INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.815

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/95)

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS SECURITY

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SECURITY FRAMEWORKS FOR OPEN SYSTEMS: INTEGRITY FRAMEWORKS

ITU-T Recommendation X.815

(Previously "CCITT Recommendation")

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the Members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation X.815 was approved on 21st of November 1995. The identical text is also published as ISO/IEC International Standard 10181-6.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

(February 1994)

ORGANIZATION OF X-SERIES RECOMMENDATIONS

Subject area	Recommendation Series
PUBLIC DATA NETWORKS	
Services and Facilities	X.1-X.19
Interfaces	X.20-X.49
Transmission, Signalling and Switching	X.50-X.89
Network Aspects	X.90-X.149
Maintenance	X.150-X.179
Administrative Arrangements	X.180-X.199
OPEN SYSTEMS INTERCONNECTION	
Model and Notation	X.200-X.209
Service Definitions	X.210-X.219
Connection-mode Protocol Specifications	X.220-X.229
Connectionless-mode Protocol Specifications	X.230-X.239
PICS Proformas	X.240-X.259
Protocol Identification	X.260-X.269
Security Protocols	X.270-X.279
Layer Managed Objects	X.280-X.289
Conformance Testing	X.290-X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300-X.349
Mobile Data Transmission Systems	X.350-X.369
Management	X.370-X.399
MESSAGE HANDLING SYSTEMS	X.400-X.499
DIRECTORY	X.500-X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600-X.649
Naming, Addressing and Registration	X.650-X.679
Abstract Syntax Notation One (ASN.1)	X.680-X.699
OSI MANAGEMENT	X.700-X.799
SECURITY	X.800-X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850-X.859
Transaction Processing	X.860-X.879
Remote Operations	X.880-X.899
OPEN DISTRIBUTED PROCESSING	X.900-X.999

CONTENTS

Summary

This Recommendation | International Standard defines a general framework for the provision of integrity services. The property that data has not been altered or destroyed in an unauthorized manner is called integrity.

Introduction

Many open systems applications have security requirements which depend upon the integrity of data. Such requirements may include the protection of data used in the provision of other security services such as authentication, access control, confidentiality, audit and non-repudiation, that, if an attacker could modify them, could reduce or nullify the effectiveness of those services.

The property that data has not been altered or destroyed in an unauthorized manner is called integrity. This Recommendation | International Standard defines a general framework for the provision of integrity services.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SECURITY FRAMEWORKS FOR OPEN SYSTEMS: INTEGRITY FRAMEWORK

1 Scope

The Recommendation | International Standard on Security Frameworks for Open Systems addresses the application of security services in an Open Systems environment, where the term "Open System" is taken to include areas such as Database, Distributed Applications, Open Distributed Processing and OSI. The Security Frameworks are concerned with defining the means of providing protection for systems and objects within systems, and with the interactions between systems. The Security Frameworks are not concerned with the methodology for constructing systems or mechanisms.

The Security Frameworks address both data elements and sequences of operations (but not protocol elements) which may be used to obtain specific security services. These security services may apply to the communicating entities of systems as well as to data exchanged between systems, and to data managed by systems.

This Recommendation | International Standard addresses the integrity of data in information retrieval, transfer, and management:

1) defines the basic concept of data integrity;
2) identifies possible classes of integrity mechanism;
3) identifies facilities for each class of integrity mechanisms;
4) identifies management required to support the class of integrity mechanism;
5) addresses the interaction of integrity mechanism and the supporting services with other security services and mechanisms.

A number of different types of standard can use this framework, including:

1) standards that incorporate the concept of integrity;
2) standards that specify abstract services that include integrity;
3) standards that specify uses of an integrity service;
4) standards that specify means of providing integrity within an open system architecture; and
5) standards that specify integrity mechanisms.

Such standards can use this framework as follows:

– standards of type 1), 2), 3), 4) and 5) can use the terminology of this framework;
– standards of type 2), 3), 4) and 5) can use the facilities identified in clause 7;
– standards of type 5) can be based upon the classes of mechanisms identified in clause 8.

Some of the procedures described in this security framework achieve integrity by the application of cryptographic techniques. This framework is not dependent on the use of particular cryptographic or other algorithms, although certain classes of integrity mechanisms may depend on particular algorithm properties.

NOTE – Although ISO does not standardize cryptographic algorithms, it does standardize the procedures used to register them in ISO/IEC 9979.

The integrity addressed by this Recommendation | International Standard is that defined by the constancy of a data value. This notion (constancy of a data value) encompasses all instances in which different representations of a data value are deemed equivalent (such as different ASN.1 encodings of the same value). Other forms of invariance are excluded.

