipher the KDs. Otherwise, a currently shared (*)KK is used. Let KDI be the key to be sent in this field. Let (*)KKY be the Key Enciphering Key to be used. Then the KD(s) is (are) computed as follows: a) Use the procedure of 12.3 to compute the (*)KKoY: using CTP in a Point-to-Point environment if no (*)KK is sent in the message and a value of one (1) if a new (*)KK is sent. In a CKT environment the count shall be set equal to a value of one (1). b) Encipher the KDI using the equation: enciphered KDI = eKKoY(KDI) or enciphered KDI = ede*KKoY(KDI) c) If brackets are used to denote field contents, the field becomes: KD/[eKKoY(KDI) (optional subfields)] or KD/[ede*KKoY(KDI) (optional subfields)] *Case 2:* KD notarised under a (*)KK (only in a Point-to-Point environment). There is no (*)KK field in the message and the KDs are to be given notarisation protection. In this case, the NOS field shall be included in the KSM and the KD fields content shall be computed as follows: a) Compute (*)KN using the contents of the ORG, RCV and CTP fields and the process defined in 12.4. b) Encipher the KDI to form the contents of the KD field using the equations: notarised KDI = eKN(KDI) or notarised KDI = ede*KN(KDI) for a KN or *KN, respectively. c) The field becomes: KD/[eKN(KDI) (optional subfields)] or KD/[ede*KN(KDI) (optional subfields)] for a KN or *KN, respectively.

Table 9 — Contents of fields in KSM *(continued)*

Field tag	Content
KDU	(CKD or CKT environment only) At least one and at most two KDs shall be sent in a KSM as KDU field(s). This field shall be present in a KSM if and only if there is no (*)KKU field in the Cryptographic Service Message. Let KDUC be the contents of the KDU field received in the RTR from which this KSM is generated. Then the contents of the KDU field in the KSM are set equal to KDUC and the field becomes: KDU/[KDUC (optional subfields)]
IV	(Optional) *Case 1:* Enciphered IV (Point-to-Point and CKT environments only). If an IV is sent in enciphered form, 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 || eKD(IV)] *Case 2:* Plaintext IV (Point-to-Point and CKT environments only). If an IV is sent in plaintext form, then the field is: IV/[P || IV] *Case 3:* Enciphered or plaintext IV (CKD environment only). Let IVC be the contents of the IV field received in the RTR from which this KSM is generated, if present, otherwise the IV (IVC) may be determined by the originating party. Then the contents of the IV field in the KSM are set equal to IVC and the field becomes: IV/IVC
EDK	(Optional) Let YYMMDDHHMMSS be the contents of the EDK field received in the RTR from which this KSM is generated, if present, otherwise the EDK field may be determined by the originating party. Then the contents of the EDK field shall be: YYMMDDHHMMSS and the field becomes: EDK/[YYMMDDHHMMSS]
CTB	(CKD or CKT environment only) CTB is the value of the counter shared by the CKD or CKT and the ultimate recipient. This CTB field is formed by setting it equal to the CTB field value received in the RTR from the CKD or CKT which initiated the generation of this KSM. If the value received in the CTB field of the RTR is "b", then the CTB field is: CTB/b

Table 9 — Contents of fields in KSM *(continued)*

Field tag	Content
CTP	(Point-to-Point environment only) If the value of the CTP before KSM preparation is "p", then the KSM shall contain the CTP field: CTP/p
MAC	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...bCTX/xb) and the field becomes: MAC/[aKDJ(MCL/KSMb...bCTX/xb)] In a Point-to-Point environment CTX is CTP and x is "p"; and in a CKD or CKT environment CTX is CTB and x is "b". 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.

13.6 Generate a Request For Service message (RFS)

A Request For Service message (RFS) is only sent to a CKT (see 13.6.4). The RFS may be initiated by the originating party or may be generated following receipt of an RSI.

All keys and IVs in an RFS shall be generated by the originator and sent to the CKT. Request For Service messages shall be generated by computing or selecting field contents in accordance with table 10.

Table 10 — Contents of fields in RFS

Field tag	Content
MCL	Insert RFS in the field. The field becomes: MCL/RFS
RCV	Insert recipient's identity (i.e., the identity of the CKT) in the field. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDU	Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
(*)KK	(Optional)
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of 13.5, above. If a (*)KK is to be sent, the (*)KK shall be generated or acquired and an associated count shall be established for key offsetting the (*)KK when other keys are enciphered using this (*)KK. This count is initially set to zero, and the value zero shall be used to key offset encipher the KD to be sent in this Cryptographic Service Message. Let (*)KKZ be the key to be enciphered (ie, to be sent to the IDU party) and *KKY be the enciphering key shared with the CKT. If a new (*)KK is to be sent, then the enciphered (*)KK is computed as follows: Use the procedure of 12.3 to compute the *KKoY Then: enciphered KKZ = ede*KKoY(KKZ) or enciphered *KKZ = ede*KKoY(*KKZ) Then the field becomes: KK/[ede*KKoY(KKZ) (optional subfields)] or *KK/[ede*KKoY(*KKZ) (optional subfields)]
KD	KDs sent in an RFS to a CKT are enciphered data keys that are used to authenticate that RFS. If no (*)KK field is present, KDs are sent to the ultimate recipient as KDUs. Note that KDs are never given notarisation protection in an RFS.

Table 10 — Contents of fields in RFS *(concluded)*

Field tag	Content
KD *(continued)* *Optional subfields* P IDK1 IDK2	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the KDs (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. *Case 1:* The RFS contains a (*)KK field. One and only one KD shall be sent in the RFS in the KD field. That KD shall be used to authenticate the RFS and to generate the MAC for inclusion in the RTR that responds to this RFS. Let KDI be the key to be sent in this field. Let (*)KKY be the Key Enciphering Key sent in the message. The KD is computed using the equations: Use the procedure of 12.3 and an offset of zero to compute (*)KKoY. a) Encipher the KDI using the equation: enciphered KDI = eKKoY(KDI) or enciphered KDI = ede*KKoY(KDI) b) If brackets are used to denote field contents, the field becomes: KD/[eKKoY(KDI).P] or KD/[ede*KKoY(KDI).P] where P is an optional subfield. *Case 2:* There is no (*)KK field in the message. At least one and at most two KDs shall be sent in a RFS as KD field(s). A KD shall be enciphered by a *KK, and the following equations are used: a) Use the procedure of 12.3 and the value of CTA to compute the *KKoY. b) Encipher the KDI using the equation: enciphered KDI = ede*KKoY(KDI) c) If brackets are used to denote field contents, the field becomes: KD/[ede*KKoY(KDI) (optional subfields)] If the value of the CTA before RFS preparation is "a", then the RFS shall contain the CTA field: CTA/a
MAC	The MAC is always computed using the KDs sent in the Cryptographic Service 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/RFS$\underline{b}$...$\underline{b}$CTA a$\underline{b}$) and the field becomes: MAC/[aKDJ(MCL/RFS$\underline{b}$...$\underline{b}$CTA/a$\underline{b}$)] 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 MAC field.

14.7 Generate Request Service Initiation message (RSI)

A Request Service Initiation message (RSI) is generated by the originating party in all environments in order to request that keys be sent in a subsequent KSM to establish a keying relationship.

In the CKD environment, the RSI is sent to the CKD to request keys which shall be sent to another party (the ultimate recipient) in a later KSM (see 13.6.3).

Request Service Initiation messages shall be generated by computing or selecting field contents in accordance with table 11.

Table 11 — 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. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDU	(Required only in a CKD environment when the RSI is sent to the centre) Insert the identity of the ultimate recipient in the field. If UUUU is the identity, the field becomes: IDU/UUUU
IDC	(Required in an RSI from Party B to Party A (see 13.6.3 and 13.6.4 in a centre environment (only)) Insert the identity of the CKD or CKT in the field. If CCCC is the identity, then the field becomes: IDC/CCCC
SVR	Insert the subfields designating the type of service requested (see 13.6 and table 2 SVR). Note that a single data key is implicitly requested by the presence of a SVR field, e.g. SVR/ to request one data key SVR/KD to request two data keys SVR/KK to request a single data key and a Key Enciphering Key SVR/KK.KD.IV to request a Key Enciphering Key, two data keys and an enciphered IV SVR/*KK.KD.IV to request a Key Enciphering Key Pair, two data keys and an enciphered IV A (*)KK shall not be requested in a CKD environment.
EDC	(Optional) The data key for EDC computation shall be:

Table 11 — Contents of fields in RSI *(concluded)*

Field tag	Content
EDC *(continued)*	KDX = 0123456789ABCDEF (see NOTE in 12.1.1) The EDC is computed using: EDC = aKDX(MCL/RSIb...bSVR/*KK.KD.IVb) and the field becomes: EDC/[aKDX(MCL/RSIb...bSVR/*KK.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.

