CMS Tabanlı Kütüphane Kullanarak ETSI Uyumlu Elektronik İmza Modülü Geliştirmek

Hasan GÖLLE, Şeref SAĞIROĞLU

***Özet*—Dijital imzanın matematiksel altyapısı iyi tanımlanmıştır ve açık anahtar altyapısının sağladığı kriptografik metodlar, gizlilik, bütünlük, inkar edememezlik ve kimlik doğrulamada etkin olarak kullanılabilmektedir. Veri nesnelerine uygulanan dijital imza için tanımlı genel standartlar (PKCS#7 veya CMS gibi) varsa da imzalanacak verinin formatını, imza formatını ve elektronik belge formatını tanımlayan bir standarda gereksinim vardır. Yalnızca uzun dönemli elektronik imza geçerliliğini destekleyen bir standart, farklı kullanıcı ortamlarda çalışan elektronik imza çözümlerinin birbirleri ile uyumlu çalışabilmelerine olanak sağlar. Bu noktada Telekomünikasyon Kurumu, atılan imzaların ETSI TS 101 733 standartına uygun olmasını tavsiye etmektedir. Bu bildiride CMS tabanlı MS CryptoApi kütüphanesinin sağladığı temel kriptografik metodlar kullanılarak ETSI uyumlu elektronik imzanın nasıl oluşturulacağı açıklanmıştır.**

***Anahtar kelimeler*—elektronik imza, CMS, ETSI, imzalama sertifikası.**

***Abstract*—The mathematical aspects related to the calculation of digital signatures are well defined and the cryptographic methods, provided by public key cryptography, are effective in achieving scalable confidentiality, integrity, authentication and non-repudiation services. Even though there have been defined so far general standards for defining digital signature applied on data objects, like PKCS#7 or CMS, there still exist the need for a standard entirely defining the format of the data that is to be signed, the format of the signature and the format of the electronic documents. Only the existence of a widely accepted standard for the format of electronic signature that could remain valid over long periods could avoid the appearance of incompatible specifications and solutions for electronic signatures in different user environments. At this time, telecommunication institute advices generating electronic signatures compatible with ETSI TS 101 733. In this paper, it is explained that how to generate ETSI compatible electronic signature by using basic criptographic methods provided from CMS based MS CryptoAPI .**

***Keywords*—electronic signature, CMS, ETSI, signing certificate.**

I. GİRİŞ

MS CryptoAPI [12] ile PKCS#7 / CMS standardı kullanılarak elektronik imza oluşturulmaktadır. Türkiyede Tübitak'ın geliştirmiş olduğu imzager (java ve .net) e-imza kütüphaneleri [1] ETSI uyumlu e-imza üretip doğrulamaktadırlar.

H. GÖLLE, TÜBİTAK, Ankara
Ş. SAĞIROĞLU, Gazi Üniversitrsi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara

ETSI TS 101 733 [2] de belirtilen BES (basic electronic signature) formatının PKCS#7 / CMS formatından farklı olarak sahip olduğu ek imza nitelikleri nedeniyle Ms Crypto API nin ürettiği elektronik imza, Tübitak imza kütüphanesi tarafından doğrulanamamaktadır.

ETSI TS 101 733, Cryptographic Message Syntax (CMS) üzerine yazılmış bir standartdır. CMS'nin güncel RFC'si RFC 3852'dır. ETSI ise RFC 3369'u kullanmıştır. CMS [3] standartında, atılan imzalara "Attribute" olarak adlandırılan çeşitli nitelikler eklemek mümkündür. ETSI 101 733, attribute'larla ilgili çeşitli kısıtlar koyarak 9 farklı imza formatı tanımlamaktadır. Bunlardan en basiti BES(Basic Elektronic Signature) yani Temel Elektronik İmzadır. BES imzası içinde bulunması zorunlu olan 3 attribute vardır. Bunlar CMS'de tanımlı Content-type, Message-digest attribute'ları ve RFC 2634'te [4] tanımlı signing-certificate ve other-signing-certificate attribute'larından biri olmalıdır.

