

# Çoklu Algoritma Desteğine Dayalı E-İmza Uygulaması (E-Signat)

**Mehmet Akif Çakar, Tuncay Yiğit**

Gazi Üniversitesi, Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Eğitimi Bölümü, 06500, Ankara  
makif\_cakar@yahoo.com, ytuncay@gazi.edu.tr

**Özet:** Elektronik imzalama uygulamaları için birçok algoritma tanımlanmıştır. Ülkemizde yasallaşma süreci yeni tamamlanan elektronik imza sürecinin yazılım boyutunun uygulamaya dayalı gösterimi araştırma kapsamı dâhilindedir. Bu makalede, çoklu algoritma desteği ile e-imzanın örnek senaryolar üzerinde gösterimi incelenmektedir. Ayrıca, bu alanda açık kaynak kodlu bir yazılım zemininin hazırlanması amaç edinilmektedir.

**Anahtar Kelimeler:** Elektronik imzala, RSA imzalama protokolü, El-Gamal, DSA, E-Sign algoritması

## E-Signing Application Based on Multiple Algorithm Support (E-Signat)

**Abstract:** Lots of algorithms are introduced for e-signing applications. Nowadays, electronic signing software is completed to become law process in our country and in the content research the application permanent of electronic signing software. In this paper, electronic signing has been examined to demonstration on the scenario examples with support of multi-algorithm. Moreover, in this field, preparation of the base software with open source code has been purposed.

**Keywords:** Electronic signing, RSA signing protocol, El-Gamal, DSA, E-Sign algorithm.

### 1. Giriş

Elektronik imza sürecini gerçekleştirmek için tanımlanmış birçok algoritma ve protokol vardır. Her ne kadar RSA[1] algoritması sunduğu avantajları nedeniyle e-imza uygulamalarında en çok kullanılan standart haline gelse de, değişik ihtiyaçlara binaen geliştirilmiş farklı algoritmalar da tercih edilebilmektedir. Farklı gereksinimlerin söz konusu olması halinde ortada kullanılacak ortak bir çözüm yapısı bulunmamakta, her ihtiyaç için özel çözüm üretilmesi yoluna gidilmekte veya standart çözümler kullanılmaktadır.

Yapılan araştırmalarda sertifikasyon standartları doğrultusunda belirlenen standart algoritmaların oldukça yoğun olarak kullanıldığı tespit edilmiştir. Farklı algoritmaların kullanılmamasındaki en büyük etkenin de standartlara[3, 7] uydurulma sorunu olduğu görülmüştür.

Çalışma sonuçları neticesinde belirlenen E-Sign, El-Gamal[4-6], DSA[8], RSA algoritmalarının belirlenen altyapı çerçevesinde sentezlenerek ortak bir yazılımsal alt yapıda bulundurulması hedeflenmiştir. Bu makalede ilgili hedef doğrultusunda, araştırma sonuçlarını içeren ve ihtiyaç sahiplerinin kullanabileceği açık kaynak kodlu bir yazılım zemininin hazırlanması amaç edinilmiştir.

### 2. E-İmza Algoritmaları

Geliştirilen e-imza uygulaması RSA algoritması başta olmak üzere EL-Gamal, DSA, E-Sign olarak belirlenmiştir.

#### 2.1. RSA algoritması

RSA şifreleme sisteminin en büyük özelliklerinden birisi olan özel anahtarın, genel anahtarı oluşturan parçalardan üretilmesinin mümkün olmamasıdır. Bu nedenle RSA geliştirme projesinde öncelikli olarak ele alınmıştır.

### **RSA Anahtar Üretim Süreci:**

İmzalamada kullanılacak anahtarları üretmek için aşağıdaki işlem basamakları kullanılır [2, 9].

1- İki adet birbirinden farklı, aynı büyüklükte, tesadüfi olarak belirlenmiş asal sayılar seçilir ve bunların adı p ve q olarak belirlenir.

2-  $n = p * q$  'dan n sayısı ve  $\Phi = (p-1) * (q-1)$  'i bulunur.

3- Bir rasgele tamsayı üretilir ve adı da "e" koyulur, bu "e" sayısı  $1 < e < \Phi$  şartını ve  $\text{obeb}(e, \Phi) = 1$  şartını sağlamalıdır.

4-  $e * d \equiv 1 \pmod{\Phi}$  ve  $1 < d < \Phi$  şartlarını sağlayan bir "d" sayısı oluşturulur.

5- A'nın genel anahtarı (n,e); özel anahtarı ise "d" dir.