14.8 Generate Response Service Message (RSM)

A Response Service Message (RSM) is generated following receipt of an acceptable DSM or KSM. Response Service Messages shall be generated by computing or selecting field contents in accordance with table 12.

Table 12 — 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. If RRRR is the identity, the field becomes: RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity, the field becomes: ORG/OOOO
IDC	(CKD or CKT environment only, when responding to a KSM) Insert the identity of the CKD or CKT used in the key distribution process of which this RSM is a step. E.g., if CCCC is the identity, the field becomes: IDC/CCCC
IDD	(Only used in response to a DSM) Copy the IDD field(s) from the DSM to which this RSM responds.
IDU	(In response to an unsolicited RTR; CKD environment only) Insert the identity of the ultimate recipient in the field. If the identity of the ultimate recipient is UUUU, then the field becomes: IDU/UUUU

Table 12 — Contents of fields in RSM *(continued)*

Field tag	Content
MAC	The MAC is always computed using the KDs sent or specified in the DSM, KSM or RTR 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 that data key is unnamed. 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/KKKKb) responding to a DSM aKDJ(MCL/RSMb...bIDC/CCCCb) responding to a KSM in a centre environment aKDJ(MCL/RSMb...bORG/OOOOb) responding to a KSM in a Point-to-Point environment aKDJ(MCL/RSMb...bIDU/UUUUb) responding to an unsolicited RTR in a CKD environment and the field becomes: MAC/[aKDJ(MCL/RSMb...bIDD/KKKKb)] responding to a DSM MAC/[aKDJ(MCL/RSMb...bIDC/CCCCb)] responding to a KSM in a centre environment MAC/[aKDJ(MCL/RSMb...bORG/OOOOb)] responding to a KSM in a Point-to-Point environment MAC/[aKDJ(MCL/RSMb...bIDU/UUUUb)] responding to an unsolicited RTR in a CKD environment 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.

14.9 Generate Response To Request message (RTR)

A Response To Request message (RTR) is generated in response to an RFS (CKT environment), an RSI (CKD environment), or an ERS (either environment) (see 13.6.3 and 13.6.4). An unsolicited RTR may be generated in a CKD environment.

Response To Request messages shall be generated by computing or selecting field contents in accordance with table 13.

Table 13 — Contents of fields in RTR

Field tag	Content
MCL	Insert RTR in the field. The field becomes: MCL/RTR
RCV	Insert recipient's identity in the field. If RRRR is the identity, the field becomes RCV/RRRR
ORG	Insert originator's identity in the field. If OOOO is the identity of the CKD or CKT, the field becomes ORG/OOOO
IDU	Insert the identity of the ultimate recipient in the field. If the identity of the ultimate recipient is UUUU, then the field becomes: IDU/UUUU
(*)KKU	(CKT environment only) (optional) This field is present if and only if there was a (*)KK field present in the RFS or ERS to which this RTR responds.
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the (*)KK being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the (*)KK (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. Let *KKY be the *KK shared between Party B and the CKT. Let (*)KKZ be the (*)KK that is to be sent to Party B, the ultimate recipient. The (*)KKZ is the Key Enciphering Key received from Party A in the RFS to which this RTR responds. The (*)KKZ shall be notarised before transmission. The (*)KKU field contents shall be computed as follows: a) Compute *KN using *KKY and the contents of the RCV, IDU and CTB fields and the process defined in 12.4. b) Encipher the (*)KKZ to form the contents of the (*)KKU field using the equations: KKU = notarised KKZ = ede*KN(KKZ) or *KKU = notarised *KKZ = ede*KN(*KKZ) c) and the field becomes, eg, KKU/[ede*KN(KKZ) (optional subfields)] or *KKU/[ede*KN(*KKZ) (optional subfields)]

Table 13 — Contents of fields in RTR *(continued)*

Field tag	Content
KD	(CKD environment only) At least one and at most two KDs shall be generated and sent in an RTR message as KD field(s).
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the KDs (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. KDs shall be enciphered using the process that follows. Let KDI be the key to be sent in this field. Let *KKY be the Key Enciphering Key shared by Party A and the CKD to be used. The following process is used: a) Compute the *KN using *KKY shared by Party A and the CKD and the contents of the RCV, IDU and CTA fields in this RTR, and the process defined by 12.4. b) Encipher the KDI using the equation: enciphered KDI = ede*KN(KDI) c) and the field becomes: KD/[ede*KN(KDI) (optional subfields)]
KDU	If, and only if, no (*)KK is being sent as a (*)KKU, then at least one and at most two notarised KDs for the ultimate recipient shall be sent as KDU field(s).
Optional subfields	
P IDK1 IDK2	If it is desired to use the odd parity feature, to name the KDs being sent in the Cryptographic Service Message, to specify the Key Enciphering Key to be used to decipher the KDs (or any combination thereof), form and insert the applicable subfields using the rules of 13.5. Let KDJ be a data key that is to be sent to the ultimate recipient. The KDU field(s) contents shall be computed as follows: a) Compute *KN using the Key Enciphering Key, *KKY, shared by Party B and the CKD, the contents of the RCV, IDU and CTB fields, and the process defined in 12.4. b) Encipher the KD to form the contents of the KDU field using the equation: KDU = notarised KD = ede*KN(KD) c) and the field becomes: KDU/[ede*KN(KD) (optional subfields)]
IV	(CKD environment only) (optional) *Case 1:* Enciphered IV. If an IV is to be sent in enciphered form, the IV shall be enciphered as follows: Let KDH be the KD sent in this Cryptographic Service Message. If two KDs are sent, the second KD (used for encipherment) shall be used. enciphered IV = eKDH(IV)

Table 13 — Contents of fields in RTR *(concluded)*

Field tag	Content
IV *(continued)*	and the field is: IV/[E || eKDH(IV)] *Case 2:* Plaintext IV. If an IV is sent in plaintext form, then the field is: IV/[P || IV]
EDK	(CKD environment only) (optional) If an EDK is sent, the field shall be EDK/[YYMMDDHHMMSS]
CTB	If the value of the CTB field before RTR preparation is "b", then the RTR shall contain the CTB field: CTB/b
CTA	(CKD environment only) CTA is the value of the counter associated with the *KK shared by the originator of the RSI or ERS to which this RTR responds and the CKD. If the value of CTA before RTR preparation is "a", then the RTR shall contain the CTA field: CTA/a
MAC	In a CKT environment, when a (*)KKU is returned in the RTR, the KD used to authenticate the RTR shall be the KD received for authenticating the RFS. In a CKD environment, the KDs used in the MAC computation are the KDs being 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/RTRb...bCTX/xb) and the field becomes: MAC/[aKDJ(MCL/RTRb...bCTX/xb)] Where CTX is CTA and x is a in a CKD environment; and CTX is CTB and x is b in a CKT environment. 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.

15 Processing Cryptographic Service Messages

15.1 Determination of message type

The message type of the 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 clause of this standard which shall be used in processing each type of Cryptographic Service Message.)

Message type	MCL field contents	Reference
Disconnect Service Message	DSM	Clause 15.2
Error Recovery Service message	ERS	Clause 15.3
Error Service Message	ESM	Clause 15.4
Key Service Message	KSM	Clause 15.5
Request For Service message	RFS	Clause 15.6
Request Service Initiation message	RSI	Clause 15.7
Response Service Message	RSM	Clause 15.8
Response To Request message	RTR	Clause 15.9

Thus, following the rules given in 13.4, 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

15.2 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 Messages shall be processed by computing or selecting field contents in accordance with table 14.

Table 14 — 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 (*)KK or KD to be discontinued. The IDD field(s) shall be inserted into the RSM generated in response to this Cryptographic Service Message. 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, ie, 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.

Table 14 — Processing of DSM *(concluded)*

Field tag	Action
IDA *(continued)*	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,ie, 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", ie, ERF/M or b) the error shall be resolved by manual means (eg, 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.

15.3 Process Error Recovery Service message (ERS)

An Error Recovery Service message (ERS) is received by a CKD or CKT which:

1) announces that a problem exists in the key or count shared by the CKD or CKT and the party identified by the IDU field (ultimate recipient), and

2) requests that further keys be processed (after appropriate corrections are made) and sent to the originator of the ERS in an RTR. The keys shall then be forwarded to the ultimate recipient in a KSM (see 13.6.3 and 13.6.4).

