

가 , 가
가 ,
3가 ,

3.

- 4.

1	1994. 2. 18	
2	1996. 1. 6	

Preface

1. Summary

This standard 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

2. Relation with other standards

This standard has its origin in the ITU-T X.509(1992) standard as a base standard and is developed as an extension to the TTA CT-X.509 standard.

3. References

- 3.1 TTA Standard : TTA CT-X509
- 3.2 ITU-T Recommendations : ITU-T X.509(1992)
- 3.3 ITU-R Recommendations : None
- 3.4 ISO Standards : ISO/IEC 9594-8

4. History

Version	Issue Date	Contents
1	1994. 2. 18	Established
2	1996. 1. 6	Revision

1

1.	-----	1
2.	-----	2
3.	-----	2
4.	-----	3
5.	-----	3

2

6.	-----	4
----	-------	---

3

7.	-----	7
8.	-----	8
9.	-----	13
10.	-----	16
11.	-----	19

A. ASN.1	-----	23
B.	-----	27
C.	-----	31
D. RSA	-----	33
E.	-----	36
F.	-----	37
G.	-----	38
H.	-----	39

CONTENTS

SECTION 1 - Introduction

1. Scope	1
2. Reference	2
3. Definition	2
4. Abbreviation	3
5. Notation	3

SECTION 2 - Simple Authentication

6. Simple Authentication Procedure	4
------------------------------------	---

SECTION 3 - Strong Authentication

7. Basis of strong Authentication	7
8. Obtaining a user's public key	8
9. Digital Signatures	13
10. Strong Authentication Procedure	16
11. Management of keys and certificates	19

Annex A - Security requirements	23
Annex B - An introduction to public key cryptography	27
Annex C - The RSA public key cryptosystem	31
Annex D - Hash functions	33
Annex E - Threats protected against by the strong authentication method	36
Annex F - Data confidentiality	37
Annex G - Authentication framework in ASN.1	38
Annex H - Reference definition of algorithm object identifiers	39

The Directory - Authentication Framework

1

1

-
-
- 가 가
- 3가 ,
-
가 ,
,
,
()
.
(DAP) , X.519
가 ,
,
가 DSA
가
C 가
7.1
,

D 가 . 가 . ,
 가 (가 , 가 .
 가 . E 가 .

2

X. 208 - OSI - 1 (ASN. 1)
 X. 219 - OSI - ,

3

X. 200 .

- a) ()
- b)
- c)
- d)
- e)
- f)
- g)
- h)
- i)
- j)
- k)
- l)
- m) ()

X. 501 .

- a)
- b)
- c)
- d)
- e)
- f)
- g)

.
 - () : .
 - () : .
 - : , , .

- : DIT .
- , : , .
 ,
- : () .
 () 가
 () .
- : y f(X)=y가 x
 () .
 x y .
- : () .
- : () .
- :
- :
- :
- : 가 가
- : CA
 CA .

4

CA
 DIB
 DIT
 PKCS

5

(1988) “1998”
 , “1992” .

X_p	X
X_s	X
$X_p [I]$	X I
$X_s [I]$	X I
$X \{I\}$	X I . 가 가 I
$CA\{x\}$	X
$CA^n(x)$	$CA(CA(...n \dots(X)))$, $n > 1$
$X_i < <X_i > >$	X_i X_i
$X_i < <X_i > > X_i < <X_i > >$	() . $X_i < <X_{i+1} > >$ $, A < > B < <C > >$ $A < <C > >$ $, A_p$ C_p .
$X_p \cdot X_i < <X_i > >$	가 () . 가 $, A \cdot A < > B < <C > >$ C_p .
$A \rightarrow B$	A B . $CA(A) < <AC^2A(A) > >$ $CA(B) < >$.

X, X_i , X_i , I

2

6

, ()
(authorization)

DUA-DSA , DSA-DSA

a) 가 ()

b) , , /

c) b) /

1 - 가

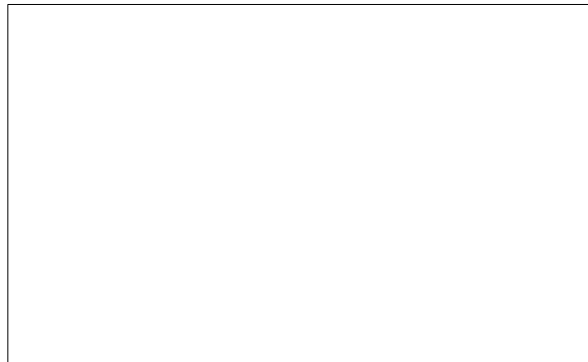
2 -

가 ,

가 1 .

