INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

X.811

(04/95)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

SECURITY

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

ITU-T Recommendation X.811

(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.811 was approved on 10th of April 1995. The identical text is also published as ISO/IEC International Standard 10181-2.

NOTE

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

© ITU 1996

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

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

Page

Introduction

Many applications have requirements for security to protect against threats to the communication of information. Some commonly known threats, together with the security services and mechanisms that can be used to protect against them, are described in ITU Rec. X.800 | ISO 7498-2.

Many Open Systems applications have security requirements which depend upon correctly identifying the principals involved. Such requirements may include the protection of assets and resources against unauthorized access, for which an identity based access control mechanism might be used, and/or the enforcement of accountability by the maintenance of audit logs of relevant events, as well as for accounting and charging purposes.

The process of corroborating an identity is called authentication. This Recommendation | International Standard defines a general framework for the provision of authentication services.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

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

1 Scope

The series of Recommendations | International Standards on Security Frameworks for Open Systems addresses the application of security services in an Open Systems environment, where the term "Open Systems" 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 are 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:

- defines the basic concepts for authentication;
- identifies the possible classes of authentication mechanisms;
- defines the services for these classes of authentication mechanism;
- identifies functional requirements for protocols to support these classes of authentication mechanism; and
- identifies general management requirements for authentication.

A number of different types of standards can use this framework including:

1) standards that incorporate the concept of authentication;
2) standards that provide an authentication service;
3) standards that use an authentication service;
4) standards that specify the means to provide authentication within an open system architecture; and
5) standards that specify authentication mechanisms.

[Note that the service in 2), 3) and 4) might include authentication but may have a different primary purpose.]

These standards can use this framework as follows:

* standard types 1), 2), 3), 4) and 5) can use the terminology of this framework;
* standard types 2), 3), 4) and 5) can use the services defined in clause 7 of this framework; and
* standard types 5) can be based on the mechanisms defined in clause 8 of this framework.

As with other security services, authentication can only be provided within the context of a defined security policy for a particular application. The definitions of security policies are outside the scope of this ITU Recommendation | International Standard.

The scope of this Recommendation | International Standard does not include specification of details of the protocol exchanges which need to be performed in order to achieve authentication.

This Recommendation | International Standard does not specify particular mechanisms to support these authentication services. Other standards (such as ISO/IEC 9798) develop specific authentication methods in greater detail. Furthermore, examples of such methods are incorporated into other standards (such as ITU Rec. X.509 | ISO/IEC 9594-8) in order to address specific authentication requirements.

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

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

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 Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.810[1] | ISO/IEC 10181-1:...[1], *Information technology – Security frameworks for open systems: Overview.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- 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.*
- ISO/IEC 10116:1991, *Information technology – Modes of operation for an n-bit block cipher algorithm*.

3 Definitions

This Recommendation | International Standard makes use of the following general security-related terms defined in Rec. X.800 | ISO 7498-2:

- audit;
- audit trail;
- authentication information;
- confidentiality;
- cryptography;
- cryptographic checkvalue;
- data origin authentication;
- data integrity;
- decipherment;
- digital signature;
- encipherment;
- key;

[1] Presently at the stage of draft.

- key management;
- masquerade;
- password;
- peer-entity authentication;
- security policy.

This Recommendation | International Standard makes use of the following term defined in ISO/IEC 10116:

- block chaining

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

- digital fingerprint;
- hash function;
- one-way function;
- private key;
- public key;
- seal;
- secret key;
- security authority;
- security certificate;
- security domain;
- security token;
- trust;
- trusted third party.

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

3.1 **asymmetric authentication method**: A method of authentication, in which not all authentication information is shared by both entities.

3.2 **authenticated identity**: A distinguishing identifier of a principal that has been assured through authentication.

3.3 **authentication**: The provision of assurance of the claimed identity of an entity.

3.4 **authentication certificate**: A security certificate that is guaranteed by an authentication authority and that may be used to assure the identity of an entity.

3.5 **authentication exchange**: A sequence of one or more transfers of exchange authentication information (AI) for the purposes of performing an authentication.

3.6 **authentication information**: Information used for authentication purposes.

3.7 **authentication initiator**: The entity that starts an authentication exchange.

3.8 **challenge**: A time variant parameter generated by a verifier.

3.9 **claim authentication information (claim AI)**: Information used by a claimant to generate exchange AI needed to authenticate a principal.

3.10 **claimant**: An entity which is or represents a principal for the purposes of authentication. A claimant includes the functions necessary for engaging in authentication exchanges on behalf of a principal.

3.11 **distinguishing identifier**: Data that unambiguously distinguishes an entity in the authentication process. This Recommendation | International Standard requires that such an identifier be unambiguous at least within a security domain.

3.12 **exchange authentication information (exchange AI)**: Information exchanged between a claimant and a verifier during the process of authenticating a principal.

3.13 off-line authentication certificate: An authentication certificate binding a distinguishing identifier to verification AI, which may be available to all entities.

3.14 on-line authentication certificate: An authentication certificate for use in an authentication exchange, obtained directly by the claimant from the authority who guarantees it.

3.15 principal: An entity whose identity can be authenticated.

3.16 symmetric authentication method: A method of authentication in which both entities share common authentication information.

3.17 time variant parameter: A data item used by an entity to verify that a message is not a replay.

3.18 unique number: A time variant parameter generated by a claimant.

3.19 verification authentication information (verification AI): Information used by a verifier to verify an identity claimed through exchange AI.

3.20 verifier: An entity which is or represents the entity requiring an authenticated identity. A verifier includes the functions necessary for engaging in authentication exchanges.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

AI	Authentication Information
OSI	Open Systems Interconnection

5 General discussion of authentication

5.1 Basic concepts of authentication

Authentication provides assurance of the claimed identity of an entity. Authentication is meaningful only in the context of a relationship between a principal and a verifier. Two important cases are:

– the principal is represented by a claimant which has a specific communications relationship with the verifier (entity authentication); and

– the principal is the source of a data item available to the verifier (data origin authentication).

This Recommendation | International Standard distinguishes between these two forms of authentication.

Entity authentication provides corroboration of the identity of a principal, within the context of a communication relationship. The principal's authenticated identity is assured only when this service is invoked. Assurance of continuity of authentication can be obtained as described in 5.2.7. An example of this is OSI peer-entity authentication as defined in CCITT Rec. X.800 | ISO 7498-2.

Data origin authentication provides corroboration of the identity of the principal that is responsible for a specific data unit.

NOTES

1 When using data origin authentication, it is also necessary to have adequate assurance that the data has not been modified. This may be accomplished by using an integrity service. For example:

a) by using environments in which data cannot be altered;

b) by verifying that the data received matches a digital fingerprint of the data sent;

c) by using a digital signature mechanism; and

d) by using a symmetric cryptographic algorithm.

2 The term communications relationship used in defining entity authentication may be interpreted in a broad way and could refer, for example, to an OSI connection, inter-process communication, or interaction between a user and a terminal.

5.1.1 Identification and authentication

A principal is an entity whose identity can be authenticated. A principal has one or more distinguishing identifiers associated with it. Authentication services can be used by entities to verify purported identities of principals. A principal's identity which has been so verified is called an authenticated identity.

Examples of principals that can be identified and hence authenticated are:

- human users;
- processes;
- real open systems;
- OSI layer entities; and
- enterprises.

Distinguishing identifiers are required to be unambiguous within a given security domain. Distinguishing identifiers distinguish a principal from others in the same domain, in one of two ways:

- at a coarse level of granularity, by virtue of membership in a group of entities considered equivalent for purposes of authentication (in this case the entire group is considered to be one principal and has one distinguishing identifier); or
- at the finest degree of granularity, identifying one and only one entity.

When authentication takes place between different security domains, a distinguishing identifier may not be sufficient to identify unambiguously an entity, as different security domain authorities may use the same distinguishing identifiers. In this case, distinguishing identifiers have to be used in conjunction with an identifier of the security domain in order to provide an unambiguous identifier for the entity.

Examples of typical distinguishing identifiers are:

- directory names (ITU-T Rec. X.509 | ISO/IEC 9594-8);
- network addresses (ITU-T Rec. X.213 | ISO/IEC 8348);
- AP-titles and AE-titles (ITU-T Rec. X.207 | ISO/IEC 9545);
- object identifiers (ITU Rec. X.208 | ISO/IEC 8824);
- names of persons (unambiguous within the context of the domain);
- passport or social security numbers.

5.1.2 Authentication entities

The term claimant is used to describe the entity which is or represents a principal for the purposes of authentication. A claimant includes the functions necessary for engaging in authentication exchanges on behalf of a principal.

The term verifier is used to describe the entity which is or represents the entity requiring an authenticated identity. A verifier includes the functions necessary for engaging in authentication exchanges.

An entity involved in mutual authentication (see 5.2.4) will assume both claimant and verifier roles.

The term trusted third party is used to describe a security authority or its agent, trusted by other entities with respect to security-related activities. In the context of this Recommendation | International Standard, a trusted third party is trusted by a claimant and/or a verifier for the purposes of authentication.

NOTE – The claimant or verifier may be refined into multiple functional components, possibly residing in different open systems.

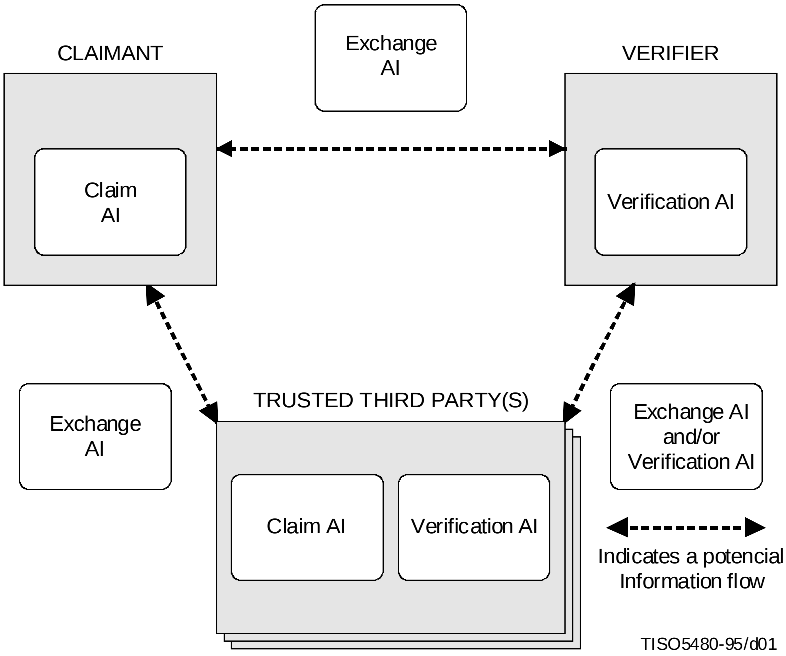
5.1.3 Authentication information

Types of authentication information are:

- exchange authentication information (exchange AI);
- claim authentication information (claim AI);
- verification authentication information (verification AI).

The term authentication exchange is used to describe a sequence of one or more transfers of exchange AI for the purposes of performing an authentication.

Figure 1 illustrates the relationship between claimant, verifier, trusted third party, and the three types of authentication information.



NOTES

1 In some scenarios no trusted third parties may be involved.

2 The verification AI may be that of the principal or that of the trusted third party (see 5.5 for further explanation).

Figure 1 – Illustration of relationship between claimant, verifier trusted third party, and types of authentication information

In some cases, in order to generate exchange AI, a claimant may need to interact with a trusted third party. Similarly, in order to verify exchange AI, a verifier may need to interact with a trusted third party. In these cases the trusted third party may hold verification AI related to a principal.

It is also possible that a trusted third party is used in the transfer of exchange AI.

The entities may also need to hold authentication information to be used in authenticating the trusted third party.

Examples of the three types of authentication information are given in 6.1.

NOTE – Because the term credentials is not always used in a consistent way in other Recommendations | International Standards, this Security Framework does not use this term. The term credentials as defined in ITU Rec. X.800 | ISO 7498-2 could be an example of exchange AI.

5.2 Aspects of authentication service

5.2.1 Threats to authentication

The goal of authentication is to provide assurance of an identity of a principal. Mechanisms to provide authentication normally must eliminate the threats of masquerade and replay.

Masquerade refers to the pretence by an entity to be a different entity. That is, the entity pretends to be another entity that is related to the verifier in a specific way (e.g. through the origin of data, or through a communications relationship). These types include replay, relay and compromise of claim AI.

A masquerade threat occurs in the context of an activity (e.g. origin of data, a communication relationship) initiated either by the claimant or the verifier. Protection against a masquerade threat to an activity requires the use of an integrity service to bind these data items to an authentication exchange. To counter masquerade-related threats, authentication must be used in conjunction with some form of integrity service, which binds the authenticated identity to the activity.

Replay refers to the repetition of exchange AI, to produce an unauthorized effect. Replay is usually used in combination with other attacks, such as data modification. Not all authentication mechanisms are equally resistant to replay. Replay can be a threat to other security services. Authentication can be used to counter replay since it offers a means to establish the source of the exchanged information.

5.2.2 Authentication forwarding

In some circumstances, a principal may have requirements to act within a system indirectly. In such cases, its representation within the system will have to be created. Moreover, before a representation for a principal can be created within a system, the principal must be authenticated.

When acting on the principal’s behalf, the representation will be authenticated in place of the principal’s identity. Because the representation acts as if it were the principal, the principal’s actions can be carried on within the system without requiring the direct involvement of the principal. See Annex A for an example.

When the principal is a human user, mechanisms can be used which limit the lifetime of the representation to the period of time in which the user is physically present at a particular location.

A claimant, in acting on behalf of a principal, may access another system which creates its own representation of the principal following authentication. The creation of this representation is referred to as authentication forwarding.

The ability to forward authentication in such a manner may be affected by security policy.

5.2.4 Unilateral and mutual authentication

Authentication can be unilateral or mutual. Unilateral authentication provides assurance of the identity of only one principal. Mutual authentication provides assurance of the identities of both principals.

Entity authentication may be either mutual or unilateral. By its very nature, data origin authentication is always unilateral.

5.2.5 Initiation of an authentication exchange

An authentication exchange can be initiated by the claimant or the verifier. The entity which starts the exchange is called the authentication initiator.

5.2.6 Revocation of authentication information

Revocation of authentication information refers to the permanent invalidation of verification AI.