Error Recovery Service messages shall be processed by computing or selecting field contents in accordance with table 15.

Table 15 — Processing of ERS

Field tag	Action
MCL	Confirm field is MCL/ERS.
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 ERS 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.
IDU	This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used in selecting the *KK to be used in generating the response to this Cryptographic Service Message. If this identity is not known, an ESM shall be sent with a "U" in the ERF field. Further Cryptographic Service Message processing may be performed in order to check keys, counts and message authentication prior to actual transmission of the ESM.
(*)KK	(CKT environment only) (When present) Using the rules of 13.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If the IDK2 subfield is not present, the *KK used to decipher the received (*)KK is the only *KK shared by the message originator and the CKT.
Optional subfields	
P	If the "P" subfield 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 ERS, with a "K" in the ERF field. All further Cryptographic Service Message processing shall cease. The "P" shall be inserted in the "P" subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message using the rules of 13.5.
IDK1	If present, the IDK1 name shall be inserted in the IDK1 subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message, using the rules of 13.5. Otherwise, no IDK1 subfield shall be sent in the (*)KKU field in the RTR.
IDK2	If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the ERS to cease and the generation and transmission to the originating party of an ESM with an "I" in the ERF field, ie, ERF/I *Decipherment of (*)KKs* The deciphered (*)KK is computed as follows: Let (*)KKZ be the key to be deciphered and *KKY be the deciphering key.

Table 15 — Processing of ERS *(continued)*

Field tag	Action
(*)KK *(continued)*	
IDK2 *(continued)*	Use the procedure of 12.4 and the contents of the CTA field to compute the *KKoY. Decipher (*)KKZ deciphered KKZ = ded*KKoY(enciphered KKZ) or deciphered *KKZ = ded*KKoY(enciphered *KKZ)
KD	(CKT environment only) At least one and at most two KDs shall be received in an ERS as KD field(s). If a new (*)KK is sent in the ERS, then only one KD shall be received in the ERS and that KD shall be deciphered using that (*)KK. The KD is to be deciphered by a *KK currently shared with the message originator. The KD field may have subfields as defined in 13.5. Select the *KK as defined below. Parse the KD field(s) to obtain the KD; and the P, IDK1, and IDK2 subfields if present.
Optional subfields	
P	If the "P" subfield 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 ERS, with a "K" in the ERF field, ie, ERF/K Further processing of the ERS shall cease. If the key parity is correct, the "P" shall be inserted in the "P" subfield of the RTR generated in response to this Cryptographic Service Message using the rules of 14.5.
IDK1	If present, this subfield names the KD. The IDK1 name shall be inserted in the IDK1 subfield of the KDU field in the RTR generated in response to this Cryptographic Service Message, using the rules of 14.5.
IDK2	The *KK used to decipher the KD is identified by the IDK2 subfield, if present. See 14.5. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the ERS to cease and the generation and transmission to the originating party of an ESM with an "I" in the ERF field, ie, ERF/I *Decipherment of the KDs:* *Case 1:* The ERS contains a (*)KK field. One and only one KD shall be received in the ERS in the KD field. That KD shall be used to authenticate the ERS and to generate the MAC for inclusion in the RTR that responds to this ERS. Let KDI be the key received in this field. Let (*)KKY be the Key Enciphering Key sent in the Cryptographic Service Message. The KD is deciphered as follows: a) The KD is deciphered by an (*)KK; an offset of zero is used. b) Decipher the KDI using the equation: deciphered KDI = dKKoY(KDI) or deciphered KDI = ded*KKoY(KDI)

Table 15 — Processing of ERS *(continued)*

Field tag	Action
KD *(continued)* IDK2 *(continued)*	*Case 2:* There is no (*)KK field in the Cryptographic Service Message. At least one and at most two KDs shall be received in an ERS as KD field(s). The KDs are deciphered by a *KK, as follows: a) Use the procedure of 12.3 to compute the *KKoY b) Decipher the KDI using the equation: deciphered KDI = ded*KKoY(KDI)
ERF	Parse the field to obtain the codes for the types of errors reported. These error types shall be utilised in generating the RTR that responds to this ERS or in the manual recovery process, if necessary.
SVR	(CKD environment only) Parse the field to obtain the designators for the types of service requested. These service types shall be utilised in generating the RTR that responds to this ERS. Note that a single data key is implicitly requested by the presence of a SVR field. See 13.6 (table 2 : SVR) for a definition of the types of service requested.
CTB	Process the value of CTB field. If the field is CTB/b, the CTB value is "b".
CTR	The value in the CTR field may be used in determining the Cryptographic Service Message to which this ERS responds. If the field is CTR/r, the CTR value is "r".
CTA	(CKT environment only) Process the value of the CTA field (see 12.2). If the field is CTA/a, the CTA value is "a". If a CTA error is detected, an ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have an "A" in the ERF field, ie, ERF/A Further processing of the ERS may continue prior to transmission of the ESM.
MAC	(CKT environment only) Compute a MAC from the message. The MAC shall always be 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/ERSb...bCTA/ab) 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". ie, 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.

Table 15 — Processing of ERS *(concluded)*

Field tag	Action
EDC	(CKD environment only) (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. ie, ERF/O or b) the field shall be disregarded and message processing may proceed. Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF) (see NOTE in 12.1.1): EDC = aKDX(MCL/ERSb...bCTB/bb) 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", ie, 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.

15.4 Process Error Service Message (ESM)

An Error Service Message (ESM) is received in response to a DSM, ERS, KSM, RSI, RSM, RFS or RTR due to the detection of one or more of the error conditions of a CSM, as listed in table 2: ERF (see 13.6.1).

Error Service Messages shall be processed by computing or selecting field contents in accordance with table 16.

Table 16 — 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 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.
IDC	(Used in a centre environment when responding to a KSM or RSM) This field contains the identity of the CKD or CKT. If the field is IDC/CCCC, the centre is CCCC.

Table 16 — Processing of ESM *(continued)*

Field tag	Action
IDU	(Used in a CKD environment when responding to an ERS, RSI (to a CKD) or RTR, and in a CKT environment when responding to an ERS, RFS or RTR) This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. If the identity of the ultimate recipient is unknown, an ESM with a "U" in the ERF field, ie. ERF/U may be generated or manual recovery may be used. Further processing of the message may be performed prior to transmission of the error message.
CTA	(Used in a CKD environment when responding to an RTR and in a CKT environment when responding to an ERS or RFS) Process the value of CTA field. If the field is CTA/a, the CTA value is "a".
CTB	(CKD or CKT environment; only when responding to a KSM) Process the value of CTB field. If the field is CTB/b, the CTB value is "b". This value (ie, "b") shall be sent as the contents of the CTB field in the ERS that is generated in response to this ESM.
CTP	(Used only in a Point-to-Point environment when responding to a KSM) Process the value of CTP field. If the field is CTP/p, the CTP value is "p".
CTR	Process the value of the 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 type(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 13.6.1 (table 2). Multiple error conditions are indicated by a string of concatenated error flags. Eg, ERF/KPM
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, ie, ERF/O or b) the field shall be disregarded and message processing may proceed. Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF) (see NOTE in 12.1.1): EDC = aKDX(MCL/ESMb...bERF/KPMb) If the computed EDC does not equal the received EDC, there is an error (possibly a transmission error) and either:

Table 16 — Processing of ESM *(concluded)*

Field tag	Action
EDC *(continued)*	a) an ESM shall be generated and returned to the originator. The ERF field shall include an "X", ie, ERF/X or b) the error shall be resolved by other means (eg, 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 EDC field.

15.5 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 (Point-to-Point environment only).

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 17.

Table 17 — 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 KSM for processing, then, the 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.
IDC	(CKD or CKT environment only) Process the value of the IDC field.If the field is IDC/CCCC, the value is CCCC. This is the identity of the CKD or CKT, and is used to select the keys and other data used to process the message. If this identity is not known, an ESM shall be sent with a "D" (CKD environment) or a "T" (CKT environment) in the ERF field. Further processing of the message shall cease.
NOS	(only used in a Point-to-Point environment) (When present) If this field is present, the (*)KK (or KDs if no (*)KK is sent in the message) has (have) been notarised.

