INTERNATIONAL
STANDARD

ISO/IEC
9594-8

First edition
1990-12-15

Information technology — Open Systems Interconnection — The Directory —

Part 8:
Authentication framework

Technologies de l'information — Interconnexion de systèmes ouverts — L'annuaire —

Partie 8: Cadre général d'authentification

Reference number
ISO/IEC 9594-8 : 1990 (E)

Contents

ISO/IEC Copyright Office ● Case postale 56 ● CH-1211 Genève 20 ● Switzerland
Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9594-8 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 9594 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — The Directory*:

- *Part 1: Overview of concepts, models and services*
- *Part 2: Models*
- *Part 3: Abstract service definition*
- *Part 4: Procedures for distributed operation*
- *Part 5: Protocol specifications*
- *Part 6: Selected attribute types*
- *Part 7: Selected object classes*
- *Part 8: Authentication framework*

Annex G forms an integral part of this part of ISO/IEC 9594. Annexes A, B, C, D, E, F and H are for information only.

Introduction

0.1 This part of ISO/IEC 9594, together with the other parts, has been produced to facilitate the interconnection of information processing systems to provide directory services. The set of all such systems, together with the directory information which they hold, can be viewed as an integrated whole, called the *Directory*. The information held by the Directory, collectively known as the Directory Information Base (DIB), is typically used to facilitate communication between, with or about objects such as OSI application-entities, people, terminals and distribution lists.

0.2 The Directory plays a significant role in Open Systems Interconnection, whose aim is to allow, with a minimum of technical agreement outside of the interconnection standards themselves, the interconnection of information processing systems:

- from different manufacturers;
- under different managements;
- of different levels of complexity; and
- of different ages.

0.3 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 briefly described in annex A. Virtually all security services are dependent upon the identities of the communicating parties being reliably known, i.e. authentication.

0.4 This part of ISO/IEC 9594 defines a framework for the provision of authentication services by the Directory to its users. These users include the Directory itself, as well as other applications and services. The Directory can usefully be involved in meeting their needs for authentication and other security services because it is a natural place from which communicating parties can obtain authentication information of each other — knowledge which is the basis of authentication. The Directory is a natural place because it holds other information which is required for communication and obtained prior to communication taking place. Obtaining the authentication information of a potential communication partner from the Directory is, with this approach, similar to obtaining an address. Owing to the wide reach of the Directory for communications purposes, it is expected that this authentication framework will be widely used by a range of applications.

INTERNATIONAL STANDARD ISO/IEC 9594-8 : 1990 (E)

Information technology — Open Systems Interconnection — The Directory —

Part 8:
Authentication framework

SECTION 1: GENERAL

1 Scope

1.1 This part of ISO/IEC 9594:

- specifies the form of authentication information held by the Directory;
- describes how authentication information may be obtained from the Directory;
- states the assumptions made about how authentication information is formed and placed in the Directory;
- defines three ways in which applications may use this authentication information to perform authentication and describes how other security services may be supported by authentication.

1.2 This part of ISO/IEC 9594 describes two levels of authentication: simple authentication, using a password as a verification of claimed identity; and strong authentication, involving credentials formed using cryptographic techniques. While simple authentication offers some limited protection against unauthorized access, only strong authentication should be used as the basis for providing secure services. It is not intended to establish this as a general framework for authentication, but it can be of general use for applications which consider these techniques adequate.

1.3 Authentication (and other security services) can only be provided within the context of a defined security policy. It is a matter for users of an application to define their own security policy which may be constrained by the services provided by a standard.

1.4 It is a matter for standards defining applications which *use* the authentication framework to specify the protocol exchanges which need to be performed in order to achieve authentication based upon the authentication information obtained from the Directory. The protocol used by applications to obtain credentials from the Directory is the Directory Access Protocol (DAP), specified in ISO/IEC 9594-5.