The usage of the term data in this Recommendation | International Standard includes all types of data structures (such as sets or collections of data, sequences of data, file-systems and databases).

This framework addresses the provision of integrity to data that are deemed to be write-accessible to potential attackers. Therefore, it focusses on the provision of integrity through mechanisms, both cryptographic and non-cryptographic that do not rely exclusively on regulating access.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Basic Reference Model: The Basic Model.*
- ITU-T Recommendation X.273 (1994) | ISO/IEC 11577:1995, *Information technology – Open Systems Interconnection – Network layer security protocol.*
- ITU-T Recommendation X.274 (1994) | ISO/IEC 10736:1995, *Information technology – Telecommunication and information exchange between systems – Transport layer security protocol.*
- ITU-T Recommendation X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview.*
- ITU-T Recommendation X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework.*
- ITU-T Recommendation X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- ITU-T Recommendation X.224 (1993), *Protocol for providing the OSI connection-mode transport service.*

  ISO/IEC 8073:1992, *Information technology – Telecommunications and information exchange between systems – Open Systems Interconnection – Protocol for providing the connection-mode transport service.*
- CCITT Recommendation X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*

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

2.3 Additional References

- ISO/IEC 9979:1991, *Data cryptographic techniques – Procedures for the registration of cryptographic algorithms.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 This Recommendation | International Standard builds on concepts developed in ITU-T Recommendation X.200 | ISO/IEC 7498-1 and makes use of the following terms defined in it:

a) (N)-connection;

b) (N)-entity;

c) (N)-facility;

d) (N)-layer;

e) (N)-SDU;

f) (N)-service;

g) (N)-user-data.

3.2 This Recommendation | International Standard builds on concepts developed in CCITT Recommendation X.800 | ISO 7498-2 and makes use of the following terms defined in it:

a) access control;

b) connection integrity;

c) data integrity;

d) decipherment;

e) decryption;

f) digital signature;

g) encipherment;

h) encryption;

i) identity-based security policy;

j) integrity;

k) key;

l) routing control;

m) rule-based security policy.

NOTE – Where not otherwise qualified, the term “integrity” in this standard is taken to mean data integrity.

3.3 This Recommendation | International Standard makes use of the following general security-related terms defined in ITU-T Rec. X.810 | ISO/IEC 10181-1:

a) digital fingerprint;

b) hash function;

c) one-way function;

d) private key;

e) public key;

f) seal;

g) secret key;

h) trusted third party.

3.4 This Recommendation | International Standard builds on concepts developed in ITU-T Rec. X.811 | ISO/IEC 10181-2 and makes use of the following terms defined in it:

– time variant parameter.

3.5 For the purpose of this Recommendation | International Standard, the following definitions apply:

3.5.1 **integrity-protected channel**: A communications channel to which an integrity service has been applied.

NOTE – Two forms of integrity services for communication channels are referred to in CCITT Rec. X.800 | ISO 7498-2. These forms (connection and connectionless integrity) are described in annex A.

3.5.2 **integrity-protected environment**: An environment in which unauthorized data alterations (including creation and deletion) are prevented or detectable.

3.5.3 **integrity-protected data**: Data and all relevant attributes within an integrity-protected environment.

3.5.4 **shield**: The conversion of data into integrity-protected data.

3.5.5 **unshield**: The conversion of integrity protected data into the data originally shielded.

3.5.6 **validate**: The checking of integrity-protected data to detect loss of integrity.

4 Abbreviations

PDU Protocol Data Unit

SDU Service Data Unit

SII Shield Integrity Information

MDII Modification Detection Integrity Information

UII Unshield Integrity Information

5 General discussion of integrity

The purpose of the integrity service is to protect the integrity of data and of their relevant attributes which can be compromised in a number of different ways:

1) unauthorized data modification;
2) unauthorized data deletion;
3) unauthorized data creation;
4) unauthorized data insertion;
5) unauthorized data replay.

The integrity service protects against these threats either by means of prevention or by detection with or without recovery. Effective integrity protection may not be possible if the necessary control information (such as keys and SII) is not integrity and/or confidentiality protected; such protection often relies, implicitly or explicitly, on principles different from the ones embodied in the mechanism that protects the data.

The notion of protected environments is explicitly used in this framework so as to capture the idea that integrity protection includes protection against unauthorized creation and/or deletion. Thus, unauthorized data creation/deletion can be seen as unauthorized modifications of some protected environment. Similarly, insertion and replays can be seen as modifications of a structured collection of data (such as a sequence, or a data structure).

We note that some alterations of data can be seen as having no impact on their integrity. For instance, if an ASN.1 description contains a **SET OF** data type, there is no integrity violation if the members of the data type are reordered. Sophisticated integrity mechanisms may recognize that some transformations of structured data do not compromise the data integrity. Such mechanisms allow transformations of signed or sealed data without necessitating recomputations of the digital signature or seal, respectively.