Table 17 — Processing of KSM *(continued)*

Field tag	Action
(*)KK	(only used in a Point-to-Point environment) (When present) Using the rules of 13.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If an IDK2 subfield is not present, the (*)KK used to decipher the received (*)KK is the only one shared by the message originator and recipient.
Optional subfields	
P	If the "P" subfield 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, ie, ERF/K All further processing of the message shall cease.
IDK1	If a (*)KK is received in the message, and no IDK1 field is present, then, immediately on decipherment of the new (*)KK, that key shall be placed into use for all subsequent processing.
IDK2	The IDK2 subfield (if present) defines the (*)KK, (*)KKY, to be used in deciphering the (*)KK contained in the KSM. Otherwise, the (*)KK to be used for decipherment of the received (*)KK is implicitly defined. See 13.5. If an IDK2 is present and the IDK2 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 "I" in the ERF field, ie, ERF/I If a (*)KK is received, an associated count shall be established for key offsetting the (*)KK when other keys are received which are enciphered using this (*)KK. The count (CTP) is initially set to one, and the value one shall be used to key offset decipher the KD which is received in this message. *Decipherment of (*)KKs* *Case 1:* (*)KK without notarisation. If a new (*)KK is received, then the deciphered (*)KK is computed as follows: Let (*)KKZ be the key to be deciphered and (*)KKY be the deciphering key. Use the procedure of 12.3 and the value of CTP to compute the (*)KKoY: Decipher (*)KKZ deciphered KKZ = dKKoY(enciphered KKZ) or deciphered KKZ = ded*KKoY(enciphered KKZ) or deciphered *KKZ = ded*KKoY(enciphered *KKZ) *Case 2:* (*)KK with notarisation. If the NOS field is present, then the (*)KK was notarised before transmission. In this case, the (*)KK subfield contents are processed as follows: a) Compute (*)KN using the contents of the ORG, RCV and CTP fields and the process defined in 12.4. b) Decipher the contents of the (*)KK field to form (*)KKZ using the equation: deciphered KKZ = dKN(notarised KKZ) or

Table 17 — Processing of KSM *(continued)*

(*)KK *(continued)* IDK2 *(continued)*	deciphered KKZ = ded*KN(notarised KKZ) or deciphered *KKZ = ded*KN(notarised *KKZ)
(*)KKU	(CKT environment only) (When present) Using the rules of 13.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields, if present. If the IDK2 subfield is not present, the *KK to be used to decipher the key in the (*)KKU subfield is the only one shared by the message recipient and the CKT.
Optional subfields	
P	If the "P" subfield 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,ie, ERF/K Further processing of the message shall cease.
IDK1	If a (*)KK is received as a (*)KKU in the message, and no IDK1 field is present, then, immediately on decipherment of the new (*)KK, that key shall be placed into use for all subsequent processing.
IDK2	The IDK2 subfield (if present) defines the *KK (*)KKY to be used in deciphering the (*)KKU. Otherwise, the *KK to be used is implicitly defined and is a *KK shared with the CKT. See 13.5. If an IDK2 is present and the IDK2 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 "I" in the ERF field, ie, ERF/I If a (*)KKU is received, an associated count shall be established for key offsetting the (*)KKU when other keys are received which are enciphered using this (*)KKU. The count (CTP) is initially set to one, and the value one shall be used to key offset decipher the KD which is received in this message. *Decipherment of the (*)KKU* Let *KKY be the *KK shared between Party B and the centre. Let (*)KKZ be the (*)KK that has been received by Party B, the ultimate recipient. The (*)KKU shall be deciphered as follows: a) Compute *KN using the (*)KKY and the contents of the ORG, RCV and CTB fields and the process defined in 12.4. b) Decipher the (*)KKU to obtain the (*)KKZ using the equation: deciphered KKZ = ded*KN(notarised KKU) or deciphered *KKZ = ded*KN(notarised *KKU)
KD	(not used in a CKD environment) At least one and at most two KDs shall be received in a KSM as KD field(s). If a new (*)KK is sent in the KSM, then the KDs shall be deciphered using that key.

Table 17 — Processing of KSM *(continued)*

Field tag	Action
KD *(continued)*	If the fields do not have subfields, proceed to case 1. Parse the field(s) to obtain the KDs; and the "P", IDK1 and IDK2 subfields if present.
Optional subfields	
P	If the "P" subfield 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, ie, ERF/K Further processing of the Cryptographic Service Message shall cease.
IDK1	If present, this subfield names the KD. See 13.5.
IDK2	The (*)KK used to decipher the KD is identified by the IDK2 subfield if present. See 13.5. If an IDK2 is present and the IDK2 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 "I" in the ERF field, ie, ERF/I If an IDK2 subfield is not present, the (*)KK used to decipher the received (*)KK is the only one shared by the message originator and recipient. *Decipherment of KDs:* *Case 1:* The KDs are enciphered by a (*)KK and are not given notarisation protection (ie, there is no NOS field in the KSM or a (*)KK *or* (*)KKU is received). Let KDI be the key received in the KD field. Let (*)KKY be the Key Enciphering Key to be used. Then the KDs are deciphered as follows: a) Use the procedure of 12.3 and the value of CTX to compute the (*)KKoY: where CTX is CTP in a Point-to-Point environment if no (*)KK is sent in the message and one (1) if a new (*)KK is sent. In a CKT environment CTX shall equal one (1). b) Decipher the KDI: deciphered KDI = dKKoY(enciphered KDI) or deciphered KDI = ded*KKoY(enciphered KDI) *Case 2:* (Only in a Point-to-Point environment) There is no (*)KK field in the Cryptographic Service Message and the KDs were notarised. In this case, the NOS field was included in the KSM and the KD contents of the KD fields shall be deciphered as follows: a) Compute (*)KN using (*)KKY and the contents of the ORG, RCV and CTP fields and the process defined in 12.4. b) Decipher the KDI using the equations: deciphered KDI = dKN(notarised KDI) deciphered KDI = ded*KN(notarised KDI)

Table 17 — Processing of KSM *(continued)*

Field tag	Action
KDU	(CKD or CKT environment only) If and only if no (*)KK is received as a (*)KKU, then at least one and at most two notarised KDs shall be received as KDU fields. Using the rules of 13.5, parse the fields to obtain the KDs; and the "P", IDK1 and IDK2 subfields if present. Parse the fields to obtain the KDUs; and the "P", IDK1 and IDK2 subfields if present.
Optional subfields	
P	If the "P" subfield 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, ie, ERF/K Further Cryptographic Service Message processing shall cease.
IDK1	If present, this subfield names the KD. See 13.5.
IDK2	The *KK used to decipher the KD is identified by the IDK2 subfield (if present). See 13.5. If the IDK2 subfield is not present, the *KK to be used for decipherment of the KDU is the only *KK shared between the recipient and the centre. If an IDK2 is present and the IDK2 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 "I" in the ERF field, ie, ERF/I *Decipherment of KDUs:* KDUs shall be deciphered using the processes that follow. a) Compute *KN using the *KK shared with the centre, the contents of the ORG, RCV and CTB fields and the process defined in 12.4. b) Decipher the KDU using the equation: deciphered KD = ded*KN(notarised KD)
IV	(When present) If the first character is "E", then the IV that follows is enciphered. If the first character is a "P", the IV does not require decipherment before use. If an IV is received in enciphered form, the IV shall be deciphered using the equation: deciphered IV = dKD(IV) where the KD is received in the Cryptographic Service Message. If only one KD is received in the message, that KD shall be used to decipher the IV. If two KDs are received, the second shall be used to decipher the IV. If the first letter is other than "E" or "P"; or the remaining characters are not a member 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, ie, ERF/F Further processing of the Cryptographic Service Message may continue prior to ESM transmission.

Table 17 — Processing of KSM *(concluded)*

Field tag	Action
EDK	(When present) The EDK, if received, is the date and time on which the KDs received in the message shall be placed in use.
CTB	(Only used in a CKD or CKT environment) Process the value of the CTB field (see 12.2). If the field is CTB/b, the CTB value is "b". If a CTB error is detected, an ESM shall be generated to notify the originating party of the CTB error condition. The ESM shall have a "B" in the ERF field, ie, ERF/B Further processing of the KSM may continue.
CTP	(Only used in a Point-to-Point environment) Process the value of the CTP field (see 12.2). 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, ie, ERF/P Further processing of the KSM may continue.
MAC	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...bCTX/xb) In a Point-to-Point environment where CTX is CTP and x is p; and in a CKD or CKT environment CTX is CTB and x is b. 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",ie, 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.

15.6 Process Request For Service message (RFS)

Request For Service messages (RFS) are only received by a CKT. Responses to the RFS are either an RTR if there are no errors detected in the RFS or an ESM if errors are detected in the RFS. An RFS shall be processed by computing or selecting field contents in accordance with table 18.