1.5 The strong authentication method specified in this part of ISO/IEC 9594 is based upon public-key cryptosystems. It is a major advantage of such systems that user certificates may be held within the Directory as attributes, and may be freely communicated within the Directory System and obtained by users of the Directory in the same manner as other Directory information. The user certificates are assumed to be formed by 'off-line' means, and placed in the Directory by their creator. The generation of user certificates is performed by some off-line Certification Authority which is completely separate from the DSAs in the Directory. In particular, no special requirements are placed upon Directory providers to store or communicate user certificates in a secure manner.

A brief introduction to public-key cryptography can be found in annex B.

1.6 In general, the authentication framework is not dependent on the use of a particular cryptographic algorithm, provided it has the properties described in 6.1. Potentially a number of different algorithms may be used. However, two users wishing to authenticate shall support the same cryptographic algorithm for authentication to be performed correctly. Thus, within the context of a set of related applications, the choice of a single algorithm will serve to maximize the community of users able to authenticate and communicate securely. One example of a public key cryptographic algorithm can be found in Annex C.

1.7 Similarly, two users wishing to authenticate shall support the same hash function (see 3.3f) (used in forming credentials and authentication tokens). Again, in principle, a number of alternative hash functions could be used, at the cost of narrowing the communities of users able to authenticate. A brief introduction to hash functions together with one example hash function can be found in annex D.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9594. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9594 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498-2:1987, *Information Processing Systems — Open Systems Interconnection —*

Basic Reference Model — Part 2: Security Architecture.

ISO/IEC 8824:1990, *Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC 8825:1990, *Information Technology — Open Systems Interconnection — Specification of Basic Encoding rules for Abstract Syntax Notation One (ASN.1)*

ISO/IEC 10021-3:1990, *Information Technology — Text Communication — Message Oriented Interchange System (MOTIS) — Part 3: Abstract Service Definition Conventions.*

3 Definitions

3.1 This part of ISO/IEC 9594 makes use of the following general security-related terms defined in ISO 7498-2:

a) *asymmetric* (encipherment);
b) *authentication exchange*;
c) *authentication information*;
d) *confidentiality*;
e) *credentials*;
f) *cryptography* ;
g) *data origin authentication*;
h) *decipherment* ;
i) *encipherment* ;
j) *key* ;
k) *password* ;
l) *peer-entity authentication* ;
m) *symmetric* (encipherment).

3.2 The following terms used in this part of ISO/IEC 9594 are defined in ISO/IEC 9594-2:

a) *attribute* ;
b) *Directory Information Base* ;
c) *Directory Information Tree* ;
d) *distinguished name* ;
e) *entry* ;
f) *object* ;
g) *root* .

3.3 For the purpose of this part of ISO/IEC 9594 the following definitions apply:

3.3.1 *authentication token (token)*: Information conveyed during a strong authentication exchange, which can be used to authenticate its sender.

3.3.2 *user certificate (certificate)* : The public keys of a user, together with some other information, rendered unforgeable by encipherment with the secret key of the certification authority which issued it.

3.3.3 *certification authority* : An authority trusted by one or more users to create and assign certificates. Optionally the certfication authority may create the users' keys.

3.3.4 *certification path* : An ordered sequence of certificates of objects in the DIT which, together with the public key of the initial object in the path, can be processed to obtain that of the final object in the path.

3.3.5 *cryptographic system, cryptosystem* : A collection of transformations from plain text into ciphertext and vice versa, the particular transformation(s) to be used being selected by keys. The transformations are normally defined by a mathematical algorithm.

3.3.6 *hash function* : A (mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A 'good' hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range.

3.3.7 *one-way function* : A (mathematical) function f which is easy to compute, but which for a general value y in the range, it is computationally difficult to find a value x in the domain such that f(x) = y. There may be a few values y for which finding x is not computationally difficult.

3.3.8 *public key* : (In a public key cryptosystem) that key of a user's key pair which is publicly known.

3.3.9 *private key* (*secret key* — deprecated): (In a public key cryptosystem) that key of a user's key pair which is known only by that user.

3.3.10 *simple authentication* : Authentication by means of simple password arrangements.

3.3.11 *security policy* : The set of rules laid down by the security authority governing the use and provision of security services and facilities.

3.3.12 *strong authentication* : Authentication by means of cryptographically derived credentials.