Policy may require the revocation of authentication information in certain situations. The decision to revoke authentication information may be based on detected security violation events, change of policy or other reasons. The revocation of authentication information may or may not imply revocation of existing access, or have other derivative effects.

In addition, the following management-related actions may be taken:

a) recording the event in the audit trail;

b) local reporting of the event;

c) remote reporting of the event; and/or

d) disconnection of a communication relationship.

The specific action to be taken for each event is dependent on the security policy in operation and other factors relating to the status of the communication relationship, e.g. whether or not an update occurred when the principal was logged in and active.

5.2.7 Assurance of continuity of authentication

Entity authentication only provides assurance of an identity at an instant of time. One way of obtaining the assurance of continuity of the authentication is by linking the authentication service with a data integrity service.

An authentication service and an integrity service are said to be linked when the principal is initially authenticated using an authentication service and further data sent on behalf of the principal is bound together with the exchange AI using an integrity service. This ensures that the later information may not be altered by any other entity and therefore must come from the initially authenticated principal. It is important that the integrity service is provided over the whole of the path that the information takes from the principal to the verifier. For example, masquerade is possible if some of the information can be produced by principals other than the one authenticated.

Another way to obtain assurance that the same remote entity is still present at a later time is to perform further authentication exchanges from time to time. However, this does not prevent intrusions during the intervals, hence no assurance of continuity is obtained. For example, the following attack is possible: an intruder, when called upon to do further authentication, allows the valid party to do the authentication actions; after these actions are complete the intruder again takes over.

If the integrity mechanism requires a key, that key may be derived from parameters specified during the authentication exchange. Having thus established that the key is associated with the authenticated principal, its use in the integrity mechanism will serve to link the two services provided as described above.

The way to derive a key for an integrity service can be specified as part of the parameters specifying which methods and algorithms should be used for the overall authentication exchange.

NOTE – When other security services are used, it is also possible to derive service information from parameters specified during the authentication exchange, e.g. a confidentiality key.

5.2.8 Distribution of authentication components across multiple domains

It is possible for security domains to enter into a relationship such that a claimant in one domain can be authenticated by a verifier in another domain. Multiple security domains may be involved, including:

- the security domain where the initiator resides;
- the security domain where the verifier resides;
- the security domains in which trusted third parties reside.

These domains need not all be distinct.

Before authentication can take place between different security domains, it is necessary to establish a secure interaction policy.

5.3 Principles used in authentication

In general, a particular authentication method will rely upon a chain of assumptions or expectations related to one or more principles.

The principles used include:

a) something known, e.g. a password;

b) something possessed, e.g. a magnetic card or a smart card;

c) some immutable characteristic, e.g. biometric identifiers;

d) accepting that a third entity (trusted third party) has established authentication; and

e) context, e.g. address of principal.

It should be noted that there are inherent weaknesses in all of the principles. For example, the authentication of something possessed is often the authentication of the possessed object rather than the authentication of its holder. In some cases the weaknesses may be overcome by the combination of several principles. For instance, when a smart card (something possessed) is used, the weakness may be overcome by the addition of a PIN (something known) in order to authenticate the user to the card. Moreover, principle e) is particularly weak and is virtually always used in conjunction with another principle.

Note that in d) there are two types of recursion:

- in order to be identified the third entity might itself require to be authenticated; and
- the authentication that the third entity establishes may use a fourth entity, etc.

Analysis of real authentication methods incorporating these principles will indicate the entities that are involved, the principles that are used, and the principals that are authenticated.

5.4 Phases of authentication

The following phases may occur in authentication:

- installation phase;
- change-authentication-information phase;
- distribution phase;
- acquisition phase;

- transfer phase;
- verification phase;
- disable phase;
- re-enable phase;
- de-installation phase.

The phases described here are not necessarily distinct in time, i.e. they may overlap.

Not all of these phases are required for a given authentication scheme. Also, in some cases, the sequencing of the phases may be different from the sequence implied by the following description.

5.4.1 Installation

In the installation phase, the claim AI and the verification AI are defined.

5.4.2 Change-authentication-information

In the change-authentication-information phase, a principal or a manager causes claim AI and verification AI to change (e.g. a password is changed).

5.4.3 Distribution

In the distribution phase verification AI is distributed to an entity (e.g. a claimant or a verifier) for use in verifying exchange AI. For example, in off-line approaches, entities may obtain authentication certificates, certificate revocation lists and authority revocation lists. The distribution phase may occur before, during or after the transfer phase.

5.4.4 Acquisition

In the acquisition phase a claimant or verifier may obtain information required to generate specific exchange AI for an instance of authentication. Different procedures may acquire exchange AI by interaction with a trusted third party or by message exchange between authenticating entities.

For example, when using an on-line key distribution centre, the claimant or verifier may obtain some information, such as an authentication certificate (see 6.1.3), from the key distribution centre to enable authentication with the other entity.

5.4.5 Transfer

In the transfer phase, exchange AI is transferred between claimant and verifier.

5.4.6 Verification

In the verification phase, the exchange AI is checked against the verification AI. In this phase, an entity which is unable to verify the exchange AI itself may contact a trusted third party which will perform the verification of the exchange AI. In this case, the trusted third party will send back a positive or negative response.

5.4.7 Disable

In the disable phase, a state is established whereby a principal that previously could be authenticated is temporarily unable to be authenticated.

5.4.8 Re-enable

In the re-enable phase, the state established in a disable phase is terminated.

5.4.9 De-installation

In the de-installation phase, a principal is removed from the population of principals.

5.5 Trusted Third Party Involvement

Authentication mechanisms can be characterized by the number of trusted third parties involved.

5.5.1 Authentication without Trusted Third Party Involvement

In the simplest situation neither the claimant nor the verifier is supported by any other entity in the generation and verification of exchange AI. In this case, the verification AI for the principal must be already installed in the verifier.

Unless most entities are restricted to a small number of possible communication partners, such an approach is of limited use in large-scale communications environments. In the worst case each verifier is required to have verification AI for all principals in a security domain, with the total information requirement growing as the square of the number of entities involved (see Figure 2).

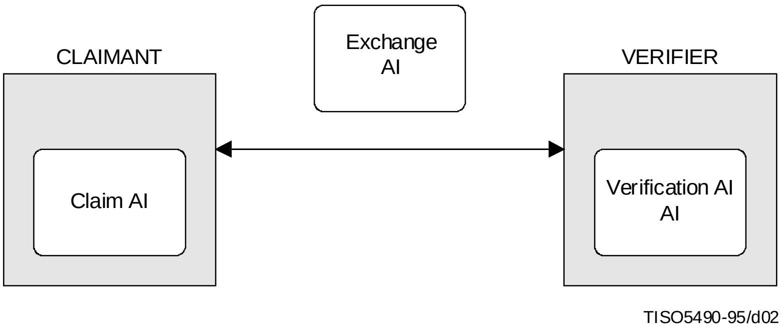


Figure 2 – Authentication without a Trusted Third Party

5.5.2 Authentication with Trusted Third Party Involvement

Verification AI can be obtained by interacting with trusted third parties. Integrity of this information must be guaranteed. It is also necessary to maintain the confidentiality of the trusted third party's claim AI, and of the verification AI if claim AI can be deduced from it.

Authentication may involve one trusted third party or a chain of trusted third parties, as covered by principle d) in 5.3. Introduction of additional trusted third parties affords authentication among a large population of entities with each maintaining information about only a limited number of entities (not about all other entities). Thus, the total information can grow linearly with the number of entities involved.

Multi-entity relationships may be characterized according to communications requirements (number of active links involved) and according to the degree of management control they have, e.g. the delay inherent in revoking authentication information.

5.5.2.1 In-line

In the case of in-line authentication, a trusted third party (an intermediary) intervenes directly in an authentication exchange between the claimant and the verifier. A principal is authenticated by the intermediary who then vouches for the identity in a subsequent in-line authentication exchange.

In-line authentication requires that the verifier trusts the intermediary to have properly authenticated the principal, and that the verifier is assured of the identity of the intermediary through authentication.

Revocation of the ability to authenticate can be controlled to the granularity of the next authentication attempt. Should the claimant have its authentication information revoked, the intermediary can immediately update the claimant's status and reject any future authentication attempts.

Occasionally, this can be extended so that a guarantee involving a chain of trusted intermediaries may be received. Depending on the security policy in force, either the verifier or the last TTP in the chain has the responsibility to determine whether or not the chain of intermediaries is valid.

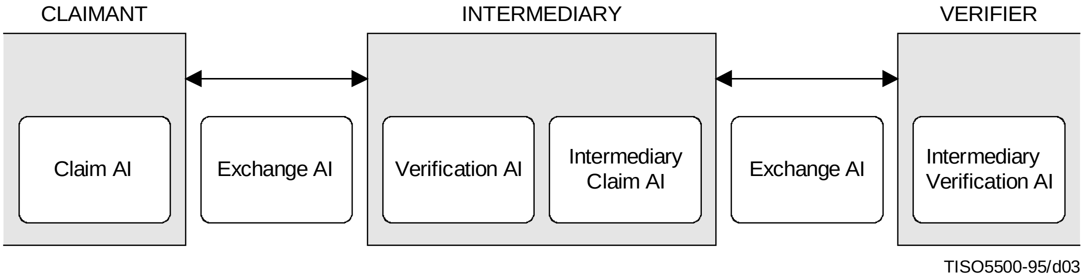


Figure 3 – In-line authentication

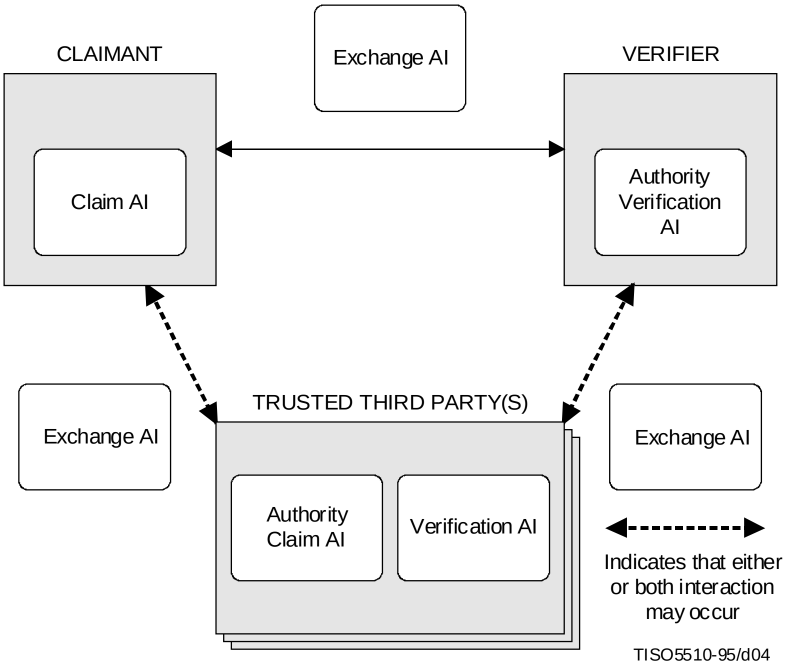
5.5.2.2 On-line

In the case of on-line authentication, one or more trusted third parties are actively involved in every instance of an authentication exchange. However, unlike in-line authentication, the on-line trusted third parties do not lie directly in the path of the authentication exchange between the claimant and the verifier. On-line trusted third parties can be requested by a claimant to generate exchange AI and can assist the verifier in its verification of exchange AI. An on-line trusted third party can generate on-line authentication certificates (see 6.1.3).

On-line authentication requires that there is a chain of trusted third parties involved in the generation of exchange AI, between the verifier and the trusted third party that can validate the principal's claim AI. In the simplest case only one trusted third party needs to interact directly with the claimant or the verifier. This, however, can be extended to a chain of trusted third parties communicating directly or indirectly with the claimant or verifier.

Revocation of the ability to authenticate can be controlled to the granularity of the next authentication attempt.

Examples of on-line trusted third parties are on-line authentication servers or key distribution centres.



NOTE – The actual exchange AI occuring between the three different entities shown in this figure is not the same.

Figure 4 – On-line authentication

5.5.2.3 Off-line

Off-line authentication is characterized by the need to use certified lists of revoked certificates, certificate lists of revoked certificates, certificate timeouts, or other non-immediate methods for the revocation of verification AI.

In the case of off-line authentication, one or more trusted third parties support authentication without being involved in each instance of authentication. The off-line trusted third party generates and distributes, in advance, off-line authentication certificates which the verifier can later use to validate an authentication exchange. The authentication exchange thus proceeds autonomously, without intervention of the authority.

Since the trusted third parties need not be able to interact directly with the claimant or the verifier at the time authentication takes place, this approach can be more efficient in terms of the number of interactions required.

Revocation must rely on additional provisions such as expiry and renewal of certificates, and certified lists of revoked certificates.

Examples of off-line trusted third parties are certification authorities which issue off-line authentication certificates (see 6.1.3).

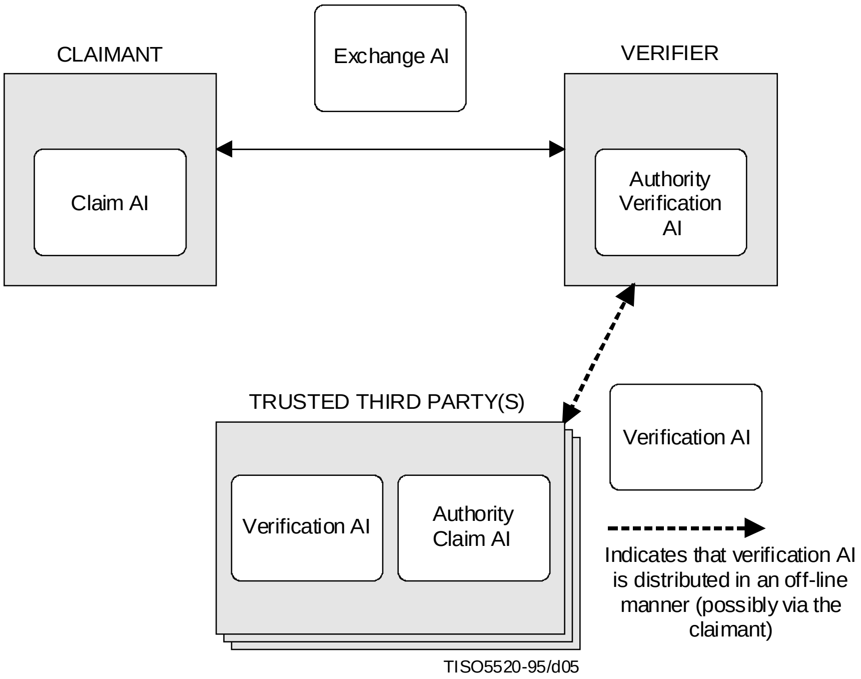

CLAIMANT
Exchange AI
VERIFIER
Claim AI
Authority Verification AI
TRUSTED THIRD PARTY(S)
Verification AI
Verification AI
Authority Claim AI
Indicates that verification AI is distributed in an off-line manner (possibly via the claimant)
TISO5520-95/d05

Figure 5 – Off-line authentication

5.5.3 Claimant trust in a verifier

Mechanisms in which it is necessary to trust a verifier are inadequate, unless all possible verifiers can be trusted. This is because, if the identity of the verifier has not been authenticated, its trustworthiness is unknown. For example, with the simple use of passwords for authentication, it is necessary to trust a verifier not to keep and re-use a password.

