Wikiprint Book

Title: revisionHerramientas:encryption

Subject: Tibisay - revisionHerramientas:encryption

Version: 7

Date: 02/05/24 11:26:01

Table of Contents

Encryption and Decryption test

- Encryption process on Android

```
        ConfigManager.init("jar://jdigidoc.cfg");
       Log.d("despues de:", "ConfigManager.init");

           /** signed doc object if used */
           SignedDoc m_sdoc;
           m_sdoc = null;

           /** encrypted data object if used */
           EncryptedData m_cdoc;
           m_cdoc = null;

           String inFile = null, outFile = null;
           String certFile = null;
           String recipient = null;
           String keyName = null;
           String carriedKeyName = null;
           String sId = null;

           inFile = str;


           outFile = "cifrado.cdoc";

           certFile = "/mnt/sdcard/chespirito.crt";


           // agregar el destinatario
           try {
                   if (m_cdoc == null){
                           Log.d("m_cdoc == null", "-");
                           m_cdoc = new EncryptedData(null, null, null, EncryptedData.DENC_XMLNS_XMLENC, EncryptedData.DEN
                   }
                   Log.d("Adding recipient", certFile);
                   X509Certificate recvCert = SignedDoc.readCertificate(new File(certFile));
                   if (recvCert != null && recipient == null)
                           recipient = SignedDoc.getCommonName(recvCert.getSubjectDN().getName());
                   Log.d("Recipient", recipient);
                   if (sId == null){
                           int n = m_cdoc.getNumKeys() + 1;
                           sId = "ID" + n;

                   }

                   EncryptedKey ekey = new EncryptedKey(sId, recipient, EncryptedData.DENC_ENC_METHOD_RSA1_5, keyName, car
                   m_cdoc.addEncryptedKey(ekey);


           }catch(Exception e){
                   Log.d("Error adding EncryptedKey: ", e.getMessage());
                   Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
           }

           // ejecutar el cifrado
           try {
                   Log.d("Encrypting file:", inFile + " to: " + outFile);
                   File fIn = new File(inFile);
                   // create a ddoc intermediate file
```

```
                m_sdoc = new SignedDoc(SignedDoc.FORMAT_DIGIDOC_XML, SignedDoc.VERSION_1_3);

                Log.d("Encrypting file:", "paso new SignedDoc");

                DataFile df = m_sdoc.addDataFile(new File(inFile), SignedDoc.xmlns_digidoc13, DataFile.CONTENT_EMBEDDED

                Log.d("Encrypting file:", "paso addDataFile");


                byte[] data = SignedDoc.readFile(new File(inFile));

                Log.d("Encrypting file:", "paso readFile");

                df.setBase64Body(data);

                Log.d("Encrypting file:", "paso setBase64Body");

                byte[] inData = m_sdoc.toXML().getBytes("UTF-8");

                Log.d("Encrypting file:", "paso toXML()");

                Log.d("Encrypting file", "Content: " + inFile + " size: " + data.length);
                Log.d("Encrypting file", "DF: " + new String(inData));

                m_cdoc.setData(inData);
                m_cdoc.setDataStatus(EncryptedData.DENC_DATA_STATUS_UNENCRYPTED_AND_NOT_COMPRESSED);
                m_cdoc.addProperty(EncryptedData.ENCPROP_FILENAME, inFile + ".ddoc");
                m_cdoc.setMimeType(EncryptedData.DENC_ENCDATA_TYPE_DDOC);
                StringBuffer sb = new StringBuffer();
                sb.append(fIn.getName());
                sb.append("|");
                sb.append(new Long(fIn.length()).toString() + " B|");
                sb.append("application/unknown|");
                sb.append("/" + fIn.getName());
                m_cdoc.addProperty(EncryptedData.ENCPROP_ORIG_FILE, sb.toString());
                //m_cdoc.addProperty(EncryptedData.ENCPROP_ORIG_SIZE, new Long(inData.length).toString());

                int nCompressOption = 0;

                m_cdoc.encrypt(nCompressOption);


                // genera el archivo cifrado en /data/data/ve.gob.cenditel/files
                FileOutputStream fos = openFileOutput(outFile, Context.MODE_PRIVATE);

                Log.d("Encrypting file", "antes de escribir archivo " + outFile);

                fos.write(m_cdoc.toXML());

                Log.d("Encrypting file", "despues de escribir archivo " + outFile);

                fos.close();

                Log.d("Encrypting file", "despues de cerrar archivo " + outFile);

                Toast.makeText(getApplicationContext(), "Cifrado correctamente: " + outFile, Toast.LENGTH_SHORT).show()


        } catch(Exception e) {
                Log.d("Error encrypting file: ", inFile + " - " + e.getMessage());
                e.printStackTrace(System.err);
```

```
                    Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
            }
```

- Decryption process with jdigidocutil (encrypted file through Android)

```
aaraujo@moe:~/desarrollo/andriod/pruebas/jdigidoc-3.7/jdigidoc$ java -jar jdigidocutil-\$\{version\}.jar -cdoc-in /tmp/cif
JDigiDoc - 3.6.0.1
Reading config file: jar://jdigidoc.cfg
Reading encrypted file: /tmp/cifrado.cdoc
Decrypting to: /tmp/salida.pdf
Using recipient: 0
SAXDigiDocFactory::readSignedDocOfType
Start reading ddoc/bdoc from file: /tmp/salida.pdf
Start Element: SignedDoc lname:  uri:
Start Element: DataFile lname:  uri:
Start collecting digest
Allocating buf: 122556 Element: DataFile lname:  uri:
Attr: ContentType ='EMBEDDED_BASE64'
Attr: Filename ='LSMDFE.pdf'
Attr: Id ='D0'
Attr: MimeType ='http://www.sk.ee/DigiDoc/v1.3.0#'
Attr: Size ='61278'
Attr: xmlns ='http://www.sk.ee/DigiDoc/v1.3.0#'
Canonicalized: '<DataFile xmlns="http://www.sk.ee/DigiDoc/v1.3.0#" ContentType="EMBEDDED_BASE64" Filename="LSMDFE.pdf" Id=
End Element: DataFile collect: 1
Digest: D0 - 5hmr2x1Q4PKrjbmkWy7K+jzYL/o= size: 61278
End Element: SignedDoc collect: 0
JDigiDoc end, time: 3 sec result: success
```