Görüldüğü gibi p,q,d,e sayılarının sadece içinde olabilecekleri bir aralık önceden bilinebilir. Bu dört sayının ne olacağı ise yazılım tarafından anahtar üretimi sırasında rasgele seçilir.

### **RSA İmzalama Süreci:**

İmzalama sürecini gerçekleştirmek için aşağıdaki işlem basamakları kullanılır[2].

1-  $m = R(m)$  hesaplanır, burada aralık değeri  $[0, n - 1]$  olur.

2-  $s = m^d \pmod{n}$  formülünden s hesaplanır.

3- Mesaj için imza değeri s'dir.

### **RSA Doğrulama Süreci:**

Doğrulama sürecini gerçekleştirmek için aşağıdaki işlem basamakları kullanılır[2].

1- Genel anahtar olan (n, e) değerleri elde edilir.

2-  $m = s^e \pmod{n}$  hesaplanır.

3- m 'in  $M_R$  elemanı olduğunu doğrulanır, değilse reddedilir.

4- m(message) =  $R^{-1}(m)$  elde edilir.

## **2.2. DSA, El-Gamal, E-Sign algoritmaları**

Geliştirilen alt yapıda seçilen şifreleme algoritmalarıdır. RSA algoritmasının anlatımında değinilen anahtar üretimi, imzalama ve doğrulama süreçleri için tanımlanan alt yapı diğer algoritmalarda da kullanılarak standardizasyon çalışmaları gerçekleştirilmiştir.

## **3. Uygulamanın Gerçekleştirilmesi**

### **3.1. Anahtar Güvenliği:**

Geliştirilen uygulamada anahtarların güvenliğinin korunması oldukça önem arz etmektedir.

Yapılan araştırmalar sonucu en uygun çözümün şifreleme süreçleri için özel olarak geliştirilen akıllı kartların kullanımı olduğu tespit edilmiştir. Uygulamanın gerçekleştirilmesinde eToken cihazı kullanılmış ve ek olarak .NET ortamında yazılım geliştiricilerin faydalanması için eToken erişim ara yüzü (ing. API) geliştirilmiştir.

### **3.2. Genel İşleyiş Süreci:**

Araştırma kapsamında tanımlanan sürecin günlük yaşantıda sıkça kullanılan e-posta uygulaması üzerinde gösterimi için genel bir yapı tanımlanmıştır. Bu yapıda süreç; anahtar üretimi, mesaj gönderimi ve mesajın doğrulanması olarak üçe ayrılmıştır.

#### **Anahtar Üretimi:**

Seçilen algoritmaya uygun anahtar çiftinin üretilmesi aşamasıdır. (Şekil 1) Üretim sürecinde girdi parametreleri; anahtar sahibinin isim ve e-posta bilgisi, kullanılacak algoritmanın adı, üretilen anahtar uzunluğu ve eToken sürücüsüne erişim için kullanılacak parola bilgisidir. Süreç başlangıcında kullanılacak eToken kartının biçimlendirilmiş olması beklenmektedir.

#### **Mesajın Gönderimi:**

Kullanıcı tarafından iletmek istenilen mesajın ve mesaj ekinin alıcıya gönderilmesi aşamasıdır. (Şekil 2) Mesaj gönderiminde uygulamanın bulunduğu ortamın port, firewall vb. ki-

sıtlamalarından etkilenmemesi için güvenilen otoritede barındırılan bir remoting uygulaması geliştirilmiştir. Bu sayede uygulamanın otorite ile port 80 üzerinden semantik veri modeline uygun şekillendirilmiş xml yapısıyla haberleşmesi sağlanmıştır.



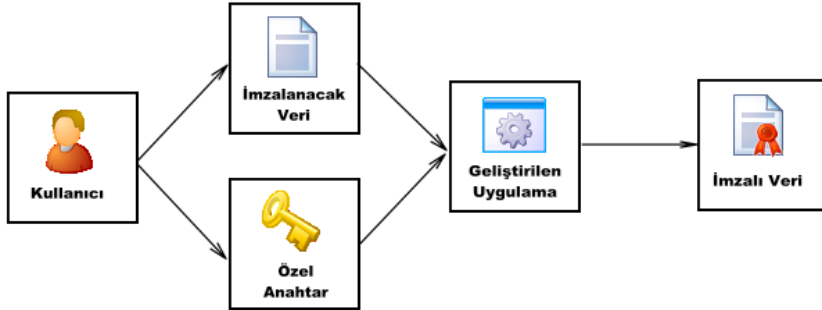
Şekil 1. Anahtar Üretim Ekranı

Mesaj gönderim sürecinde girdi parametreleri alıcının mail adresi, mesajın başlığı ve içeriği, eToken kartının bulunduğu sürücü adı ve şifresidir.