5.6 Types of principal

Principals can be categorized in various ways, such as:

a) those with passive characteristic(s), e.g. fingerprint, retinal characteristics;

b) those with information exchange and processing capability;

c) those with information storage capability; and

d) those with a unique fixed location.

Principals may fit more than one category [for example, human entities fit a), b) and c)]. A different method of authentication applies in each case:

a) measurement of the passive characteristic(s);

b) complex challenge and response evaluation;

c) memory of a secret (such as a password);

d) determination of position.

5.7 Human user authentication

In an instance of authentication, it may be necessary to authenticate the ultimate human user, rather than a process acting on behalf of the human user.

Methods for authentication of human users must be acceptable to human users as well as being economical and safe. Unacceptable methods may encourage human users to find ways to avoid the procedures, so that the potential for intrusion increases.

Approaches to human user authentication are based on principles described in 5.3. Procedures for human user authentication are based on the phases described in 5.4.

Annex A provides further information on human user authentication and on processes acting on behalf of a human user.

5.8 Types of attack on authentication

Three forms of attack are considered:

– *replay attacks*, in which exchange AI is read and later replayed;

– *relay attacks*, initiated by an intruder; and

– *relay attacks*, in which an intruder responds.

A relay attack is one in which exchange AI is intercepted and then immediately forwarded.

5.8.1 Replay attacks

There are two cases of replay to be considered. These are the replay of some exchange AI:

– on the same verifier; or

– on another verifier.

The latter case is possible when the (same) verification AI of a principal is known by several verifiers. When a successful replay may be achieved, this is a specific case of masquerade.

Both cases of replay can be countered using challenges. Challenges are generated by the verifier. The same challenge must never be issued twice by the same verifier. This can be achieved in several ways (see Annex C).

5.8.1.1 Replay on the same verifier

Replay on the same verifier may be countered using unique numbers or challenges.

Unique numbers are generated by the claimant. The same unique number must never be accepted twice by the same verifier. This can be achieved in several ways (see Annex C).

5.8.1.2 Replay on a different verifier

Replay on a different verifier may be countered using challenges. Alternatively it may be countered using, in the computation of the exchange AI, any characteristic which is unique to the verifier. Such characteristic may be the name of the verifier, its network address or in general any attribute unique with respect to the verifiers sharing the same verification authentication information.

5.8.2 Relay attacks

5.8.2.1 Intruder initiated relay attacks

This type of attack involves the intruder as the initiator of the authentication. This attack is possible only if both claimant and verifier can initiate the authentication. With this attack the claimant and verifier exchange authentication information via an intruder without being aware of this, i.e. the intruder pretends to be a certain verifier towards a claimant and to be this claimant towards that verifier.

For example, suppose intruder C wants to pretend to verifier B that he is claimant A. C starts an interaction with both A and B. C tells A that he is B, asks A to authenticate towards B, and also tells B that he is A and that he wants to authenticate himself (see Figure 6).

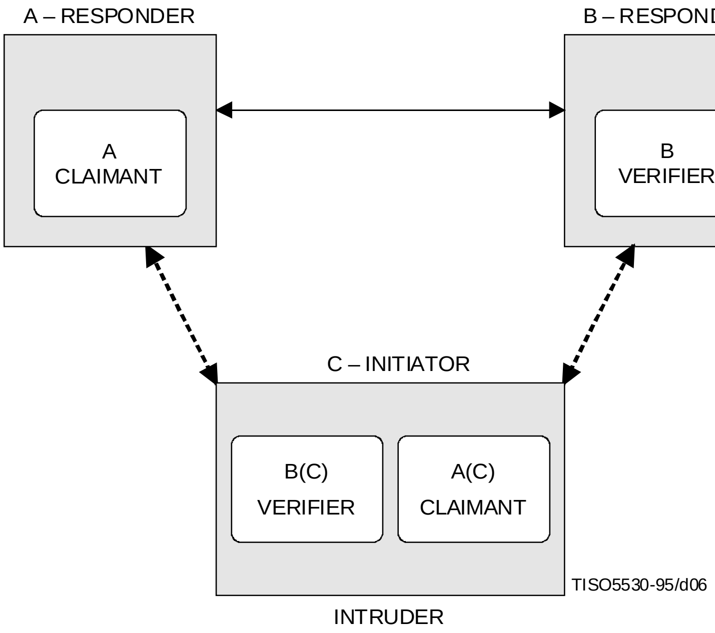


Figure 6 – Intruder initiated relay attack

During the process of authentication, A acts as a claimant towards B (actually to C acting as B) and, therefore, supplies information that C can use to authenticate to B. B acts as the verifier and also supplies the information that C needs to play the role of verifier. Following the authentication, intruder C will appear to B as the authenticated A.

This type of attack may be countered if:

a) the entity that starts an interaction is either always the claimant or always the verifier (note this may not be possible when mutual authentication is used); or

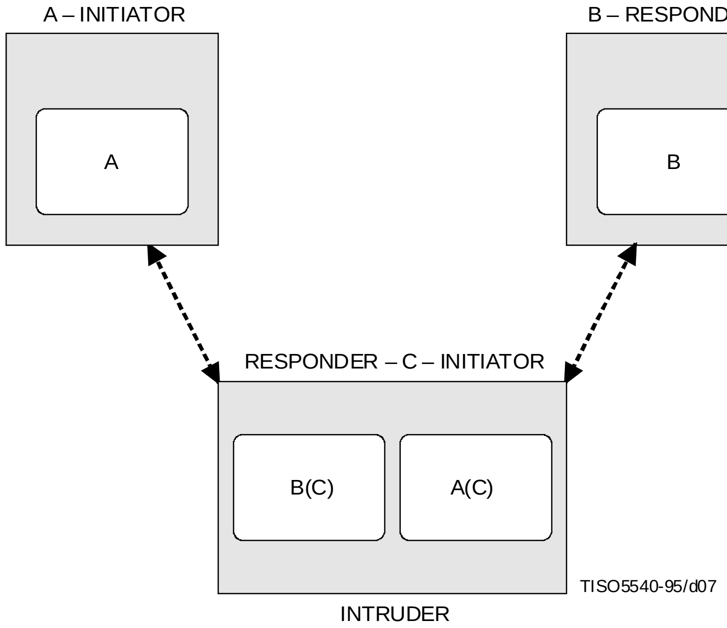
b) the exchange AI provided by the claimant differs according to his role as initiator of an authentication request or responder to an authentication invitation. This difference allows the verifier to detect the interception described. See Annex D for further details.

5.8.2.2 Relay attacks in which an intruder responds

In this type of attack the intruder sits in the middle of an authentication exchange, intercepts the authentication information and forwards it, taking over the role of initiator. This type of attack may occur either opportunistically, in which case the intruder(s) waits to be mistaken as the responder; or, systematically, in which case the intruder advertises itself as the responder (for example, in a central resource location table).

The general way to counter this type of attack requires the use of a complementary service (integrity or confidentiality) for further data exchanges. The exchange AI is combined with some other information that enables the claimant and verifier, provided they are the legitimate parties, to derive a key. The derived key can then be used as a key for a cryptographic-based integrity or confidentiality mechanism.

Another way to counter this type of attack is relevant where the communications network is not subject to interceptions internally, i.e. it always delivers data unchanged to the correct address. In this situation, the attack can be countered by integrating the network addresses into the exchange AI (e.g. signing the network address).



NOTES

1 Even if intruder-initiated attacks are countered using methods a) or b) from 5.8.2.1, an authentication method will still be vulnerable to intruder responding attack.

2 The notation X(Y) indicates that Y is pretending to be X.

Figure 7 – Relay attacks in which an intruder responds

6 Authentication information and facilities

6.1 Authentication information

6.1.1 Claim authentication information

Claim AI is information used to generate exchange AI needed to authenticate the principal.

Examples of claim AI are:

a) *Password*.

b) *Secret key* – This is for use with authentication mechanisms using symmetric algorithms.

c) *Private key* – This is for use with authentication mechanisms using asymmetric algorithms.

6.1.2 Verification authentication information

Verification AI is information that is used to verify an identity claimed through exchange AI.

Examples of verification AI are:

a) *Password* – Related to the identity of a principal.

b) *Secret key* – Related to the identity of a principal or authority. This is for use with authentication mechanisms using symmetric algorithms.

c) *Public key* – Related to the identity of a principal or authority. This is for use with authentication mechanisms using asymmetric algorithms.

Verification AI can be provided in the form of an authentication table and/or an off-line authentication certificate (see 6.1.4.2).

An authentication table is a set of entries which is directly accessible by the verifier. The path used to access the table has integrity protection, and additionally, for symmetric mechanisms, confidentiality protection.

Examples of elements which may be contained within an authentication table entry are:

- the identity of the principal;
- verification AI, for example, a password, a secret key, or a public key;
- validity period for the entry;
- security policy applicable to the entry;
- authority responsible for the entry.

6.1.3 Exchange authentication information

Exchange AI is information exchanged between a claimant and a verifier during the process of authenticating a principal. Examples of exchange authentication information are:

- claimed distinguishing identifier;
- password;
- challenge;
- response to challenge;
- unique number;
- verifier distinguishing identifier;
- the result of a transformation function applied to or using claim AI and other data (e.g. time stamp, random number, counter, challenge, identity of verifier, digital fingerprint, identity of claimant); example transformation functions are a one-way function, asymmetric encipherment function, and symmetric encipherment function;
- on-line certificate;
- off-line certificate.

Some or all of the exchange AI conveyed in one transfer may be in the form of a security token.

6.1.4 Authentication certificates

A common form of authentication information is an authentication certificate. An authentication certificate is a particular type of security certificate, certified by a trusted authority, which may be used for authentication.

Different types of authentication certificate are:

- on-line authentication certificates;
- off-line authentication certificates;
- authentication revocation certificates; and
- revocation authentication certificate lists.

Off-line certificates (see 6.1.4.2) are primarily applicable to public key related verification AI. The validity of an off-line certificate may be revoked through use of either a revocation certificate or a revocation certificate list.

Examples of elements which may be contained within any authentication certificate are:

- Identification of the method and/or key which has been used to generate a cryptographic checkvalue.
- The identity of the authentication authority and the identity of the agent that has issued the authentication certificate (when an authority is represented by several agents, the identity of the agent allows precise knowledge of which agent’s key has been used).

- The creation time of the authentication certificate (the creation time may be used for audit purposes or may be used when the validity period of the authentication certificate is not present; after a time period which is security policy dependent, excessively old authentication certificates may be rejected).
- The validity period (not before, not after) of the authentication certificate (this time period may be considered if the security policy of the receiver allows for it, otherwise the expiration time will be derived from the creation time according to the security policy of the receiver).
- The security policy applicable for the authentication certificate.
- A certificate reference number which is unique to this authentication certificate, with respect to all authentication certificates of the same authority agent.
- Type of certificate.
- The identity or attributes of the verifier for which the authentication certificate is intended (entities may check this value if present and incorrect ones may be rejected. Identities/attributes may be, for example, human user names, application processes and/or physical machine identities).

Additional elements for the different types of authentication certificate are identified in the following subclauses.

Profiles may be defined in application standards to specify which elements are mandatory and which elements are optional.

6.1.4.1 On-line authentication certificates

An on-line authentication certificate is created by a trusted third party by direct request of a claimant. An on-line authentication certificate is usually passed to the verifier as part of the exchange AI.

Examples of additional elements that may be contained within an on-line authentication certificate are:

- Distinguishing identifier of the principal.
- Digital fingerprint of the data, when data origin authentication is used.
- A symmetric key assigned to the principal for authentication, together with identification of the algorithm to be used in conjunction with this key. It will be necessary to maintain the confidentiality of this information.
- The authentication method used to obtain this authentication certificate.
- The authentication method(s) with which this authentication certificate may be used.
- Identification of the method which must be used to protect the authentication certificate while in transit and any associated parameters necessary to achieve this protection. (Examples of such protection parameters are a challenge, a unique number or a protection key.)

6.1.4.2 Off-line authentication certificates

An off-line authentication certificate binds an identity to a cryptographic key. It is created by an authority, without either the claimant or verifier needing to interact directly with the authority. Off-line authentication certificates are commonly applied to authentication mechanisms using asymmetric algorithms. An off-line authentication certificate may be passed to the verifier as part of the exchange AI.

Examples of additional elements that may be contained within an off-line authentication certificate are:

- distinguishing identifier of the principal;
- a public key assigned to the principal by the authentication authority, together with identification of the algorithm to be used in conjunction with this public key.

An off-line authentication certificate may be revoked before the end of its validity period through use of either a revocation certificate or a revocation certificate lists.

6.1.4.3 Revocation certificates

A revocation certificate is a security certificate issued by a security authority to indicate that a particular off-line authentication certificate has been revoked. This information is stored and referred to whenever a certificate is presented to determine whether the presented authentication certificate is still valid.

Examples of additional elements that may be contained within a revocation certificate are:

- the identity of a principal, group of principals, or authority;
- the time and date when the off-line authentication certificate was revoked.
- the reference number of the revoked certificate.

6.1.4.4 Revocation certificate lists

A revocation certificate list is a certified list of all the authentication certificates revoked by a given security authority together with the time and date of issue of the list. This information is stored and referred to whenever a certificate is presented to determine whether the presented authentication certificate is still valid.

A revocation certificate list may comprise the following:

- revocation certificates;
- reference identifiers of revocation certificates;
- the authentication certificates revoked;
- reference identifiers of the revoked authentication certificates;
- the date the list was issued;
- the date when the next list will be issued.