가 .

- 1) A B .
- 2) B A , A
- 3) UserPassword (B) .
- 4) 가 A .



1 -

6.1

가 2 . f_1 f_2 ()

,

- 가 1) , B가
- 4) .

6.2

가 3 .

가 (f_1)

- 1) A (Authenticator 1) B . 2
- (f_1) 가 , /

A .

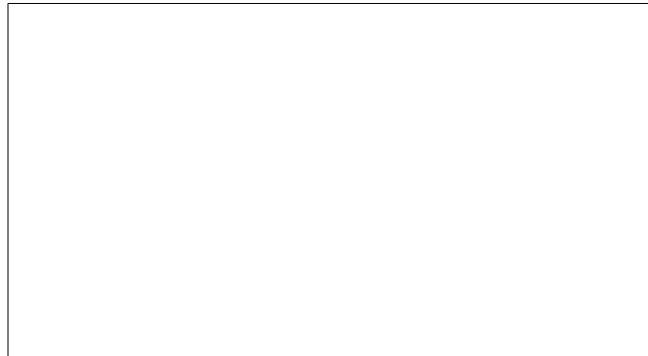
$$\text{Protected1} = f_1(t1^A, ql^A, \text{passwdA})$$

B .

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

B (A가) , / A
) A (Protected1) A가
 . B (Protected1)
 () .

2) B A .



2 -

f_1, f_2 가 .

1) A 가 (Authenticator2) B . 2
 f_2 가 . 가
 .

Protected2 = $f_2(t2^A, g2^A, Protected1)$

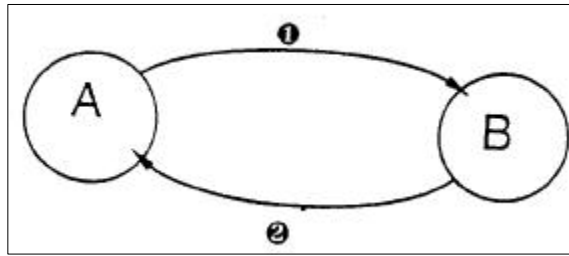
B .

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

, B A 가 Protected2
 () . (6.4.1 1) .)

2) B A .

- A, B . (X.511, X.518)
 , A DSA B DUA , A DSA B DSA
 .



3 -

6.3

가 .

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

6.4

ASN. 1

PROTECTED MACRO ::= SIGNATURE

3

7

(PKCS)

가 ,

가 .

C
 PKCS가

가

가

가

$X_0 \circ X = X \circ X_0$,

X_0/X

X

/

/

PKCS

PKCS

가

가

가
(CA)

.

.

.

. (

가 가).

가 가

- CA가 DIT , CA DIT 가

.

(9)

,

, , , 가 UCA

A 가 UA 가

가 .

CA < <A> > = CA {V, SN, AI, CA, A, UA, Ap, T^A}

, V , SN , AI

, T^A

24 가 UTC

가 CA,

ASN. 1

Certificate ::= SIGNED SEQUENCE {

version	[0] Version DEFAULT v1,
serialNunber	CertificateSerialNunber,
signature	AlgorithmIdentifier,
issuer	Name,
validity	Validity,
subject	Name,
subjectPublicKeyInfo	SubjectPublicKeyInfo,
issuerUniqueIdentifier	[1] IMPLICIT BIT STRING OPTIONAL, -- if present, version must be v2 --
subjectUniqueIdentifier	[2] IMPLICIT BIT STRING OPTIONAL -- if present, version must be v2 --}

Version ::= INTEGER {v1(0), v2(1)}

CertificateSerialNunber ::= INTEGER

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

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

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

, CA

CA가

CA가

A A

DIT A A

, CA(A) (A가) CA . A CA

, .