MS CryptoAPI ile ETSI uyumlu imza atamamadaki problem de bu noktada ortaya çıkmaktadır. MS CryptoAPI, BES yapısı içine konması gereken 3 attribute'tan ilk ikisini imza içine koymasına rağmen, üçüncü olarak konması gereken signing-certificate veya other-signing certificate attribute'larının ikisini de koymamaktadır. Bu nedenle MS CryptoAPI içinde bu atribute'ları direkt olarak tanımlamak mümkün değil. Fakat Microsoft, imza yapısına bir attribute eklemek için gerekli alt yapıyı oluşturmuştur ve eğer encoding'i de dahil olmak üzere attribute'u oluşturup MS CryptoAPI ye verilirse istenen attribute'un eklenmesi mümkündür.

II. PKCS#7 / CMS

En önemli kriptografik standartlardan biri, RSA veri güvenliğinin çıkardığı PKCS#7 dir. Bu standart, S/MIME [5] eposta güvenliği, kredi kartı ödemeleri için (Secure Electronic Specification Transaction-SET) standartı yada gizli anahtar ve sertifikanın güvenli taşımasında kullanılan PKCS#12 standardı gibi mekanizmalarda geniş kabul görmüş ve temel alınmıştır.

PKCS#7 standartı, (Internet Engineering Task Force-IETF)'un S/MIME çalışma grubunun ürettiği ve sıradan bir mesajın dijital imzalanması, özetinin alınması, doğrulanması yada şifrelenmesi amacıyla kullanılan Kriptografik Mesaj Sözdizimi (Cryptographic Message Syntax- CMS) standardının evrimleşmiş halidir. Temelde PKCS#7 /CMS, elektronik verilere dijital imza gibi kriptografik düzenlemeler eklemek için farklı kullanışlı biçimler önermektedir. Standart, imzalama zamanı, mesaj içeriği ile birlikte doğrulanabilme, çoklu imza gibi niteliklere izin verir. Standart, PKIX çalışma grubunun [6] tanımladığı X.509 tabanlı PKI gibi farklı sertifika tabanlı mimarilere

destek verecek şekilde tasarlanmıştır. Tüm veri tipleri ASN.1 sözdizimine uygun olarak açıkça tanımlanmıştır ve şifreleme kuralları (DER ve/veya BER şifreleme nerede ve ne zaman kullanılır gibi.) da ayrıca belirlenmiştir.

Herhangi bir içeriğe dijital imza eklenmesi gerektiğinde PKCS#7/CMS standardında bulunan imzalı veri (signed-data) içerik tipini paralel olarak çok sayıda imzacı kullanabilir. Imzalanan veri formatı blobdur. PKCS#7/CMS de belge eklerine dijital imza desteklenmemiştir. Mesaj kendi içinde imza doğrulama için gerekli sertifikaları ve sertifika iptal listelerini (CRL) [7] içerebilir. Standartla ortaya çıkabilecek olumsuz bir durum da içerikte imzalayan olmadığı durumda sertifikalar ve CRL' ler anlamsızlaşabilir. Dahası veri ve imzasının *signed-data* içerik tipi içinde birarada gruplanmadığı ayrık imzalar da ortaya çıkabilir. Bu son durumda imza doğrulama işlemi uygulama bağımlıdır. Örneğin S/MIME'da tanımlı imzalı belgeler için tanımlı iki *format application/pkcs7-mime with SignedData* ve *multipart/signed'dır*. Sonuç olarak ya *multipart/signed* MIME içerik tipi[8] kullanarak ayrık imza oluşturulabilir ya da *application/pkcs7-mime with SignedData*[5] tipi kullanarak tek CMS nesnesi oluşturulabilir.