The objective of the integrity service is to protect against or to detect unauthorized data modifications, including unauthorized data creation and deletion. The provision of the integrity service is accomplished through the following activities:

1) **shield**: the generation of integrity protected data from data;
2) **validate**: the checking of integrity-protected data to detect integrity failure;
3) **unshield**: the regeneration of data from integrity-protected data.

These activities do not necessarily employ cryptographic techniques. When they do use cryptographic techniques, they do not necessarily transform the data. For instance, the **shield** operation may be provided by appending a seal or a digital signature to the data. In this case, after successful **validation**, **unshielding** is performed through seal/digital signature removal.

The integrity service applies to Information Retrieval, Transfer, and Management as follows:

1) For information being transferred in an OSI environment, the integrity service is provided by combining **shielding,** transfer using an (N-1)-facility, and **unshielding** to form the transmission part of an (N)-service.
2) For data storage and retrieval, the integrity service is provided by combining **shielding**&storage and retrieval&**unshielding**.

Both **shielding** and **unshielding** can be provided as parallel operations such that the same data [for example, all the data of an (N)-connection] may consist simultaneously of parts that have not yet been **shielded**, parts that are held as integrity-protected data, and parts that have been **unshielded**.

Integrity mechanisms provide protected environments and hence both the **shield** and **unshield** stages involve the transfer of data between protected environments. Where integrity-protected data is transferred from an environment protected by one integrity mechanism to another the **shield** of the second mechanism should precede the **unshield** of the first in order for the data to be continuously protected.

5.1 Basic concepts

Several types of integrity service can be distinguished depending on what data activity is addressed (creation, deletion, modification, insertion, and/or replay), on whether the prevention or just the detection of a violation is required, and on whether they support data recovery in the event of an integrity violation. The different types of integrity service are described in 5.2.

The mechanisms that can be used as a means to provide these services can be split into broad categories that depend on the level of systematic activity assumed in an attempted integrity violation. These different types of mechanism are described in 5.3.

5.2 Types of integrity services

Integrity services are classified according to the following criteria:

1) By the type of violation they protect against. The types of violation are:
   a) unauthorized data modification;
   b) unauthorized data creation;
   c) unauthorized data deletion;
   d) unauthorized data insertion;
   e) unauthorized data replay;
2) By the type of protection they support. The types of protection are:
   a) prevention of integrity compromise;
   b) detection of integrity compromise;
3) By whether they include recovery mechanisms or not:

   In the former case (with recovery), the **unshield** operation may be able to recover the original data (and possibly signal a recovery action or an error for purposes such as audit) whenever the **validate** operation indicates alteration.

   In the latter (without recovery), the **unshield** operation is unable to recover the original data whenever the **validate** operation indicates alteration.

5.3 Types of integrity mechanisms

As a rule, the ability to protect data depends on the medium in use. Some media are, by their very nature, hard to protect (such as removable storage media or communication media) and as a result unauthorized parties can obtain access and engineer data modifications at will. In such media, the purpose of the integrity mechanism is to provide detection of modification and, possibly, restoration of the affected data. The following instances of integrity mechanism are therefore distinguished:

1) Those that prevent access to the medium. Such mechanisms include:
   a) physically isolated, noise free, channels;
   b) routing control;
   c) Access Control.
2) Those that detect unauthorized modifications of data or sequences of data items, including the cases of unauthorized data creation, data deletion, and data replication. Such mechanisms include:
   a) seals;
   b) digital signatures;
   c) data replication (used as a means of combatting other types of violation);
   d) digital fingerprints in conjunction with cryptographic transformations;
   e) message sequence numbers.

In terms of strength of protection these mechanisms can be classified as follows:

1) no protection;
2) detection of modification and of creation;
3) detection of modification, creation, deletion, and replication;
4) detection of modification and of creation with recovery; and
5) detection of modification, creation, deletion, and replication with recovery.

5.4 Threats to integrity

In terms of the services provided, the threats can be classified as follows:

1) Unauthorized creation/modification/deletion/insertion/replay in environments which support data integrity through prevention.

   Example: Tapping into a secure channel.

2) Unauthorized and undetected creation/modification/deletion/insertion/replay in environments which support integrity through detection.

   Example: The integrity of data may be ensured by enciphering the protected data and associated checksums as described in ITU-T Rec. X.274 | ISO/IEC 10736. If the communicating entities A and B are using the same key to support encipherment and the origin of the data is not integrity protected, then integrity protected data that were sent from A to B can be later submitted to A as if originating from B (a reflection attack).