B A가 CA(B)

. A가 CA(B) , X

X , 2가 X

. 가

가

. . A

B (A B) .

- Xⁱ CA(A) , , CA(A) < <Xⁱ > >

- X < <Xⁱ⁺¹ > >

- B .

DIT
 A B B A A, B가
 CA , CA DIT
 CA
 , CA 1 CA
 1

UserCertificate, CACertificate, CrossCertificatePair

3.3

UserCertificate ::= ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX Certificate
 CACertificate ::= ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX CertificatePair
 CertificatePair ::= SEQUENCE {
 forward [0] Certificate OPTIONAL,
 reverse [1] Certificate OPTIONAL
 -- at least one shall be present -- }

ASN.1

Certificates ::= SEQUENCE {
 userCertificate Certificate,
 certificationPath ForwardingCertificationPath
 OPTIONAL }

CertificationPath ::= SEQUENCE {
 userCertificate Certificate,
 theCACertificates SEQUENCE OF CertificatePair
 OPTIONAL }

ASN.1

ForwardingCertificationPath ::= SEQUENCE OF CrossCertificates

8.1

가

a)

가

b)

CA가

DIT

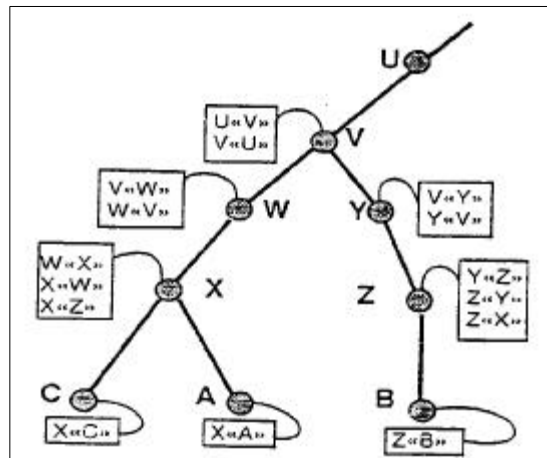
3, 4

c) CA 가 CA

d) 가

e) 가

8.2



4 - CA 가

DIT 가 가 4 , CA 가 CA

가

CA가 , A B

$X < \langle W \rangle$, $W < \langle V \rangle$, $V < \langle Y \rangle$, $Y < \langle Z \rangle$, $Z < \langle B \rangle$

, A Bp B

$Bp = Xp \cdot \langle \langle W \rangle \rangle W < \langle V \rangle \rangle V < \langle Y \rangle \rangle Y < \langle Z \rangle \rangle Z < \langle B \rangle \rangle$

, B A

A

$$Z < <Y> >, Y < <V> >, V < <W> >, W < <X> >, X < <A> >$$

$$A \quad , B \quad A_p \quad A$$

$$A_p = Z_p \cdot Z < <Y> > Y < <V> > V < <W> > W < <X> > X < <A> >$$

8.1

$$a) \quad A \quad C \vdash \quad X_p \quad , \quad A \quad C \quad .$$

$$C_p = X_p \cdot X < <C> >$$

$$A_p = X_p \cdot X < <A> >$$

$$b) \quad A \vdash W < <X> >, \forall p, V < <W> >, \forall p, U < <V> >, U_p \quad , \quad A \vdash \quad \text{가}$$

$$V < <Y> >, Y < <Z> >, Z < >$$

$$A \vdash$$

$$Z < <Y> >, Y < <V> >$$

$$c) \quad Z \quad , \quad (\quad b) \quad) \quad V < <Y> >, Y < <V> >, Y < <Z> >, \quad Z < <Y> > \quad .$$

$$B \quad Z < > \quad .$$

$$d) \quad X \quad Z \quad \text{가} \quad , \quad X < <Z> > \quad X \quad . \quad (\quad 4 \quad .) \quad A \vdash B$$