III. ETSI ELEKTRONİK İMZA STANDARDI (CADES)

Elektronik imza, imzalayanla doğrulayan arasında uzun yıllar sonra bile oluşabilecek anlaşmazlıklarda kullanılabilir. Bu standartta elektronik imza üretmede açık anahtar altyapısı baz alınmıştır. Standartta ileri elektronik imza için kullanılan format olarak ASN.1 (Abstract syntax notation 1) belirtilmiştir. Bu format CMS (Cryptographic message syntax) tabanlı olup RFC 3852 [3] de tanımlıdır. Elektronik imzalar bu nedenle CadES "CMS Advanced electonic signature" olarak adlandırılır. Şekil 1 de temel elektronik imza formatı görülmektedir.

Standartta CMS (RFC 3852[3]) tabanlı Elektronik imza formatları tanımlanmıştır. Bu bağlamda temel olarak şu tanımlara yer verilmiştir:

- İmzalayan
- Doğrulayan
- Güvenilir servis sağlayıcı (trusted service providers)
- Hakem

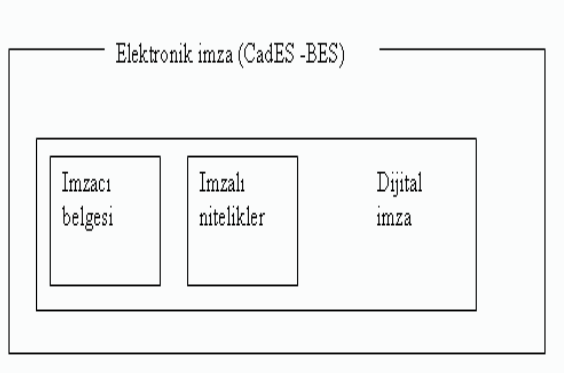
Güvenilir servis sağlayıcılar (TSPs), imzalayanla doğrulayan arasında güven ilişkisi sağlamaya yarayan varlıklardır. Bu güven ilişkisini şu hizmetleri sürdürerek sağlarlar:

- Kullanıcı sertifikaları
- Çapraz sertifikalar, zaman damgası tokenları
- CRL[7] ve OCSP[11] sorguları

Bu hizmetler aşağıdaki güvenilir servis sağlayıcıları vasıtasıyla sürdürülür:

- Sertifika makamı
- Kayıt makamı
- CRL yayınlayıcı
- OCSP sunucu
- Iptal makamı
- Zaman damgası makamı
- Time-marking makamı
- Imza politikası yayınlayıcı

Cades-BES yapısı şu şekildedir:



Şekil 1: Elektronik imza (Cades -BES) formatı

Cades-BES, Avrupa elektronik imza yönergesi yasal gereksinimlerini karşılar. Temel kimlik doğrulama ve bütünlük denetimi sağlar.

IV. CADES-BES DE BULUNMASI ZORUNLU NİTELİKLER

Aşağıdaki niteliklerin imzalı belgede bulunması zorunludur.

A İçerik tipi (Content-type)

Content-type niteliği imzalı içeriğin tipini belirtir. Content-type niteliğinin sözdizimi CMS (RFC 3852 [3]) de tanımlıdır. Content-type niteliği, signed-data yada authenticated-data içindeki ContentInfo'nun içerik tipini gösterir. Content-type niteliği imzali-veride (signed-data) imzalı nitelikler varsa yada doğrulanmış-veride (authenticated-data) doğrulanmış nitelikler varsa mutlaka bulunmalıdır. Content-type niteliği imzalı-veride ya da doğrulanmış verideki encapContentInfo eContentType değeriyle uyuşmalıdır. Content-type niteliği imzalı olmalıdır.

