Poder Judiciário do Estado do Piauí

Assinador Digital

Para utilizar alguns dos serviços do selo digital, é necessário assinar digitalmente a requisição, isto é, atestar computacionalmente os dados garantindo integridade e autenticidade à comunicação com o Poder Judiciário.

A seguir um exemplo de como isso pode ser realizado - os exemplos a seguir foram idealizados na linguagem Java 1.8, apenas com a própria API da linguagem:

1. De posse da chave privada do certificado digital (pode ser que possua a versão Hexadecimal da mesma) é necessário transformá-la em um objeto usuável, faz-se:

```
String key = "308204BE020100300D06092A864886F70D0101010500048204A8308
byte[] encoded = hexStringToByteArray(key);
PKCS8EncodedKeySpec spec = new PKCS8EncodedKeySpec(encoded);
KeyFactory kf = KeyFactory.getInstance("RSA");
RSAPrivateKey privKey = (RSAPrivateKey) kf.generatePrivate(spec);
```

O método hexStringtoByteArray é descrito a seguir:

```
/**
 * Converte uma String hexa no array de bytes correspondente.
 *
 * @param hexa
 *            - A String hexa
 * @return O vetor de bytes
 * @throws IllegalArgumentException
 *             - Caso a String não seja uma representação haxadecimal válida
 */
public static byte[] hexStringToByteArray(String hexa) throws IllegalArgumentException {
    if (hexa.length() % 2 != 0) {
        throw new IllegalArgumentException("String hexa inválida");
    }
    byte[] b = new byte[hexa.length() / 2];
    for (int i = 0; i < hexa.length(); i += 2) {
        b[i / 2] = (byte) (("0123456789ABCDEF".indexOf(hexa.charAt(i)) << 4)
                | ("0123456789ABCDEF".indexOf(hexa.charAt(i + 1))));
    }
    return b;
}
```

2. Para a comunicação correta, o usuário deve assinar digitalmente o corpo da mensagem (campo Body), no exemplo a seguir, usou-se o método GetTiposSelo.

```
<soapenv:Body xmlns:S=\"http://schemas.xmlsoap.org/soap/envelope/\" >
   <selo:getTiposSelo/>
</soapenv:Body>
```

3. Lembrar de retirar todos os espaços em branco, quebras de linha e tabulações da mensagem anterior, além de remover os namespaces da tag Body, resultando em:

```
<soapenv:Body><selo:getTiposSelo/></soapenv:Body>
```

Observe que tudo ficou na mesma linha (espaços internos às tags foram preservados).

4. Dado o corpo da mensagem, é necessário realizar o Hash da mesma, isto é, codificá-la para que a mesma diminua de tamanho (isso por que podem existir requisições enormes no processo de Selo Digital). Para se fazer o hash de uma mensagem irá se usar o Algoritmo “SHA-256” como segue:

```
public static String hash(String data)
        throws UnsupportedEncodingException, NoSuchAlgorithmException {
    byte[] bytesOfMessage = data.getBytes("UTF-8");
    MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
    byte[] theDigest = sha256.digest(bytesOfMessage);
    return byteArrayToHexString(theDigest);
}
```

O método “byteArrayToHexString é descrito a seguir:

```java
/**
 * Converte o array de bytes em uma representação hexadecimal.
 *
 * @param b
 *            - O array de bytes a ser convertido.
 * @return Uma String com a representação hexa do array
 */
public static String byteArrayToHexString(byte[] b) {
    StringBuffer buf = new StringBuffer();

    for (int i = 0; i < b.length; i++) {
        int j = ((int) b[i]) & 0xFF;
        buf.append("0123456789ABCDEF".charAt(j / 16));
        buf.append("0123456789ABCDEF".charAt(j % 16));
    }
    return buf.toString();
}
```

5. Para assinar digitalmente um texto, pode-se realizar o seguinte procedimento, usando o algoritmo de assinatura “SHA1withRSA”:

```java
/**
 * Assina um determinado dado passado e retorna sua assinatura
 *
 * @param data
 * @param pk
 * @return a assinatura digital em formato hexadecimal ou nulo
 */
public static String assinar(String data, PrivateKey pk) {
    try {
        Signature s = Signature.getInstance("SHA1withRSA");
        s.initSign(pk);
        s.update(data.getBytes());
        byte[] sign = s.sign();
        return byteArrayToHexString(sign);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return null;
}
```

O retorno desse método é assinatura digital do elemento “data” passado como parâmetro.

6. A requisição que deve ser realizada para os serviços do selo digital final deve ser:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:selo="http://www.selo.tjpi.jus.br">
   <soapenv:Header>
      <selo:sign>
      2688788D7C32888BB9150EAD71EFB79EDE65B8A73511FABF077D15B3BDF03FB5391B89FCA
      4639A382389AA7B40E2D9DB59772B2DF7A19F481472CECEED94573921F132F835E64159FB
      4AA2B3A1B32107BDAF8DBE03823989BB952C1C0961E93E5ADF8398A8E9176E76864B5C705
      B09115CBF8CFBABAB2C4CEF63DBA788312B007BA4780EE7CD11C2599A3C73159617C41794
      229690D7510A7296BACB43120769F4FDB16DACC189A575847F1483F8F3B1CD6B66C5D1503
      07EA11BACC14CE33A9075F25AFF39F7505916E9B595F83D83CD6AAE2EA064850E68AA221E
      2FBB7654C64D482D6347CEE415D9263883035F18FA456841B5FBC7B7EFC15000F47BA8595
      5</selo:sign>
      <selo:cns>123456</selo:cns>
      <selo:user>ficticio</selo:user>
   </soapenv:Header>
   <soapenv:Body>
      <selo:getTiposSelo/>
   </soapenv:Body>
</soapenv:Envelope>
```