3.3.13 *trust*: Generally, an entity can be said to "trust" a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects. This trust may apply only for some specific function. The key role of trust in the authentication framework is to describe the relationship between an authenticating entity and a certification authority; an authenticating entity shall be certain that it can trust the certification authority to create only valid and reliable certificates.

3.3.14 *certificate serial number*: An integer value, unique within the issuing CA, which is unambiguously associated with a certificate issued by that CA.

4 Notation and Abbreviations

4.1 The notation used in this part of ISO/IEC 9594 is defined in table 1 below.

Note - in the table, the symbols X, X_1, X_2 etc. occur in place of the names of users, while the symbol I occurs in place of arbitrary information

4.2 The following abbreviations are used in this part of ISO/IEC 9594:

CA	Certification Authority
DIB	Directory Information Base
DIT	Directory Information Tree
PKCS	Public key cryptosystem

Table 1 - Notation

NOTATION	MEANING
Xp	public key of a user X.
Xs	secret key of X.
Xp[I]	encipherment of some information, I, using the public key of X.
Xs[I]	encipherment of I using the secret key of X.
X{I}	the signing of I by user X. It consists of I with an enciphered summary appended.
CA(X)	a certification authority of user X.
CA^n(X)	(where n>1): CA(CA(...n times...(X)))
X_1«X_2»	the certificate of user X_2 issued by certification authority X_1.
X_1«X_2» X_2«X_3»	a chain of certificates (can be of arbitrary length), where each item is the certificate for the certification authority which produced the next. It is functionally equivalent to the following certificate X_1«X_{n+1}». For example, possession of A«B»B«C» provides the same capability as A«C», namely the ability to find out Cp given Ap.
X_1p • X_1«X_2»	the operation of unwrapping a certificate (or certificate chain) to extract a public key. It is an infix operator, whose left operand is the public key of a certification authority, and whose right operand is a certificate issued by that certification authority. The outcome is the public key of the user whose certificate is the right operand. For example: Ap •A«B» B«C» denotes the operation of using the public key of A to obtain B's public key, Bp, from its certificate, followed by using Bp to unwrap C's certificate. The outcome of the operation is the public key of C, Cp.
A→B	a certification path from A to B, formed of a chain of certificates, starting with CA(A)«CA^2(A)» and ending with CA(B)«B».

Section 2: Simple Authentication

5 Simple Authentication Procedure

5.1 Simple authentication is intended to provide local authorization based upon a Distinguished Name of a user, a bilaterally agreed (optional) password, and a bilateral understanding of the means of using and handling this password within a single domain. Utilization of simple authentication is primarily intended for local use only, i.e. for peer entity authentication between one DUA and one DSA or between one DSA and one DSA. Simple authentication may be achieved by several means:

a) the transfer of the user's distinguished name and (optional) password in the clear (non-protected) to the recipient for evaluation;

b) the transfer of the user's distinguished name, password, and a random number and/or a timestamp, all of which are protected by applying a one-way function;

c) the transfer of the protected information described in b) together with a random number and/or a timestamp, all of which is protected by applying a one-way function.

Notes

1. There is no requirement that the one-way functions applied be different.

2. The signalling of procedures for protecting passwords may be a matter for extension to the document.

5.2 Where passwords are not protected, a minimal degree of security is provided for preventing unauthorized access. It should not be considered a basis for secure services. Protecting the user's distinguished name and password provides greater degrees of security. The algorithms to be used for the protection mechanism are typically non-enciphering one-way functions that are very simple to implement.

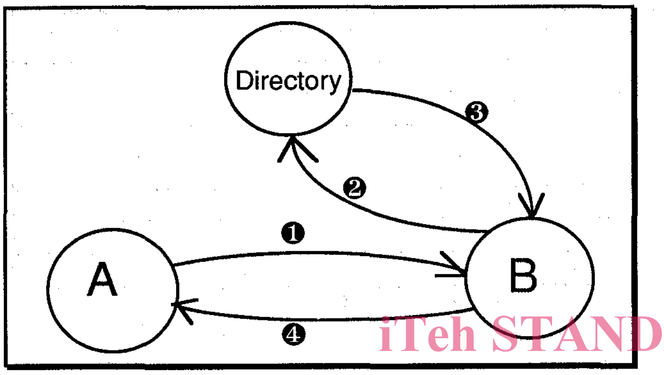


Figure 1— The Unprotected Simple Authentication Procedure

5.3 The general procedure for achieving simple authentication is shown in figure 1.

5.3.1 The following steps are involved:

❶ an originating user A sends its distinguished name and password to a recipient user B;