Şu nesne belirteci content-type niteliğini belirtir:

id-contentType OBJECT IDENTIFIER ::= { iso(1) member-body(2)
us(840) rsadsi(113549) pkcs(1) pkcs9(9) 3 }

ContentType ::= OBJECT IDENTIFIER

B Mesaj özeti (message-digest)

Mesaj özeti (Message-digest) niteliğinin sözdizimi CMS (RFC 3852 [3]) de tanımlıdır. Mesaj-özeti (message-digest) niteliği imzalı-veride (signed-data) imzalanacak encapContentInfo eContent OCTET STRING verisinin özetini gösterir. Imzalı-veri için mesaj-özeti, imzalayanın mesaj özeti algoritması ile hesaplanır. Imzalı-veride imzalı niteliklerin bulunması durumunda mutlaka mesaj özeti niteliği de bulunmalıdır. Mesaj özeti mutlaka imzalı olmalıdır.

Şu nesne belirteci content-type niteliğini belirtir:

id-messageDigest OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9) 4 }

MessageDigest ::= OCTET STRING

C İmzalama sertifikası (signing-certificate) niteliği

İmzalama sertifikası (signing-certificate) niteliği, ESS imzalama-sertifikası (ESS signing-certificate) yada ESS imzalama-sertifikası-v2 (ESS signing-certificate-v2) ile desteklenmektedir. Bu niteliklerde imzalayan sertifikasına referans bulunmalıdır ve bu nitelikler yerine koyma ve yeniden yayınlama saldırılarına karşı koyan ve imza doğrulamada kısıtlı sayıda sertifikaya izin veren yapıda tasarlanmalıdır. Tam bir sertifikadan daha yoğun ve küçük bir yapıdadırlar ve kolayca ayırdedilebilirler.

- İmzalama sertifikası (ESS signing certificate) niteliği (Enhanced Security Services -ESS) RFC 2634 [4] de tanımlanmıştır ve özet algoritması olarak yalnızca SHA-1 e izin verir.
- İmzalama sertifikası-V2 (ESS signing certificate-V2) niteliği ise (ESS update: Adding CertID algorithm agility) RFC 5035 [9] de tanımlanmıştır ve diğer özet algoritmaları kullanıldığında kullanılmalıdır.

Imza doğrulamada kullanılacak sertifika, sertifika zincirinde bulunmalıdır ve sertifika zinciri de boş olmamalıdır. Signing-certificate RFC 2634 [4] (Enhanced Security Services for S/MIME) içinde şöyle tanımlanıyor:

```
SigningCertificate ::= SEQUENCE {
        certs SEQUENCE OF ESSCertID,
        policies SEQUENCE OF PolicyInformation OPTIONAL
}

id-aa-signingCertificate OBJECT IDENTIFIER ::= { iso(1)
        member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
        smime(16) id-aa(2) 12 }


ESSCertID ::= SEQUENCE {
        certHash Hash,
        issuerSerial IssuerSerial OPTIONAL
}

Hash ::= OCTET STRING -- SHA1 hash of entire certificate

IssuerSerial ::= SEQUENCE {
        issuer GeneralNames,
        serialNumber CertificateSerialNumber
}
```

Sertifika zincirindeki ilk sertifika imza doğrulama sertifikası olmalıdır. Bu sertifika için EssCertID kodlamasında issuerSerial alanı mutlaka bulunmalıdır. Böylece imzayı atan kişinin sertifikasına ulaşmak mümkün olabilir. İmzalamaSertifikası (SigningCertificate) niteliği mutlaka imzalı olmalıdır.

V. ASN1 ENCODE

Bu yapıyı encode edebilmek için DER encode fonksiyonları hazırlamak gereklidir. Bu fonksiyonlar MS CryptoAPI'de [12] tanımlı bir veri yapısı olan CRYPT_ATTR_BLOB üzerinde işlem yapmaktalar. Programda kullanılan EncodeLen, MakeTLV ve BasaTagEkle metodları öncelikle oluşturulmuştur. TLV, Tag, Length, Value kelimelerinin baş harflerinden oluşmaktadır. Der kodlaması böyle bir yapı kullanmaktadır.