6.1.4.5 Chains of certificates

Authentication certificates are always protected to provide data origin authentication from a trusted third party. If the verifier does not hold the verification AI to check the origin of the certificate, then a chain of certificates may be used. A certificate, originating from another authority, certifies the verification AI used to validate the origin of the first certificate.

A chain of certificates can be used recursively, each one certifying verification AI used to validate the origin of the previous certificate. The chain provides a *certification path* of authorities from the verifier to the claimant. The verifier has to make its own decision whether or not each certificate of the chain may be trusted, based on information it holds or can obtain from a trusted third party.

6.2 Facilities

This clause provides a general model of authentication in terms of generic facilities.

6.2.1 Authentication state information

Authentication state information represents the authentication state retained between invocations of the authentication services. Authentication state information can include:

- session cryptographic keys;
- message sequence numbers.

Authentication state information needs to be stored securely. This information is held by providers of these services.

6.2.2 Management related services

Authentication management-related facilities may involve distribution of descriptive information, passwords or keys (using key management) to entities required to perform authentication. It may also involve use of a protocol between communicating entities and other entities providing authentication services. Authentication management may also involve the revocation of authentication information.

6.2.2.1 Install

The Install facility installs claim AI and verification AI. This facility may be further refined in terms of Enrol, Validate, and Confirm facility.

6.2.2.1.1 Enrol

The Enrol facility causes a security authority to record some verification authentication information associated with a principal. This information includes a distinguishing identifier which is provided either by the principal or by the security authority. The facility is invoked by the principal, by another entity, or by a security authority. (The recording security authority may require the principal to provide assurance in support of validation of enrolment.) At this point in time, the principal is a candidate to enter a security domain but is not yet recognized as a member of the security domain. No authentication exchange is possible at this point of time.

6.2.2.1.2 Validate

The Validate facility, performed on behalf of the security domain authority, introduces a principal into a security domain.

The validation of the verification AI associated with a principal may involve communication between the security authority and another entity, which is not necessarily performed using OSI communications. The Validate facility causes a distinguishing identifier to be bound to verification AI.

6.2.2.1.3 Confirm

The Confirm facility is invoked after the Validate facility. It returns specific information to the principal or other entities. The simplest form of information returned is an acknowledgement or a rejection for the installation. Other forms are:

- an off-line authentication certificate;
- the accepted distinguishing identifier; or
- claim AI.

Following confirmation, the principal can be authenticated.

6.2.2.2 Change-AI

The Change-AI facility is invoked on behalf of a principal or a manager, to cause authentication information to change.

6.2.2.3 Distribute

The Distribute facility enables any entity to obtain sufficient verification AI upon which to verify exchange AI.

6.2.2.4 Disable

The Disable facility, which is invoked on behalf of a security authority, causes a state to be established whereby a principal is temporarily unable to be authenticated.

6.2.2.5 Re-enable

The Re-enable facility, which is invoked on behalf of a security authority, causes the state established by the Disable service to be terminated.

6.2.2.6 De-install

The De-install facility causes a principal to be removed from a population of authenticatable principals. This facility may be further refined in terms of Invalidate, Notify and Unenrol facilities.

6.2.2.6.1 Invalidate

The Invalidate facility is an action performed by a security authority which consists of the revocation of the verification AI and/or a change to the status information associated with a principal. The Invalidate facility prevents a principal from authenticating.

6.2.2.6.2 Notify

The Notify facility may be invoked by the security authority after the Invalidate facility. It returns a notification to the principal of its invalidation and possibly information on how to re-enrol.

6.2.2.6.3 Unenroll

The Unenroll facility causes a principal to be suppressed from a security domain. It corresponds to the removal of the principal’s identity and associated verification AI. The facility is invoked by a security authority.

6.2.3 Operational related facilities

6.2.3.1 Acquire

The Acquire facility allows a claimant or verifier to obtain information required to generate specific exchange AI for an instance of authentication. This may require interacting with a trusted third party (e.g. an authentication server).

Candidate inputs are:

- authentication exchange type;
- distinguishing identifier of principal;
- identity of the verifier;
- type of claim AI (e.g. password, key);
- claim AI (e.g. password value);
- type of exchange AI;
- validity (start/expiry times).

Candidate outputs are:

- status (success or failure);
- information required to generate exchange AI;
- validity (start/expiry times).

6.2.3.2 Generate

The Generate facility is invoked by a claimant to generate exchange AI, and/or to process received exchange AI.

Candidate inputs are:

- authentication exchange type;
- distinguishing identifier of principal;
- information required to generate exchange AI as output from an Acquire;
- reference to retained authentication state information;
- exchange AI received from the verifier;
- type of exchange AI;
- identity of the verifier;
- claim AI.

Candidate outputs are:

- status (success, further transfers required, or failure);
- reference to retained authentication state information;
- exchange AI for transfer to verifier.

The authentication exchange type may be provided as an input, on the first invocation of the Generate facility in an authentication exchange, when the claimant is the authentication initiator. On the same invocation, a reference to retained authentication state information is returned as an output. In subsequent invocations of the Generate facility for the same authentication exchange, this input and output need not be present, but the reference to retained authentication state information may be provided as an input.

The authentication state information is retained within the facility for later use for authentication until success or failure is returned.

If “further transfers required” is returned, the claimant will need to invoke the Generate facility following receipt of exchange AI from the other entity. The claimant may be required to perform many such operations (i.e. invoking the Generate facility with the previous authentication state information and received exchange AI) until success or failure is indicated. In this manner, this facility accommodates many schemes including n-way challenge-response exchanges, as well as the exchanges required by certain zero knowledge schemes.

6.2.3.3 Verify

A verifier invokes the Verify facility to verify received exchange AI from the claimant, and/or to generate exchange AI for transfer to the claimant.

Candidate inputs are:

- authentication exchange type;
- information required to generate exchange AI as output from an Acquire;
- reference to retained authentication state information;
- exchange AI received from the claimant;
- verification AI.

Candidate outputs are:

- status (success, further transfers required, or failure);
- reference to retained authentication state information;
- exchange AI for transfer to claimant (if status is "further transfers required");
- distinguishing identifier of principal (if status is "success");
- validity (start/expiry times);
- mutual authentication indicator.

The authentication exchange type may be provided as an input, on the first invocation of the Verify facility in an authentication exchange, when the verifier is the authentication initiator. On the same invocation, a reference to retained authentication state information is returned as an output. In subsequent invocations of the Verify facility for the same authentication exchange, this input and output need not be present, but the reference to retained authentication state information may be provided as an input.

The authentication state information is retained within the facility for later use for authentication until success or failure is returned.

If a status of "success" is returned, the authenticated identity of the principal is also returned.

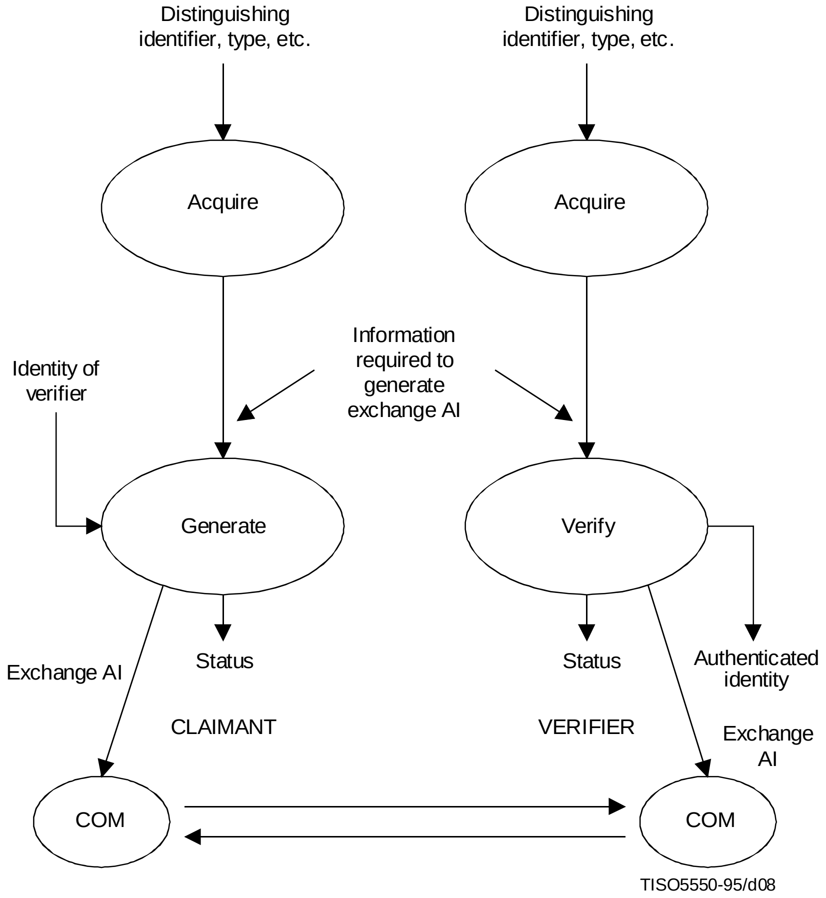
6.2.3.4 Generate and Verify

In the case of mutual authentication, the Generate and Verify facilities may be merged into a single facility. The candidate inputs and outputs are the union of the inputs and outputs of the two facilities.

NOTE – The Generate facility and the Verify facility do not transfer any data. The transfer of data is dependent upon the environment where authentication is used. This is outside the scope of this Recommendation | International Standard.

6.2.3.5 Example of information flows

Figure 8 gives an example of the information flows associated with the invocation of Acquire, Generate, and Verify facilities, as used to model the provision of authentication (e.g. to application processes).



NOTE – In this example, the Acquire service is shown to be invoked by both claimant and verifier. In practice it will usually be invoked by only one of these entities, or not invoked at all. Although Information flows between Generate and Verify, neither of these services are intended to invoke communications primitives.

Figure 8 – Example of information flows in operational related services

7 Characteristics of authentication mechanisms

Authentication mechanisms in the scope of this Recommendation | International Standard can be based on principles a), d), and e) of 5.3. Principle d) involves the use of a trusted third party as described in 5.5.2, but these mechanisms will ultimately rely upon principles a) or e). Otherwise, in open systems, authentication of remote principals is often based on principle a), in which secrets in the form of a key or password are used.

7.1 Symmetry/Asymmetry

Authentication of remote principals is often based on secrets which take the form of a password or key. Authentication involves demonstrating knowledge of the secret. Methods for such demonstration fall into two broad categories:

- *symmetric*, in which both entities share common authentication information; and
- *asymmetric*, in which not all authentication information is shared by both entities.

Examples of symmetric authentication methods are:

- password; and
- a challenge enciphered using a symmetric key technique.

Examples of asymmetric authentication methods are:

- asymmetric key techniques; and
- techniques by which possession of information can be verified without any part of that information being revealed.

7.2 Use of cryptographic/Non-cryptographic techniques

Authentication mechanisms based on something known (see 5.3) may be further characterized by their use of cryptographic algorithms to protect authentication information. Symmetric, asymmetric or a hybrid of cryptographic techniques may be used to provide integrity and, in some cases, confidentiality protection of authentication information.

Non-cryptographic techniques include use of passwords or challenge-and-response tables. Examples of cryptographic techniques include the use of encipherment to protect passwords during transmission.

7.3 Types of authentication

Entity authentication involves two entities. In unilateral authentication, one entity acts as a claimant and the other acts as a verifier. For mutual authentication, each entity is acting at the same time as a claimant and a verifier. Mutual authentication can be obtained using the same or a different authentication mechanism in either direction.

7.3.1 Unilateral authentication

Unilateral authentication may be obtained using either:

- a single transfer of authentication information, for example, when unique numbers are used;
- three transfers of authentication information when challenges are used; or
- more than three transfers of authentication information. This case is applicable to some specific mechanisms using zero knowledge techniques.

The above cases assume the claimant is the authentication initiator. If the verifier is the authentication initiator, the number of transfers is different; for details see 8.2.

7.3.2 Mutual authentication

Mutual authentication does not necessarily imply the doubling of the number of transfers, nor does it imply using the same authentication mechanism in both directions.

For authentication mechanisms using three transfers of authentication information for unilateral authentication, mutual authentication does not require any further exchange; the request for a challenge may be combined with the sending of another challenge used by the verifier (then acting as a claimant) to authenticate to the claimant (then acting in a verifier role).

7.3.3 Authentication acknowledgement

In some cases, an acknowledgement of the fact that an entity authentication has been accepted or refused is useful. This acknowledgement may be guaranteed or simply be a yes or no answer without any guarantee. This will require an additional transfer.

8 Authentication mechanisms

8.1 Classification by vulnerabilities

Authentication mechanisms themselves may be vulnerable to attacks, which limit their effectiveness (see 5.8).

In this subclause, authentication mechanisms which can be employed to support authentication in the transfer phase are classified according to the threat(s) against which they are resistant. The mechanisms described are all based on the authentication principle of “something known” [see 5.3 a)].

All mechanisms described are applicable to entity authentication, and some are also applicable to data origin authentication, for example, a digital fingerprint of the data in the authentication exchange.

The following classes of authentication mechanism are defined:

- Class 0: Unprotected;
- Class 1: Protected against disclosure;
- Class 2: Protected against disclosure and replay on different verifiers;
- Class 3: Protected against disclosure and replay on the same verifier;
- Class 4: Protected against disclosure and replay on the same verifier or different verifiers.

NOTE – In classes 1 to 4, "protected against disclosure" means protected against of disclosure of claim AI.

Additional classes may be defined as appropriate. Sub-classes are identified for some classes of mechanism. The sub-classes are not necessarily exhaustive.

The exchange AI for each of the mechanism classes is as shown on the diagrams.

Where an encipherment function is used as part of the Generate facility, the claim AI, possibly along with other information, is used to form the key. Where a decipherment function is used as part of the Verify facility, verification AI, possibly along with other information received in the authentication exchange, is used to form the key.

The following authentication exchanges are described from the perspective of the claimant and always initiated by the claimant. For exchanges initiated by the verifier, see 8.2. The exchanges described are applicable to unilateral authentication. For exchanges applicable to mutual authentication, see 8.4. In some cases, an acknowledgement of the fact that authentication was successful or unsuccessful is needed. An additional transfer of data may be necessary for this. This is not described in this clause. The facilities referred to in this clause are as defined in 6.2.

In the diagrams below, the notation of a bracket pair [ ] indicates an optional component of the information transferred, included only under some conditions.