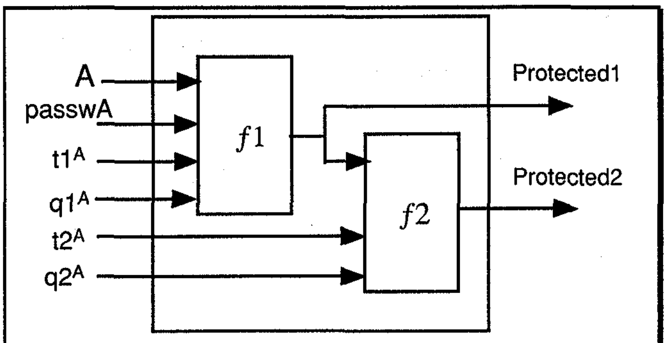
❷ B sends the purported distinguished name and password of A to the Directory, where the password is checked against that held as the **UserPassword** attribute within the directory entry for A (using the Compare operation of the Directory);

❸ the Directory confirms (or denies) to B that the credentials are valid;

❹ the success (or failure) of authentication may be conveyed to A.

5.3.2 The most basic form of simple authentication involves only step ❶ and after B has checked the distinguished name and password, may include step ❹.

5.4 Figure 2 illustrates two approaches by which protected identifying information may be generated. $f1$ and $f2$ are one-way functions (either identical or different) and the timestamps and random numbers are optional and subject to bilateral agreements.



Legend

A	=	user's distinguished name
t^A	=	timestamps
passwA	=	password of A
q^A	=	random numbers, optionally with a counter included

Figure 2 — Protected Simple Authentication

5.4.1 Figure 3 illustrates the procedure for protected simple authentication.

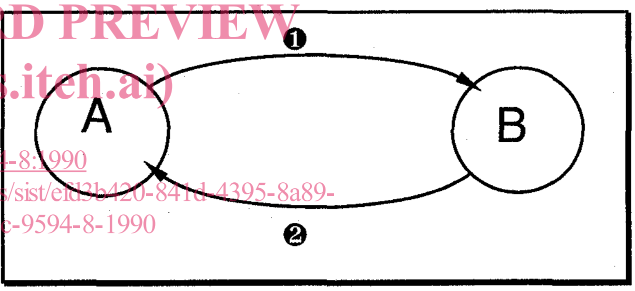


Figure 3 — The Protected Simple Authentication Procedure

The following steps are involved (initially using $f1$ only):

❶ an originating user, user A, sends its protected identifying information (Authenticator1) to user B. Protection is achieved by applying the one-way function ($f1$) of figure 2, where the timestamp and/or random number (when used) is used to minimize replay and to conceal the password.

The protection of A's password is of the form:

Protected1 = $f1$ ($t1^A$, $q1^A$, A, passwA)

The information conveyed to B is of the form:

Authenticator1 = $t1^A$, $q1^A$, A, Protected1

B verifies the protected identifying information offered by A by generating (using the distinguished name and optional timestamp and/or random number provided by A, together with a local copy of A's password) a local protected copy of A's password (of the form Protected1). B compares for equality the purported identifying information (Protected1) with the locally generated value.

❷ B confirms or denies to A the verification of the protected identifying information.

5.4.2 The procedure of 5.4.1 can be modified to afford greater protection using $f1$ and $f2$. The main differences are as follows:

❶ A sends its additionally protected identifying information (Authenticator2) to B. Additional protection is achieved by applying a further one-way function, $f2$, as illustrated in figure 2. The further protection is of the form:

Protected2 = $f2$ (t2^A, q2^A, Protected1)

The information conveyed to B is of the form:

Authenticator2 = t1^A, t2^A, q1^A, q2^A, A, Protected2

For comparison, B generates a local value of A's additionally protected password and compares it for equality with that of Protected2 (this is similar in principle to step ❶ of 5.4.1).