VI. ETSI UYUMLU ELEKTRONİK İMZA OLUŞTURMAK İÇİN KULLANILAN ALGORİTMA

Ms CryptoApi'nin[12] sağladığı en temel kriptografik medodlar kullanılarak şu işlem adımları ile ETSI uyumlu elektronik imza gerçekleştirilebilir. Şekil 2 de bu işlem adımları gösterilmektedir.

Değişkenleri tanıtıp ilk değer atadıkdan sonra kullanıcıya sertifika seçtirilir.

- Sertifika store açmak için *CertOpenSystemStore* kullanılır.
- Store'daki sertifikaları liste halinde gösterip kullanıcının seçmesine imkan tanımak için *CryptUIDlgSelectCertificateFromStore* kullanılır.

Seçilen sertifika için iptal kontrolü yapılır. Sertifika geçerliyse devam edilir.

Microsoft'un ekleyemediği *signingCertificate* attribute oluşturulur. Bu attribute RFC 2634'de [4] tanımlıdır.

- Sertifika icinden blob olarak issuer'i ve serial'i alınır.
- Blob halindeki issuerdan *GeneralNames* yapisi olusturulur.
- Blob halindeki serial'dan *CertificateSerialNumbe*r=INTEGER yapisi olusturulur.
- *IssuerSerial* yapisi olusturulur.
- Hash'i hesaplamak ve Hash yapısını olusturmak için *CryptAcquireContext, CryptCreateHash, CryptHashData, CryptGetHashParam, CryptDestroyHash, CryptReleaseContext* metodları kullanılır.
- Hash ve *IssuerSerial* kullanarak *ESSCertID* yapısı olusturulur.
- *SigningCertificate* yapısı elde edilir.

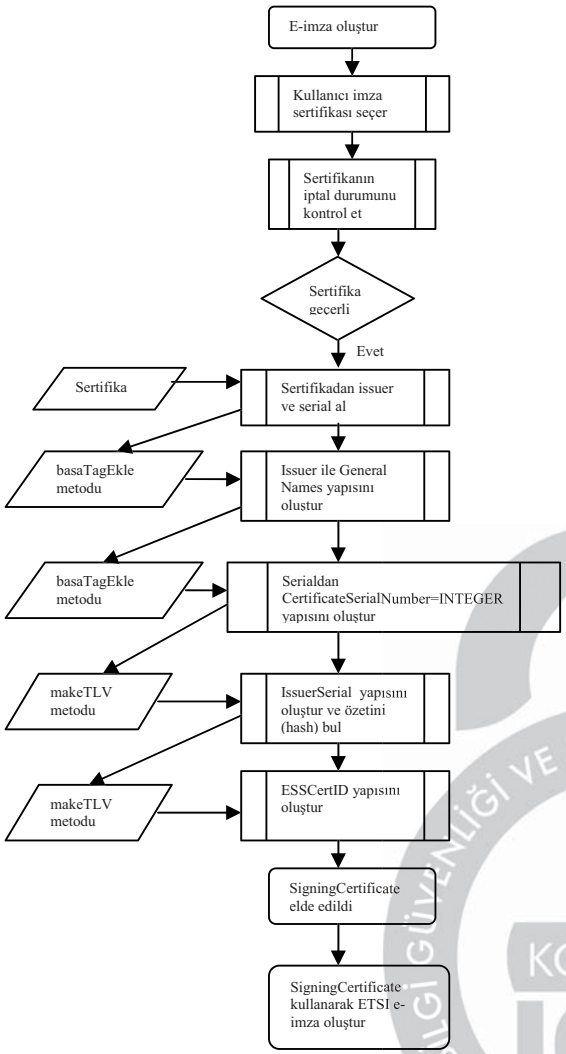
İmza sonucunun kaç byte olduğunu belirlemek için *CryptSignMessage* kullanılır.

- CryptSignMessage(&SigParams,false,1,MessageArray, MessageSizeArray,NULL, &DataSize)

İmzayı atmak için *CryptSignMessage* kullanılır.

