Elektronik İmza ve ESYA API

Orçun ERTUĞRUL

Araştırmacı
E-Posta: orcunertugrul@uekae.tubitak.gov.tr

TÜBİTAK UEKAE
ULUSAL ELEKTRONİK ve KRİPTOLOJİ ARAŞTIRMA ENSTİTÜSÜ

Sunum Planı

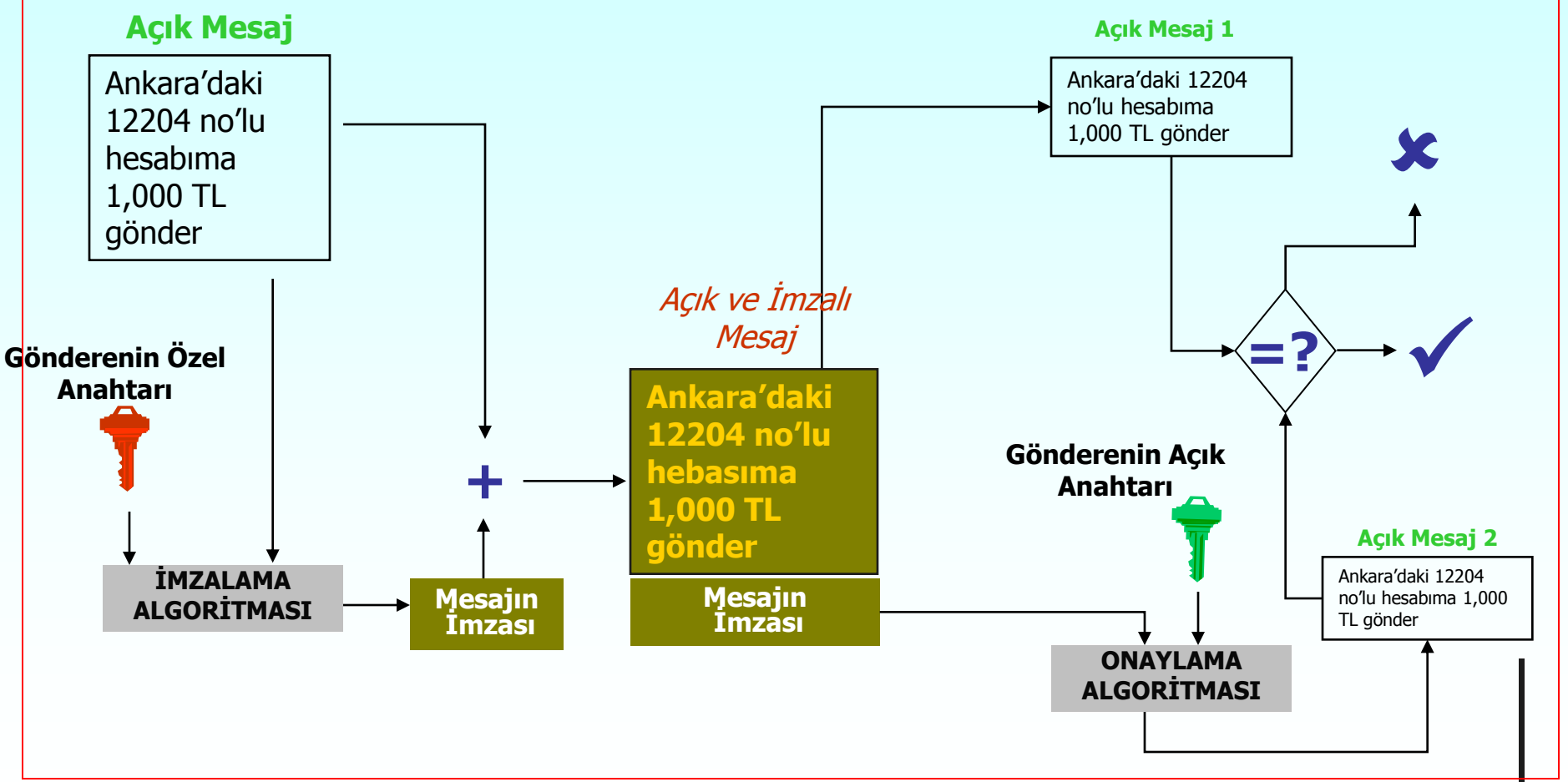
- Sayısal İmza
- Sertifikalar
- ESYA API Elektronik İmza Kütüphanesi
  - İmza Atma İşlemleri
  - İmza Doğrulama İşlemleri
  - İmza Doğrulama Sonucunun Yorumlanması
  - Akıllı Kart İşlemleri
- ESYA API Paket İçeriği
- ESYA API ile Applet Geliştirme
- C#'taki farklılıklar ve ActiveX
- Sorular