❷ B confirms or denies to A the verification of the protected identifying information.

Note — The procedures defined in these clauses are specified in terms of A and B. As applied to the Directory (specified in ISO/IEC 9594-3 and ISO/IEC 9594-4), A could be a DUA binding to a DSA, B; alternatively, A could be a DSA binding to another DSA, B.

5.5 A User Password attribute type contains the password of an object. An attribute value for the user password is a string specified by the object.

```
UserPassword ::=     ATTRIBUTE
                     WITH ATTRIBUTE-SYNTAX
                     OCTET STRING (SIZE (0..ub-user-password))
                     MATCHES FOR EQUALITY
```

5.6 The following ASN.1 macro may be used to define the data type arising from applying a one-way function to a given other data type:

```
PROTECTED MACRO ::= SIGNATURE
```

SECTION 3: STRONG AUTHENTICATION

6 Basis of Strong Authentication

6.1 The approach to strong authentication taken in this part of ISO/IEC 9594 makes use of the properties of a family of cryptographic systems, known as public-key cryptosystems (PKCS). These cryptosystems, also described as asymmetric, involve a pair of keys, one secret and one public, rather than a single key as in conventional cryptographic systems. Annex B gives a brief introduction to these cryptosystems and the properties which make them useful in authentication. For a PKCS to be usable in this authentication framework at this present time, it shall have the property that both keys in the key pair can be used for encipherment, with the secret key being used to decipher if the public key was used, and the public key being used to decipher if the secret key was used. In other words, $X_p \bullet X_s = X_s \bullet X_p$, where X_p/X_s are encipherment/decipherment functions using the public/private keys of user X.

Note — Alternative types of PKCS, i.e., ones which do not require the property of permutability and that can be supported without great modification to this part of ISO/IEC 9594, are a possible future extension.

6.2 This authentication framework does not mandate a particular cryptosystem for use. It is intended that the framework shall be applicable to any suitable public key cryptosystem, and shall thus support changes to the methods used as a result of future advances in cryptography, mathematical techniques or computational capabilities. However, two users wishing to authenticate shall support the same cryptographic algorithm for authentication to be performed correctly. Thus, within the context of a set of related applications, the choice of a single algorithm shall serve to maximize the community of users able to authenticate and communicate securely. One example of a cryptographic algorithm can be found in annex C.

6.3 Authentication relies on each user possessing a unique distinguished name. The allocation of distinguished names is the responsibility of the Naming Authorities. Each user shall therefore trust the Naming Authorities not to issue duplicate distinguished names.

6.4 Each user is identified by its possession of its secret key. A second user is able to determine if a communication partner is in possession of the secret key, and can use this to corroborate that the communication partner is in fact the user. The validity of this corroboration depends on the secret key remaining confidential to the user.

6.5 For a user to determine that a communication partner is in possession of another user's secret key, it shall itself be in possession of that user's public key. Whilst obtaining the value of this public key from the user's entry in the Directory is straightforward, verifying its correctness is more problematic. there are many possible ways for doing this: clause 7 describes a process whereby a user's public key can be checked by reference to the Directory. This process can only operate if there is an unbroken chain of trusted points in the Directory between the users requiring to authenticate. Such a chain can be constructed by identifying a common point of trust. This common point of trust shall be linked to each user by an unbroken chain of trusted points.

7 Obtaining a User's Public Key

7.1 In order for a user to trust the authentication procedure, it shall obtain the other user's public key from a source that it trusts. Such a source, called a certification authority (CA), uses the public key algorithm to certify the public key, producing a *certificate*. The certificate, the form of which is specified in 7.2, has the following properties:

- any user with access to the public key of the certification authority can recover the public key which was certified;
- no party other than the certification authority can modify the certificate without this being detected (certificates are unforgeable).

Because certificates are unforgeable, they can be published by being placed in the Directory, without the need for the latter to make special efforts to protect them.