$$, \quad A \quad .$$

$$X < <Z> >, Z < >$$

$$Z < <X> >$$

$$e) \quad , \quad A \quad C$$

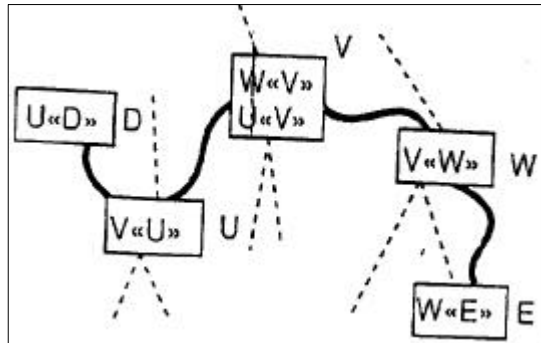
$$C_p = X_p \cdot X < <C> >$$

$$Ap = Xp \cdot X < <A> >$$

가 , U , D가 W E
 , U , D U < <U> >가 E
 W < <E> >가

V CA U 가 CA
 U < <V> >, V < <U> >, W < <V> > V < <W> >
 U < <V> > W < <V> >가 V , V < <U> >가 U , V < <W> >가 W
 가

D E
 CA (node), (arc)
 , D U < <V> >, V < <W> >, W < <E> > U E
 W < <V> >, V < <U> >, U < <D> >



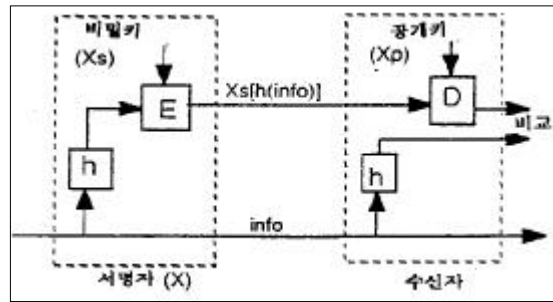
5 - -

9

(Info) 가
 , (6 .)

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

(,)



6 -

가 가 , , , 가 E 가 .

ASN. 1

```

ENCRPTED MACRO ::=
BEGIN
    TYPE NOTATION ::= type (ToBeEnciphered)
    VALUE NOTATION ::= value (VALUE BIT STRING)
END
  
```

ToBeEnciphered (ASN. 1)

1- ,

2- 가

3-

가 ,
ASN. 1 가

```

SIGNED MACRO ::=
BEGIN
    TYPE NOTATION ::= type (ToBeSigned)
    VALUE NOTATION ::= value (VALUE
        SEQUENCE {
            ToBeSigned,
            AlgorithmIdentifier,
            -- of the algorithm used to compute the signature --
            ENCRYPTED OCTET STRING
            -- where the octet string is the result of the hashing
            -- of the value of ToBeSigned -- } )
END

```

ASN.1 가 .

```

SIGNATURE MACRO ::=
BEGIN
    TYPE NOTATION ::= type (OfSignature)
    VALUE NOTATION ::= value (VALUE
        SEQUENCE {
            AlgorithmIdentifier,
            -- of the algorithm used to compute the signature --
            ENCRYPTED OCTET STRING
            -- the octet string is a function of the value --
            -- 'OfSignature', which may include the identifier --
            -- of the algorithm used to compute the signature --
        } )
END

```

END

SIGNED SIGNATURE 가 . SIGNED

SIGNATURE X.209 “ ”

가 .

a) 가 , .

b) , .

c) , .

d) SET .

e) SET OF .

f) TRUE , ‘FF’₁ .

g) 0 .

h) 8, 10 , 16 , 2

0 .

10

10.1

가 가
3가
-

3가 가 가 ,

a) 10.2 (A)가 (B)

- A A

- B B

- (2)

가

b) 10.3 B A 가가

, 가

- B A

-

- ()

c) 10.4 3 A B 가 ,

, .

, A B B A

7

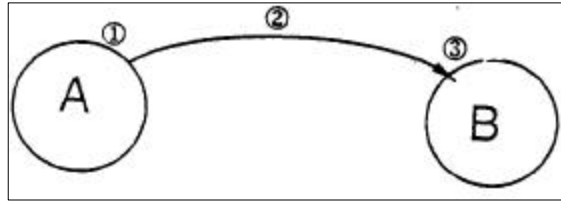
, .

UTC

3 , A

가

10.2



7 -

1) A , r^A , .

2) A B .

B A, $A\{t^A, r^A, B\}$

, t^A ()
sgnData가 , .

B A, $A\{t^A, r^A, B, \text{sgnData}\}$

encData가 , .

