Prototype PKD
Interface Specification

2nd Edition

2 March 2005

Ministry of Economy, Trade and Industry

New Media Development Association

History: 2 March, 2005 by H.Shimada

P10: Modification of 6 Tree structure and delete Fig.2

1. LDAP Protocol

This protocol uses LDAPv3 and if necessary, access with LDAPv2 is also possible. However, when using LDAPv2, "2" must be specified in the version parameter of the BindRequest and attribute options including ";binary" must not be used.

The LDAP protocol carries out server authentication by SSL. The port number and default option of the Common Reference Directory is 636. The port number of the Common Registration Directory is 736.

The FQDN for both the Common Reference Directory and the Common Registration Directory is “p-pkd.nmda.or.jp”.

The type of LDAP operations which are supported

The type of LDAP operations which are supported are shown below.

- Bind Request / Response
- Unbind Request / Response
- Search Request
- Search ResEntry / ResDone / ResRef (only for LDAPv3)
- Search Response (only for LDAPv2)
- Modify Request / Response
- Add Request / Response
- Delete Request / Response
- ModifyDN Request / Response (only for LDAPv3)
- ModifyRDN Request / Response (only for LDAPv2)

2. Algorithms

The algorithms which can be verified by PKD are as follows.

(1)	Hash algorithm	SHA1,256,512	
(2)	Encryption algorithm	RSA or RSA-PSS	Key length 1024 ~ 4096 Bit
		DSA	Key length 1024 ~ 4096 Bit
		ECDSA	Key length 256 Bit

3. SSL Communication

The 3.0 version of the SSL communication is supported.

The Handshake Protocol for the SSL communication which is supported is stipulated in RFC2246. The option of client authentication is not used this time.

" Fig. 1　Handshake Protocol sequence" shows a flow diagram.

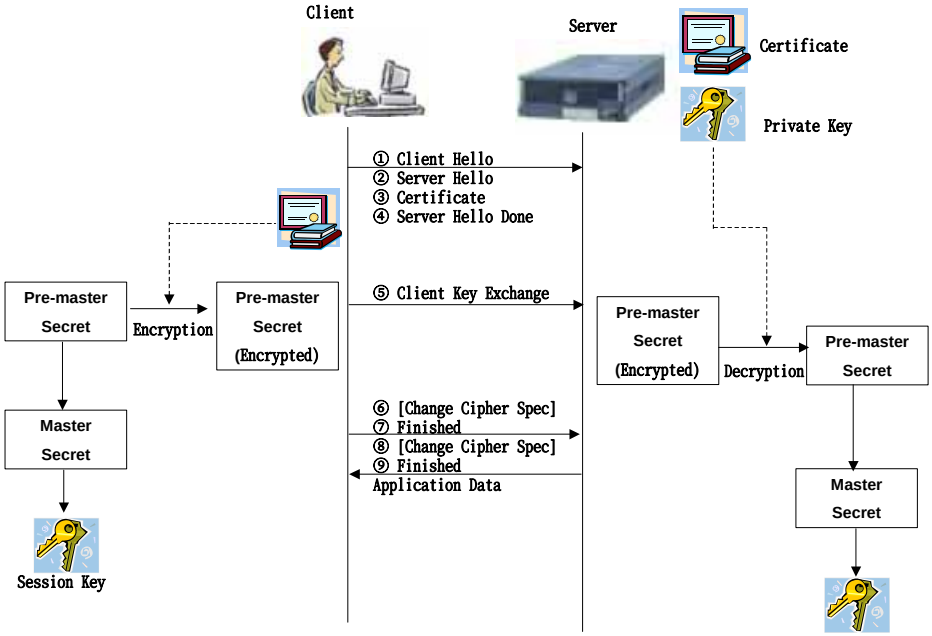

Fig. 1　Handshake Protocol sequence

The Handshake Protocol flow is explained as follows.

(1) Client Hello

Notifies the server of the start of the communication. Sends a list of encryption and compression algorithms for the client to use.