Note - Although the CAs are unambiguously defined by a distinguished name in the DIT, this does not imply that there is any relationship between the organization of the CAs and the DIT.

7.2 A certification authority produces the certificate of a user by signing (see clause 8) a collection of information, including the user's distinguished name and public key. Specifically, the certificate of a user with distinguished name A, produced by the certification authority CA, has the following form:

$$CA«A» = CA\{SN, AI, CA, A, Ap, T^A\}$$

where SN is the serial number of the certificate, AI is the identifier of the algorithm used to sign the certificate and, T^A indicates the period of validity of the certificate, and consists of two dates, the first and last on which the certificate is valid. Since T^A is assumed to be changed in periods not less than 24 hours, it is expected that systems would use Coordinated Universal Time as a reference time base. The signature in the certificate can be checked for validity by any user with knowledge of CAp. The following ASN.1 data type can be used to represent certificates:

```
Certificate           ::=    SIGNED SEQUENCE{
        version              [0]    Version DEFAULT v1988,
        serialNumber                CertificateSerialNumber,
        signature                   AlgorithmIdentifier,
        issuer                      Name,
        validity                    Validity,
        subject                     Name,
        subjectPublicKeyInfo        SubjectPublicKeyInfo}

Version        ::=     INTEGER {v1988(0)}
```

```
CertificateSerialNumber  ::=        INTEGER

Validity        ::=
        SEQUENCE{
                notBefore       UTCTime,
                notAfter        UTCTime}

SubjectPublicKeyInfo            ::=
        SEQUENCE{
                algorithm       AlgorithmIdentifier,
                subjectKey      BIT STRING,}

AlgorithmIdentifier ::=
        SEQUENCE{
                algorithm       OBJECT IDENTIFIER,
                parameters      ANY DEFINED BY algorithm
                                    OPTIONAL}
```

7.3 The directory entry of each user, A, who is participating in strong authentication, contains the certificate(s) of A. Such a certificate is generated by a Certification Authority of A, which is an entity in the DIT. A Certification Authority of A, which may not be unique, is denoted CA(A), or simply CA if A is understood. The public key of A can thus be discovered by any user knowing the public key of CA. Discovering public keys is thus recursive.

7.4 If user A, trying to obtain the public key of user B, has already obtained the public key of CA(B), then the process is complete. In order to enable A to obtain the public key of CA(B), the directory entry of each Certification Authority, X, contains a number of certificates. These certificates are of two types. First there are forward certificates of X generated by other Certification Authorities. Second there are reverse certificates generated by X itself which are the certified public keys of other certification authorities. The existence of these certificates enables users to construct certification paths from one point to another.

7.5 A list of certificates needed to allow a particular user to obtain the public key of another, is known as a *certification path*. Each item in the list is a certificate of the certification authority of the next item in the list. A certification path from A to B (denoted A→B):

- starts with a certificate produced by CA(A), namely $CA(A)«X^1»$ for some entity X^1;
- continues with further certificates $X^i«X^{i+1}»$;
- ends with the certificate of B.

A certification path logically forms an unbroken chain of trusted points in the Directory Information Tree between two users wishing to authenticate. The precise method employed by users A and B to obtain certification paths A→B and B→A may vary. One way to facilitate this is to arrange a hierarchy of CAs, which may or may not coincide with all or part of the DIT hierarchy. The benefit of this is that users who have CAs in the hierarchy may establish a certification path between them using the Directory without any prior information. in order to allow for this each CA may store one certificate and one reverse certificate designated as corresponding to its superior CA.

7.6 Certificates are held within directory entries as attributes of type **UserCertificate**, **CACertificate** and **CrossCertificatePair**. These attribute types are known to the Directory. These attributes can be operated on using the same protocol operations as other attributes. The definition of these types can be found in clause 3.3 of this part of ISO/IEC 9594; the specification of these attribute types is as follows:

```
UserCertificate      ::=     ATTRIBUTE
          WITH ATTRIBUTE-SYNTAX Certificate

CACertificate  ::=     ATTRIBUTE
          WITH ATTRIBUTE-SYNTAX Certificate

CrossCertificatePair ::=     ATTRIBUTE
          WITH ATTRIBUTE-SYNTAX CertificatePair

CertificatePair::=
                    SEQUENCE{
                              forward  [0] Certificate OPTIONAL,
                              reverse  [1] Certificate OPTIONAL
                              - - at least one shall be present - - }
```

A user may obtain one or more certificates from one or more Certification Authorities. Each certificate bears the name of the Certification Authority which issued it.