The optional component [digital fingerprint] is present in the case of data origin authentication, and absent otherwise. A digital fingerprint can be achieved, for example, using an asymmetrical encipherment algorithm, either simply by enciphering the data or by providing a cryptographic check value of the data using the signer's private key. For data origin authentication, the transfer of the data to which the digital fingerprint refers may occur completely independently of, or may share the use of, the means of communication used for the following mechanisms.

8.1.1 Class 0 (Unprotected)

In Class 0, the claim AI is simply sent, along with the distinguishing identifier, as claimant-to-verifier exchange AI. A prime example is sending a password. Class 0 is a case of symmetric authentication. This class of mechanisms is vulnerable to disclosure of authentication information and replay attack.

The Generate facility produces exchange AI, as shown in Figure 9, directly from its inputs.

The Verify facility verifies that the received claim AI (e.g. a password) matches the verification AI associated with the received distinguishing identifier.

Class 0 mechanisms are applicable to both data origin and entity authentication.

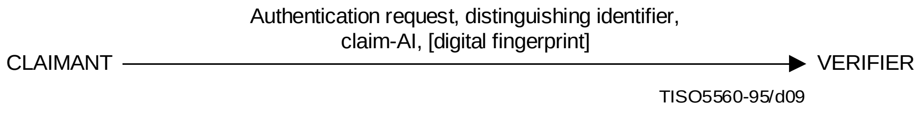

Authentication request, distinguishing identifier,
claim-AI, [digital fingerprint]
CLAIMANT
VERIFIER
TISO5560-95/d09

Figure 9 – Class 0 mechanism (Unprotected)

8.1.2 Class 1 (Protected against disclosure)

This class of mechanisms provides protection against disclosure of claim AI. Class 1 mechanisms are applicable to both data origin and entity authentication.

These mechanisms employ a transformation function in which the claim AI, possibly combined with the distinguishing identifier, is transformed under the function and transferred along with the distinguishing identifier. The actual claim AI is not transmitted over the communications channel. Examples include:

- sending a password transformed under a one-way function (e.g. a cryptographic checkvalue or hash function);
- sending a digital fingerprint enciphered under a secret key;
- sending a password encrypted under a confidentiality key; and
- sending a digital fingerprint signed using a private key.

Mechanisms of this type are applicable to both data origin and entity authentication. They are vulnerable to replay attacks but provide protection against disclosure of claim AI. For example, the transformed password may be replayed at the protocol exchange level but the cleartext password, which is usable at the system interface level, is not disclosed.

The Generate facility uses the claim AI and, if required, the distinguishing identifier and/or a digital fingerprint as inputs to a cryptographic transformation to generate the exchange AI as shown in Figure 10.

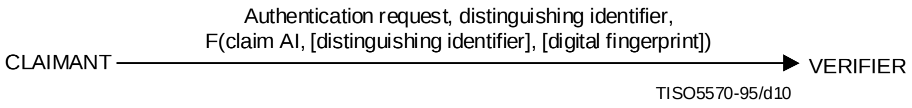

Authentication request, distinguishing identifier,
F(claim AI, [distinguishing identifier], [digital fingerprint])
CLAIMANT
VERIFIER
TISO5570-95/d10

Figure 10 – Class 1 – Mechanism protected against disclosure

Three examples of transformation functions (F) are as follows:

a) In the case of a one-way function, the Verify facility repeats the one-way function using verification AI instead of claim AI, and matches this against the received exchange AI.

b) In the case of the use of symmetric encipherment, the Verify facility uses the verification AI to decipher the received exchange AI and then verifies the correctness of the decipherment by checking that it contains distinguishing features such as the distinguishing identifier of the claimant, the correct digital fingerprint, a password or a constant value.

c) In the case of a digital signature, the Verify facility re-calculates the digital fingerprint from the received data and uses the verification AI to check that the received signature is a valid signature for that fingerprint.

In addition, for data origin authentication, the digital fingerprint in the exchange AI is matched against a regenerated digital fingerprint of the data requiring authentication.

NOTE – Where the principal's distinguishing identifier is combined with the claim AI, this makes an exhaustive attack more difficult. Only an attack on a specific principal may be performed at one time instead of an attack over all the principals together.

In order to provide confidentiality, the transformation function must either have no inverse or, if it has, the inverse must be computationally intractable to parties from which the claim AI (and digital fingerprint) is to be kept confidential.

8.1.3 Class 2 (Protected against disclosure and replay on different verifiers)

This class of mechanisms provides protection against disclosure of claim AI and replay on different verifiers but not against replay on the same verifier. This class of mechanism is identical to Class 1, except that a data item containing a characteristic unique to the intended verifier is included as an input to the transformation function. This provides additional protection.

8.1.4 Class 3 (Protected against disclosure and replay on the same verifier)

This class of mechanisms provides protection against disclosure of claim AI and against replay on the same verifier.

Unique number mechanisms in this class use transformation functions in combination with unique information to provide protection against replay on one verifier. The claim AI and the unique number are transformed and transferred, along with the distinguishing identifier.

Three examples of sources for a unique number are:

a) *Random or pseudo-random number* – Such a number is not intentionally repeated within the lifetime of the claim AI. A random or pseudo-random number from a sufficiently large range can reduce the risk (probability) that the same number has already been used.

b) *Time stamps* – The unique number is a time stamp, obtained from a trusted source, which is unique within the lifetime of the claim AI. Old time stamps or time stamps which were previously used will be rejected.

c) *Counter* – The unique number is the value of a counter which is continuously incremented as long as the same claim AI is used.

d) *Cryptographic chaining* – The unique number is a value derived from the contents of previous data exchanges between claimant and verifier by block chaining.

The uniqueness of this number outside the claimant can be ensured by concatenating it with data unique to the claimant (such as its own distinguished identifier).

It is also possible to use combinations of these techniques to produce a unique number.

Three examples of transformation functions (F) are:

a) *One-way function* – The unique number, the claim AI and, optionally, the distinguishing identifier, are transformed under a one-way function. The unique number is also transmitted so that the verifier can perform the same transformation.

b) *Asymmetric algorithm* – When the claim AI is a private key, the unique number is signed under that private key.

c) *Symmetric algorithm* – When the claim AI is a secret key, the unique number is enciphered under that secret key.

This sub-class is applicable to both data origin and entity authentication.

The Generate facility generates a unique number. It then carries out encipherment using the following as inputs:

- unique number;
- claim AI;
- distinguishing identifier (optional);
- digital fingerprint (if data origin authentication),

and produces exchange AI as shown in Figure 11.

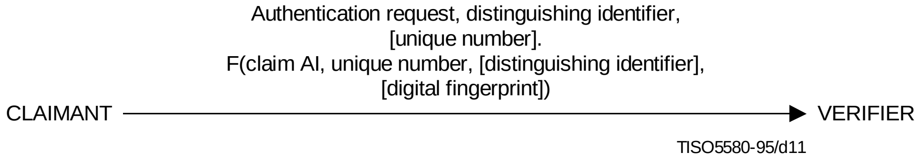

Authentication request, distinguishing identifier,
[unique number].
F(claim AI, unique number, [distinguishing identifier],
[digital fingerprint])
CLAIMANT
VERIFIER
TISO5580-95/d11

Figure 11 – Sub-class 3 – Unique number mechanism

The Verify facility deciphers and checks the exchange AI for validity using verification AI as described in Class 1. It also checks that the received unique number has not been received before. If the number has been received before, this indicates there has been a replay. In addition, for data origin authentication, the digital fingerprint in the exchange AI is matched against a regenerated digital fingerprint of the received data.

NOTE – The use of the term *cryptographic chaining* here corresponds to the definition of *block chaining* in ISO/IEC 10116.

8.1.5 Class 4 (Protected against disclosure and replay on the same verifier or different verifiers)

8.1.5.1 Sub-class 4a – Unique number mechanisms

This sub-class of mechanism is identical to class 3, except that a data item containing a characteristic unique to the intended verifier is included as an input to the transformation function in the exchange. This provides additional protection.

8.1.5.2 Sub-class 4b – Challenge mechanisms

The purpose of a challenge mechanism is to counter replay attacks, i.e. to ensure that any attempt to authenticate by replaying exchange AI will not succeed. In response to an authentication request, the verifier issues a challenge to the claimant in the form of a data item with a unique value. The claimant transforms the challenge information and the claim AI under some function, and returns the result of this transformation to the verifier.

Challenge mechanisms therefore involve a three-way transfer of information:

- sending an authentication request;
- issuing a challenge; and
- sending a response which contains a value obtained from the claim AI, possibly combined with the distinguishing identifier, and the challenge information, transformed under some appropriate function (F).

In the general case, the distinguishing identifier may be sent either with the authentication request or with the final response.

Three examples of transformation functions (F) used in challenge mechanisms are:

a) *One-way function* – The challenge and the claim AI are transformed under a one-way function.

b) *Asymmetric algorithm* – When the claim AI is a private key, the challenge is signed under that private key.

c) *Symmetric algorithm* – When the claim AI is a secret key, the challenge is enciphered under that secret key.

As a special case of a challenge mechanism, the challenge generated may depend upon the identity received in the authentication request. This is known as a dedicated challenge mechanism. In this case, the distinguishing identifier is mandatory with the authentication request. In addition, a fourth possible transformation function is:

d) *Non-cryptographic* – One example is to use a table of challenge-response pairs; the challenging entity requests a particular response. Another example is a biometric scheme, such as a voice repeat system.

This sub-class is applicable to both data origin and entity authentication.

The Generate facility produces an authentication request (which, in the dedicated challenge case, must be accompanied by a distinguishing identifier). On receipt of this authentication request the Verify facility generates a unique challenge as exchange AI.

The Generate facility then produces exchange AI as a transformation of the input data as shown in Figure 12.

In the case of a one-way function, the Verify facility repeats the transformation using verification AI instead of claim AI and checks this against the received exchange AI. In order to repeat this function, the verifier must have available the distinguishing identifier, and the data to which the service applies. In the case of other transformations, the Verify facility either repeats the transformation or does an inverse function and checks the contents using the verification AI.

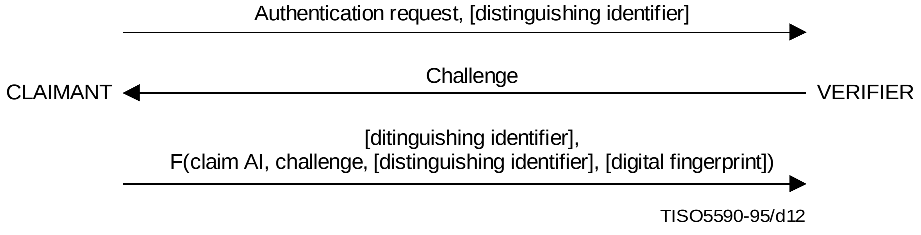

Authentication request, [distinguishing identifier]
CLAIMANT
Challenge
VERIFIER
[ditinguishing identifier],
F(claim AI, challenge, [distinguishing identifier], [digital fingerprint])
TISO5590-95/d12

Figure 12 – Sub-class 4b – Challenge mechanism

8.1.5.3 Sub-class 4c – Dedicated enciphered challenge mechanisms

Dedicated enciphered challenge mechanisms also involve a three-way transfer of information:

- sending an authentication request and distinguishing identifier;
- issuing a challenge and verification AI, possibly combined with the distinguishing identifier, transformed under some suitable function (F); and
- sending a response consisting of the challenge information.

Two examples of dedicated enciphered challenge mechanisms are:

a) *Asymmetric algorithm* – When the claim AI is a private key, the challenge is enciphered under the corresponding public key.

b) *Symmetric algorithm* – When the claim AI is a secret key, the challenge is enciphered under that secret key. The challenge is enciphered by the challenging entity.

This type of mechanism is applicable to entity authentication but not data origin authentication.

The Generate facility produces an authentication request. On receipt of the authentication request and a distinguishing identifier, the Verify facility generates an unpredictable challenge. This is then operated on by the transformation function to produce exchange AI as shown in Figure 13.

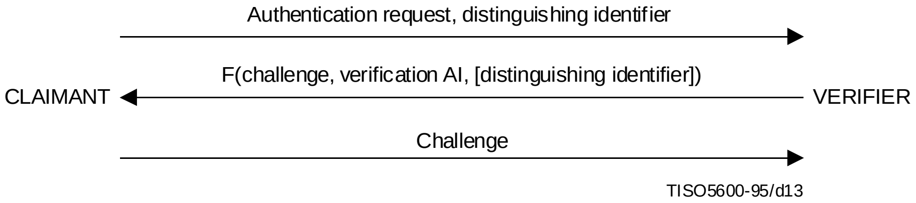

Authentication request, distinguishing identifier
F(challenge, verification AI, [distinguishing identifier])
CLAIMANT
VERIFIER
Challenge
TISO5600-95/d13

Figure 13 – Sub-class 4c – Dedicated enciphered challenge mechanism

The Generate facility then does the inverse transformation using claim AI instead of verification AI to obtain the challenge which is returned for use as exchange AI. Note that only encipherment transformations are relevant to this scheme.

The Verify facility finally checks the challenge against that generated earlier.

8.1.5.4 Sub-class 4d – Computed response mechanisms

This class of mechanism also involves a three-way transfer of information:

- sending an authentication request with a choice of values to be selected and identity information;
- issuing a challenge which indicates which values were selected by the verifier; and
- sending a response consisting of the unique number, the challenge, or the values selected to compute the response, and the claim AI, transformed under some appropriate function.

One example is a zero knowledge technique in which the verifier selects one from a set of “problems” which the claimant must solve without revealing exactly how.

The exchanges may be repeated to provide a higher level of assurance of the identity. This protects against masquerade attacks by an intruder who can compute the correct response for some, but not all, of the values that a verifier might select. If there is only one exchange, the verifier may, by chance, select a value for which the intruder knows the correct response. Increasing the number of exchanges decreases the probability of such an attack succeeding.

The Generate facility firstly generates a unique number and choice of values and then puts these in exchange AI as shown in Figure 14.

