http://dergipark.gov.tr/tujes
Trakya University Journal of Engineering Sciences, 18(1): 43-52, 2017
ISSN 2147–0308
Araştırma Makalesi / Research Article

MOBİL CİHAZLARDA RSA ALGORİTMASININ PERFORMANS OPTİMİZASYONU

Tarık YERLİKAYA[1*], Hakan GENÇOĞLU[2]

[1] *Trakya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Edirne*

[2] *İstanbul Aydın Üniversitesi Fen Edebiyat Fakültesi Matematik-Bilgisayar Bölümü, İstanbul*

Özet: Asimetrik şifreleme algoritmaları, simetrik algoritmalara göre çok yavaş çalışırlar. Fakat anahtar dağıtım problemleri yoktur. Optimize edilmedikleri sürece fazla sayıda işlem yaparak şifreleme işlemini gerçekleştirirler. Ayrıca, örneğin RSA algoritması, tatmin edici bir güvenlik için, çok büyük asal sayılar kullanarak şifreleme yapar. Bu durum da donanım için ekstra işlem yükü demektir. Genel kullanımda asimetrik algoritmalar küçük veri paketlerini şifrelemek için kullanılır. Bu çalışmamızda mobil cihazlar üzerinde RSA algoritmasının çalışması test edilmiştir. Mobil cihazların kısıtları göz önüne alınarak RSA algoritması optimize edilmiş ve algoritmanın hızlı çalışması sağlanmıştır.

Anahtar Kelimeler: Mobil iletişim; RSA; Mobil RSA; Mobil Veri Güvenliği; Biginteger

OPTIMIZATION OF PERFORMANCE OF THE RSA ALGORITHM ON MOBİLE DEVICES

Abstract: Asymmetric encryption algorithms encrypt data very slowly according to symmetric encryption algorithms but there is no key distrubiton problem in asymmetric algorithms. Asymmetric algorithms make a lot of process to encrypt the data unless they are optimized. Besides RSA algorithm uses very big prime numbers for satisfactory security and it means extra process for the hardware. Asymmetric algorithms is usullay used for encryption of very small data packets. In this study we tested the performance of RSA algorithm on mobile devices. Considering boundaries of mobil devices we optimised the RSA algorithm, and fast operation of the algorithm was provided.

Keywords: Mobile communications; RSA; Mobile RSA; Mobile Data Security; Biginteger

[*]Corresponding Author: Tarık YERLİKAYA
e-mail: tarikyer@trakya.edu.tr

GİRİŞ

Teknolojinin gelişmesi, internetin yaygınlaşması, mobil cihazlar ve bilgisayarların hayatımızda vazgeçilemez bir hal almaya başlamasıyla veri güvenliği önem kazanmıştır. Veri güvenliği artık iletişim güvenliği olarak anılmaya başlanmıştır. Bulut teknolojisi platform bağımsız ve kesintisiz iletişimin gerçekleşmesini sağlar. Bununla birlikte verilerimizin herkese açık güvensiz internet ortamında bulunması gerekmektedir. Özel bilgilerimizin herkese açık ortamlarda korunabilmesi için şifreleme tekniklerinden yararlanılabilir. Veriler internet ortamına girmeden önce mobil cihazlar veya bilgisayarlarda şifrelenerek internete verilirse, verileri ele geçirmeye çalışan üçüncü şahıslar için anlamsız veri katarları haline gelecektir.

Kısıtlı kaynakları dolayısıyla mobil cihazlarda şifreleme algoritmalarının doğru bir şekilde kullanılması gerekir. Matematik tabanlı asimetrik şifreleme algoritmalarının içerdiği çözülmesi zor problemler çok fazla işlem ve kaynak gerektirmektedir. Bu algoritmaların mobil cihazlarda kullanılabilmesi için optimize edilmeleri gerekir. Bu çalışmamızda RSA algoritmasının içerdiği üs alma işlemi optimize edilerek mobil cihazlardaki performansı ölçülmüştür. Üs alma işlemi için İkili üs alma (Binary Method), çok büyük asal sayıların depolanabilmesi ve işlemleri için BigInteger sınıfı kullanılmış Windows platformunu içeren masaüstü pcler ile Android platformunu içeren tabletlere yönelik yazılan uygulamalar ile performans testi gerçekleştirilmiştir.

ÖNCEKİ ÇALIŞMALAR

Mobil cihazlarda şifreleme ve veri iletimi üzerine farklı çalışmalar yapılmıştır. “Bluetooth üzerinden güvenli veri iletimi” isimli çalışmada küçük boyutlu verilerin akıllı olmayan mobil cihazlarda bluetooth üzerinden gönderilmesi simetrik algoritma kullanılarak yapılmıştır. [1] Başka bir çalışmada simetrik yapıda bir algoritma düşünülerek bu algoritma için akıllı telefonlara yönelik uygulama yazılmış ve analizi yapılmıştır.[2]

Çalışmamız gücü kanıtlanmış olan yüksek işlem gücü gerektiren RSA algoritmasının optimizasyon sonrasındaki performansını ölçecektir.