In terms of the medium in which the data reside, threats can be classified as:

1) threats particular to the medium in which data is stored;
2) threats particular to the medium in which data is transmitted;
3) threats that are independent of the medium.

Insofar as this Recommendation | International Standard regards integrity violations as unauthorized actions, it does not address the issue of authorized modifications which may violate the external consistency of the data as described in Annex B (e.g. false accounting). Consequently, this Recommendation | International Standard, unlike the confidentiality framework, does not address the issue of insider attacks (the confidentiality framework addresses issues pertaining to the continuous protection of information such as the possibility that authorized access may be followed by intentional or unintentional unauthorized release of the confidentiality-protected information).

5.5 Types of integrity attacks

To each of the threats enumerated above correspond one or more attacks, i.e. instantiations of the threat in question. Attacks aim at defeating the mechanism(s) used to provide integrity and can be classified as follows:

1) Attacks aimed at defeating cryptographic mechanisms or at exploiting weaknesses of such mechanisms. Such attacks include:
   a) penetration of the cryptographic mechanism;
   b) (selective) deletion and replication.
2) Attacks aimed at defeating the contextual mechanism used (contextual mechanisms exchange data at specific times and/or locations). Such attacks include:
   a) massive, coordinated changes of data-item replicas;
   b) penetration of the context establishing mechanism.
3) Attacks aimed at defeating detection and acknowledgement mechanisms. Such attacks include:
   a) false acknowledgements;
   b) exploitation of faulty sequencing between the acknowledgement mechanism and the treatment of the received data.
4) Attacks aimed at defeating, subverting, or suborning prevention mechanisms. Such attacks include:
   a) attacks on the mechanism itself;
   b) penetration of the services the mechanism relies upon;
   c) exploitation of utilities with unintended side-effects.

6 Integrity policies

An integrity policy is the part of a security policy which deals with the provision and use of the integrity service.

Data whose integrity is protected are subject to control over which entities may create, alter, and delete them. An integrity policy must therefore identify the data that is subject to controls and indicate which entities are intended to be allowed to create, alter, or delete the data.

Depending on the relative importance of the integrity of different types of data, an integrity policy may also indicate the type and strength of mechanisms that are to be used to provide the integrity services for each of the different types of data.

The management of an integrity security policy is not addressed in this Recommendation | International Standard.

6.1 Policy expression

In expressing an integrity policy means are required for identifying the information involved and the entities involved.

A security policy can be thought of as a set of rules. Each rule in an integrity policy can associate a data characterization, an entity characterization, and a set of allowed data activities (typically, create, alter, and delete). In some policies these rules are not explicitly stated but can be derived from the policy expression.

The following subclauses describe a number of ways in which integrity policies may be expressed. Note that, although some integrity mechanisms will have a parallel in specific kinds of policy expression, the way in which the policy is expressed does not directly imply the use of a specific mechanism to implement the policy.

6.1.1 Data characterization

A policy may identify data in a variety of ways. For example:

1) by identifying the entities authorized to create/alter/delete such data;
2) by its location; or
3) by identifying the context in which the data is presented (e.g. its intended function).

6.1.2 Entity characterization

There are many ways to characterize the entities to which an integrity policy applies. Two main examples are given.

6.1.2.1 Identity based policies

In this form of policy expression entities are identified individually; are identified as part of a group of equivalent entities (for the purposes of the integrity policy); or are identified by the role in which they are acting. Thus, each entity allowed successfully to be involved in a data activity will possess an individual identity, group identity, or role identity that is used to characterize it.

In the case of roles, an integrity policy may specify exclusive groups of roles available to each entity whereby roles from different groups may not be claimed simultaneously.

6.1.2.2 Rule based policies

In this form of integrity policy expression, attributes are associated with each entity and with each integrity protected data item. Global rules operating on attributes of the entities and of the data determine which actions are permitted. Rule Based Policies are discussed in more detail in the Security Framework Overview (see ITU-T Rec. X.810 | ISO/IEC 10181-1).

7 Integrity information and facilities

This clause classifies the information necessary for the correct operation of an integrity service as well as the facilities that either use or generate the information in question.

7.1 Integrity information

In order that data may be **shielded**, **validated**, or **unshielded** auxiliary information may be employed. This auxiliary information is known as Integrity Information. Integrity information can be classified as follows.

7.1.1 Shield integrity information

Shield Integrity Information (SII) is information used to **shield** data. Examples include:

1) private keys;
2) secret keys;
3) algorithm identifier and relevant cryptographic parameters;
4) time variant parameters (e.g. time stamps).

7.1.2 Modification detection integrity information

Modification Detection Integrity Information (MDII) is information used to **validate** integrity protected data. Examples include:

1) public keys;
2) secret keys.