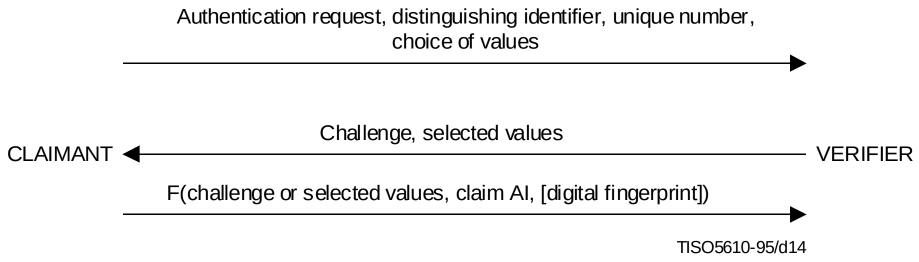

Authentication request, distinguishing identifier, unique number, choice of values
Challenge, selected values
CLAIMANT
VERIFIER
F(challenge or selected values, claim AI, [digital fingerprint])
TISO5610-95/d14

Figure 14 – Sub-class 4d – Computed response mechanism

The Verify facility then selects values from this choice and generates a challenge to form the second exchange AI.

The Generate facility carries out a transformation on the challenge or selected values using claim AI.

The Verify facility then finally carries out an inverse transformation using the verification AI and checks the received values.

8.2 Initiation of transfer

In 8.1, the exchanges are described with the claimant initiating the exchange using an *authentication request*. However, for entity authentication, the same sub-classes of mechanisms could involve the verifier initiating the exchange using an *authentication invitation*. In this case, the number of transfers will be different. Table 1 in 8.5 gives the number of transfers for each of these cases.

8.3 Use of authentication certificates

Authentication mechanisms may be classified in terms of the means used to acquire verification AI. Possible means include:

- on-line authentication certificates;
- off-line authentication certificates; and
- verification AI provided in advance, e.g. by using secure channels.

An authentication certificate can be used to provide a proof of authentication using the principle described in 5.3, d). The authentication certificate provides proof that a trusted third party has associated a given distinguishing identifier with specific verification AI.

8.4 Mutual authentication

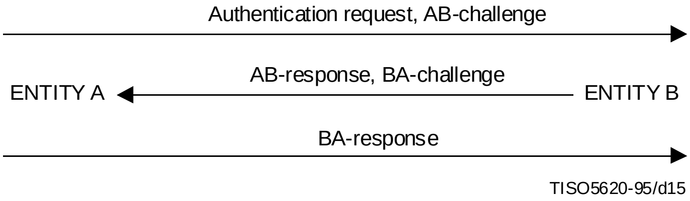
For sub-classes of mechanisms involving a one-way exchange (i.e. sub-classes 1, 2, 3 and 4a) an exchange of the same form may be used in either direction for mutual authentication.

For sub-class 4b, the same type of mechanism can be used in both directions. The first challenge may be sent along with the authentication request, and the transformation of the first challenge sent along with the second challenge (see Figure 15). This requires the same number of exchanges as for unilateral authentication.

Similarly, for sub-class 4c, the transformation of the first challenge may be sent along with the authentication request and the transformation of the second challenge may be sent along with the first challenge.

Sub-class 4b can be used in conjunction with a 4c mechanism. The two challenges are placed within the transformed data. In the case of symmetric encipherment, the claim and verification AI at each end are the same, and the transformation is only carried out once. In the case of asymmetric encipherment, the two transformations are carried out at each end.

For sub-class 4d, three or more transfers are needed for unilateral authentication. Mutual authentication requires four or more transfers.


Authentication request, AB-challenge
ENTITY A
AB-response, BA-challenge
ENTITY B
BA-response
TISO5620-95/d15

ENTITY A	Authentication Initiator
ENTITY B	Authentication Responder

NOTE – For details of responses and transfer of distinguishing identifiers, see sub-class description and figures.

Figure 15 – Mutual authentication using challenge mechanisms

8.5 Summary of class characteristics

Table 1 summarizes the vulnerabilities and the characteristics of the different classes and sub-classes. The characteristics are as described in clause 7.

Table 1 – Vulnerabilities and characteristics of mechanisms

Sub-class	0	1	2	3	4a	4b	4c	4d
Vulnerabilities								
Disclosure	Yes	No	No	No	No	No	No	No
Replay on different verifiers	Yes	Yes	No	Yes	No	No	No	No
Replay on same verifier	Yes	Yes	Yes	No	No	No	No	No
Intruder initiated relay attack	No	No	No	No	No	No	No	No
Intruder responding relay attack	Yes	No	No	No	No	No	No	No
Characteristics								
Symmetry/asymmetry	Sym	Any	Any	Any	Any	Any	Any	Asym
Cryptographic (Yes)/ Non-cryptographic (No)	No	Any	Any	Any	Any	Any	Yes	Yes
Number of Transfers								
– claimant initiator	1	1	1	1	1	3	3	3
– verifier initiator	2	2	2	2	2	2	4	4
Support data origin authentication	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

8.6 Classification by configuration

When entities wish to authenticate, they may need to involve one or more trusted third parties. The nature of trust between each entity and any trusted third party has to be defined. The simplest model involving trusted third parties has a single trusted third party. Other models may consider a set of trusted third parties which trust each other, while the more general model involves a set of trusted third parties which do not trust each other.

8.6.1 Modelling principles involving trusted third parties

In some cases the verifier may only be assured of the identity of the principal if it receives assurance of this identity through multiple trusted third parties.

When three or more trusted third parties are involved, it is possible to protect against the corruption of one or more trusted third parties. Under some security policies, a majority rule may apply.

The following only considers the simplest case, where a single trusted third party is involved.

The relationships between the claimant, the verifier and a single third entity can be modelled in terms of:

- phases, as identified in 5.4 (in particular, the distribution, acquisition, transfer, and verification phases); and
- initial information knowledge.

8.6.1.1 Phase model

The phases relate to the various entities as follows:

- the distribution phase is applicable between the claimant, the verifier and the trusted third party;
- the acquisition phase is applicable between the claimant and the trusted third party or the verifier and the trusted third party;
- the transfer phase is applicable between any pair of: claimant, verifier, and trusted third party;
- the verification phase is applicable between the verifier and the trusted third party.

The acquisition, transfer, and verification phases may use an authentication mechanism from a class identified in 8.1.

The distribution phase may be on-line or off-line. When it is off-line, it will generally happen prior to the authentication exchange. In these cases, there is no assurance that the claim AI is still valid (that is, has not been revoked).

A number of different authentication schemes may be identified, as illustrated in Figure 16. In this figure, entity A corresponds to the claimant and entity B corresponds to the verifier. This figure is for illustrative purposes only and is not necessarily exhaustive.

In Scheme A, entity A obtains its claim AI from the trusted third party after an authentication exchange with the trusted third party, and entity B obtains the verification AI from the trusted third party. Then entity B performs the verification locally.

In Scheme B, entity A obtains its claim AI from the trusted third party after an authentication exchange with the trusted third party, and entity B presents the exchange AI received from entity A to the trusted third party for verification.

In Scheme C, entity A obtains its claim AI from the trusted third party after an authentication exchange with the trusted third party, as well as the verification AI necessary for entity B to perform the verification locally.

In Scheme D, entity A obtains the verification AI necessary for entity B to perform the verification locally, and generates locally the exchange AI. The exchange AI and the verification AI are presented together to entity B.

In Scheme E, entity A generates locally its exchange AI and presents it to entity B, then entity B obtains from the trusted third party the verification AI necessary to perform the verification locally.

In Scheme F, entity A generates locally its exchange AI and presents it to entity B, then entity B presents the exchange AI received from entity A to the trusted third party for verification.

In Scheme G, which is an in-line trust relationship, entity A generates locally its exchange AI and presents it to the trusted third party, then the trusted third party sends to entity B an authentication certificate together with the verification AI necessary to perform the verification locally.

In Scheme H, which is another case of in-line trust relationship, entity A generates locally its exchange AI and presents it to the trusted third party, then the trusted third party sends to entity B a notification that the identity of entity A was verified.

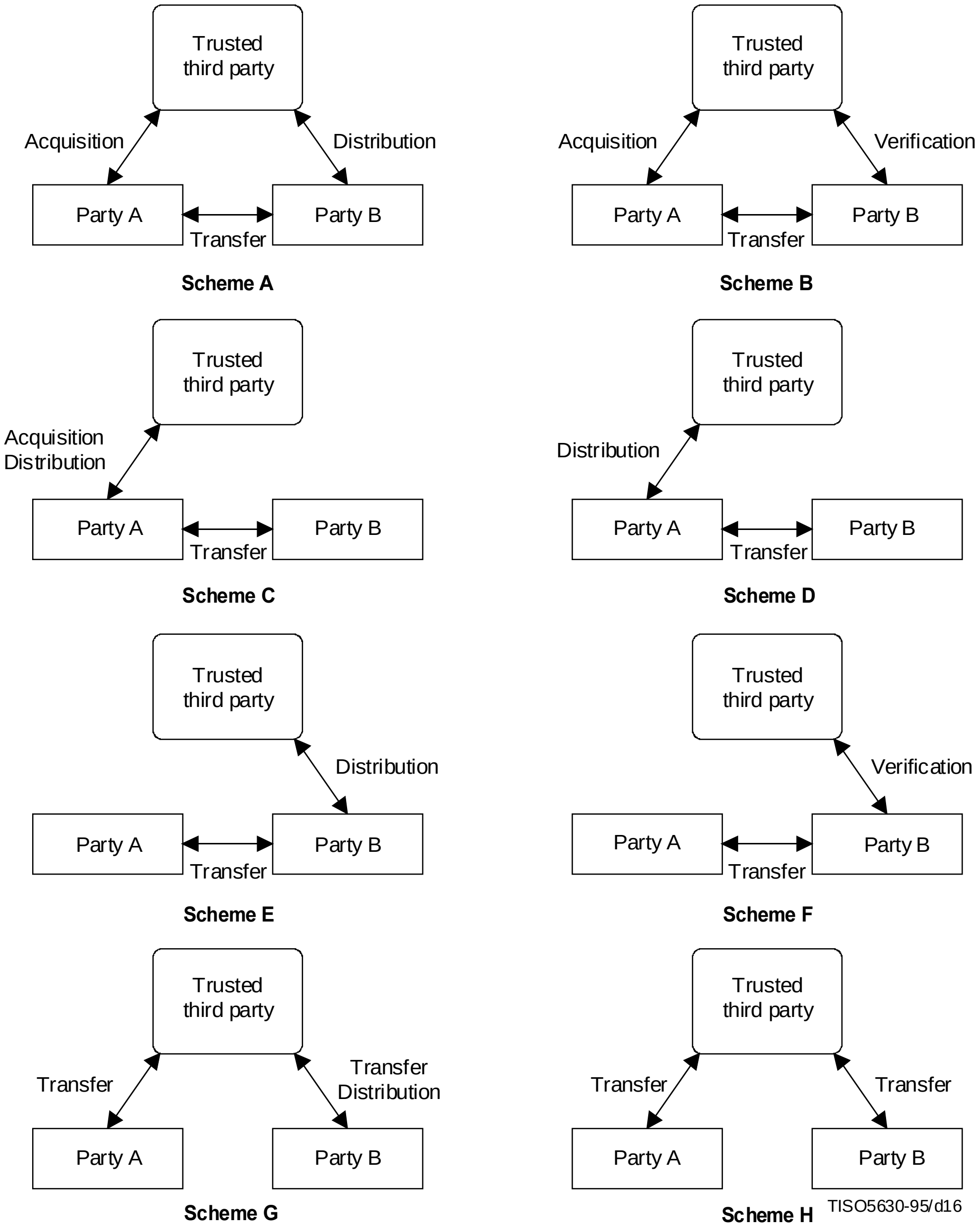


Figure 16 – Schemes for authentication

8.6.1.2 Modelling using initial information knowledge

The claimant (entity A) and the verifier (entity B) must use some initial information before an authentication exchange may take place. If a trusted third party is involved, it means that the claimant does not know directly a public key or a secret key usable by the verifier. Different kinds of initial knowledge, as described below, may be considered.

8.6.1.2.1 Initial information shared between the claimant and the trusted third party

Different cases are:

a) secret key shared between the claimant and the trusted third party known by the claimant and by the trusted third party (secret key techniques);

b) claimant private key known only by the claimant (entity A); claimant public key known by the trusted third party (asymmetric techniques);

c) claimant private key known by the claimant and by the trusted third party (some zero-knowledge techniques).

8.6.1.2.2 Initial information shared between the verifier and the trusted third party

Different cases are:

a) secret key shared between the verifier (entity B) and the trusted third party known by the verifier and by the trusted third party (secret key techniques);

b) trusted third party public key known by the verifier (entity B) (asymmetric and zero-knowledge techniques).

8.6.2 Relationships between trusted third parties involved with authentication

8.6.2.1 On-line trusted third parties

On-line trusted third parties may be necessary so that an authentication exchange may take place. On-line trusted third parties of the same security domain may be holding the claim AI and/or the verification AI of the entities which have been previously registered in the domain.

Protocols and/or procedures are required to ensure that, within a given security domain, different principals cannot register under the same name.

Availability of on-line trusted third parties is an important concern, otherwise authentication exchanges using on-line third entities would be subject to denial of service. Replication of the authentication information in a number of different third entities may minimize this problem. Protocols are also required to replicate that authentication information. When verification AI needs to be exchanged then an integrity service and, in some cases, a confidentiality service are necessary between the authentication trusted third parties. When claim AI needs to be exchanged, then an integrity service and a confidentiality service are necessary between the trusted third parties.

In addition, it may be necessary to consider the exchange of the audit trails maintained by the different on-line authentication trusted third parties of the security domain. Protocols are required to send and receive the audit trails.

8.6.2.2 Off-line trusted third parties

Off-line trusted third parties are often referred to as certification authorities as they may issue off-line authentication certificates. No specific protection is needed to protect an off-line authentication certificate as it is self-protected. Availability of off-line authentication certificates is an important concern, otherwise authentication exchanges using off-line authentication certificates would be subject to denial of service. Replication of this information in a number of different repositories (e.g. the Directory) may minimize this problem.

9 Interactions with other security services/mechanisms

9.1 Access control

Users may need to be authenticated before being allowed to obtain access control information which will permit access to resources subject to an access control policy. Accordingly, the authentication service may pass the results of authentication to the access control service, to be used by the access control service.

The revocation of authentication information may imply revocation of existing access.

9.2 Data integrity

Authentication may be used in conjunction with data integrity to provide assurance of continuity of authentication and to establish corroboration of the source of data.

Some authentication mechanisms can be used to distribute, either implicitly or explicitly, key material that can be used for an integrity service. When this key material is implicitly defined, the way to derive this key material from the data transferred must be known or specified during the authentication exchange. When this key material is explicitly defined, additional data must be transferred in either direction during the authentication exchange.