ŞİFRELEME ALGORİTMALARI

Simetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmaları şifreleme ve şifre çözme işlemleri için aynı anahtarı kullanır. Dolayısıyla anahtar gizli olmak zorundadır. Algoritma ne kadar güçlü olursa olsun anahtar bilindiği takdirde şifreli metin hemen çözülebilir. İki tür simetrik algoritmadan söz edilebilir.

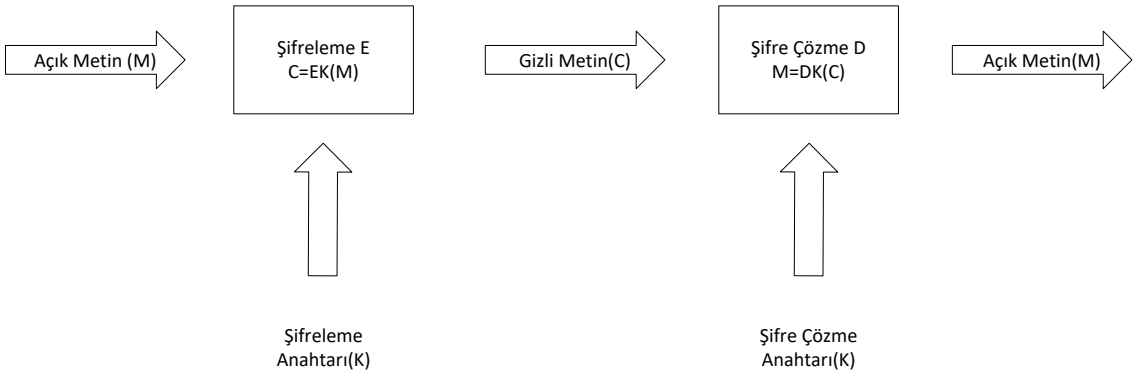
- Blok şifreleme algoritmaları
- Akış şifreleme algoritmaları

Blok şifreler yaygın olarak Feistel Ağı veya Substitution-Permutation Ağı nı kullanır. Feistel mimarisine örnek olarak DES verilebilir, AES-Rijndael algoritması da Sustituon-Permutation Ağı nı kullanır.

Hangi ağı kullanırlarsa kullansınlar simetrik algoritmalar asimetrik algoritmalara göre çok hızlıdırlar. Basit yer değiştirme, xor lama, öteleme gibi işlemlerden oluşurlar.

Akış şifreler ise veri uzunluğuna eşit veya büyük, periyodik olmayan (Bir bölümü kullanılarak anahtarın tahmin edilemeyeceği), rastgele üretilmiş tek kullanımlık anahtarlar ile verinin işleme sokulması ile şifrelenirler. Buradaki en büyük problem tek kullanımlık tamamen rastgele üretilmiş tek kullanımlık anahtarı üretmektir.

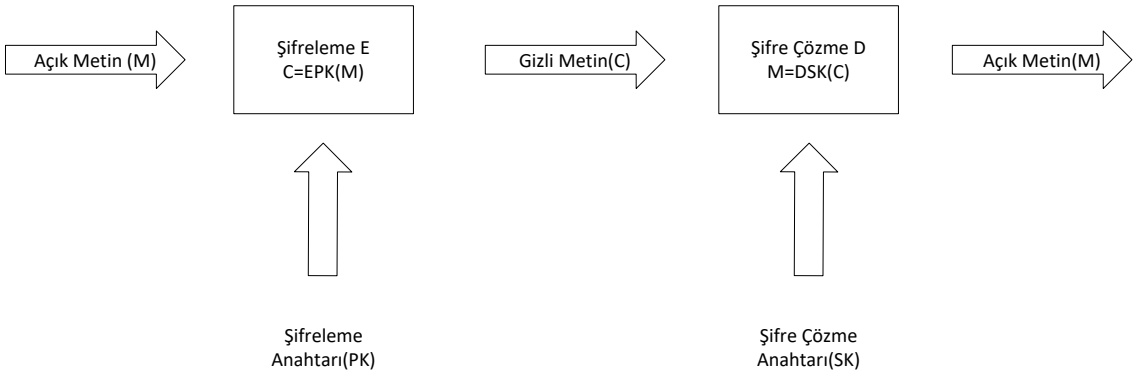
Bu performanslarına rağmen şifreleme ve şifre çözme işlemleri için tek anahtar kullanmaları bu anahtarı alıcıya iletme işleminde büyük problem doğurur. Simetrik algoritmaların bu problemi farklı yöntemlerle çözülmeye çalışılmıştır.



