Encryption/decryption

Summary

Encryption/decryption provides the function of encryption/decryption of data through GPKI (Government Public Key Infrastructure) in order to encode the data or to decrypt the encoded data for the sake of security. Additionally, this provides electronic signature and electronic signature confirmation function through GPKI.

Precondition

In order to use and separately use the encryption/decryption function of GPKI, the encryption/decryption module and server certificate of GPKI shall be issued through the administrative electronic signature certificate management center (Government Certification Management Authority).

For detailed information, refer to the GPKI certificate login service of common component or refer to the administrative electronic signature certificate management center (http://www.gpki.go.kr).

Description

GPKI encryption/decryption is the function of providing the encryption and decryption of data through the service and it does not provide separate screen. Only the JSP screen for test is provided.

Package Dependency

Encryption/decryption package has direct functional dependency only for the common package (cmm) of element technology.

- Dependency between packages: Security Package Dependency

Related Sources

Type	Corresponded Source	Remarks
Controller	egovframework.com.sec.pki.web.EgovGPKITestController.java	Controller class for encryption/decryption test
Service	egovframework.com.sec.pki.service.EgovGPKIService.java	Service interface for encryption/decryption
ServiceImpl	egovframework.com.sec.pki.service.impl.EgovGPKIServiceImpl.java	Service implementation class for encryption/decryption
JSP	/WEB-INF/jsp/egovframework/com/sec/pki/EgovGpkiTest.jsp	jsp page for encryption/decryption test

Configuration

The item and its environment setup for utilizing the encryption/decryption function of GPKI are as follows.

Confirmation of GPKI API installation file

First of all, for GPKI certificate login function, the GPKI API which is suitable to the system shall be applied for issuance. The standard API which has been configured at the server is for IBM AIX, therefore, it cannot be used for WINDOWS series or other UNIX system.

Configuration element of standard API

Classification	Type	File name/folder	Description
Standard API Native module	Library	libgpkiapi64.a	For IBM AIX (for administration)
Standard API Native module	Library	libgpkiapi64_jni.a	For IBM AIX (for administration)
Standard API Native module	Library	libibmldap64n.a	For IBM AIX (for civilian)
Environment file (conf)	Environment file	gpkiapi.conf	Including the information required for certificate verification
Test program (sample)	Code	/java	Cert.java, Cms.java, Crypto.java, Ivs.java, Main.java, Tsa.java, Util.java (source code)
Test program (sample)	Execution file	/class	/Sample (the data to turn the test program)Cert.class, Cms.class, Crypto.class, Ivs.class, Main.class, Tsa.class, Util.class (test program)
Standard API	jar file	libgpkiapi_jni.jar	Standard security API

Setup of class, library route

```
export GPKI_HOME=/product/jeus/egovProps/libgpkiapi
export CLASSPATH=$GPKI_HOME/libgpkiapi_jni.jar:$CLASSPATH
export LIBPATH=/product/jeus/egovProps/libgpkiapi/gpkiapi
export PATH=$PATH:/product/jeus/egovProps/libgpkiapi/gpkiapi
```

In order to use the standard API (libgpkiapi_jni.jar) for JAVA, jar file shall have been taken at the class route, and the route of JNI file which is called from the standard API for JAVA shall be taken. At this moment, this JNI file is connected with the standard security API for C/C++ and LDAP library, therefore, these 2 routes of library shall also be taken.

Location of certificate (example)

```
/product/jeus/egovProps/gpkisecureweb/certs/SVR..._env.cer
/product/jeus/egovProps/gpkisecureweb/certs/SVR..._env.key
/product/jeus/egovProps/gpkisecureweb/certs/NPKIRootCA1.der
/product/jeus/egovProps/gpkisecureweb/certs/GPKIRootCA1.der
```

Setup of property file

globals.properties

In order to designate the information for certificate, additional attribute shall be set up to the attribute file of globals.properties.