7.7 In the general case, before users can mutually authenticate, the Directory shall supply the complete certification and return certification paths. However, in practice, the amount of information which shall be obtained from the Directory can be reduced for a particular instance of authentication by:

a) if the two users that want to authenticate are served by the same certification authority, then the certification path becomes trivial, and the users unwrap each others' certificates directly;

b) if the CAs of the users are arranged in a hierarchy, a user could store the public keys, certificates and reverse certificates of all certification authorities between the user and the root of the DIT. Typically, this would involve the user in knowing the public keys and certificates of only three or four certification authorities. The user would then only require to obtain the certification paths from the common point of trust ;

c) if a user frequently communicates with users certified by a particular other CA, that user could learn the certification path to that CA and the return certification path from that CA, making it necessary only to obtain the certificate of the other user itself from the directory;

d) certification authorities can cross-certify one another by bilateral agreement. The result is to shorten the certification path;

e) if two users have communicated before and have learned one another's certificates, they are able to authenticate without any recourse to the Directory.

In any case, having learned each others' certificates from the certification path, the users shall check the validity of the received certificates.

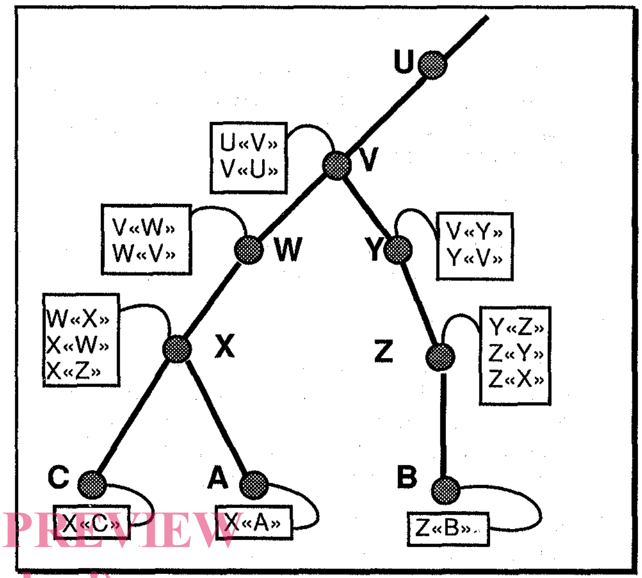


Figure 4 —CA Hierarchy – A Hypothetical Example

7.8 (Example). Figure 4 illustrates a hypothetical example of a DIT fragment, where the CAs form a hierarchy. Besides the information shown at the CAs, we assume that each user knows the public key of its certification authority, and its own public and secret keys.

7.8.1 If the CAs of the users are arranged in a hierarchy, A can acquire the following certificates from the directory to establish a certification path to B:

X«W», W«V», V«Y», Y«Z», Z«B»

When A has obtained these certificates, it can unwrap th certification path in sequence to yield the contents of th certificate of B, including Bp:

Bp = Xp • X«W» W«V» V«Y» Y«Z» Z«B»

In general A also has to acquire the following certificate: from the directory to establish the return certification patl from B to A:

Z«Y», Y«V», V«W», W«X», X«A».