(2) Server Hello

Determines the encryption and compression algorithms which are used and notifies the client. The algorithms are chosen from the list sent by the client.

(3) Certificate

Sends the server certificate, including a list of certificates up to the root CA (certificate chain).

(4) Server Hello Done

Notifies the client that the message series beginning from Server Hello has been completed.

(5) Client Key Exchange

Generates the information – Pre-Master Secret – to create the session key used for encryption, encrypts and sends to the server. For the encryption of the Pre-Master Secret, the public key which is included in the server certificate is used.

(6) [Change Cipher Spec]

The client creates the Master Secret from the Pre-Master Secret and then creates the session key from the Master Secret.

Then, the client notifies the server that the preparation of the encryption algorithm to be used has been completed. The Change Cipher Spec is not a part of the Handshake Protocol but an independent protocol (Change Cipher Spec Protocol).

(7) Finished

Notifies the server that the exchanging of keys and authentication processing have been successful. Encrypts the message by using the exchanged session key and then sends the message.

(8) [Change Cipher Spec]

The server decrypts the encrypted Pre-Master Secret received from the client by using its own secret key. The server then creates the Master Secret from the decrypted Pre-Master Secret and creates a session key (common key) from the Master Secret. This session key is the same as the session key created by the client.

Then, the server notifies the client that the preparation of the encryption algorithm to be used has been completed. Change Cipher Spec is not a part of the Handshake Protocol but an independent protocol (Change Cipher Spec Protocol).

(9) Finished

Notifies the server that the exchanging of keys and authentication processing have been successful. Encrypts the message by using the exchanged session key and then sends the message.

After this, sends and receives the application data in an encrypted state.

Algorithms which are used

“ Table 1　Cipher suites used in the TLS” shows a list of the cipher suites which are used.

Table 1　Cipher suites used in the TLS

Cipher suite	Key exchange	Cipher	Hash
TLS_RSA_EXPORT_WITH_RC4_40_MD5	RSA_EXPORT	RC4_40	MD5
TLS_RSA_WITH_RC4_128_MD5	RSA	RC4_128	MD5
TLS_RSA_WITH_RC4_128_SHA	RSA	RC4_128	SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA_EXPORT	DES40_CBC	SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	DHE_RSA_EXPORT	DES40_CBC	SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	AES_128_CBC	SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE_RSA	AES_128_CBC	SHA
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	AES_256_CBC	SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE_RSA	AES_256_CBC	SHA

The server certificate used for the demonstration experiment was issued by the CSCA in Japan.

4. I/F for Updating

Access control is assumed to be authenticated with an ID / password entry unit for each country.

Type of LDAP operations which are supported

To the Common Registration Directory, the following LDAP operations are supported.

BindRequest

Connects when there is authentication.

BindRequest parameters are as follows:

Version	3
Name	User identification name
authenticaton.simple	User password

Bind Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.2.3

Unbind Request / Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.3

Search Request

Is assumed to be in accordance with the stipulations defined in RFC2251 4.5.1

SearchResEntry / ResDone / ResRef

Is assumed to be in accordance with the stipulations defined in RFC2251 4.5.2 and 4.5.3

Modify Request / Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.6

Add Request/Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.7

Delete Request/Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.8

Modify DN Request/Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.9

5. I/F for Referring

The Common Reference Directory is assumed to be set for only reading with an anonymous connection.

Type of the LDAP operations which are supported

To the Common Reference Directory, the following LDAP operations are supported.

BindRequest

Connects when there is no authentication.

BindRequest parameters are as follows.

Version	3
Name	“” (a character string with length 0)
authenticaton.simple	“” (a character string with length 0)

Bind Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.2.3

Unbind Request / Response

Is assumed to be in accordance with the stipulations defined in RFC2251 4.3

Search Request

Is assumed to be in accordance with the stipulations defined in RFC2251 4.5.1

SearchResEntry / ResDone / ResRef

Is assumed to be in accordance with the stipulations defined in RFC2251 4.5.2 and 4.5.3

In addition, the Directory server connection parameters are set as follows:

Maximum search number	25,000
Maximum connection number	1,024
Time limit	8 minutes

6. Tree structure

Common registration Directory executes the access control that enables only writing the home country under the control of entry (c=××) of the country by ID/password authentication. Therefore, when a new country registers in PKD, it is necessary to request additional registration of the user who does the addition and the access control of entry (c=××) of the country to the operations manager of PKD.