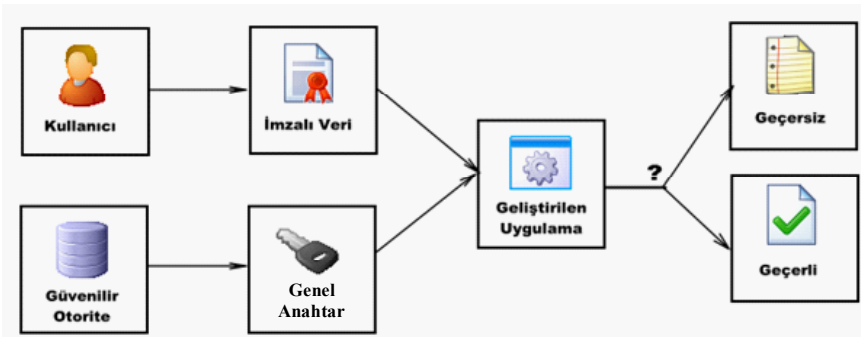
E-posta gönderim uygulaması girdi parametrelerini gönderen kişinin özel anahtarı ile imzalayarak serileştirerek imzalı veri katarını oluşturur. Bu katarı simetrik şifrelemeden geçirerek otorite adresine yönlendirilmiş bağlantı bilgisine dönüştürür. Dönüşen bu bağlantı bilgisini e-posta mesajının içerisine yerleştirerek alıcıya gönderir.

### Mesajın Doğrulaması:

Alıcının aldığı e-posta mesajını doğrulaması aşamasıdır. (Şekil 3) Alıcı imzalı e-postayı aldığı anda mesaj içerisindeki otorite bağlantısına tıklayarak güvenilen otorite sunucusuna bağlanır otorite kendisine gelen parametreyi işleyerek alıcıya ilgili dönütü verir.



Şekil 2. İmzalama Süreci



Şekil 3. Doğrulama Süreci

## Yazılım Mimarisi:

Yazılımın geliştirilmesinde çok katmanlı mimari esas alınmıştır. (Şekil 4) Geliştirilen uygulamada toplam dört temel on yedi yan katmandan mevcuttur.



Şekil 4. Yazılım Mimarisi

## 4. Sonuç

Ülkemizde henüz yeni yaygınlaşma sürecine başlayan elektronik imza uygulamaları hakkında teorik bilginin kolayca bulunabilmesine rağmen pratiğe dönük açık kaynak kodlu uygulamaları elde etmek oldukça güçtür. Bu bağlamda gerçekleştirilen çalışma başlangıcında çoklu algoritma desteği için ilk aşama olarak belirlenen E-Sign, El-Gamal, DSA, RSA algoritmaları sentezlenmiş ve ortak bir altyapı çerçevesinde toplanmıştır. E-İmza alanında araştırma yapmak isteyenlerin birinci elden edinebilecekleri bir kaynak ortaya çıkartılmıştır. E-Signat uygulaması bunun yanında küçük ve orta ölçekli işletmelerin iç uygulamalarında kullanabilecekleri bir çalışma niteliğinde olduğu saptanmıştır.

## 5. Kaynaklar

[1] Rivest, R. L., Shamir, A., ve Adleman, L. A method for obtaining digital signatures and public key cryptosystems. Commun. ACM 21 294-299, 1978

[2] A.J. Menez, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Inc., USA, Ch. 11, 1997

[3] The digital signature standart, Comm. ACM, 35(7), 36-40, 1992

[4] T. ElGamal, Cryptography and logarithms over finite fields, PhD thesis, Stanford University, 1984.

[5] \_\_\_\_, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, 31 (1985), 469–472.

[6] \_\_\_\_, “A public key cryptosystem and a signature scheme based on discrete logarithms”, Advances in Cryptology–Proceedings of CRYPTO 84 (LNCS 196), 10–18, 1985.

[7] FIPS 186, “Digital signature standard”, Federal Information Processing Standards Publication 186, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994

[8] K. Nyberg, R. Rueppel, “A new signature scheme based on the DSA giving message recovery”, 1st ACMConference on Computer and Communications Security, 58–61, ACM Press, 1993.

[9] Gilboa, N.: Two Party RSA Key Generation. Proc. of Crypto’99, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, 116–129, 1999.