Sayısal İmza

- Mesajı gönderenin kimliğinin doğrulamasını ve mesajın bütünlüğünün kontrolünü sağlar.
- İnkar edememezlik hizmeti sağlar.
- Asimetrik kriptografi kullanır.
  - Özel anahtar – Açık Anahtar çifti
  - Gönderen özel anahtarı ile işlem yapar.
  - Alıcı gönderenin açık anahtarı ile işlem yapar.

Sayısal İmza Kullanım Senaryosu

Açık Anahtarlı İmzalama



Özetleme Fonksiyonu

Özetleme Fonksiyonu

- Büyük bir veriyi kısa bir veri ile temsil etmek.
- Sabit çıkış uzunluğu (mesajdan çok kısa)
- Mesajdaki küçük değişiklikler bile özette büyük değişikliklere yol açabilir
- Md5, Sha-1, Sha-256, Sha-512

E-imzalı Mesaj Oluşturma

Bora, Ayşe'ye Elektronik İmzalı Bir Mesajı Nasıl Gönderir?

Açık Mesaj

Ankara'daki 12204 no'lu hesabıma 1,000 TL gönder

Özetleme Algoritması

Mesaj Özeti

Özel Anahtar

Bora

Bora

Güvenli Elektronik İmza Aracı

Elektronik İmza

Gönderilen Paket

Açık Mesaj

Ankara'daki 12204 no'lu hesabıma 1,000 TL gönder

Elektronik İmza

Elektronik Sertifika

E-imzalı Mesaj Doğrulama

Ayşe, Bora'dan Gelen Elektronik İmzalı Bir Mesajı Nasıl Doğrular?

Alınan Paket

Açık Mesaj

Ankara'daki 12204 no'lu hesabıma 1,000 TL gönder

Elektronik İmza

Elektronik Sertifika

Açık Mesaj

Ankara'daki 12204 no'lu hesabıma 1,000 TL gönder

Özetleme Algoritması

Mesaj Özeti

Sertifika Kontrolleri (SİL – OCSP)

Elektronik İmza

Açık Anahtar

Bora

Onaylama Algoritması

Onaylanmış Mesaj Özeti

=?

Sertifikalar

- Sertifika sahibi hakkında bilgileri barındırırlar.
  - Adı – Soyadı, TC Kimlik Nosu, Açık Anahtarı
- Ecertificate sınıfı
  - getSubject(), kime verildiği bilgisi
- Kullanıma giriş ve bitiş tarihi vardır.
- Güvenilir bir kurum tarafından yayınlanır.
  - Bu kurum tarafından imzalanmıştır.
- Yayıncı kuruluş (Issuer) ve seri numarası (serial number) bir sertifikayı tekil yapar.
  - SubjectKeyIdentifier'da tekildir. (Zorunlu alan değil)

Açık Anahtar Sertifikaları
Örnek Sertifika

Seri No	2368
Sertifika Sahibi	Ahmet Uzun
Şirket/Kurum	Uzun A.Ş.
Yayınlayan	UEKAE
E-posta Adresi	auzun@uzun.com.tr
Yayın Tarihi	05.02.2005
Son Kullanım	05.02.2008
Açık Anahtar	2489349e894859f45489450dab45454 ca0908d8809

UEKAE Sayısal İmzası	ae89349c989893e8989548d0 823048b08023f9e903

Sertifika İptal Bilgisi

- Sertifika iptal edilmiş olabilir.
- İptal kontrolü yapılmalı.
  - Sil veya OCSP hizmetleri kullanılır.

Sertifika İptal Listeleri
Örnek SİL

Yayınlayan	**UEKAE**
Yayın Tarihi	**22.08.2007**
Son Kullanım	**24.08.2007**

İptal Olan Sertifikaların Listesi
55, 678, 2164, 3403, 4034, 5677 ....