- CryptSignMessage(&SigParams,false,1,MessageArray,MessageSizeArray,sonuc->pbData,&DataSize))

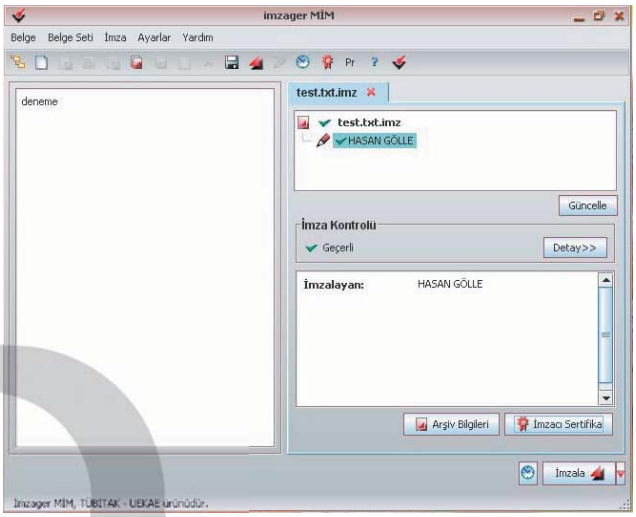
Her iki işlem için *CryptSignMessage* metodu kullanılıyor fakat girilen parametreler nedeniyle birinde imza büyüklüğü alınırken diğerinde imza atılıyor.



Şekil 2: ETSI uyumlu imzada kullanılmak üzere SigningCertificate attribute oluşturulması

VII. SONUÇ

Yukarda işlem adımlarını gerçekleştirerek oluşturduğum elektronik imza modülü, dll uzantılı bir kütüphane programıdır. Microsoft .Net C++ ortamında, Ms CryptoAPI kullanılarak hazırlanmıştır. Bu kütüphane kullanılarak ETSI uyumlu elektronik imza oluşturulabilir ve oluşturulan imzaların iptal kontrolleri yapılabilir. Masaüstü ve web tabanlı kullanıcı arayüzü yazılımlarında ve uygulama yazılımlarında bu kütüphanenin dışarıdan kullanılması (import edilmesi) yeterlidir. Uygulama geliştiren kişinin karmaşık kriptografik algoritmaları bilmesi ve kullanması gerekmeden güvenli elektronik imza uygulaması geliştirebilir. Sonuçta elde edilen imzalı dosyalar Tübitak'ın dağıttığı Imzager MIM [10] ile açılmış ve imzaların doğrulandığı ve ETSI standardına uygun olduğu görülmüştür. Şekil 3 de bu programın ekran görünümü verilmiştir.



Şekil 3. Elektronik imzalı belgeler Tübitak Imzager programı ile başarılı olarak doğrulanmıştır.

KAYNAKLAR

[1] Imzager API [Online] Available: http://www.kamusm.gov.tr/tr/Urunler/Eimza/
[2] ETSI TS 101 733 V1.5.1 (2003-7 12) "Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats".
[3] IETF RFC 3852 (2004): "Cryptographic Message Syntax (CMS)".
[4] IETF RFC 2634 (1999): "Enhanced Security Services for S/MIME".
[5] B. Ramsdell, "S/MIME Version 3 Message Specification", RFC-2633, 1999
[6] [Online] Available: http://www.ietf.org/html.charters/pkix-charter.html
[7] R. Housley, et al., "Internet X. 509 Public Key Infrastructure Certificate and CRL Profile", RFC-2459, 1999
[8] J. Galvin, et al., "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC-1847, 1995
[9] IETF RFC 5035 (2007): "ESS Update: Adding CertID Algorithm Agility"
[10] [Online] Available: Imzager MIM, http://www.kamusm.gov.tr/tr/Urunler/Imzager/
[11] M. Myers, et al., "X509 Internet Public Key Infrastructure Online Certificate Status Protocol –OCSP", RFC-2560, 1999
[12] [Online] Available: http://msdn.microsoft.com/en-us/library/aa380256(VS.85).aspx