7.1.3 Unshield integrity information

Unshield Integrity Information (UII) is information used to **unshield** integrity protected data. Examples include:

1) public keys;
2) secret keys.

7.2 Integrity facilities

A number of Integrity Facilities have been identified in Annex C. They can be distinguished into those that relate to operational aspects and those that relate to management aspects.

7.2.1 Operational related facilities

These facilities are:

1) **shield**

   This facility applies integrity protection to data. Candidate inputs include:

   – data to be protected;
   – SII;
   – mechanism-specific identifiers such as those mentioned in Annex C,

   while candidate outputs include:

   – integrity protected data;
   – completion/return codes.

2) **validate**

   This facility checks integrity-protected data for modifications. Candidate inputs include:

   – integrity protected data;
   – MDII;
   – mechanism specific identifiers such as those mentioned in Annex C,

   while candidate outputs include:

   – an indication as to whether the integrity of the data has been violated.

3) **unshield**

   This facility converts integrity-protected data into the data originally **shielded**. Candidate inputs include:

   – integrity-protected data;
   – UII;
   – mechanism specific identifiers such as those mentioned in Annex C,

   while candidate outputs include:

   – data;
   – completion/return codes.

7.2.2 Management related facilities

The integrity management facilities allow a user to obtain, modify, and remove information (such as keys) which is necessary for the provision of integrity. In broad terms these facilities are:

1) install management information;
2) modify management information;
3) delete management information;
4) list management information;
5) disable management information;
6) re-enable management information.

8 Classification of integrity mechanisms

This clause classifies Integrity Mechanisms by the means they use to provide an integrity service.

8.1 Integrity provision through cryptography

There are two classes of cryptographic integrity mechanism to be considered:

1) integrity mechanisms based on symmetric cryptographic techniques in which the validation of integrity-protected data is possible through knowledge of the same secret key that was used to shield the data; and
2) integrity mechanisms based on asymmetric cryptographic techniques in which the validation of integrity-protected data is possible through knowledge of the public key corresponding to the private key that was used to shield the data.

Mechanisms of the first type correspond to seals while mechanisms of the second correspond to digital signatures.

Time variant parameters can be used in conjunction with integrity mechanisms based on cryptographic techniques so as to protect against replay.

8.1.1 Integrity provision through sealing

Sealing provides integrity by appending a cryptographic checkvalue to the data to be protected. In sealing, the same secret key is used to protect and to validate the integrity of the data. When this class of mechanisms is used, either all potential validators are known in advance, or they must have the means of accessing the secret key.

The set of entities capable of sealing the data and the set of entities capable of validating the data are, by the very definition of the mechanism, coincident.

This mechanism supports modification detection as follows:

- **Shield** is achieved by attaching a cryptographic checkvalue on the data to be integrity-protected (e.g. a one-way function is computed over the data to be protected and this value is transformed through an encipherment mechanism).
- **Validate** is achieved by using the data, the cryptographic checkvalue, and the secret key to determine if the data match the seal (e.g. the facility could submit the data and the secret key to the shield facility and compare the resultant seal to the value that was actually affixed to the data). When they do, the data is considered not to have been modified.
- **Unshield** is performed by removing the cryptographic checkvalue, once the data have been validated.

8.1.2 Integrity provision through Digital Signatures

Digital Signatures are computedusing a private key and an asymmetric cryptographic algorithm. The shielded data (data plus the appended digital signature) can be validated using the corresponding public key. In general, the public key can be made publicly available.

Digital Signatures permit the set of entities which can **validate** the data to be arbitrary in size and composition.

This mechanism supports modification detection as follows:

- **Shield** is achieved by attaching cryptographic checkvalues on the data to be integrity-protected (e.g. a digital fingerprint of the data to be protected is computed and this value is combined with the private key and, possibly, other parameters to generate one or more values which, collectively, form the digital signature).
- **Validate** is achieved by using a digital fingerprint of the the data received, the digital signature, and the public key with an algorithm that verifies the digital signature. If verification of the digital signature fails, then the data is considered to have been altered.
- **Unshield** is performed by removing the cryptographic checkvalue, once the data have been validated.

This mechanism may also support data origin authentication and non-repudiation.

8.1.3 Integrity provision through encipherment of redundant data

The integrity of Redundant Data (e.g. natural language) can be supported through encipherment. Data that include error detection codes and digital fingerprints are redundant and their integrity can be protected through encipherment (provided that the encipherment algorithm used makes it impossible to predict how changes of the enciphered data, will be reflected on the original data after decipherment).

The lower layer security protocols (see ITU-T Recs. X.273 | ISO/IEC 11577 and X.274 | ISO/IEC 10736) use this type of mechanism to provide integrity protection to confidentiality protected data.