UEKAE Sayısal İmzası	6656e345200cde989228d0823aec8b08023f9

Çevrimiçi Sertifika Durum Protokolü (ÇisDup) (Online Certificate Status Protocol=OCSP)

- SİL gerçek zamanlı değil.
- Çevrimiçi Sertifika Durum Protokolü gerçek zamanlı
  - İstemci isteği: Bu sertifika iptal edilmiş mi?
  - Boyut küçük
  - Gerçek zamanlı
  - Çevrimiçi olmak gerekiyor.

ESYA API Elektronik İmza Kütüphanesi

Elektronik İmza Standartları

- RFC 3852
  - Cryptographic Message Syntax (CMS)
- ETSI TS 101 733 V1.8.1
  - RFC 3852 üzerine hazırlanmış
  - European Telecommunications Standards Institute
    - Electronic Signatures and Infrastructures (ESI)
  - CMS Advanced Electronic Signatures (CAdES)

İmza Atma

- Sertifikaya sahip olmak gerekiyor.
  - Nitelikli Sertifika
    - Attığınız imzanın kanuni geçerliliği oluyor.
    - ECertificate sınıfı isQualified()
  - Niteliksiz Sertifika (Kurum İçi)
- Gizli anahtara sahip olmak gerekiyor
  - Donanım'da saklanabilir. (Akıllı kart, HSM)
    - Nitelikli imza atmak için kanuni şart.
  - Yazılım tabanlı (Pfx Dosyası – Parola tabanlı şifreli)
    - Örn: PfxSigner

ESYA API ile İmza Atma

- Akıllı kart için sürücüler kurulmalı. (http://kamusm.gov.tr/islemler/surucu_yukleme_servisi/)
  - Kart okuyucu sürücüsü
    - Omnikey, ASC
  - Kart sürücüsü
    - Akis, GemPlus, …
- BaseSignedData sınıfı imza yapısından sorumlu.
- İmza atma sırasında sertifika geçerlilik kontrolü yapılıyor. (Devre dışı bırakılabilir.)
  - CertificateValidationException fırlatılır.
- Örn.ler: Sign paketi altında

ESYA API ile İmza Atma

```
BaseSignedData bs = new BaseSignedData();

bs.addContent(getSimpleContent()); // Bütünleşik imza
//bs.addContent(getSimpleContent(), false); //Ayrık imza


HashMap<String, Object> params = new HashMap<String, Object>();
params.put(EParameters.P_VALIDATE_CERTIFICATE_BEFORE_SIGNING,
   false);


bs.addSigner(ESignatureType.TYPE_BES, getSignerCertificate(),
       getSignerInterface(), null, params);

byte []signature = bs.getEncoded();
```

Bütünleşik ve Ayrık İmza

- Bütünleşik İmza
  - İmza ve imzalanan içerik (content) bir arada.
  - İmzalanan içeriğin yanlışlıkla değiştirilmesi zor.
  - İmzalanan içerik java heap-size ile sınırlı.
- Ayrık imza
  - İmza ve imzalanan içerik ayrı dosyalar.
    - Zip yapısında birleştirilebilir.
    - ContentIdentifierAttr ile ilişkilendirilebilirler.
  - İmza formatından anlamayan biri bile içeriği görebiliyor.
  - Sonsuz büyüklükte dosya imzalanabilir.

İmzalı Dökümana İmza Eklenmesi

- İmzanın eklenmesi istenen imza veri yapısı BaseSignedData constructor'una verilmeli.
  - Verilen yapı imza yapısı değilse NotSignedDataException fırlatılır.
    - BaseSignedData.isSigned(InputStream) ile kontrol yapılabilir.