B A, $A\{t^A, r^A, B, \text{sgnData}, Bp[\text{encData}]\}$

encData sgnData

3) B .

a) A B A A .

b) .

c) B .

d) “ ” .

e) r^A 가 , , r^A 가 , .

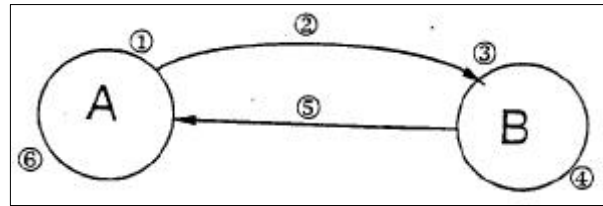
r^A t^A , r^A 가
A가 t^A r^A

B가 , , t^A

10.3

8

가



8 -

1) 10.2

2) 10.2

3) 10.2

4) B r^A

r^B

5) B

A

$B\{t^B, r^B, A, r^A\}$

, t^B t^A

sgnData가

$B\{t^B, r^B, A, r^A, \text{sgnData}\}$

encData가

$B\{t^B, r^B, A, r^A, \text{sgnData}, \text{Ap}[\text{encData}]\}$

encData

sgnData

6) A

a)

b) A

c)

t^B 가 “ ”

d)

r^B 가

. (10.2

3) d)

.)

10.4 3

9 가 .

1) 10.3 .

2) 10.3 . t^A 가 0 .

3) 10.3 . , .

4) 10.3 .

5) 10.3 . t^B 가 0 .

6) 10.3 . , .

7) A r^A 가 r^A .

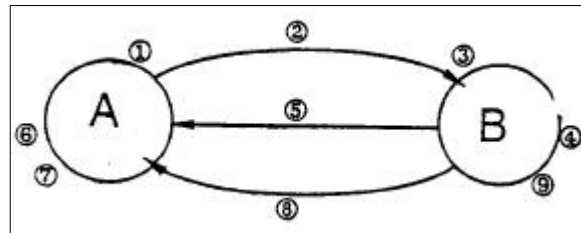
8) A B .

$A\{r^B\}$

9) B .

a) .

b) r^B 가 B가 r^B .



9 - 3

11

11.1

, .

.

, 가 ,

, () ,

PIN (

) 가 . ,

11.1.1

3 가 .

a) 가 . 가
D .

b) 3 . 3
3

c) CA . b) , 가 .

-
가 . CA 가

() 가 .

11.2

가 .

a) .

b) .

- CA 가 , / CA , CA CA

가 ,

a) 가 CA가 ,

b) 11.1 c) , .

c) 11.1 a) b)
CA (- , -)
CA 가

CA가

가 .

CA

가

CA

CA 가 가 가 CA
 가 가

CA

CA

. CA

- CA

a)

b) CA CA

- CA

- () CertificateRevocationList Authority-
RevocationList

CertificateRevocationList ::= ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX CertificateList

AuthorityRevocationList ::= ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX CertificateList

CertificateList ::= SIGNED SEQUENCE {
 signature AlgorithmIdentifier,
 issuer Name,
 lastUpdate UTCTime,
 revokedCertificates SIGNED SEQUENCE OF SEQUENCE {
 signature AlgorithmIdentifier,
 issuer Name,
 subject CertificateSerialNumber,
 revocationDate UTCTime } OPTIONAL }
 }

1 -

.

2 -

가 CA가

,

(

) CA

.