9.3 Data confidentiality

Some authentication mechanisms can be used to distribute, either implicitly or explicitly, key material that can be used for a confidentiality service. When this key material is implicitly defined, the way to derive this key material from the data transferred must be known or specified during the authentication exchange. When this key material is explicitly defined, additional data must be transferred in either direction during the authentication exchange.

9.4 Non-repudiation

Some authentication mechanisms can be used to distribute, either implicitly or explicitly, key material that can be used for a non-repudiation service. When this key material is implicitly defined, the way to derive this key material from the data transferred must be known or specified during the authentication exchange. When this key material is explicitly defined, additional data must be transferred in either direction during the authentication exchange.

9.5 Audit

Authentication-related information which may be used for audit may include:

a) the results of authentication (i.e. assured identification);

b) information related to revocation of authentication information;

c) information on assurance of continuity of authentication;

d) other information relating to the process of authentication.

Annex A

Human user authentication

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

A.1 General

Correct authentication of human users may be essential in Open Systems security when the open system supports the actions of people. The dialogue between human users and computer systems may increase the possibility of intrusion by masquerading. Methods for authentication of human users must be acceptable to the human users, as well as being economical and safe. Inconvenient methods sometimes encourage human users to find ways to avoid the procedures, so that the potential for intrusion increases.

Authentication of a human user relies on authentication principles in one or more of the following categories:

a) something known;

b) something possessed;

c) characteristics of the individual human user;

d) accepting that an identified trusted third party has established the human user's identity; and/or

e) context (e.g. source address of request).

In general, the process of human user authentication involves matching credentials presented by the user with authentication information obtained in the installation phase.

A.1.1 Authentication by means of something known

In this category, the most commonly used authentication information is a password. When accessing a system, the human user presents the password and the authenticating system compares it with the corresponding value in a password list, so as to corroborate the identity of the human user. Passwords should be difficult to guess and should be carefully managed. Otherwise, they are potentially subject to unintentional disclosure.

A.1.2 Authentication by means of something possessed

In this category, a physical token is used, such as:

a) magnetic stripe card; or

b) IC card.

With magnetic stripe cards, when accessing a system, the human user presents the physical token and the authenticating system reads authentication information from the physical token to compare it with stored authentication information, in order to corroborate the identity of the human user.

One vulnerability of magnetic stripe cards is that they may be easily copied. Another vulnerability is that if the magnetic stripe card is possessed by the wrong person, the authentication approach fails.

With IC cards, when accessing a system, the human user presents the physical token and the authenticating system uses information stored in the physical token to produce the exchange AI, in order to corroborate the identity of the human user. An advantage of IC cards is that they may not be easily copied.

Two variants may be considered, as to whether or not the IC card is able to authenticate the card holder:

– when the IC card is able to authenticate the card holder, then there is a double authentication scheme where the user is authenticated by the verifier; this is equivalent, by transitivity, to authenticating the user directly;

– when the IC card is not able to authenticate the card holder and if the object is possessed by the wrong person, then the authentication approach fails.

A.1.3 Time-dependent password generator

One type of human user authentication mechanism is a hand-held device which functions as a time-dependent password generator. Exchange AI is generated using a combination of:

– secret information stored internal to the device;

– time;

– a PIN entered directly by the user into a PIN-pad on the device.

Exchange AI so generated is displayed on the device. It is then sent by the user (in cleartext form) to the verifying system. This system may need to maintain synchronization with the card. This type of human user authentication mechanism requires a person attempting to be authenticated with such a device to *both*:

a) possess the correct device; and

b) know the PIN.

A.1.4 Authentication using characteristics of individual human user

Passwords are susceptible to disclosure if treated carelessly, and physical tokens are susceptible to theft or, in the case of magnetic stripe cards, unauthorized copying. There is a class of human user authentication method without these weaknesses, which is based on characteristics of individual users such as:

– hand-written signature;

– fingerprint;

– vocal pattern;

– retinal pattern; or

– dynamic keyboard characteristics.

Two important classes of hand-written signature method are static and dynamic systems. In the latter, pressure, time, and direction information may be available.

The analysis of dynamic keyboard characteristics provides a continuous form of authentication.

In the enrolment phase, a human user registers his or her identity at an enrolment system. The user executes the required procedure, for example, writing a signature on a pad, pressing a finger on a pad, or pronouncing specified words. The procedure is repeated as necessary until reliable reference information can be acquired. The system analyses the characteristic value of action of the human user and records it as a profile.

In the transfer/verification phase, the human user presents his or her identity and again executes the required procedure. The verification system compares the pattern obtained from the user with the profile stored for this user.

A.2 Processes acting on behalf of a human user

In some circumstances a user may wish to act without being present. In such cases the user will have a representation within the system which may have a lifetime independent of the effective presence of the user.

Because the representation acts as if it were the user, the user's actions can be continued without requiring the direct involvement of the user. For example, a human user may log on and then may use different computers without further logging on.

Rather than supporting representation with independent lifetimes, representations can also be used with additional mechanisms that make the lifetime of representations dependent upon the presence of the user.

Annex B

Authentication in the OSI 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 ISO 7498-2. This annex summarizes what is relevant to authentication.

Two security services are considered:

– peer-entity authentication;

– data origin authentication.

B.1 Peer-entity authentication

Peer-entity authentication may be used at the establishment of, or at times during, the data transfer phase of a connection to confirm the identities of one or more of the entities connected to one or more of the other entities. This service is available in both connection-oriented and connectionless protocols. One-way and mutual peer entity authentication are possible.

B.2 Data origin authentication

Data origin authentication provides the corroboration of the source of a data unit. The service does not provide protection against duplication of data units.

B.3 Use of authentication within the OSI layers

Peer-entity authentication and data origin authentication are only relevant to the following OSI layers:

– Network Layer (layer 3);

– Transport Layer (layer 4);

– Application Layer (layer 7).

B.3.1 Use of authentication at the network layer

Peer-entity authentication, when used at the Network Layer, allows the confirmation of the identities of network entities. This service allows the authentication of network nodes, subnetwork nodes or relays.

Data origin authentication, when used at the Network Layer, allows the confirmation of the identities of the source of a data unit. The source may be a network node, a subnetwork node, or a relay.

The mechanisms used by the Network Layer are within that layer.

B.3.2 Use of authentication at the Transport Layer

Peer-entity authentication, when used at the Transport Layer, allows the confirmation of the identities of transport entities. This service allows the authentication of end systems. Different applications supported by the same end systems may not be authenticated.

Data origin authentication, when used at the Transport Layer, allows the confirmation of the identities of the source of a data unit. The source is an end system.

The mechanisms used by the Transport Layer are within that layer.

B.3.3 Use of authentication at the Application Layer

Peer-entity authentication, when used at the Application Layer, allows the confirmation of the identities of application entities supported by end systems. This service allows the authentication of application entities or application processes. Different application entities or application processes supported by the same end system may be authenticated.

Data origin authentication, when used at the Application Layer, allows the confirmation of the identities of the source of a data unit. The source may be an application entity or an application process.

The mechanisms used by the Application Layer may be at the Application Layer or at the Presentation Layer. Authentication, when invoked at the Application Layer, may also make use of the authentication services provided by the Network Layer or the Transport Layer.

Annex C

Countering replay using unique numbers or challenges

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

C.1 Unique numbers

Unique numbers are generated by the claimant. The same unique number must never be accepted twice by the same verifier. This can be achieved in several ways. Some techniques which would be valid in theory may not be usable in practice. The straightforward example of such a technique would be to keep track of all received unique numbers which have been successfully used during an authentication exchange. This would lead to an amount of memory growing with the number of successful authentications achieved. This may not be acceptable for cost reasons and/or performance reasons.

One way to reduce the amount of memory required on the verifier side is to keep track of all the successfully used unique numbers but only during some period of time. This leads to the introduction of a time stamp as part of the unique number so that only "recent" unique numbers have to be remembered by the verifier. In practice, a time window of several minutes may be adequate for both limiting the amount of memory needed and for minimizing the time synchronization problem between the two different time references used by the principal and the verifier.

In order to prevent denial of service, it is better to prevent unintentional collisions between unique numbers generated by two different principals. For that, the unique number should be from a sufficiently large range. The range of the unique number is related to the maximum number of authentications per period of time (e.g. per second) that needs to be achieved on the particular verifier where authentication is attempted. When the time reference used by the principal does not directly provide such a large number, a random number may be added to the time stamp in order to enlarge the range of the unique number.

C.2 Challenges

Challenges are generated by the verifier. The same challenge must never be issued twice by the same verifier. This can be achieved in several ways.

Some techniques which would be valid in theory may not be usable in practice. The straight-forward example of such a technique would be to keep track of all issued challenges. This would lead to an amount of memory growing with the number of successful authentications achieved using these challenges. This may not be acceptable for cost reasons and/or performance reasons.

There are several ways to reduce the amount of memory required on the verifier side:

- to issue sequential values for the challenges and to keep track of only the last sequential value;
- to issue random numbers for the challenges. Although it violates the rule that "the same challenge must never be used twice", the likelihood of such an occurrence may be made acceptably small by using random numbers from a sufficiently large range;
- to use a time stamp for the challenge;
- to use a combination of a time stamp and a random number.

Annex D

Protection against some forms of attack on authentication

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

D.1 Listen-and-replay attacks

There are two cases of replay to be considered. These are the replay of some exchange AI:

- on the same verifier; or
- on another verifier.

The latter case is possible as soon as the verification AI of a principal is known by several verifiers. When a successful replay may be achieved, this is a specific case of masquerade.

Both cases of replay can be countered using challenges. Challenges are generated by the verifier. The same challenge must never be issued twice by the same verifier. This can be achieved in several ways (see Annex C).

D.2 Replay on the same verifier

Replay on the same verifier may be countered using unique numbers or challenges.

Unique numbers are generated by the claimant. The same unique number must never be accepted twice by the same verifier. This can be achieved in several ways (see Annex C).

D.3 Replay on a different verifier

Replay on a different verifier may be countered using challenges. Alternatively it may be countered using, in the computation of the exchange AI, any characteristic which is unique to the verifier. Such characteristic may be the name of the verifier, his network address or in general any attribute unique with respect to the verifiers sharing the same verification authentication information.

D.4 Interception-and-relay attacks

D.4.1 Direct attacks

One type of attack (a direct attack) involves the intruder as the initiator of the authentication. This attack is possible only if both claimant and verifier can initiate the authentication. With this attack the claimant and the verifier exchange authentication information via an intruder without being aware of this, i.e. the intruder pretends to be a certain verifier towards a claimant and to be this claimant towards that verifier.

For example, suppose intruder C wants to pretend to verifier B that he is claimant A. C starts an interaction with both A and B. C tells A that he is B, asks A to authenticate towards B, and also tells B that he is A and that he wants to authenticate himself.

During the process of authentication A acts as a claimant towards B (actually to C acting as B) and, therefore, supplies information that C can use to authenticate to B. B acts as the verifier and also supplies the information that C needs to play the role of verifier. Following the authentication, intruder C will appear to B as the authenticated A.

Possible ways to counter this type of attack require replay protection on a different verifier:

a) the entity that starts an interaction is always the claimant; or

b) the exchange AI provided by the claimant differs according to his role as initiator of an authentication request or responder to an authentication invitation. This difference allows the verifier to detect the interception described. See Annex D for further details.

D.4.2 Opportunistic attacks

One type of attack is where the intruder sits in the middle of an authentication exchange, intercepts the authentication information and forwards it, taking over the role of claimant.

The general way to counter this type of attack requires the use of a complementary service (integrity or confidentiality). The exchange AI is combined with some other information that enables the claimant and verifier, provided they are the legitimate parties, to derive a key. The derived key can then be used as a key for a cryptographic-based integrity or confidentiality mechanism.

Another way to counter this type of attack is relevant where the communications network is not subject to interceptions internally, i.e. it always delivers data unchanged to the correct address. In this situation the attack can be countered by using the network addresses as an additional input to the generate service. In this way the exchange AI will be dependant on the network address.

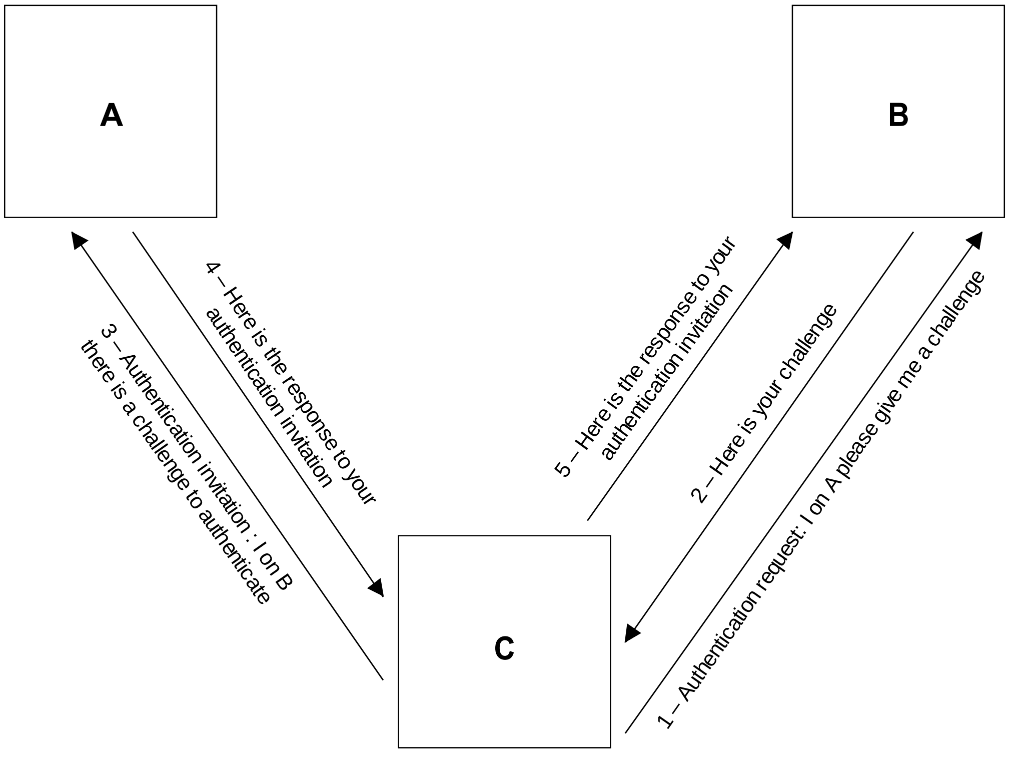
D.5 A limited form of protection against an intruder attack

The second type of attack described in D.4 is possible either when challenges are used or when unique numbers are used. The protection involves the use, by the claimant, of an indicator stating if the response is following an authentication invitation or an authentication request. The indicator may either state (e.g. when set to one) that the response is following an authentication invitation or state (e.g. when set to zero) that the response is following an authentication request. As the indicator is part of the computation of the response, this means that the value of the response given by the claimant is dependant upon the value of the indicator. In the following, the indicator is called the invitation/request indicator.