İmzalı Dökümana İmza Eklenmesi – Paralel İmza

```
BaseSignedData bs = new BaseSignedData(getSign());

HashMap<String, Object> params = new HashMap<String, Object>();
params.put(EParameters.P_VALIDATE_CERTIFICATE_BEFORE_SIGNING,
   false);

//Harici İmza
//File file = new File(movieFile);
//ISignable externalContent = new SignableFile(file);
//params.put(EParameters.P_EXTERNAL_CONTENT, externalContent);

bs.addSigner(ESignatureType.TYPE_BES, getSignerCertificate(),
       getSignerInterface(), null, params);

byte []signature = bs.getEncoded();
```

İmza Yapısına göre İmza Türleri

- Paralel İmza
  - Dökümanın eklendiği seviyeye eklenir
  - Atılan imza çıkartılabilir.
  - Örn: ParallelSign
- Seri İmza
  - Bir İmzacıya eklenir.
    - Eklendiği imzacının imzasi imzalanır.
  - Eğer imza çıkartılırsa, altında bulunan bütün imzalar çıkartılır.
  - Örn: SerialSign

Seri İmza Eklenmesi

```
BaseSignedData bs = new BaseSignedData(getSign());

HashMap<String, Object> params = new HashMap<String, Object>();
params.put(EParameters.P_VALIDATE_CERTIFICATE_BEFORE_SIGNING, false);

List<Signer> signerList =  bs.getSignerList();
Signer lastSigner = null;

if(signerList!=null && signerList.size() > 0)
{
   lastSigner = signerList.get(signerList.size() - 1);
   while(lastSigner.getCounterSigners().size() > 0)
        lastSigner = lastSigner.getCounterSigners().
   get(lastSigner.getCounterSigners().size()-1);
}

lastSigner.addCounterSigner(ESignatureType.TYPE_BES,
   getSignerCertificate(), getSignerInterface(), null, params);
```