The Common Registration Directory, under the country's entry (o=×× Government), allows access control which enables each country to only write after ID / password authentication. Therefore, when a new country registers for the PKD, it must contact PKD and request for user registration to carry out the country's entry (o=×× Government) and access control.

Table 2 Identification attribute type and attribute value for every Directory

Hierarchy	Identification attribute type	Minimun	Maximum length	The value which can be obtained as attribute value (example)
1st Directory	c	2	2	Defines the 2-letter country code of the ISO 3166. Example: "JP"
2nd Directory	o	1	64	Example: stores the "Japanese Government"; a government name.
3rd Directory	ou	1	64	Example: stores the "e-Passport Center"; the name of the concerned authorities where a certificate is issued.
4th Directory	cn	1	64	Example: stores the "Document Signer Certificate"; the name of DS certificate. Example: stores "CRL"; the name of CRL.

7. Schema

Participating countries use the object class and attributes shown below.

(1) Country

For the object class consisting of an entry representing a country, the country object class stipulated in RFC2256 is used. A c attribute, an essential attribute of the

country object class, is also used as stipulated in RFC2256. For this attribute, a 2-letter country code of the ISO 3166 is used.

The definition of the country object class is shown in “Table 3 Definition of the country object class”.

The definition of a c attribute is in accordance with RFC2256 stipulations.

Table 3 Definition of the country object class

【Name of object class】	country
【Object identifier】	joint-iso-itu-t(2) ds(5) objectClass(6) country(2)
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	c

(2) Organization

For the object class consisting of an entry representing an organization, the organization object class stipulated in RFC2256 is used. An o attribute, an essential attribute of the organization object class, is also used as stipulated in RFC2256.

The definition of organization object class is shown in “Table 4　Definition of the organization object class”.

The definition of an o attribute is in accordance with RFC2256 stipulations.

Table 4　Definition of the organization object class

【Name of object class】	organization
【Object identifier】	joint-iso-itu-t(2) ds(5) objectClass(6) organization(4)
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	o

(3) Organizational Unit

For the object class consisting of an entry representing an organizational unit, the organizationalUnit object class stipulated in RFC2256 is used. An ou attribute, an essential attribute of organizationalUnit object class, is also used as stipulated in RFC2256.

The definition of the organizationalUnit object class is shown in “Table 5 Definition of the organizationalUnit object class”.

The definition of an ou attribute is in accordance with RFC2256 stipulations.

Table 5 Definition of the organizationalUnit object class

【Name of object class】	organizationalUnit
【Object identifier】	joint-iso-itu-t(2) ds(5) objectClass(6) organizationalUnit(5)
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	ou

(4) DS Certificate

For the object class consisting of an entry representing the DS certificate, the inetOrgPerson object class stipulated in RFC2798, the pkiUser object class stipulated in RFC2587, or the device object class stipulated in RFC2256 is used.

A cn attribute and sn attribute, essential attributes of the inetOrgPerson object class, are used as stipulated in RFC2256 and in order to store the certificate, userCertificate – an optional attribute – is used as stipulated in RFC 2256.

Since pkiUser object class is an auxiliary type, it is used by combining with other structure type object classes.

When a device object class is used, by combining with the above-mentioned pkiUser object class, it is possible to use a serialNumber attribute, an optional attribute, as stipulated in RFC2256.

The definition of the inetOrgPerson object class is shown in “Table 6 Definition of the inetOrgPerson object class”.

The definition of the pkiUserobject class is shown in “Table 7 Definition of the pkiUser object class”.

The definition of the device object class is shown in “Table 8 Definition of the device object class”.