A

ASN. 1

```
ASN.1 , "AuthenticationFramework" ASN.1

AuthenticationFramework {joint-iso-ccitt ds(5) modules(1)
    authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    informationFramework, selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
        usefulDefinitiond(0)}

    Name, ATTRIBUTE, ATTRIBUTE-SYNTAX
        FROM InformationFramework informationFramework

    up-user-password
        FROM Upper Bounds upperBounds;

-- types

Certificate ::= SIGNED SEQUENCE {
    version [0] Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,
    validity Validity,
    subject Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueIdentifier [1] IMPLICIT BIT STRING OPTIONAL,
    -- if present, version must be v2
    subjectUniqueIdentifier [2] IMPLICIT BIT STRING OPTIONAL,
    -- if present, version must be v2 --}

Version ::= INTEGER {v1(0), v2(1)}

CertificateSerialNumber ::= INTEGER
```

```

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

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

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

Certificates ::= SEQUENCE {
    certificate     Certificate,
    certificationPath ForwardCertificationPath OPTIONAL }

ForwardCertificationPath ::= SEQUENCE OF CrossCertificates

CertificationPath ::= SEQUENCE {
    userCertificate      Certificate,
    theCACertificates    SEQUENCE OF CertificatePair OPTIONAL}

CrossCertificates ::= SET OF Certificate

CertificateList ::= SIGNED SEQUENCE {
    signature      AlgorithmIdentifier,
    issuer         Name,
    lastUpdate     UTCTime,
    revokedCertificates SIGNED SEQUENCE OF SEQUENCE {
        signature      AlgorithmIdentifier,
        issuer         Name,
        userCertificate SerialNumber,
        revocationDate UTCTime } OPTIONAL }

CertificatePair ::= SEQUENCE {
    forward [0]      Certificate OPTIONAL,
    revers [1]      Certificate OPTIONAL
    -- at least one of the pair must be present -- }

-- attribute types

UserCertificate ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CACertificate ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CrossCertificatePair ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificatePair

```

```
CertificateRevocationList ::= ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAX CertificateList
```

```
AuthorityRevocationList ::= ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAX CertificateList
```

```
UserPassword ::= ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAX
                               OCTETSTRING(SIZE(0..up-user-password))
                               MATCHES FOR EQUALITY
```

```
-- macros
```

```
ALGORITHM MACRO ::=
BEGIN
TYPE NOTATION ::= "PARAMETER" type
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
END
```

```
ENCRYPTED MACRO ::=
BEGIN
TYPE NOTATION ::= type(ToBeEnciphered)
VALUENOTATION ::= value(VALUE BIT STRING)
END
```

```
SIGNED MACRO ::=
BEGIN
TYPE NOTATION ::= type (ToBeSigned)
VALUE NOTATION ::= value (VALUE
                        SEQUENCE {
                        ToBeSigned,
                        AlgorithmIdentifier,
                        -- of the algorithmused to generate the signature
                        -- ENCRYPTED OCTET STRING
                        -- where the octet string is the result
                        -- of the hashing of the result "ToBeSigned"-- }
END
```

```
SIGNATURE MACRO ::=
BEGIN
TYPE NOTATION ::= type(OfSignature)
VALUE NOTATION ::= value(VALUE
                        SEQUENCE {
                        AlgorithmIdentifier,
                        -- of the algorithmused to compute the signature
                        -- ENCRYPTED OCTET STRING
                        -- where the octet string is a function (e.g.
                        -- compressed of hashed version) of the value
                        -- "OfSignature",
```

```
-- which may include the identifier of the algorithm  
-- used to compute the signature -- }
```

```
END
```

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

```
END -- of Authentication Framework Definitions
```

B

OSI , CCITT가 , CCITT가
가 . 가 .

B.1

a) (identity interception) :

b) (masquerade) : 가

c) (replay) :

d) (data interception) : 가

e) (manipulation) : 가 , , ,

f) (repudiation) :

g) (denial of service) :

- ,
.

h) (mis-routing) : 가

- OSI 1-3 . ,
.

i) (traffic analysis) : (, / , , , ,)가

- OSI
 ()

B.2

B.3 가

a) (peer entity authentication) : 가

- ()
 - , 가

b) (access control) : , 가

c) (data confidentiality) :

d) (data integrity) :

e) (non-repudiation) : 3
 - 가 가 -

B.3

B.2

a) (authentication exchange) : 2가

: 가 가

:

b) (encipherment) :

c) (data integrity) :

d) (digital signature) :

B.4

B.5

	* ()			
		*		
	*			
	* ()		* ()	*
			*	*
				*

C

가 , 가 .

(PKCS) ,

X . (Xp)

(Xs) 가 X

(D=Xs [Xp [D]]).