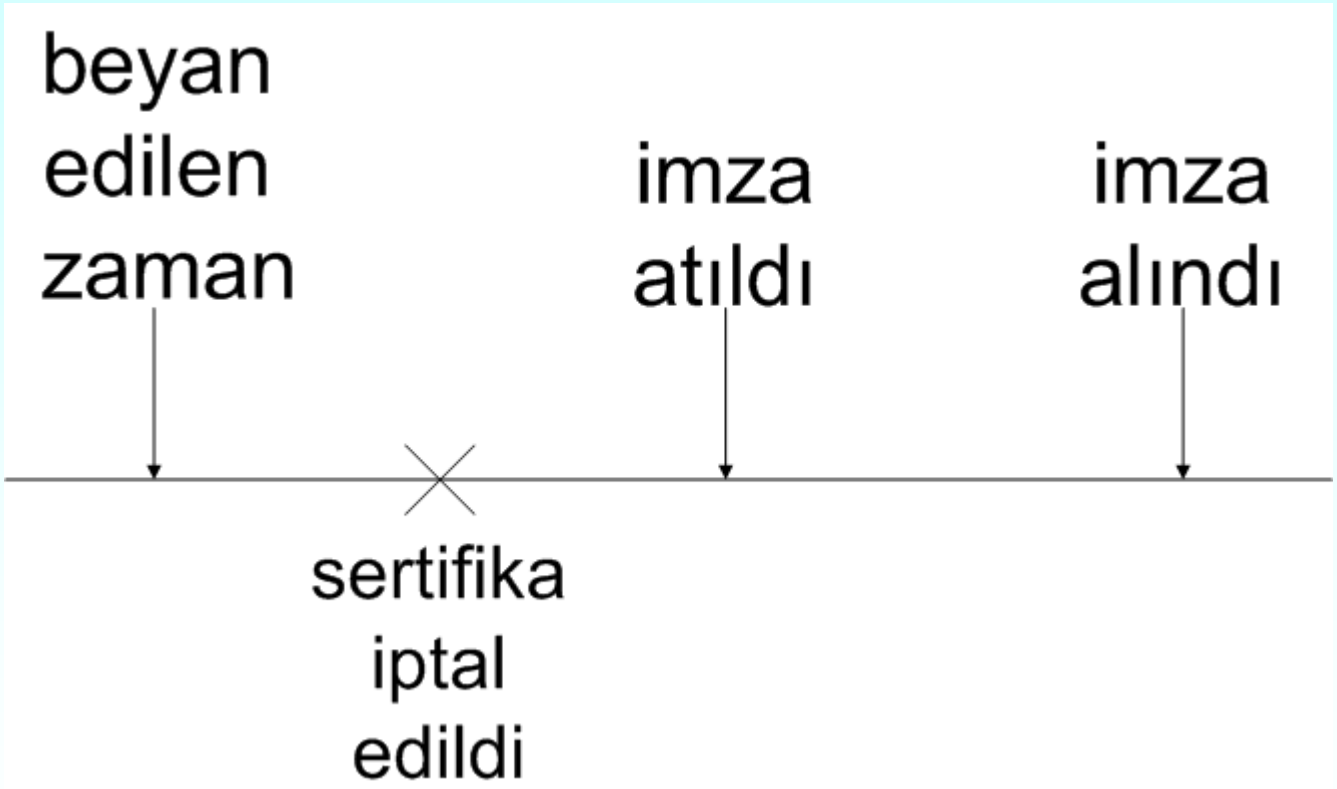
İmza Tipleri

- ETSI'nin belirlediği standartlara göre çeşitli imza tipleri vardır.
  - BES, EPES, EST, ESC, ESXLONG, ESX-Type1, ESX-Type2, ESX-Long-Type1, ESX-Long-Type2, ESA
- ETSI ve Türkiye profil dökümanlarında kullanılan imza tipleri.
  - BES, EST, ESX-Long, ESA
- Tipler arasında dönüşüm mümkün.

İmza Tipleri

- BES
  - En basit imza.
  - İmza zaman bilgisi yok.
    - Çevrimdışı imza atılabilir.
    - Beyan Edilen Zaman bilgisini içerebilir.
    - İmzayı aldığınız andaki zaman sizin için imzanın zamanıdır.
      - Beyan edilen zaman güvenilmez olabilir.

Beyan Edilen Zaman



EST

- BES + imza zamanını içerir
  - Zaman Damgası vardır.
- Kısa ömürlüdür.
  - Sertifikanın son kullanılma tarihine kadar geçerlidir.
- Geçmişte sertifika doğrulama yapılabilir.
- Çevrimiçi imza atılmalı.
- Zaman Damgası kontörü satın alınmalıdır.

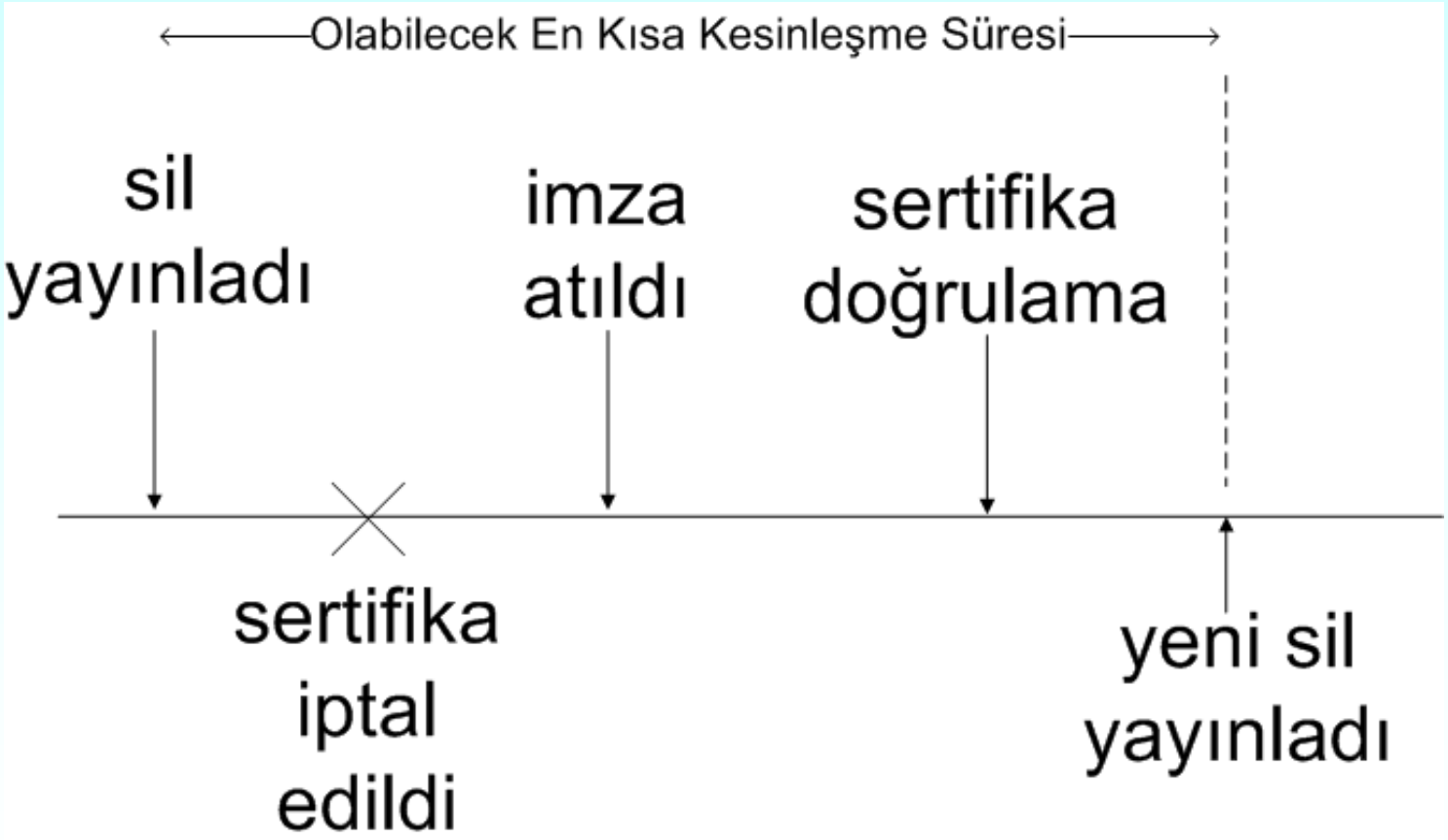
EST İmza Örneği

```
TSSettings tsSettings = new
   TSSettings("http://tzd.kamusm.gov.tr", 1901, "12345678");

params.put(EParameters.P_TSS_INFO, tsSettings);
params.put(EParameters.P_TS_DIGEST_ALG, DigestAlg.SHA1);

bs.addSigner(ESignatureType.TYPE_EST, getSignerCertificate(),
getSignerInterface(), null, params);
```