Table 6 Definition of the inetOrgPerson object class

【Name of object class】	inetOrgPerson
【Object identifier】	2.16.840.1.113730.3.2.2
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	cn sn userCertificate

Table 7　Definition of the pkiUser object class

【Name of object class】	pkiUser
【Object identifier】	joint-iso-itu-t(2) ds(5) attributeType(4) pkiUser(21)
【Classification】	auxiliary type
【Attribute supported by the PKD for the experiment】	userCertificate

Table 8　Definition of the device object class

【Name of object class】	device
【Object identifier】	joint-iso-itu-t(2) ds(5) attributeType(4) device (14)
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	cn serialNumber

(5) CRL, ARL and Certificate

As a method of specifying where the CRL and ARL are stored, the following two kinds of application can be used.

– Using a CRL Distribution Point Extensions of the certificate

– Storing in the object class where the CA certificate is stored

In the ICAO TECHNICAL REPORT, using the CRL Distribution Point Extensions of the certificate is given as an option; if these extensions are not used, they can be stored in the object class where the CA certificate is stored. On the other hand, the ICAO TECHNICAL REPORT states that the Country Signing CA certificate is not stored within the Directory and therefore, an empty certificate object class should be used.

The ICAO TECHNICAL REPORT also states that it is possible to use a link certificate, but storing within the Directory is not mentioned; therefore, link certificates can be stored within the Directory.

■ In a case where the CRL Distribution Point Extensions of the certificate are used

For the object class consisting of an entry representing CRL and ARL, the cRLDistributionPoint object class stipulated in RFC2587 is used. A cn attribute, an essential attribute of the cRLDistributionPoint object class, is also used. In addition, in order to store CRL, certificateRevocationList, an optional attribute, is used as stipulated in RFC2256 and in order to store ARL, authorityRevocationList, an optional attribute, is used as stipulated in RFC2256.

The definition of the cRLDistributionPoint object class is shown in “Table 9 Definition of cRLDistributionPoint object class”.

The definition of a certificateRevocationList attribute and an authorityRevocationList attribute is in accordance with RFC2256 stipulations.

Table 9 Definition of the cRLDistributionPoint object class

【Name of object class】	cRLDistributionPoint
【Object identifier】	joint-iso-itu-t(2) ds(5) objectClass(6) cRLDistributionPoint(19)
【Classification】	structure type
【Attribute supported by the PKD for the experiment】	cn certificateRevocationList authorityRevocationList

■ In a case where the link certificate is registered or stored in the object class where the CA certificate is stored

For the object class consisting of an entry representing a link certificate, the pkiCA object class stipulated in RFC2587 is used. In order to store the link certificate, a cACertificate, an optional attribute, is used as stipulated in RFC2256.

In order to store CRL, certificateRevocationList, an optional attribute, is used as stipulated in the RFC2256. In order to store ARL, authorityRevocationList, an optional attribute, is used as stipulated in RFC2256.

Since pkiCA object class is an auxiliary type, it can be combined with other structure type object classes.

The definition of the pkiCA object class is shown in “Table 10 Definition of the pkiCA object class”.

The definition of a cACertificate attribute, a certificateRevocationList attribute and

an authorityRevocationList attribute is in accordance with RFC2256 stipulations.

Table 10　Definition of the pkiCA object class

【Name of object class】	pkiCA
【Object identifier】	joint-iso-itu-t(2) ds(5) attributeType(4) pkiCA(22)
【Classification】	auxiliary type
【Attribute supported by the PKD for the experiment】	cACertificate certificateRevocationList authorityRevocationList

The CRL is divided into revoked information of the certification authority – the CSCA – and revoked information of the end entity certificate – DS certificate.

To use the CRL, there are two methods.

– By dividing into the CRL – revoked information of the end entity certificate and the ARL – revoked information of the certification authority

– By putting both pieces of information into the CRL

ICAO-PKD supports both methods and is not limited to either method.

As a CRL – revoked information of end entity certificate – issuing method, there is a method to issue only difference information; this is known as Delta CRL, but is not supported.