Table 18 — Processing of RFS

Field tag	Action
MCL	Confirm field is MCL/RFS.
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 RFS for processing, then, the 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.
IDU	This field contains the identity of the ultimate recipient. If the field is IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used in selecting the (*)KK to be used in generating the RTR that responds to this Cryptographic Service Message. If the identity of the ultimate recipient is not known to the CKT, an ESM shall be generated and sent to the originator with a "U" in the ERF field, ie, ERF/U Further processing of the Cryptographic Service Message may be performed.
(*)KK	(When present) Using the rules of 13.5, parse the field to obtain the (*)KK; and the P, IDK1 and IDK2 subfields if present. If an IDK2 subfield is not present, the *KK used to decipher the received (*)KK is the only one shared by the message originator and recipient (centre).
Optional subfields	
P	If the "P" subfield 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 RFS, with a "K" in the ERF field,ie, ERF/K Further processing of the Cryptographic Service Message shall cease. If the key parity is correct, the "P" shall be inserted in the "P" subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message using the rules of 13.5.
IDK1	If present, the IDK1 shall be inserted in the IDK1 subfield of the (*)KKU field in the RTR generated in response to this Cryptographic Service Message, using the rules of 13.5. Otherwise, no IDK1 subfield shall be sent in the (*)KKU field in the RTR.
IDK2	If an IDK2 is present and the IDK2 is not known to the CKT, this shall cause the processing of the RFS to cease and the generation and transmission to the originating party of an ESM with an "I" in the ERF field, ie, ERF/I

Table 18 — Processing of RFS *(continued)*

Field tag	Action
(*)KK *(continued)* IDK2 *(continued)*	*Decipherment of (*)KKs* If a new (*)KK is received, then the deciphered (*)KK is computed as follows: Let (*)KKZ be the key to be deciphered and *KKY be the Key Enciphering Key shared between the CKT and the originator of the RFS. Use the procedure of 12.3 and CTA to compute the *KKoY. Decipher (*)KKZ deciphered KKZ = ded*KKoY(enciphered KKZ) or deciphered *KKZ = ded*KKoY(enciphered *KKZ)
KD	At least one and at most two KDs shall be received in an RFS as KD fields. If a new (*)KK is sent in the RFS, then only one KD shall be received in the RFS and that KD shall be deciphered using the received (*)KK. If the KD fields do not have subfields, proceed to case 1. Parse the field(s) to obtain the KD; and the "P", IDK1 and IDK2 subfields if present. If the IDK2 subfield is not present, the *KK used to decipher the KD is the only *KK shared by the message originator and the CKT.
Optional subfields	
P	If the "P" subfield 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 RFS, with a "K" in the ERF field,ie, ERF/K Further processing of the Cryptographic Service Message shall cease. If the key parity is correct, the "P" shall be inserted in the "P" subfield of the RTR generated in response to this Cryptographic Service Message using the rules of 13.5.
IDK1	If present, this subfield names the KD. The IDK1 shall be inserted in the IDK1 subfield of the KDU field in the RTR generated in response to this Cryptographic Service Message, using the rules of 13.5.
IDK2	The *KK used to decipher the KD is identified by the IDK2 subfield if present. See 13.5. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the RFS to cease and the generation and transmission to the originating party of an ESM with an "I" in the ERF field, ie, ERF/I *Decipherment of the KDs:* *Case 1:* The RFS contains a (*)KK field. One and only one KD shall be received in the RFS in the KD field. That KD shall be used to authenticate the RFS and to generate the MAC for inclusion in the RTR that responds to this RFS. Let KDI be the key received in this field. Let (*)KKY be the Key Enciphering Key sent in the Cryptographic Service Message. The KD is deciphered as follows: a) An offset of zero is used to compute the (*)KKoY.

Table 18 — Processing of RFS *(concluded)*

Field tag	Action
KD *(continued)* IDK2 *(continued)*	b) Decipher the KDI using the equation: deciphered KDI = dKKoY(KDI) or deciphered KDI = ded*KKoY(KDI) *Case 2:* There is no (*)KK field in the Cryptographic Service Message. At least one and at most two KDs shall be received in an RFS as KD field(s). The KDs are deciphered by a *KK shared with the message originator, and the following equations are used: a) Use the value of CTA and the procedure of 12.3 to compute the *KKoY b) Decipher the KDI using the equation: deciphered KDI = ded*KKoY(KDI)
CTA	Process the value of the CTA field (see 12.2). If the field is CTA/a, the CTA value is "a". If a CTA error is detected, an ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have a "A" in the ERF field, ie, ERF/A Further processing of the RSM may continue.
MAC	Compute a MAC from the message. The MAC shall always be computed using the KDs received in the Cryptographic Service Message. If one KD is received, that KD, KDJ, 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/RFSb...bCTA/ab) If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M", ie, 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.

Table 19 — Processing of RSI *(concluded)*

Field tag	Action
EDC *(continued)*	a) an ESM shall be generated and returned to the originator with an "O" in the ERF field, ie, ERF/O or b) the field shall be disregarded and message processing shall proceed. Compute an EDC from the message using the data key for EDC computation (KDX = 0123456789ABCDEF) (see NOTE in 12.1.1). Eg, EDC = aKDX(MCL/RSIb...bSVR/*KK.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", ie, 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.

15.8 Process Response Service Message (RSM)

Response Service Messages (RSM) are received as an authenticated acknowledgement of a DSM, a KSM or an unsolicited RTR (CKD environment only) and shall be processed by computing or selecting the field contents in accordance with table 20.

Table 20 — 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 RSM for processing, then, the Cryptographic Service Message has been misrouted and further processing shall cease. 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.
IDC	(Used in a CKD or CKT environment when responding to a KSM). Process the value of the IDC field. If the field is IDC/CCCC, the value is CCCC. This is the identity of the CKD or CKT, and is used to select the keys and other data used to process the Cryptographic Service Message. If two RSMs are received from the same originator in response to two KSMs from that originator that used two different centres (CKD or CKT), the identity of the centre used is needed to identify the KSM to which the RSM responds. If this identity is not known, an ESM shall be sent with a "D" (CKD environment) or a "T" (CKT environment) in the ERF field. Further processing of the Cryptographic Service Message shall cease.

Table 20 — Processing of RSM *(concluded)*

Field tag	Action
IDD	(When present) If no IDD field is present, this RSM is in response to a KSM or an unsolicited RTR (CKD environment only). 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 KK, *KK or 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.
IDU	(In response to an unsolicited RTR; CKD environment only). If the field is IDU/UUUU, the ultimate recipient is UUUU. If the identity of the ultimate recipient is not known to the CKD, an ESM shall be generated and sent to the originator with a "U" in the ERF field,ie, ERF/U Further processing of the RSM may continue prior to transmission of the ESM.
MAC	Compute a MAC from the Cryptographic Service Message. The MAC is always computed using the KDs used to generate the MAC in the DSM, KSM or RTR to which the RSM responds. If one KD is received, that KD (KDJ) shall be used. If two KDs were used (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/KKKKb) responding to a DSM aKDJ(MCL/RSMb...bIDC/CCCCb) responding to a KSM in a centre environment aKDJ(MCL/RSMb...bORG/OOOOb) responding to a KSM in a Point-to-Point environment aKDJ(MCL/RSMb...bIDU/UUUUb) responding to an unsolicited RTR in a CKD environment 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. If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M", ie, ERF/M Following receipt of an RSM that responds to a DSM, the KD used to authenticate the message shall be discontinued.

15.9 Process Response To Request message (RTR)

A Response To Request message (RTR) is received from a CKD or a CKT (see 13.6.3 and 13.6.4). A correct RTR shall result in a KSM being generated and sent to the party identified in the IDU field. If an error is detected in the RTR, an ESM shall be generated and returned to the centre that originated the RTR.

Response To Request messages shall be processed by computing or selecting the content in accordance with table 21.