ESXLong

- EST + Sertifika doğrulama verisini içerir.
- İmza doğrulama sırasında sertifika doğrulama verisinin toplanmasına gerek yoktur.
- Uzun ömürlü.
- İmza boyutu büyür.
  - Sil kullanılırsa çok büyüyebilir.
  - OCSP kullanılırsa boyutu makül.
    - KSM sistemi için: BES (4 kb), ESX-Long (22 kb)
- Sil kullanılarak doğrudan ESXLong atılamaz.
  - Sil gerçek zamanlı değil.
    - Kesinleşme süresi geçtikten sonra doğrulama verisi eklenebilir.
  - EST'den dönüştürmek lazım.

Kesinleşme Zamanı (Grace Period)



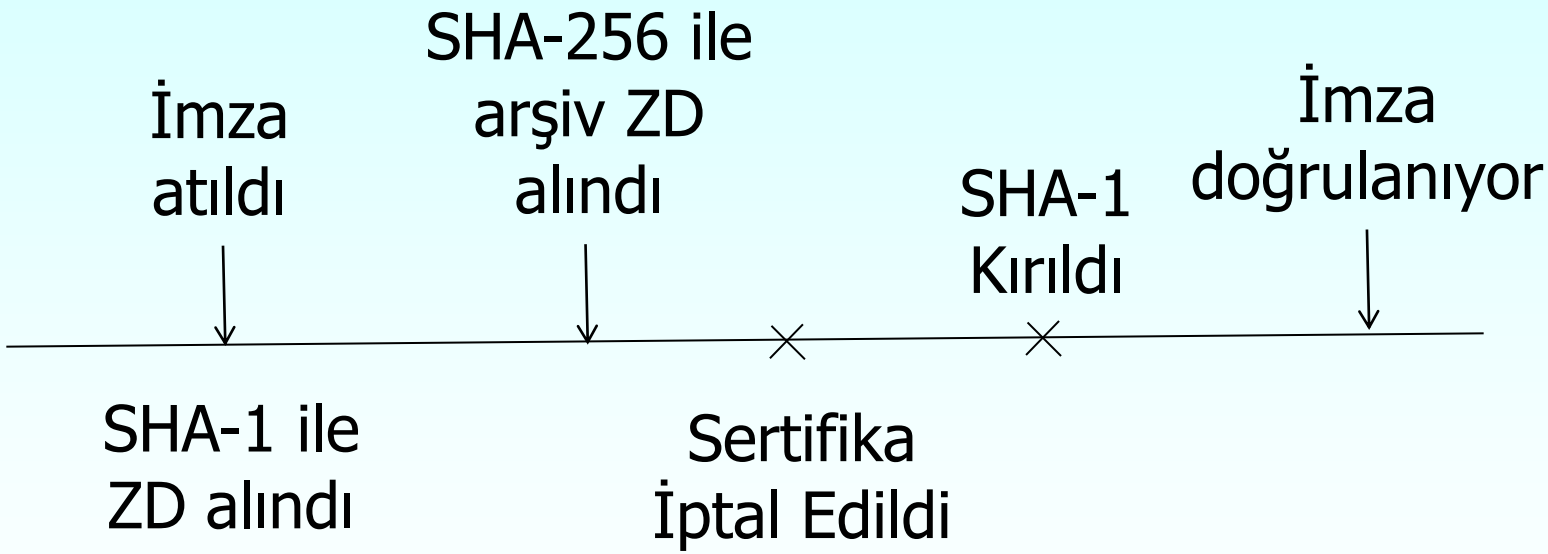
Kesinleşme Zamanı (Grace Period)