For the contents related with globals.properties, refer to the element technology property and instruction shell script section.

```
...
Globals.GPKIConfPath = /product/jeus/egovProps/conf/gpki.properties
...
```

gpki.properties (Example)

```
#-------------------------
# for GPKI LDAP access
#-------------------------
gpki.ldap.ip=ldap.gcc.go.kr
#gpki.ldap.ip=10.1.7.140
```

```
gpki.ldap.port=389
gpki.ldap.basedn=ou=Group of Server,o=Government of Korea,c=kr
gpki.ldap.attribute=usercertificate;binary

#-------------------------
# Certificate information
# Actual certificate related files are obtained by combining each of the attributes.
#-------------------------
gpki.certificate.path = /product/jeus/egovProps/gpkisecureweb/certs/
gpki.certificate.server = 1310123456
gpki.privatekey.password = test
```

Certificate for the subject system is obtained through the ldap of GPKI, and corresponding server information etc. are processed with the codes as follows.

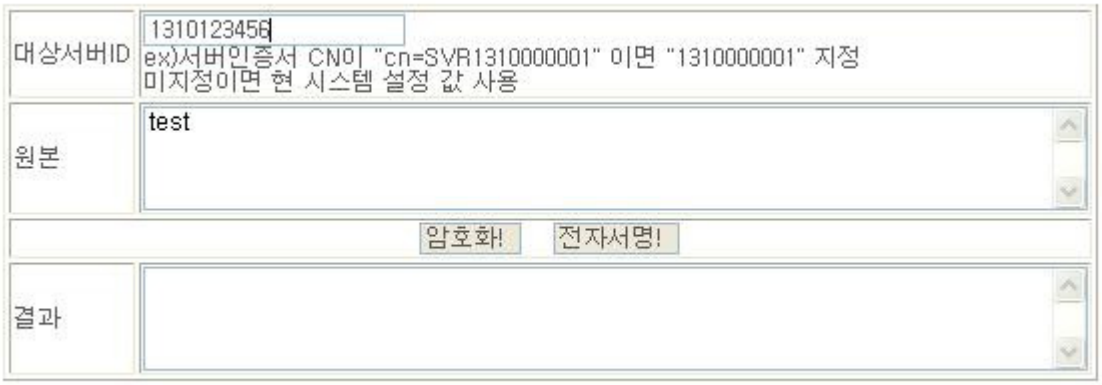
```
        //--------------------------------
        // Obtaining LDAP related information
        //--------------------------------
        String serverIp = EgovProperties.getProperty(config, "gpki.ldap.ip");
        String serverPort = EgovProperties.getProperty(config, "gpki.ldap.port");
        String basedn = EgovProperties.getProperty(config, "gpki.ldap.basedn");
        String readEntry = "cn=SVR" + code;
        String attribute = EgovProperties.getProperty(config, "gpki.ldap.attribute");
...
        //--------------------------------
        // Setup information (Certificate information for encryption is required.)
        //--------------------------------
        String path = EgovProperties.getProperty(config, "gpki.certificate.path");

        String certForEnvFile = path + "/SVR" + EgovProperties.getProperty(config,
"gpki.certificate.server") + "_env.cer";
        String keyForEnvFile = path + "/SVR" + EgovProperties.getProperty(config,
"gpki.certificate.server") + "_env.key";
        String pinForEnv = EgovProperties.getProperty(config, "gpki.privatekey.password");
...
        //--------------------------------
        // Setup information (Certificate information for electronic signature is required.)
        //--------------------------------
        String path = EgovProperties.getProperty(config, "gpki.certificate.path");

        String certForSignFile = path + "/SVR" + EgovProperties.getProperty(config,
"gpki.certificate.server") + "_sig.cer";
        String keyForSignFile = path + "/SVR" + EgovProperties.getProperty(config,
"gpki.certificate.server") + "_sig.key";
        String pinForSign = EgovProperties.getProperty(config, "gpki.privatekey.password");
```

Related screen

Test screen for encryption/decryption is as follows.



For test, decryption and electronic signature confirmation processing are provided regarding the encryption for the original and electronic signature processing and result.