Table 21 — Processing of RTR

Field tag	Action
MCL	Confirm field is MCL/RTR.
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 RTR 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.
IDU	Process the value of the IDU field. If IDU/UUUU, the ultimate recipient is UUUU. The field contents shall be used as the contents of the RCV field in the following KSM. If the identity of the ultimate recipient is not known to the recipient, an ESM shall be generated and sent to the centre with a "U" in the ERF field,ie, ERF/U Further processing of the RTR may continue prior to transmission of the ESM.
(*)KKU	(CKT environment only) (When present) The field contents shall be used as the contents of the (*)KKU field in the KSM that is generated and returned in response to this Cryptographic Service Message.
KD	(CKD environment only) At least one and at most two KDs shall be received in a RTR as KD field(s). The *KK used to decipher the KD is a *KK shared with the CKD. Parse the field(s) to obtain the KD; and the "P", IDK1 and IDK2 subfields if present.
Optional subfields	
P	If the "P" subfield 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 parity, further processing of the Cryptographic Service Message shall cease, and an ESM shall be generated in response to the RTR, with a "K" in the ERF field, ie, ERF/K
IDK1	If present, this subfield names the KD. See 13.5.
IDK2	The *KK used to decipher the KD is identified by the IDK2 subfield if present. See 13.5. If the IDK2 subfield is not present, the *KK used for decipherment is the only *KK shared by the message recipient and the CKD. If an IDK2 is present and the IDK2 is not known to the recipient, this shall cause the processing of the RTR to cease and the generation and transmission to the originating party of an ESM with an "I" in the ERF field, ie,

Table 21 — Processing of RTR *(continued)*

Field tag	Action
KD *(continued)* IDK2 *(continued)*	ERF/I Decipherment of KDs: Let KDI be the key received in the KD field. a) Compute *KN using the contents of the RCV, IDU and CTA fields and the process defined in 12.4. b) Decipher the KDI: deciphered KDI = ded*KN(notarised KDI)
KDU	The field contents shall be used as the contents of the KDU field in the KSM that is generated in response to this Cryptographic Service Message.
IV	(CKD environment only) (When present) The IV is generated by the CKD. On receipt of the RTR, the IV is stored for retransmission to the ultimate recipient and deciphered for use by the recipient using the process given below. Process the contents of the field. If the first character is "E", then the IV that follows is enciphered. If the first character is a "P", the IV does not require decipherment before use. If the first letter is other than an "E" or "P"; or the remaining characters are not a member of the set (0-9), (A-F), then an error condition exists. An ESM shall be generated and sent to the originator of the Cryptographic Service Message with an "F" in the ERF field,ie, ERF/F Further processing of the Cryptographic Service Message may continue prior to transmission of the ESM. *Case 1:* Enciphered IV. If an IV is received in enciphered form, the IV shall be deciphered as follows: Let KDH be the KD received in the Cryptographic Service Message. If two KDs are received, the second KD shall be used as the KDH. deciphered IV = dKDH(enciphered IV) *Case 2:* Plaintext IV. If an IV is received in plaintext form, then it does not require further processing.
EDK	(CKD environment only) (When present) This field contains the date and time that the KDs become effective. The field contents shall be used by the recipient and as the contents of the EDK field in the KSM that is generated in response to this Cryptographic Service Message.
CTB	If the field is CTB/b, "b" shall be processed and then sent as the contents of the CTB field in the KSM that is generated in response to this RTR (see 12.2).
CTA	(CKD environment only) Process the value of CTA (see 12.2). If the field is CTA/a, the CTA value is "a".

Table 21 — Processing of RTR *(concluded)*

Field tag	Action
CTA *(continued)*	If a CTA error is detected, an ESM shall be generated to notify the originating party of the CTA error condition. The ESM shall have a "A" in the ERF field, ie, ERF/A Further processing of the RTR may continue.
MAC	Compute a MAC from the Cryptographic Service Message. In a CKT environment, when a (*)KK is returned in the Cryptographic Service Message, the KD used to authenticate the Cryptographic Service Message shall be the KD sent in the ERS or RFS to which this message responds. Otherwise, the MAC is always computed using the KDs received in the message. If 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/RTRb...bCTA/ab) for a CKD environment and aKDJ(MCL/RTRb...bCTB/bb) for a CKT environment If the computed MAC does not equal the received MAC, the Cryptographic Service Message fails to authenticate. An ESM shall be generated and returned to the originator. The ERF field shall include an "M", ie, 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.

Annex A

An example of manual key distribution and control procedures

(This annex does not form a part of this standard)

A.1 General

This annex describes one example of management and control of keys that are manually distributed, stored, used or destroyed.

A.2 Appointment of authorised personnel

A.2.1 Personnel (hereinafter referred to as "custodians") responsible for key management should be designated by the management of all the sending and receiving institutions.

A.2.2 No one person should ever have overall control of keying material and physical keys for encipherment devices.

A.2.3 At least two custodians should be available for each operational shift. Personnel assigned these responsibilities should be present to carry out their duties within a prescribed elapsed time after notification of a non-planned need to change keys.

A.2.4 Any alternate custodian may perform the duties of a custodian in the absence of a regularly assigned custodian. To provide proper coverage for multiple-shift operations, management should designate personnel at subsidiary levels, such as "Key Inserters", "Key Controllers".

A.3 Responsibilities of custodians

A.3.1 Responsibility for the receipt, verification of contents, and storage of keying material, should be under the dual control of two custodians, with split knowledge.

A.3.2 Custodians should:

1) receive and store keying material;

2) maintain a record of personnel authorised to utilise keying material;

3) verify, under dual control and split knowledge, that the received new keying material is intact;

4) destroy, or witness the destruction of the old keying material;

5) prepare a record of keying material used; and

6) enter and change keying material.

A.3.3 At least two custodians should control the physical key(s) which allow(s) entry to the encipherment or authentication device.

A.4 Distribution of keying material

A.4.1 All necessary accompanying letters and receipts should be prepared in advance of generating the keys which need to be held under dual control at all times.

A.4.2 Whenever a key is issued, identification details of the recipient, the issue date and the signature of the issuer should be recorded.

A.4.3 All components of a key should not be despatched on the same day. Every key component should be accompanied by its receipt.

A.4.4 If magnetic storage media or specially designed devices containing electronically protected memory (eg. key loaders) are used, then any passwords for the storage media should be despatched independently of the key components. If such devices are distributed by hand, then they should be transported and loaded under dual control.

A.4.5 Keying material should be packaged in envelopes marked to indicate their content, which are then sealed in such a way that any subsequent interferences may be detected. These envelopes should be placed in outer envelopes addressed to the recipient. A courier may be used.

A.4.6 An acknowledgement from the recipient bearing authorised signatures should be received before the key is used.

A.5 Receipt of keying material

A.5.1 Upon receipt, the custodians should closely examine the inner envelope to ensure that no access has taken place.

A.5.2 The custodian should verify under dual control the contents and the authenticity of the signatures of the originator's custodians on the accompanying documentation.

A.5.3 The accompanying receipt form should be signed by each receiving custodian. One copy should be returned to the originator and one retained on file.

A.5.4 The originator should be notified immediately if there is a question regarding the integrity of the keying material. In the event of notification, the originator in consultation with the recipient should decide on the action to be taken.

Transfer of keying material should be accomplished under dual control and at a mutually acceptable time. Transfer of control should only be to properly designated individuals and should be documented.

A.6 Storage of keying material and encipherment/ authentication device physical keys

A.6.1 All keying material under control of the custodian should be stored in a secure receptacle, such as a safe located in a secure area.

A.6.2 Access to the receptacle should require at least two custodians, with separate keys or combinations.

A.6.3 A log should be maintained by the custodians recording the deposits and withdrawals for each access to the receptacle.

A.6.4 Back-up physical keys (or combination codes) for the locked receptacle should be kept in a secure location under control. Access to such back-up keys or combination should require the presence of one custodian and a third party.

A.6.5 Physical keys for the encipherment or authentication device should be controlled by custodians. Where the device has two locks, a custodian should have access to only one key.

A.6.6 Physical keys for the encipherment or authentication device should be kept in a secure location.

A.6.7 A short term power failure should not result in the loss of a key.

A.7 Use of keying material

A.7.1 Whenever it is necessary to change a key, the materials should be removed from the locked receptacle by the custodians. The next key or key component (eg, page) should be removed by the custodian and a record of the removal recorded.

A.7.2 A custodian should not be aware of the content of the key component which is not the custodian's responsibility.

A.7.3 Each of the current key components should be sealed in a separate envelope by the respective custodian and placed back into the locked receptacle after use.

A.7.4 A record should be kept of the return of the key component information to storage.

A.8 Destruction of keying material

A.8.1 Keying material should not be destroyed by the custodian until instructed to do so by an authorised person.

A.8.2 Keying material needs to be destroyed by either crosscut shredding or any other method which destroys it beyond reconstruction.

A.8.3 The destruction of the keying material should be under dual control, and without disclosure.