- İmza atılma zamanından sonra sertifika doğrulamanın kesin sonuç üretmesi için bir süre geçmeli.
  - Sil gerçek zamanlı değil.
  - OCSP'de bu süre yok sayılabilir.

ESA – Arşiv Tipi

- Ne zaman kullanılmalı?
  - Doğrudan ESA atmak anlamlı değil.
    - Var olan imzalar ESA'ya çevrilmeli
  - İmzada kullanılan algoritmalar kırılmadan.
  - İmzadaki son zaman damgasının sertifikası dolmadan.
- Convert etmek gerekiyor.
- Yeni çıkan algoritmalar kullanılarak arşiv tipi zaman damgası alınarak çevrilir.
  - SHA-1 ile atılmış imzalar SHA-256 ile arşivlenmeli

ESA - Gereklilik



SmartCard Modülü ile BES-İmza

- Sadece Java tarafında var.
- SmartCard modülünün sadece common jar'ına bağlantısı var.
- BES tipi imza atabiliyor.
- Sertifika doğrulama yok.
  - Sertifika Deposuna, politika deposunu ihtiyacı yok.
- Applet için uygun.
- Örn: PKCS7SignatureTest

İmza Tipi Dönüştürme

- BES-EST-ESXLong-ESA
- İmza tipi yükseltilecek *Signer* tipinde nesnenin *convert(…)* fonksiyonu çağrılır.
- EST ve ESA için zaman damgası ayarları verilmelidir.
- Örn: Convert

İmza Tipi Dönüştürmeye Örnek

```
BaseSignedData bs = new BaseSignedData(getSign());

TSSettings tsSettings = new
   TSSettings("http://tzd.kamusm.gov.tr", 1901, "12345678");

Map<String,Object> params = new HashMap<String, Object>();
params.put(EParameters.P_TS_DIGEST_ALG, DigestAlg.SHA1);
params.put(EParameters.P_TSS_INFO, tsSettings );
params.put(EParameters.P_VALIDATE_CERTIFICATE_BEFORE_SIGNING,
   false);

bs.getSignerList().get(0).convert(ESignatureType.TYPE_EST,
   params);
```

İmza Doğrulama

- İmzanın yapısal olarak doğrulanması
- Sertifika doğrulamasının yapılması
  - Sertifikanın yapısal olarak doğrulanması
    - İmzasının, zamanının geçerli olması
  - Sertifika iptal kontrolü.
  - Politika dosyası ile doğrulamanın nasıl yapılacağı tanımlanıyor.
  - Güvenilir bir kökten verilmiş olması.
- Örn: ValidationUtil

Sertifika Deposu

- Güvenilir köklerin saklandığı dosya.
  - Sqlite formatında.
  - Varsayılan yer olarak "user home" klasörünün altında, .sertifikadeposu klasöründe.
  - Politika dosyasında belirtilebilir.
- Sertifika doğrulaması için mutlaka olmalı.
- Microsoft, Firefox kendi depoları vardır.
  - Firefox: Tools→Options→View Cert.
  - Microsoft: mmc.exe (Microsoft Management Console)

Politika Dosyası

- Sertifika doğrulamayı tanımlar.
  - policy.xml
- Güvenilir bir şekilde korunmalı, değiştirilmesi önlenmeli.
  - İmzalı
  - Sunucu'da tutulabilir.
    - Özet değeri sunucuda. (*DigestUtil*)
  - Şifreli
    - Parola tabanlı olarak.

