

Zaman Damgası İstemcisi Kullanım Kılavuzu

Kamu Sertifikasyon Merkezinden zaman damgası almak için "MA3 API E-İmza Kütüphane" paketini kullanabilirsiniz. Bu paket içerisindeki bütün dll/jar'lara ihtiyacınız olmayacaktır. Aşağıda belirtilen dosyalar yeterli olacaktır.

Zaman damgası güvenilir bir otoritenin zaman damgası alınmak istenen verinin özetini imzalamasıyla oluşur. İmzayı atan otorite imza içersine imzanın ne zaman atıldığı bilgisini ekler. Böylelikle zaman damgasının alındığı zamanda verinin varlığı garanti edilmiş olur.

MA3 API ile zaman damgası alınabilmesi için;

Java teknolojisi kullanılıyorsa:

asn1rt.jar, log4j.jar, ma3api-asn.jar, ma3api-common.jar, ma3api-crypto.jar, ma3api-crypto-gnuprovider.jar ve ma3api-infra.jar jarlarına

.Net teknolojisi kullanılıyorsa:

asn1rt.dll, log4net.dll, ma3api-asn.dll, ma3api-common.dll, ma3api-crypto.dll, ma3api-crypto-bouncyprowider.dll ve ma3api-infra.dll dlllerine ihtiyacınız vardır.

Alınan zaman damgasının doğrulanabilmesi için;

Java teknolojisi kullanılıyorsa :

asn1rt.jar, log4j.jar, ma3api-asn.jar, ma3api-certstore.jar, ma3api-certvalidation.jar, ma3api-cmssignature.jar, ma3api-signature.jar ma3api-common.jar, ma3api-crypto.jar, ma3api-crypto-gnuprovider.jar ve ma3api-infra.jar jarlarına

.Net teknolojisi kullanılıyorsa :

asn1rt.dll, log4net.dll, ma3api-asn.dll, ma3api-certstore.dll, ma3api-certvalidation.dll, ma3api-cmssignature.dll, ma3api-signature.dll, ma3api-common.dll, ma3api-crypto.dll, ma3api-crypto-bouncyprowider.dll ve ma3api-infra.dll dlllerine ihtiyacınız vardır.

Örnek Kullanım

Bir veriye zaman damgası alabilirsiniz veya zaman damgası alınmış verinin zaman damgasını doğrulayabilirsiniz.

Bir veriye zaman damgası almak için aşağıdaki örnek kod bloğunu kullanabilirsiniz.

Not: KULLANICI_ID, KULLANICI_PAROLA alanlarına KamuSM'den edinilen kullanıcı numarası ve parolası girilmelidir.

Java

```
FileInputStream file = new FileInputStream("T:\\MA3\\api-  
cmssignature\\testdata\\support\\policy.xml");  
byte [] digest = DigestUtil.digestStream(DigestAlg.SHA256, file);  
  
TSClient tsClient = new TSClient();  
//Test sistemi, kanuni bir geçerliliği yok.  
TSSettings settings = new TSSettings("http://tzd.kamusm.gov.tr",  
KULLANICI_ID, "KULLANICI_PAROLA", DigestAlg.SHA256);  
ETimeStampResponse response = tsClient.timestamp(digest, settings);  
  
AsnIO.dosyayaz(response.getContentInfo().getEncoded(), "T:\\MA3\\api-  
cmssignature\\testdata\\support\\policy.xml.zd");
```

C#

```
byte[] data = Encoding.ASCII.GetBytes("test");  
byte[] digest = DigestUtil.digest(DigestAlg.SHA256, data);  
TSClient tsClient = new TSClient();  
TSSettings settings = new TSSettings("http://tzd.kamusm.gov.tr", KULLANICI_ID,  
"KULLANICI_PAROLA", DigestAlg.SHA256);  
ETimeStampResponse response = tsClient.timestamp(digest, settings);  
byte[] tsBytes = response.getContentInfo().getEncoded();
```

Zaman damgası alınmış verinin zaman damgasını kontrol etmek için aşağıdaki örnek kod bloğunu kullanabilirsiniz. Yalnız zaman damgasının kontrolü için alınan zaman damgasının doğru dosya için mi alındığı ,zaman damgasının imza ve sertifika kontrolü ve zaman damgası veren sertifikanın zaman damgası vermeye yetkisinin kontrol edilmesi gerekmektedir. Bu işlemler için "MA3 API CMSSignature" paketine ve lisansına ihtiyacınız olacaktır.

