

Assinatura digital – arquivos XML RPS

O processo de assinatura visa manter a **integridade e autenticidade** do documento XML submetido ao webservice.

A integridade é mantida pelo DigestValue, pois caso alguma informação no documento XML seja alterada após o mesmo já ter sido assinado, o webservice ao fazer a verificação irá calcular um DigestValue diferente ocasionando a falha de integridade.

Por outro lado, **SignatureValue** e **X509Certificate** tem por fim garantir a autenticidade do documento através do processo de criptografia de chave assimétrica, ou seja, SignatureValue irá conter um HASH criptografado com a chave privada do certificado digital e X509Certificate a chave pública deste certificado para que o webservice possa fazer a decriptação.

O processo de assinatura de um documento XML é uma especificação do W3C, os mesmos possuem toda a documentação sobre o processo.

- Necessito entender o que compõe a tag **DIGESTVALUE** (o que pegar para efetuar o calculo e qual tipo de cálculo deve ser aplicado nesta tag).

É um HASH base64 codificado sobre o SHA1 da string a ser calculada.

Tomemos como exemplo o XML abaixo

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <EnviarLoteRpsEnvio xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.abrasf.org.br/nfse.xsd">
3      <LoteRps Id="Lote532526430004311001174" versao="2.02">
4          <NumeroLote>174</NumeroLote>
5          <CpfCnpj>
6              <Cnpj>000000000000100</Cnpj>
7          </CpfCnpj>
8          <InscricaoMunicipal>1001</InscricaoMunicipal>
9          <QuantidadeRps>1</QuantidadeRps>
10         <ListaRps>
11             <Rps>
12                 <InfDeclaracaoPrestacaoServico Id="Rps18131">
13                     <Rps>.....
```

Para assinar o RPS a string a ser calculada se inicia em **<InfDeclaracaoPrestacaoServico** e para assinar o Lote a string se inicia em **<LoteRps**. Observe que ao assinar o Lote todos os Signature deverão estar presentes em todos os RPS que compõem o documento.

O primeiro passo é **canonicalizar** a string. É uma especificação do W3C para simplificar (normalizar) documentos XML a fim de que os octetos utilizados para gerar o hash no lado cliente sejam os mesmos que serão utilizados pelo receptor ao aplicar o hash para conferência. Normalmente as linguagens de programação possui funções para realizar este processo de forma automática.

Com a string canonicalizada aplique o SHA1 e em seguida o base64, desta forma você terá o DigestValue.

- Necessito entender o que compõe a tag **SIGNATUREVALUE** (o que necessito pegar para efetuar o calculo e qual tipo de cálculo deve ser aplicado também aqui).

```
1   <OptanteSimplestNacional>2</OptanteSimplestNacional>
2   <IncentivoFiscal>2</IncentivoFiscal>
3   </InIDeclaracaoPrestacaoServico>
4   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
5     <SignedInfo>
6       <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
7       <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
8       <Reference URI="#Rps18131">
9         <Transforms>
10        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
11        <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
12      </Transforms>
13      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
14      <DigestValue>9oo6kOnBTAmTjAzzX17ll7bRtXo=</DigestValue>
15    </Reference>
16  </SignedInfo>
17  <SignatureValue>
18    DFScOaq1YDMqvifeploN4jyBdqoAbb1oXRsFo8wE2XKlwoT98cCZpyPepBjLBEPMSWVmz2rjDAUtyaLjYm2xeGB9oA/3bujbG+L6wn
19    vyCx86RiTaG3jsmZla4cDV79ftK2dSn4u3ZZqYU/AcnufT72gChl+fPNcwiev7osn1aY=</SignatureValue>
20
21  <X509Data>
22    <KeyInfo>
23      <X509Certificate>
MIIc2zCCAkQCCQC22aNn4hoPwTANBgkqhkiG9w0BAQUFADCbsTELMAkGA1UEBhMCQIkxDTALBgNVBAgTBERibW8xGjAYBg
NVBAcTEURibW9uc3RyYWNhbyBDaxR5MRswGQYDVQQKEjJEZW1vbN0cmFjYVW8gTHRkYS4xFTATBgNVBAstDERibW9uc3
RyYWNhbzEVMBMGA1UEAxMMRGVtb25zdHJhY2FvMSwwKgYJKoZhvcNAQkBFh1kZW1vbN0cmFjYVW9AZGVtb25zdHJhY2Fv
LmNvbTAeFw0xMzA3MjlxNDQ0MzRaFw0xODA3MjExNDQ0MzRaMIGxMQswCQYDVQQGEwJCUjENMAsGA1UECBMERGVtbz
EaMBgGA1UEBxMRRGVtb25zdHJhY2FvENpdHkxGzAZBgNVBAoTEkRibW9uc3RyYWNhbyBMdGRhLjEVMBMGA1UECxMMRG
Vtb25zdHJhY2FvMRUwEwYDVQQDEwxEZW1vbN0cmFjYVW8xLDAqBgkqhkiG9w0BCQEWHWRibW9uc3RyYWNh0BkZW1vbN
0cmFjYVW8uY29tMIGfMA0GCSqGSIb</X509Certificate>
24      </X509Data>
25    </KeyInfo>
26  </Signature>
```

Com o DigestValue já calculado você é capaz de montar “**SignedInfo**”, utilize esta como string para calcular o SignatureValue.

Novamente obtenha a forma canonicalizada, porem desta vez aplicará o SHA1 utilizando como a chave privada do seu certificado digital como chave.

Aplique o base64 e você terá SignatureValue.

Necessito entender o que compõe a tag **X509Certificate** (o que necessito pegar do certificado e se é feito algum cálculo ou só colocado na tag).

É a chave pública do seu certificado digital.