This mechanism supports modification detection as follows:

- **Shield** is achieved by enciphering the redundant data.
- **Validate** is achieved by deciphering the shielded data and determining if they satisfy whatever invariances the original data satisfied. Examples include:
  1) if the original data were a sequence of ASCII characters, the recovered data should have the same property;
  2) if the original data were presumed to be a human utterance in a given language, the recovered data should be (or made to be with small modifications) a commonly accepted human utterance in the same language;
  3) if the original data contained checksums, one can compute the same checksums over the relevant portions of the protected data and see if the results match the checksum values embedded in the deciphered data.
- **Unshield** is performed by deciphering the enciphered data.

  NOTE – This mechanism is known to be unsuitable for several common forms of redundancy (e.g. data with error detection codes) used in conjunction with common forms of encipherment (e.g. block ciphers, such as DES, with or without chaining). A relatively simple example of potential pitfalls is the following:

  If ASCII text (e.g. common electronic mail) were to be protected through DES encryption, the message could be undetectably modified through truncation (unless, of course, the mail header contains a length indicator, as it commonly does).

8.2 Integrity provision through context

Integrity can be supported by mechanisms that store or transmit the data in one or more pre-agreed contexts. Such mechanisms can protect data as well as data structures (e.g. sequences of data units). They include:

8.2.1 Data Replication

This class of integrity mechanisms is based on the replication of the data over space (e.g. several storage areas) or time (e.g. at different times). It is assumed that potential attackers cannot simultaneously compromise more than a limited number of replicas and that whenever attacks are detected, the data can be reconstructed from the genuine copies.

As an example, such a mechanism may be used by databases resistant to penetration attacks.

These mechanisms provide deletion integrity detection with recovery and can be used in conjunction with other security mechanisms. They provide integrity as follows:

- **Shield** is achieved by providing many copies of the same data either successively in time or at different locations.

- **Validate** is achieved by gathering a copy of data at each of the given times or locations. They are compared and if they are not all identical, an integrity violation is judged to have occurred.
- **Unshield** (with recovery) can be achieved when some pre-established criterion is met (e.g. “90% or more of the values agree”) by choosing as the correct value the one that minimizes some pre-assigned measure (e.g. the probability of erroneous recovery).

It should be noted that our recovery criterion must be satisfied when all values agree and that under these circumstances the value that minimizes the measure should be the common value.

8.2.2 Pre-agreed context

These mechanisms provide deletion detection of integrity-protected data and are often used in conjunction with other integrity mechanisms:

- **Shield** is achieved by providing data at a specific time and/or location within a given level of variation.
- **Validate** is achieved by expecting the data at the given time and/or location. If it is not present, an integrity violation is judged to have occurred.

In order to prevent the substitution of alternative data, this mechanism must, as noted above, be combined with other integrity mechanisms.

8.3 Integrity provision through detection and acknowledgement

These mechanisms use an integrity detection whenever performing idempotent operations with positive feedback (an operation is deemed idempotent if multiple, successive executions of the operation give the same result as a single one). Examples of such mechanisms are transmissions with positive acknowledgements (e.g. Transport class 4 as described in ITU-T Rec. X.224 | ISO/IEC 8073) and remote operations with feedback. These mechanisms assume that **shield** and **validate/unshield** operate over the same duration of time and are not, as a rule, suitable for data storage:

- **shield** is achieved by repeating the same action until the integrity policy dictates otherwise or until a positive acknowledgement is received.
- **validate** is performed on each instance of shielded data (unless the integrity policy otherwise dictates); successful validations result in signalling positive acknowledgements to the entity performing the shield operation.

If a correction can be made to deliver correct information from the **unshield** of the modification integrity protection mechanism used, this mechanism’s unshield can signal a positive acknowledgement.

An efficiency can result if, when **validate** indicates a negative result (either a modification or a deletion having taken place) a negative acknowledgement is indicated to the **shield** operation.

These mechanisms assume that data modifications can be detected through other means and, therefore, can be seen as enhancing mechanisms protecting against data modification.

8.4 Integrity provision through prevention

Integrity can be provided by preventing physical access to data storage or transmission media and through access control.

The specification of means to prevent physical access is outside the scope of this framework.

Access control is described in the Access Control Framework.

9 Interactions with other security services and mechanisms

This clause describes how other Security Services and Mechanisms can be used to support Integrity. The use of Integrity to support other security services is not described here.

9.1 Access Control

Access Control can be used to create integrity-protected environments.

9.2 Data origin authentication

Data Origin Authentication can be used to support integrity, e.g. if the origin of a PDU cannot be authenticated, the PDU may be deemed compromised. Similarly, if the presumed source of a PDU is not authorized to create the PDU, an integrity violation has taken place.