가 . Xp X

C-1

-
-
-
-
-
-
-
-

C.1 - PKCS

- A가 Ap As , B , Bp Bs . A B
- ,
- . (C-1).
- 1) A B x . A B x
- e B .
- $e = E_p[x]$
- 2) B x e Bs
- B Bs ,
- 가 x 가 . Bs 가 B 가 .
- .
- $x = B_s[e]$ $x = B_s[B_p[X]]$
- 3) B 가 A , Ap x' A .

$$e' = \text{Ap}[x']$$

$$4) A \quad e' \quad x' \quad .$$

$$x' = \text{As}[e'], \quad x' = \text{As}[\text{Ap}[X']]$$

$$\begin{matrix} A & B \\ A & B \end{matrix} \quad \begin{matrix} x & x' \\ x & x' \end{matrix} \quad . \quad \text{가}$$

$$\begin{matrix} A & B \\ A & B \end{matrix} \quad \begin{matrix} \text{As} & \text{Bs} \\ \text{As} & \text{Bs} \end{matrix} \quad . \quad \begin{matrix} A & B \\ A & B \end{matrix} \quad \begin{matrix} x' \\ x' \end{matrix} \quad ,$$

, B가

$$\begin{matrix} A \\ A \end{matrix} \quad . \quad \begin{matrix} B & A \\ B & A \end{matrix} \quad .$$

$$\begin{matrix} \text{PKCS} \\ X \end{matrix} \quad \begin{matrix} D = \text{Xp}[Xs[D]] \\ (Xp) \end{matrix} \quad \text{가} \quad .$$

D

D

RSA

RSA(Rivest - Shamir - Adleman)

D.1

RSA 가 ,

D.2

D.2.1 : (Public Exponent) (Arithmetic Modulus)

- BIT STRING (G) ASN.1 subjectPublicKey RSA

SEQUENCE {INTEGER, INTEGER}

, (Arithmetic Modulus)
(Public Exponent) ASN.1

D.2.2 : (Secret Exponent)

D.3

X, Y ()

n (Arithmetic Modulus)

e (Public Exponent)

d (Secret Exponent)

b, q (product) (n)

- 가 2 , 3

lcm

mod n n

D.4

$$Y = Y^e \bmod n, \quad 0 < X < n$$

$$X = Y^d \bmod n, \quad 0 < Y < n$$

$$ed \bmod \text{lcm}(p-1, p-1) = 1$$

$$ed \bmod (p-1)(p-1) = 1$$

가
(x)
(가 1)
2^{x-1} 가 2

(pad) 가 가

D.5

D.5.1

가 , 가 , 512 가 , 가

D.5.2

RSA n 가 p q 가 가
, (“ ”)

- a) .
- b) .
- c) .
- d) p-q .
- e) (p+1) 가 .
- f) (q+1) 가 .
- g) (p-1) , r 가 .
- h) (q-1) , s 가 .
- i) (r-1) 가 .
- j) (s-1) 가 .

E

.

E.1

.

.

a)

. ,
가 .

b)

. , 2
가 .

E.2

square-mode .

F

B

가 가

가

- CA

"need to know"

CA

CA

CA - CA가
가

CA가

- 가 CA
CA

가 CA

CA

CA가

가

. 3

가

G

G.1

9

G.2

가

G.3

9

가

H

ASN.1 "AlgorithmObjectIdentifiers"

AlgorithmObjectIdentifiers

{joint-iso-ccitt ds(5) modules(1) algorithmObjectIdentifiers(8) }

DEFINITIONS ::=

BEGIN

-- EXPORTS ALL --

IMPORTS

algorithm authenticationFramework

FROM UsefulDefinitions

{joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

ALGORITHM

FROM AuthenticationFramework authenticationFramework;

-- categories of object identifier

encryptionAlgorithm OBJECT IDENTIFIER ::= {algorithm 1}

hashAlgorithm OBJECT IDENTIFIER ::= {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER ::= {algorithm 3}

-- algorithms

rsa ALGORITHM

PARAMETER KeySize

::= {encryptionAlgorithm 1}

KeySize ::= INTEGER

sqMdn ALGORITHM

PARAMETER BlockSize

::= {hashAlgorithm 1}

BlockSize ::= INTEGER

sqMdnWithRSA ALGORITHM

PARAMETER BlockSize

::= {signatureAlgorithm 1}

KeyAndBlockSize := INTEGER

END