A.8.4 A record of destruction should be maintained. This record should contain sufficient information to identify the material destroyed (eg, the keying material book number, keying material page number), signature of person destroying the page (custodian), date/time, and signature of a witness.

A.8.5 The record of destruction should be stored in the locked receptacle used to contain the keying material.

A.9 Archival storage of keys

Where it is necessary that keying material be stored for archival purposes, that keying material should be handled in accordance with the requirements of A.5 and A.6 if the confidentiality of the data or secrecy of the key or both are required. An alternative is to protect the keys by encipherment in a key used only for archival storage. In this case, only the storage key needs to be protected.

Where keying material needs to be stored for archival purposes and that material has no further security value (security life), it should be stored using the usual procedures in a place for the storage and protection of vital records.

Annex B

Notation

(This annex forms a part of this standard)

This annex describes the notation used in clauses 12 to 15 of this International Standard.

B.1 *Operators* are represented by the following lower case letters:

a for authentication
e for encipherment
d for decipherment

"( )" indicates a choice of characters

"[ ]" denotes representation of field contents

B.2 *Concatenation* is indicated by "||"

B.3 *Modulo-2* is indicated by "+"

B.4 *Fields* are separated by "b" space

B.5 *Subfields* within a field are separated by "." full stop

B.6 Field type and usage is defined by a character sequence. The sequence for keys begins with "K"; that for an IV begins with a "IV"; identifications begin with "ID". For keys, the second letter is defined by the table:

D for data key
K for Key Enciphering Key
N for notarising key

B.7 *For keys,* the third letter is defined by the table:

o the key has been key offset
l the key is the left key of a key pair
r the key is the right key of a key pair
U the key has been notarised for the ultimate recipient
X the data key is used in the computation of the Error Detection Code

otherwise, the third letter is unassigned and may be used for any purpose.

B.8 *The types of key* are:

KD data key, plaintext or enciphered
KDU data key, notarised for the ultimate recipient
KDX data key, plaintext, for computation of the Error Detection Code, hexadecimal 0123456789ABCDEF (see NOTE in 12.1.1)
KK Key Enciphering Key, plaintext or enciphered
KKU Key Enciphering Key, notarised for the ultimate recipient
KN notarising key

B.9 *IVs* are denoted by:

IV initialisation vector, plaintext or enciphered

B.10 *Designation of specific keys and key pairs*

Keys may be single keys or may be used as key pairs. A single key is expressed as:

K = key

whereas:

*KK = KKl || KKr

is a key pair consisting of two quantities, part "l" (left) and part "r" (right). An asterisk "*" preceding the character sequence is used to specify that the key is a key pair.

(*)KK designates the use of either a KK or *KK.

Annex C

Pseudo-random key and IV generator

(This annex does not form a part of this standard)

C.1 Purpose

The purpose of the annex is to present an example, using the algorithm DEA, of a pseudo-random key and IV generator.

C.2 Algorithm

Let ede*X(Y) represent a DEA multiple encipherment of Y under key *X. Let *K be a DEA key pair reserved only for the generation of other keys, let V be a 64-bit seed value which is also kept secret, and let + be the modulo-2 operator. Let DT be a date/time vector which is updated on each key generation. I is an intermediate value. A 64-bit vector R is generated as follows:

$$I = ede*K(DT)$$

$$R = ede*K(I + V)$$

and a new V is generated by $V = ede*K(R + I)$

To obtain a DEA key, every eighth bit is reset to odd parity. To obtain an initialisation vector (IV), R may be used directly.

Annex E

Dual Key Translation Centre application

(This annex does not form a part of this standard)

E.1 General

When a party subscribing to one Key Translation Centre (CKT1) wishes to establish a keying relationship with another party which does not subscribe to CKT1 but does subscribe to a second Key Translation Centre (CKT2), the concept of using a Key Translation Centre to establish a Key Enciphering Key relationship between parties can be extended. The dual CKT application allows the initiating party to obtain a keying relationship with any subscriber to the second centre.

Note that this augments the existing automated key management architecture by adding a fourth layer; an automatically distributed Key Enciphering Key is used to establish the Key Enciphering Key relationship with the second centre. This can be accomplished by an iterative use of the protocol in the CKT environment.

E.2 Process

Two Key Translation Centres are used to establish Key Enciphering Keys (or key pairs) and data keys between Parties A and B which do not share a Key Enciphering Key (or key pair) and do not share a Key Enciphering Key Pair with the same Key Translation Centre. There are three manually distributed Key Enciphering Key Pairs, one shared by the two Key Translation Centres, CKT1 and CKT2; one shared by Party A and CKT1 and the third shared by Party B and CKT2.

In order for Party A to establish a keying relationship with Party B, Party A sends a request for translation of a Key Enciphering Key Pair (*KKA2) to his Key Translation Centre (CKT1) with CKT2 identified as the ultimate recipient. CKT1 then returns a message to Party A containing *KKA2 enciphered under the manually distributed Key Enciphering Key Pair shared by the two centres (CKT1 and CKT2).

Party A sends this enciphered key to CKT2 and thus establishes a Key Enciphering Key Pair relationship with CKT2 (the key is *KKA2).

Party A then sends a request for translation of a Key Enciphering Key, (*)KKAB, and a data key, KD, or a request for translation of a data key only, to CKT2 with Party B as the ultimate recipient. (*)KKAB is enciphered under *KKA2, and the KD is enciphered under (*)KKAB.

E.3 Message flow in a dual Key Translation Centre environment

Figures 11 and 12 show the flow of Cryptographic Service Messages necessary to establish communications between two parties that have Key Enciphering Key relationships with different Key Translation Centres. The two CKTs must have a Key Enciphering Key Pair relationship with each other.

Initially, one of the parties may establish a Key Enciphering Key Pair relationship with the other party's CKT. Once this is accomplished, keys may be defined for communication between the two parties according to the steps outlined in 13.6.4

Cryptographic Service Message flow for establishing the Key Enciphering Key relationship with the other party's centre may be as follows:

1) Let Party B be a party that wishes to communicate with Party A, does not currently share a key with Party A, may not have a key generation capability, and does not share a Key Enciphering Key Pair with the same CKT as Party A.

A RSI message may be sent from Party B to Party A requesting the type of key(s) and (optionally) an IV to be provided. The RSI may also identify the CKT to be used.

If the RSI received by Party A from Party B contains error(s), then Party A may return an ESM to Party B, and Party B may try sending another RSI to Party A. If Party A has no key generation capability and cannot otherwise acquire keys, then Party A may return an ESM to Party B and communications cannot be established.

2) Let Party A be a party that desires to send keys to Party B with whom there is no commonly shared (*)KK (perhaps in response to an RSI message from Party B).

Party A sends an RFS message to CKT1 containing the *KK to be sent to CKT2 (the ultimate recipient). The *KK is enciphered under a *KK shared between Party A and CKT1. A KD is also sent, enciphered under the *KK sent in the message, and is used to authenticate the Cryptographic Service Message. If an error(s) is detected in the RFS message by CKT1, then CKT1 returns an ESM to Party A.

3) The RFS message is received by CKT1. The *KK received in the message may be deciphered using the *KK shared by the centre and Party A, and notarised using the identities of Party A and CKT2 and the *KK shared between CKT1 and CKT2 and the count associated with that *KK. The result is inserted in the *KKU field of an RTR message returned to Party A. The KD contained in the received RFS message is used to authenticate both the RFS and the subsequent RTR messages. If an error(s) is detected in the RTR message received by Party A, and ESM is returned to the Key Translation Centre.

4) Party A sends a KSM to CKT2 containing the *KKU field received in the RTR message. Party A also sends a KD enciphered under the *KK sent in the KSM in order to allow the centre to authenticate the message. If CKT2 receives the KSM from Party A without error, then an RSM is returned to Party A, otherwise, an ESM is returned to Party A.

Party A may resend a KSM to Party B an arbitrary number of times, but Party A does not send a new KSM (i.e., utilising new keys or a new count for the (*)KK specified in a KSM) until the old KSM is acknowledged by an RSM or an ESM.

5) At this point, Party A has established a *KK relationship with CKT2. Further establishment of keys between Party A and Party B is accomplished using the procedure of 13.6.4.