When B receives these certificates from A, it can unwra the return certification path in sequence to yield the content of the certificate of A, including Ap:

Ap = Zp • Z«Y» Y«V» V«W» W«X» X«A»

7.8.2 Applying the optimizations of 7.7:

a) taking A and C, for example: both know Xp, so that A simply has to directly acquire the certificate of C. Unwrapping the certification path reduces to:

   Cp = Xp • X«C»

   and unwrapping the return certification Path reduces to:

   Ap = Xp • X«A»

b) assuming that A would thus know W«X», Wp, V«W», Vp, U«V», Up, etc., reduces the information which A has to obtain from the directory to form the certification path to:

   V«Y», Y«Z», Z«B»

   and the information which A has to obtain from the directory to form the return certification path to:

   Z«Y», Y«V».

c) assuming that A frequently communicates with users certified by Z, it can learn (in addition to the public keys learned in b) above) V«Y», Y«V», Y«Z», and Z«Y». To communicate with B, it need therefore only obtain Z«B» from the directory.

d) assuming that users certified by X and Z frequently communicate, then X«Z» would be held in the directory entry for X, and vice versa (this is shown in figure 4). If A wants to authenticate to B, A need only obtain:

   X«Z», Z«B»

   to form the certification path, and:

   Z«X»

   to form the return certification path.

e) assuming users A and C have communicated before and have learned one anothers certificates, they may use each other's public key directly, i.e.,

   Cp = Xp • X«C»

   and

   Ap = Xp • X«A»

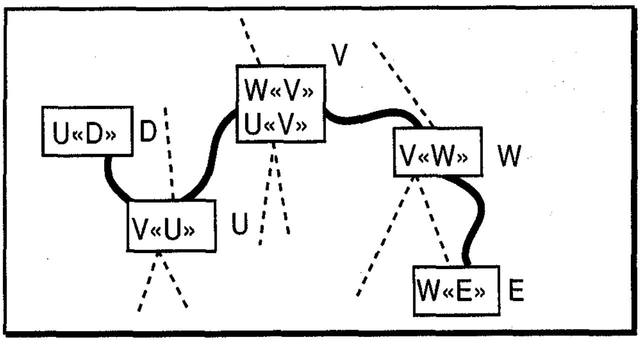


Figure 5 — Non-hierarchical Certification Path – An Example

7.8.3 In the more general case the Certification Authorities do not relate in a hierarchical manner. Referring to the hypothetical example in figure 5, suppose a user D, certified by U, wishes to authenticate to user E, certified by W. The directory entry of user D shall hold the certificate U«D» and the entry of user E shall hold the certificate W«E».

Let V be a CA with whom CAs U and W have at some previous time exchanged public keys in a trusted way. As a result, certificates U«V», V«U», W«V» and V«W» have been generated and stored in the Directory. Assume U«V» and W«V» are stored in the entry of V, V«U» is stored in U's entry, and V«W» is stored in W's entry.

User D must find a certification path to E. Various strategies could be used. One such strategy would be to regard the users and CAs as nodes, and the certificates as arcs in a directed graph. in these terms, D has to perform a search in the graph to find a path from U to E, one such being U«V», V«W», W«E». When this path has been discovered, the reverse path W«V», V«U», U«D» can also be constructed.

8 Digital Signatures

This clause is not intended to specify a standard for digital signatures in general, but to specify the means by which the tokens are signed in the Directory.

8.1 Information (info) is signed by appending to it an enciphered summary of the information. The summary is produced by means of a one-way hash function, while the enciphering is carried out using the secret key of the signer (see figure 6). Thus

X{Info} = Info, Xs[h (Info)]

Note — The encipherment using the secret key ensures that the signature cannot be forged. The one-way nature of the hash function ensures that false information, generated so as to have the same hash result (and thus signature), cannot be substituted.

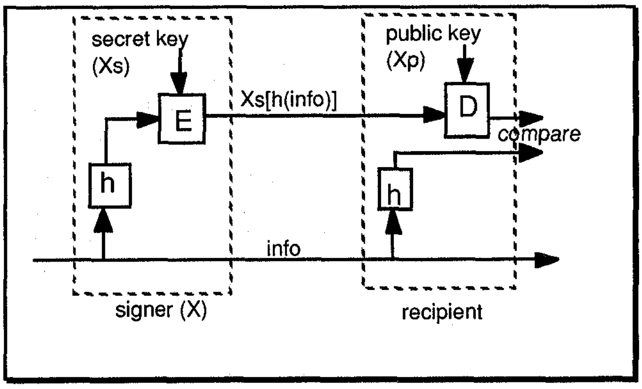


Figure 6 — Digital Signatures