D.6 Protocol using challenges

When challenges are used, C pretends to be A and sends an authentication request to B (first transfer). B gives a challenge to C (second transfer). C sends an authentication invitation to A and forwards the challenge received from B to A (third transfer). A computes his response using both the challenge received from C and the invitation/request indicator set to “invitation”. C forwards to B the response received from A. B checks the response. As it has received originally from C an authentication request, it is waiting for an invitation/request indicator set to “request”. As it receives a response computed with an invitation/request indicator set to “invitation”, it rejects the authentication (see Figure D.1).

If B is supporting both authentication requests and authentication invitations, additional care must be taken by B: for every authentication invitation issued by B, B must remember to which claimant a given challenge has been given so that C cannot use it for another claimant when sending its authentication invitation (third exchange).


A
B
4 – Here is the response to your authentication invitation
3 – Authentication invitation : I on B there is a challenge to authenticate
5 – Here is the response to your authentication invitation
2 – Here is your challenge
1 – Authentication request: I on A please give me a challenge
C
TISO5640-95/d17

NOTE – Direct attacks, as explained in D.4.1, even if countered using method a) or b) are still vulnerable to the opportunistic attack.

Figure D.1 – Protection against an intruder attack when using challenges

D.7 Protocol using unique numbers

When unique numbers are used, C pretends to be B and sends an authentication invitation to A (first transfer). A computes his response using a unique number and the invitation/request indicator set to “invitation” (second transfer). C forwards to B the response received from A (third transfer). B checks the response. It contains an invitation/request indicator set to “invitation” but B has not issued any authentication invitation, so it rejects the authentication (see Figure D.2).

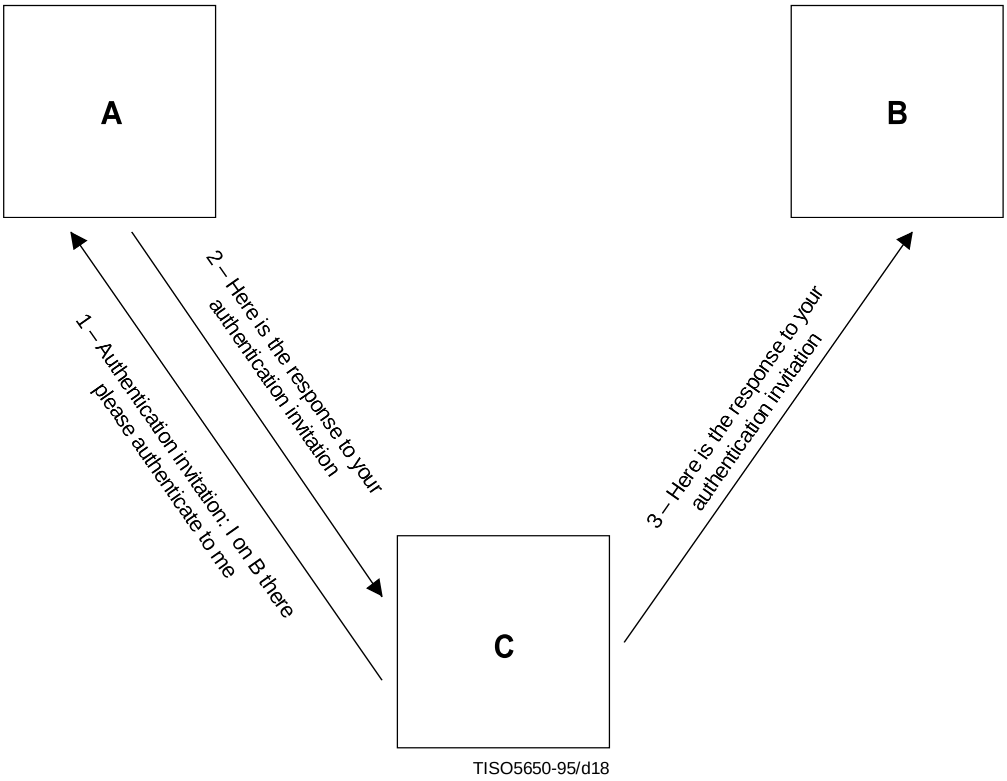

A
B
C
1 – Authentication invitation: I on B there please authenticate to me
2 – Here is the response to your authentication invitation
3 – Here is the response to your authentication invitation
TISO5650-95/d18

Figure D.2 – Protection against an intruder attack when using unique numbers

Annex E

Bibliography

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

ISO/IEC 9798-1:1991, *Information technology – Security techniques – Entity authentication mechanisms – Part 1: General model*.

ISO/IEC 9798-2:1994, *Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms.*

ISO/IEC 9798-3:1993, *Information technology – Security techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public key algorithm.*

ISO/IEC 9798-4:1995, *Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function.*

ITU-T Recommendation X.509 (1993) | ISO/IEC 9594-8:1995, *Information technology – Open Systems Interconnection – The Directory: authentication framework.*

Annex F

Some specific examples of authentication mechanisms

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

This annex provides two specific examples of the use of authentication mechanisms.

F.1 Specific example of a unique number mechanism with on-line authentication certificate

This example illustrates the use of a unique number mechanism as described in Class 3 of 8.1. In this example, employing on-line authentication certificates, the distinguishing identifier, a protection method, a protection parameter, and a validity period are included in the authentication certificate. This example needs only one transfer and allows a given authentication certificate to be used more than once.

The protection method indicates the relationship between the protection parameter held in the certificate and the external control parameter to be used to protect the authentication certificate against unauthorized use. The external control parameter may be related to the protection parameter through a one-way relationship such as:

- the external control parameter is a validation value and the protection parameter is the result of a one-way function applied to the validation value; or
- the external control parameter is a private key and the protection parameter is the corresponding public key.

When a validation value is used as the external control parameter, it is sent to the verifier as proof of ownership of the authentication certificate. While in transit, the validation value must be confidentiality protected, e.g. it is sent enciphered by the claimant to the verifier using an external confidentiality key associated with the communication channel or with the receiving end of the communication channel.

Ownership and Replay protection are achieved using a unique number and a transformation function. Three different transformation functions (F) can be used depending upon the nature of the external control parameter:

a) *One-way function* – When the external control parameter is a validation value, then the unique number and the validation value are transformed under a one-way function. The result and the unique number are transmitted so that the verifier can perform the same transformation.

b) *Asymmetric algorithm* – When the external control parameter is a private key, the unique number is signed under that private key.

c) *Symmetric algorithm* – When the external control parameter is a secret key, the unique number is enciphered or sealed under the validation value used as a secret key.

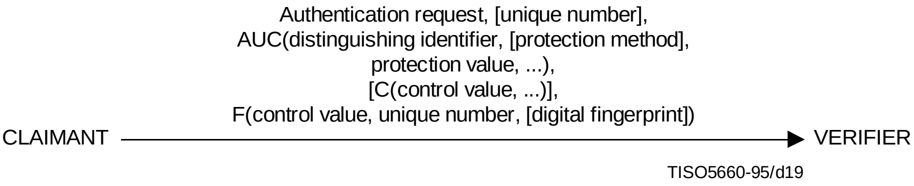
This example is applicable to both data origin and entity authentication. For data origin authentication, the data, or a digital fingerprint of the data, may also be transformed under the function F.

The Acquire service is used to obtain the on-line authentication certificate and an external control parameter. The Generate service then generates a unique number and carries out a transformation using the following as inputs:

- unique number;
- external control parameter;
- distinguishing identifier (optional);
- digital fingerprint (if data origin authentication).

In addition, when the external control parameter is a validation value or a secret control key, the Generate service sends this value enciphered so that only the intended verifier may decipher it, and produces exchange AI as shown in Figure 14.

The Verify service checks the exchange AI for validity using the protection value within the authentication certificate. In addition, when a validation value or a secret control key is used, the Verify service deciphers the enciphered validation value or secret control key and verifies that it matches the protection value. It also checks that the unique number has not been successfully received before.


Authentication request, [unique number],
AUC(distinguishing identifier, [protection method],
protection value, ...),
[C(control value, ...)],
F(control value, unique number, [digital fingerprint])
CLAIMANT
VERIFIER
TISO5660-95/d19

NOTES

1 AUC(....) is used to denote an on-line authentication certificate including the give parameters.

2 C(....) is used to denote the application of a confidentiality service. This is only applicable when the control parameter is a validation value.

Figure F.1 – Unique number mechanism with on-line authentication certificate

F.2 Challenge mechanism with on-line certificate

This mechanism uses an authentication certificate to provide a proof of authentication using the principle described in 5.3, d) and the challenge mechanism described in 8.1.5.2. The authentication certificate provides proof that a trusted third party has authenticated its holder with a specific distinguishing identifier. The mechanism provides a means to prove that an authentication certificate for a given distinguishing identifier is held by the claimant.

In this example employing on-line authentication certificates, the distinguishing identifier, a protection method, a protection parameter, and a validity period are included in the authentication certificate. This example allows a given authentication certificate to be used more than once.

The protection method indicates the relationship between the protection parameter held in the certificate and the external control parameter to be used to protect the authentication certificate against unauthorized use. The external control parameter may be related to the protection parameter through a one-way relationship such as:

– the external control parameter is a validation value and the protection parameter is the result of a one-way function applied to the validation value;

– the external control parameter is a private key and the protection parameter is the corresponding public key.

When a validation value is used as the external control parameter, it is sent to the verifier as proof of ownership of the authentication certificate. While in transit, the key must be confidentiality protected, e.g. it is sent enciphered by the claimant to the verifier using an external confidentiality key associated with the communication channel or with the receiving end of the communication channel.

Ownership and Replay protection are achieved using a challenge and a transformation function. Three different transformation functions (F) can be used depending upon the nature of the external control parameter:

a) *One-way function* – When the external control parameter is a validation value, then the challenge and the validation value are transformed under a one-way function. The result and the challenge are transmitted so that the verifier can perform the same transformation.

b) *Asymmetric algorithm* – When the external control parameter is a private key, the challenge is signed under that private key.

c) *Symmetric algorithm* – When the external control parameter is a secret key, the challenge is enciphered or sealed under the validation value used as a secret key.

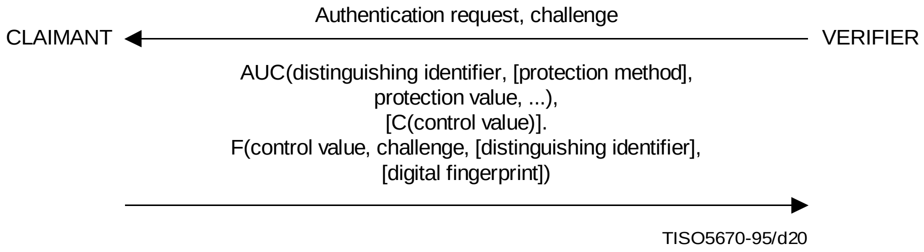
This example is applicable to both data origin and entity authentication. For data origin authentication, the data or a digital fingerprint of the data, may also be transformed under function F.

The Acquire service is used to obtain the on-line authentication certificate and an external control parameter. The Generate service produces an authentication request. On receipt of the authentication request, the Verify service generates a challenge as exchange AI. The Generate service then carries out a transformation using the following as inputs:

- challenge;
- external control parameter;
- distinguishing identifier (optional);
- digital fingerprint (if data origin authentication).

In addition, when the control value is a validation value or a secret control key, the Generate service sends this value enciphered so that only the intended verifier may decipher it, and produces exchange AI as shown in Figure 16.

The Verify service checks the exchange AI for validity using the protection parameter within the authentication certificate. In addition, when a validation value or a secret control key is used, the Verify service deciphers the enciphered validation value or secret control key and verifies that it matches the protection value. It also checks that the challenge matches the one sent.


Authentication request, challenge
CLAIMANT
VERIFIER
AUC(distinguishing identifier, [protection method],
protection value, ...),
[C(control value)].
F(control value, challenge, [distinguishing identifier],
[digital fingerprint])
TISO5670-95/d20

NOTES

1 AUC(....) is used to denote an on-line authentication certificate including the given parameters.

2 C(....) is used to denote the application of a confidentiality service. This is only applicable when the control parameter is a validation value.

Figure F.2 – Challenge mechanism with on-line authentication certificate

Annex G

Authentication facilities outline

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

<table>
<tr><td colspan="2" rowspan="3">Security facilities outline</td><td rowspan="2">Element</td><td>Entity: Claimant, Verifier, Trusted Third Party, Principal, Manager</td></tr>
<tr><td>Info Object: Authentication information</td></tr>
<tr><td colspan="2">Goal of Entity: To provide assurance of the claimed identity of an entity</td></tr>
<tr><td rowspan="6">A
C
T
I
V
I
T
Y</td><td>Entity</td><td colspan="2">Security authority, principal, manager</td></tr>
<tr><td>Function</td><td colspan="2"></td></tr>
<tr><td>Management related activity</td><td colspan="2">– Install – Re-enable
– Change-AI – De-install
– Distribute</td></tr>
<tr><td>Entity</td><td colspan="2">– claimant
– verifier
– TTP</td></tr>
<tr><td>Function</td><td colspan="2"></td></tr>
<tr><td>Operational related activity</td><td colspan="2">– Acquire
– Generate
– Verify
– Generate;
– and Verify</td></tr>
<tr><td rowspan="3">I
N
F
O
R
M
A
T
I
O
N</td><td>Input/Output Data element managed by SDA</td><td colspan="2">Descriptive information, e.g. password, keys, use of a protocol, challenge-and-response table, acknowledgement or rejection, off-line certificate, status information, AI</td></tr>
<tr><td>Information type used in operation</td><td colspan="2">Claim AI
Exchange AI
Verification AI</td></tr>
<tr><td>Control information</td><td colspan="2">Validity
Authentication state information</td></tr>
</table>