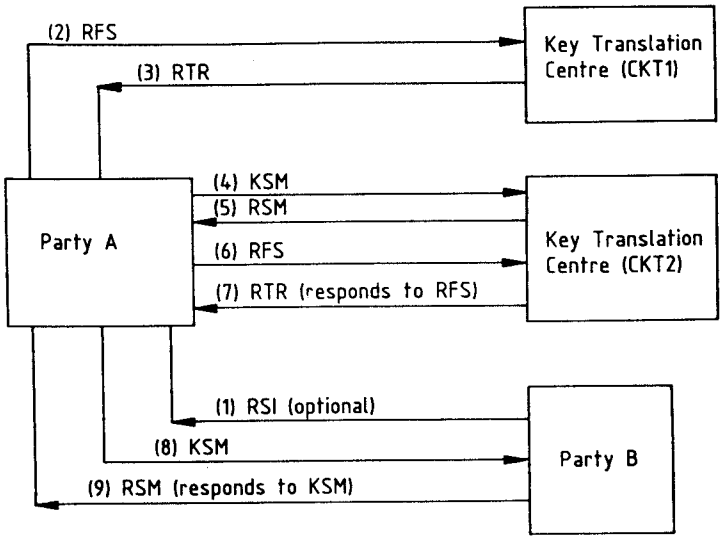


Figure 11 — Dual Key Translation Centre application (normal message flow)

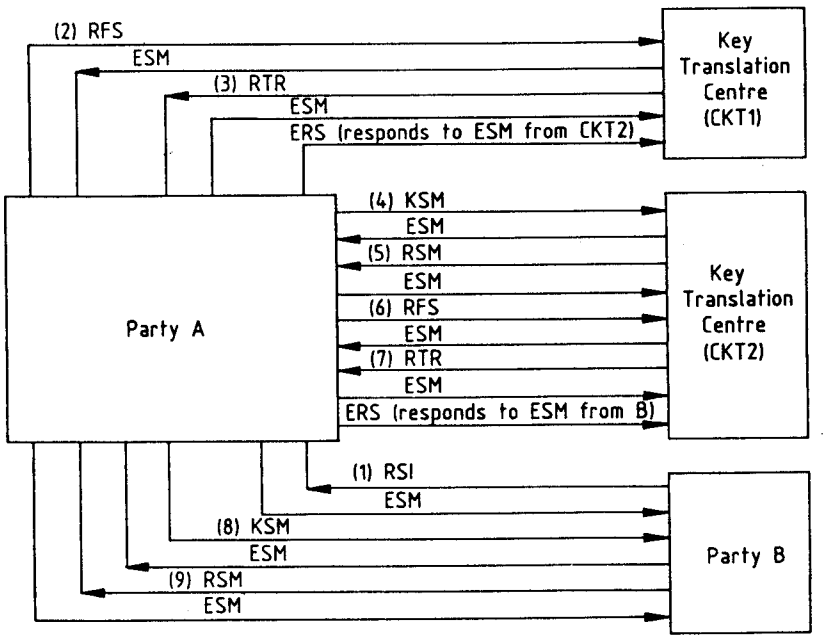


Figure 12 — Dual Key Translation Centre application (message flow with errors)

Annex F

Keying material. Guidance on clearing and destruction procedures.

(This annex does not form a part of this standard)

F.1 Purpose

To establish guidance on uniform erasing (zeroising) (eg degaussing, erasing and overwriting), clearing and destruction procedures for storage material used so that unauthorised access to or compromise of the data is prevented.

F.2 General

Due to the physical properties and retentive capabilities of storage media and devices (eg magnetic cores, drums, disks and various microelectronic circuits) used to store, record or manipulate sensitive data, special precautions should be taken to safeguard against the compromise of possible residual information. This annex presents recommended procedures for such zeroisation prior to reuse, or for destruction.

F.3 Cathode Ray Tube (CRT)

A display CRT can be considered zeroised if, after visual inspection, it is determined that no sensitive information has been etched into the CRT phosphor coating.

If there is any doubt after inspection of the screen, the CRT surface should be highlighted by filling the screen with vectors to create a raster effect to light up the entire screen. The brightness of the raster can be varied with the intensity control. Any burns or uneven illumination of the phosphor coating that could be considered compromising should then be easily detected. Random burns on the CRT should not necessitate automatic classification of the CRT as containing sensitive information.

Should any area of the CRT be determined to contain sensitive information, the CRT should remain classified at the highest level of residual information.

If the CRT becomes defective and cannot be purged of sensitive information, it should be destroyed.

F.4 Magnetic core memory

To zeroise a magnetic core memory, overwrite all data bit locations. All data bit locations should be set to zeros and verified for successful entry; then all locations should be set to ones and the verification repeated. This overwrite procedure (with random hard copy readout or other equivalent verification at conclusion) should be executed alternately with zeros and ones for 1000 cycles. Finally, non-sensitive, arbitrary data should be written in all data bit locations and left in the core.

Alternative procedures:

a) The magnetic core memory should be destroyed by pulverising, melting, incinerating.

b) Expose all cores to a recomended magnet. The magnet should be held within 1 cm of each core.

F.5 Disk pack

To zeroise magnetic disk packs, overwrite all data bit locations three times by setting zeros and ones alternately. Verify successful entry of the overwrites through a random hard copy readout or equivalent verification. Write non-sensitive arbitrary data on all data locations on all tracks of the disk and leave it there.

If the disk has failed in such a way that it cannot be overwritten or the overwrite cannot be verified, clear the disk by exposing the recording surface to a permanent magnet assembly. Cover the magnet assembly with a lintless wiping tissue to prevent damage to recording surfaces.

Wipe the entire surface *at least* three times with the magnet.

Alternative procedures:

a) Apply an emery wheel or sander to the recording surface of an inoperative disk. Ensure that the entire surface is completely removed prior to disposal.

b) The resin binder and ferric oxide surface can be completely removed/stripped (chemically destroyed) from the disk with concentrated hydrochloric acid (55-58%).

c) Meltdown,

F.6 Drum

To zeroise a magnetic drum, overwrite all data bit locations three times by setting zeros and ones alternately. Verify successful entry of the overwrites through a random hard copy readout or other equivalent verification.

Write non-sensitive arbitrary data on all data locations, verify that the data have been written to these locations, and leave it there.

If the drum has failed in such a way that it cannot be overwritten or the overwrite cannot be verified, zeroise the drum by exposing the recording surface to a permanent magnet assembly. Cover the magnet with a lintless wiping tissue to prevent damage to the recording surface. Wipe the entire surface at least three times with the magnet. Ensure that all recording areas of the drum are exposed to the active area of the magnet assembly.

F.7 Magnetic tapes

Magnetic tapes shcuid be zeroised with a degausser. Magnetic

tapes may be cleared by overwriting one time with any one character. However, cleared magnetic tapes should be safeguarded, controlled, and marked at the level commensurate with the most sensitive information recorded on them. Before release of a zeroised magnetic tape for destruction, it should be subjected to two degaussing cycles and removed from the reel, then destroyed by disintegration into pieces 9mm or smaller, or incineration.

F.8 Internal memory, buffers and registers

Internal memory, buffers, or registers should be zeroised initially by use of a hardware clear switch or power-on/off reset cycle; secondarily by overwriting all data bit locations with continuously changing random data for 1000 cycles. Periodic verification should be made that the method(s) are working correctly. Verify successful entry of the overwrites through a random hard copy readout or through other equivalent verification. Finally, all locations should be overwritten with non-sensitive, random data and verified.

F.9 Semiconductor memory

1) Random Access Memory (RAM) should be initialised by use of a power-on/off reset cycle. Overwrite the storage area by alternately setting each data bit location to all zeros then all ones for 1000 cycles. Periodic verification should be made that the method is working correctly. Verification may take the form of random sampling or use of a read and compare program. Finally, all locations should be overwritten with non-sensitive data and verified.

2) Erasable Programmable Read Only Memory (EPROM) should be initialised by optical ultraviolet erasing the entire array. Zeroisation should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

3) Electrically Alterable Read Only Memory (EAROM) should be initialised by pulsing all gates. Zeroisation should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

4) Electrically Erasable Programmable Read Only Memory (EEPROM) should be initialised by pulsing the erase control gate. Zeroisation should be verified. All storage locations should be overwritten with non-sensitive random data and verified.

5) Read Only Memory (ROM) is physically programmed during manufacture. Physical destruction is the only recommended method to ensure erasure.

F.10 Paper materials

Paper materials should be destroyed by burning, pulverising, or crosscut shredding. When material is pulverised, all residue should be reduced to pieces 5mm or smaller. When material is burned, the residue should be reduced to white ash.

F.11 Platens and ribbons

The printer platen and ribbon should be removed from a printer before the printer is released. Platens (only the rubber surface should be physically removed for destruction) and ribbons should be destroyed (eg by incineration).