9.3 Confidentiality

Redundancy can in some instances be combined with encipherment to yield data that cannot be modified without detection.

Indeed redundant data (such as natural language and data including checksums and hash values) have invariant properties. As a rule, changes in the enciphered data will not preserve, with high probability, these properties once the altered data is “deciphered”.

(Another way to say this is that when k bits of information are encoded in k + m bit sequences, the valid encodings constitute a sparse subset of all possible bit sequences; if changes in the enciphered data result, after decipherment, in changes that from the point of the attacker are seen as random, the probability that altered data will be deciphered as a valid encoding is of the order of 2^{-m}.)

Thus redundancy (including checksums and hash functions) in conjunction with cryptography based confidentiality can support integrity.

Annex A

Integrity in the OSI Basic Reference Model

(This annex does not form an integral part of this Recommendation | International Standard)

The relationship of security services to the OSI Reference Model is defined in CCITT Rec. X.800 | ISO 7498-2. This annex summarizes what is relevant to integrity.

Different security services are considered:

- connection integrity with recovery;
- connection integrity without recovery;
- selective field connection integrity;
- connectionless integrity;
- selective field connectionless integrity.

A.1 Connection integrity with recovery

Connection integrity provides for the integrity of all (N)-user-data on an (N) connection and detects any modification, insertion, deletion and replay of any data within an entire SDU sequence (with recovery attempted).

A.2 Connection integrity without recovery

Connection integrity provides for the integrity of all (N)-user-data on an (N) connection and detects any modification, insertion, deletion and replay of any data within an entire SDU sequence (with no recovery attempted).

A.3 Selective field connection integrity

Selective field integrity provides for the integrity of selected fields within the (N)-user-data on an (N)-SDU transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted or replayed.

A.4 Connectionless integrity

Connectionless integrity when provided by the (N)-layer provides integrity assurance to the requesting (N + 1) entity.

A.5 Selective field connectionless integrity

Selective field integrity provides for the integrity of selected fields within a single connectionless SDU and takes the form of determination of whether the selected fields have been modified.

A.6 Use of integrity within OSI layers

The integrity services are relevant to the following OSI layers:

- data link layer (layer 2);
- network layer (layer 3);
- transport layer (layer 4);
- application layer (layer 7);

A.6.1 Use of integrity at the data link layer

Integrity services can be supported at the data link layer as described in IEEE 802.10.

A.6.2 Use of integrity at the network layer

Connection integrity without recovery and connectionless integrity are the only integrity services provided at the network layer. Connection integrity without recovery and connectionless integrity may be provided by a data integrity mechanism sometimes in connection with an encipherment mechanism. These services allow integrity between network nodes, subnetwork nodes or relays.

A.6.3 Use of integrity at the transport layer

Connection integrity with recovery, connection integrity without recovery and connectionless integrity are the only integrity services provided at the transport layer. Connection integrity with or without recovery and connectionless integrity may be provided by a data integrity mechanism sometimes in connection with an encipherment mechanism. These services allow integrity between end systems.

A.6.4 Use of integrity at the application layer

All the integrity services, namely, connection integrity with recovery, connection integrity without recovery, selective field connection integrity, connectionless integrity and selective field connectionless integrity may be provided at the application layer. Connection integrity with or without recovery and connectionless integrity can be supported using a lower layer data integrity mechanism (sometimes in conjunction with an encipherment mechanism). Selective field integrity can be supported using a lower layer data integrity mechanism (sometimes in conjunction with an encipherment mechanism) at the presentation layer.

Annex B

External Data Consistency

(This annex does not form an integral part of this Recommendation | International Standard)

NOTE – The text that follows addresses the issues of internal/external integrity (as developed in the original paper of Clark and Wilson and other subsequent work) and explores the impact that these notions could have on the integrity framework. Whenever possible, and in the interest of limiting paraphrasing to a minimum, the text lifts sentences out of the quoted publications.

Non-normative references 1, 2, and 3 below contain a number of references to an integrity model proposed by D.D. Clark and D.R. Wilson. What follows is an attempt to summarize the model and to highlight those of its points that may impact this framework.

This framework considers integrity in the sense of maintaining a specific invariant on data (its constant value). The model of Clark and Wilson considers additional invariants. Namely, it assumes that a computer system reflects and emulates data and processes that are external to the computer. Therefore, the final test of integrity is to ensure that the data within the computer are consistent with the world they are intended to represent. It follows then that integrity controls can never be a matter strictly internal to the computer.

As a result, the integrity of data in a Clark-Wilson model can be seen as a two step approach:

1) adequate mechanisms must exist to initiate changes, when changes are needed, so that the external consistency of the data will be maintained; and

2) adequate mechanisms must exist that ensure that when changes are initiated, these changes are carried out as an atomic operation of a well-formed transaction.