İmza Doğrulama Örneği

```
Hashtable<String, Object> params = new Hashtable<String,
   Object>();

params.put(EParameters.P_CERT_VALIDATION_POLICY, getPolicy());
if(externalContent != null)
   params.put(EParameters.P_EXTERNAL_CONTENT, externalContent);

SignedDataValidation sdv = new SignedDataValidation();
SignedDataValidationResult sdvr = sdv.verify(getSign(), params);

System.out.println(sdvr);
```

İmzalı Doküman Doğrulama Sonucu

SignedDataValidationResult nesnesinden erişilir.

- *toString()* methodu detaylı olarak imza sonuçlarını verir.
- SignedData_Status doğrulama durumunu verir.
  - *ALL_VALID:* Bütün imzalar doğrulanmış.
  - *NOT_ALL_VALID:* En az imzalardan bir tanesi doğrulanmamış.
- *getSDValidationResults(),* imzacıların sonuçlarını döner.
  - *SignatureValidationResult* listesi döner.

Herbir İmzacının Doğrulama Sonucu

SignatureValidationResult nesnesinden erişilir.

- *toString()* methodu detaylı olarak imza sonucunu verir.
- Signature_Status herbir imzanın doğrulama sonucunu verir.
  - *VALID:* Geçerli
  - *INVALID:* Geçersiz.
  - *INCOMPLETE:* Bilinmiyor. Sertifika iptal kontrolü yapılamadı.
- *getCheckerResults(),* kontrolcü sonuçlarını dönüyor.

ESYA API SmartCard Kütüphanesi

Akıllı Kart İşlemleri - 1

- İmza için gerekli olan işlemler;
  - Akıllı karttan sertifika okunması.
  - Akıllı karttan imzacının oluşturulması.
- SmartCardManager sınıfı
  - SmartCardManager.getInstance().getCertificate(...)
  - SmartCardManager.getInstance().getSigner(PIN, Sertifika)
- SmartCard sınıfı

SmartCard Sınıfı

```
Pair<Long, CardType> card = SmartOp.findCardTypeAndSlot();
Long slot = card.getObject1();
CardType cardType = card.getObject2();
System.out.println("Slot: " + slot + "Card Type: " + cardType);

SmartCard sc = new SmartCard(cardType);
long session = sc.openSession(slot);
sc.login(session, "12345");
ECertificate cert = new ECertificate(sc.getSignatureCertificates(session).get(0));

SignatureAlg signAlg = SignatureAlg.RSA_SHA256;

SCSignerWithCertSerialNo signer = new SCSignerWithCertSerialNo(sc, session, slot,
    cert.getSerialNumber().toByteArray(), signAlg.getName());
```

Java 5 – Java 6 Farkı

- Java 5
  - Karta erişim sadece dll üzerinden
  - Hangi kart olduğu API'ye söylenmeli
    - Akis, GemPlus, Tcard, …
- Java 6
  - Kartın hangi kart olduğu öğrenilebiliyor.
  - Karta komut (APDU) gönderilebiliyor. (C#'ta yok)
    - Akis için APDU ile işlem yapılabiliyor. (APDUSmartCard)
- 64 bit jre 64 bit akıllı kart sürücüsü olduğu sürece destekleniyor.
  - Extra konfigurasyon (Java)
  - Dll değişikliği (C#)

PAKET İÇERİĞİ

ESYA API ile Applet Geliştirme

- Örnek Applet uygulaması paket içersinde
  - Test lisansı değiştirilmeli.
  - BES Imza atıyor, türü değiştirilebilir.
- Applet için kullanılan jarlar imzalanmalı.