Java

```
byte[] tsBytes = AsnIO.dosyadanOKU("T:\\MA3\\api-
cmssignature\\testdata\\support\\policy.xml.zd");

ETimeStampToken tsToken = new ETimeStampToken(tsBytes);
byte [] digestInTS = tsToken.getTSTInfo().getHashedMessage();

DigestAlg digestAlg =
DigestAlg.fromAlgorithmIdentifier(tsToken.getTSTInfo().getHashAlgorithm
());
FileInputStream file = new FileInputStream("T:\\MA3\\api-
cmssignature\\testdata\\support\\policy.xml");
byte [] digestOfFile = DigestUtil.digestStream(digestAlg, file);

//Zaman damgası doğru dosyaya mı ait kontrolü yapılıyor.
if(!Arrays.equals(digestInTS, digestOfFile))
    throw new Exception("Özetler uyuşmuyor. Zaman damgası bu dosyanın
değil.");

//Zaman damgası imzası ve sertifikası doğrulanıyor.
Hashtable<String, Object> params = new Hashtable<String, Object>();
params.put(EParameters.P_CERT_VALIDATION_POLICY, getPolicy());
SignedDataValidation sdv = new SignedDataValidation();
SignedDataValidationResult sdvr = sdv.verify(tsBytes, params);
if(sdvr.getSDStatus() != SignedData_Status.ALL_VALID)
    throw new Exception("Zaman damgası doğrulanamadı.");

//Zaman damgasını veren sertifikanın zaman damgası yetkisi kontrol
ediliyor.
BaseSignedData bs = new BaseSignedData(tsBytes);
ECertificate tsCert = bs.getSignerList().get(0).getSignerCertificate();
if(!tsCert.isTimeStampingCertificate())
    throw new Exception("Zaman damgası veren sertifika zaman damgası
vermeye yetkili değil.");

System.out.println("Zaman Damgası Doğrulandı.");
Calendar tsTime = tsToken.getTSTInfo().getTime();
System.out.println("Zaman Damgası Alınma Tarihi: " + tsTime.getTime());
```

C#

```
EContentInfo ci = new EContentInfo(tsBytes);
ESignedData sd = new ESignedData(ci.getContent());
ETSTInfo tstInfo = new ETSTInfo(sd.getEncapsulatedContentInfo().getContent());

byte[] digestInTS = tstInfo.getHashedMessage();
DigestAlg digestAlg = DigestAlg.fromAlgorithmIdentifier(tstInfo.getHashAlgorithm());

byte[] digest = DigestUtil.digest(digestAlg, data);

if (!Arrays.AreEqual(digest, digestInTS))
    throw new Exception("Özetler uyuşmuyor. Zaman damgası bu dosyanın değil.");

Dictionary<String, Object> params_ = new Dictionary<String, Object>();
params_[EParameters.P_CERT_VALIDATION_POLICY] = CadesSampleBase.getPolicy();

SignedDataValidation sdv = new SignedDataValidation();
SignedDataValidationResult sdvr = sdv.verify(tsBytes, params_);
if (sdvr.getSDStatus() != SignedData_Status.ALL_VALID)
    throw new Exception("Zaman damgası doğrulanamadı.");

//Zaman damgasını veren sertifikanın zaman damgası yetkisi kontrol ediliyor.

BaseSignedData bs = new BaseSignedData(tsBytes);
ECertificate tsCert = bs.getSignerList()[0].getSignerCertificate();
if(!tsCert.isTimeStampingCertificate())
    throw new Exception("Zaman damgası veren sertifika zaman damgası vermeye yetkili değil.");

Console.WriteLine("Zaman Damgası Doğrulandı.");
DateTime tsTime = tstInfo.getTime();
Console.WriteLine("Zaman Damgası Alınma Tarihi: " + tsTime.ToLocalTime());
```

Zaman Damgası Upgrade İşlemleri

Zaman damgasını imzalayan sertifikanın, geçerlilik süresi dolduktan sonra sertifika iptal verilerine erişilemeyeceği için zaman damgası sertifikası doğrulanamaz. Zaman damgası, içeriğinde zaman bilgisi taşıyan BES tipi imzaya denk gelmektedir. Zaman damgasını imzalayan sertifikanın geçerlilik süresi dolmadan ilk olarak BES tipi imza, üzerine doğrulama verileri ve yeni bir zaman damgası eklenerek XLONG tipi imzaya upgrade edilir. Böylelikle imzanın geçerlilik süresi yeni alınan zaman damgası sertifikasının bitiş tarihine kadar uzatılmış olur. Bu bitiş tarihi dolmadan XLONG tipi imza, üzerine arşiv tipi zaman damgası eklenerek ESA tipi imzaya upgrade edilmelidir. Oluşturulan ESA tipi imza ile birlikte imzanın geçerlilik süresi arşiv tipi zaman damgası sertifikasının bitiş tarihine kadar uzatılmış olur. Bu aşamadan sonra imza üzerine yeni arşiv zaman damgaları eklenerek imza geçerlilik süresi uzatılabilir.

Zaman damgası upgrade işlemleri için MA3 E-imza Kütüphanelerinin "...\\ornekler\\src\\tr\\gov\\tubitak\\uekae\\esya\\api\\timestamp\\example\\..." dizininde bulunan örnek kod kullanılabilir. İmza upgrade'leri zaman damgası sertifikalarının geçerlilik süre bitimlerine 3 ay kala yapılacak şekilde düzenlenmiştir. Kullanıcı tarafından bu süre değiştirilebilmektedir.