Assuming that above points accurately reflect our intuitive understanding of Integrity, we are led to draw the following semantic distinctions:

– **Internal data consistency**: A datum is internally consistent if and only if all modifications of this item satisfy the relevant integrity security policies.

– **External data consistency**: A datum is externally consistent whenever its value conforms to the real world situation it describes.

If the above distinction is accepted, then it may be combined with the following classification of the strength of the integrity protection:

– Strong protection maintains the internal and external consistency of the data. Weak protection detects the violation of the internal and/or external consistency of the data.

Moreover and to the extent that the changes in the data are carried out as atomic operations by trusted processes, one may wish to address the properties of these operations, e.g.:

1) can the atomicity of the operation be violated?; and

2) are there assurances that the operation will indeed be carried out?

Finally, one may wish to classify the integrity mechanisms with respect to the maintenance of the following properties:

– internal/external consistency;

– weak/strong protection;

– atomicity/assurance of operations on protected data.

Thus, when specifying an integrity mechanism, the following should be considered:

1) Which form of consistency (internal, external, both) does the mechanism address?

2) Does it provide weak or strong integrity protection to data? Is it resistant to engineered attacks, to random changes, or both? Does it provide for recovery?

3) Does it protect the atomicity of operations or does it provide assurance (as well) that the operation will be carried out?

References

1) CLARK, David and WILSON, David: A Comparison of Commercial and Military Computer Security Policies, *Proceedings of 1987 IEEE Symposium on Security and Privacy*.

2) NIST Special Publication 500-160: Report on the Invitational Workshop on Integrity Policy in Computer Information Systems (WIPCIS); edited by Stuart W. Katzke and Zella G. Ruthberg.

3) NIST Special Publication 500-168: Report on the Invitational Workshop on Data Integrity, edited by Zella G. Ruthberg and William T. Polk.

Annex C

Integrity Facilities Outline

(This annex does not form an integral part of this Recommendation | International Standard)

<table>
<tr><td colspan="3" rowspan="4">Security Facilities Outline</td><td rowspan="3">Element</td><td colspan="3">Entity: Initiator, Verifier, Integrity-TTP</td></tr>
<tr><td colspan="3">Function:</td></tr>
<tr><td colspan="3">Info. object: Integrity-protected data</td></tr>
<tr><td>Goal of Service</td><td colspan="3">To protect data against unauthorized modification/deletion/creation/insertion/replication</td></tr>
<tr><td rowspan="5">ACTIVITY</td><td rowspan="2"></td><td>Entity</td><td colspan="4">Security Domain Authority (SDA)</td></tr>
<tr><td>Function</td><td colspan="4"></td></tr>
<tr><td colspan="2">Management related activity</td><td colspan="4">– Install management information
– Modify management information
– Delete management information
– List management information
– Disable management information
– Re-enable management information</td></tr>
<tr><td rowspan="3"></td><td>Entity</td><td>Initiator</td><td>Verifier</td><td>Integrity-TTP</td></tr>
<tr><td>Function</td><td></td><td></td><td></td></tr>
<tr><td>Operational related activity</td><td>– Shield data
ACL label
Checksum
Crypto check value
Digital signature</td><td>– Validate integrity
ACL label
Checksum
Crypto check value
Digital signature
– Unshield (recover data)
Crypto check value
Digital signature</td><td>– Tissue
ACL
Crypto check value
Checksum
Certificate</td></tr>
<tr><td rowspan="3">INFORMATION</td><td colspan="2">Input/ Output Data element managed by SDA</td><td colspan="3">– IDs (initiator, verifier, integrity-TTP)
– Crypto keys
– Time variant value</td></tr>
<tr><td colspan="2">Information type used in operation</td><td colspan="3">– Shield Integrity Information (SII)
– Modification Detection Integrity Information (MDII)
– Unshield Integrity Information (UII)</td></tr>
<tr><td colspan="2">Control Information</td><td colspan="3">– Time period
– Route
– Location
– System status</td></tr>
</table>

This annex makes use of the following concepts:

C.1 Integrity entities

Integrity in Open Systems involves some basic entities:

- Initiator;
- Verifier;
- Trusted Third Party for Integrity facilities.

C.1.1 Initiator

The entity that generates integrity-protected data by shielding the data and sending or storing them.

C.1.2 Verifier

The entity that receives or retrieves data from the initiator, checks its value after validating it to detect integrity failure, and, when necessary, regenerates the value of the data.

C.1.3 Trusted Third Party for integrity facilities

An entity that distributes SII or MDI and/or performs integrity relevant operations on behalf of the initiator or the verifier.