Şekil 1. Simetrik Algoritma Şifreleme – Şifre Çözme Süreci

Asimetrik Şifreleme Algoritmaları

Asimetrik şifreleme algoritmalarında şifreleme için ayrı, şifre çözme için ayrı anahtar kullanılır. Şifreleme için kullanılan anahtar açıktır ve herkes tarafından bilinebilir. Şifre çözmek için kullanılan anahtar sadece şifreli metni çözecek olan alıcı tarafından bilinmelidir ve gizlidir. Bu iki anahtar matematiksel olarak birbirine bağlı olmakla birlikte açık anahtar kullanılarak gizli anahtarı elde etmek imkânsızdır.



Şekil 2. Asimetrik Algoritma Şifreleme – Şifre Çözme Süreci

Rivest-Shamir-Adleman (RSA) Algoritması

1977 yılında duyurulan algoritma ayrık logaritma problemine dayanır. Çok büyük sayıları oluşturma ve işleme zorluğu üzerine kuruludur. Anahtar oluşturma işlemi asal sayılar ile yapılır.

- P ve Q gibi çok büyük iki asal sayı seçilir.
- Bu iki asal sayının ve bir eksiklerinin çarpımı $N = P.Q$, $\varphi(N)=(P-1)(Q-1)$ hesaplanır.
- 1'den büyük $\varphi(N)$'den küçük $\varphi(N)$ ile aralarında asal bir E tamsayısı seçilir.
- Seçilen E tamsayısının mod$\varphi(N)$'de tersi alınır sonuç D gibi bir tamsayıdır.
- E ve N tamsayıları genel anahtarı, D ve N tamsayıları ise özel anahtarı oluşturur. [3]

RSA Sisteminin Güvenliği

RSA algoritmasına yapılabilecek saldırılardan birisi P ve Q sayılarını hesaplamaya çalışmaktır. P ve Q sayılarının tespit edilebilmesi durumunda $\varphi(N)$ sayısı ve dolayısıyla D gizli anahtarı hesaplanabilir. Bu durumda şifrelenmiş metinler kolaylıkla çözülebileceği gibi imzalar da taklit edilebilir. N= PQ olduğundan çarpanlara ayırma yapılarak P ve Q sayıları tahmin edilmeye çalışılır. Büyük sayıların çarpanlara kolayca ayrılabilmesiyle ilgili henüz kesin ve hızlı bir yöntem bulunmadığından P ve Q sayıları çok büyük seçilirse bu işlem imkansızlaşır.[4]

RSA algoritmasında şifreleme işlemi şifrelenmesi düşünülen karakterin ASCII karakter tablosundaki

karşılıklarının belirlenen E anahtarı üssünün N sayısına göre mod alma işlemi ile gerçekleştirilir. İşte bu noktada iki adet problem ortaya çıkmaktadır.

- E ve D anahtarlarının büyüklüğü nedeniyle üs alma işleminin uzun sürmesi
- Seçimler ve işlemler sonucunda ortaya çıkan çok büyük sayıların depolanması

Bu çalışmamızda bu iki probleme yönelik masaüstü pcler için Windows platformu mobil cihazlar için android platformu kullanılarak uygulama geliştirilmiş ve performans tespiti yapılmıştır.

MODÜLER ÜS ALMA

Bilgi teknolojilerinde klasik üs alma işlemleri kullanılacak olursa üs adet çarpma işlemi yapılması gerekir. Örneğin RSA algoritmasında şifrelemek için kullanılacak olan E=230910291 anahtarı 230910291 defa çarpma işlemi yapılacağı anlamına gelir. Bu işlemin şifrelenmesi istenen metnin her bir karakteri için yapıldığı düşünülecek olursa sonucun ne kadar uzun sürdüğü tahmin edilebilir.

Modüler üs alma algoritmaları kullanıldığında ise bu işlem çok daha kısa sürede sonlandırılır. Bu çalışmamızda ikili üs alma metodu (Binary Method) kullanılmıştır. İkili üs alma metodu üs değerini ikili sistemde değerlendirerek her bit değerine göre işlem yapar. Örneğin 230910291 sayısının bit karşılığı 11011100001101101001010010011 ve üs alma işlemi için 15 defa çarpma işlemi yapılacaktır. [3]

Örneğin 3^15 (mod 10) işlemini ele alalım.

3^15=14348907 ve 14348907 (mod 10) =7 olduğundan 3^15 mod 10 = 7 dir. Klasik üs alma mantığında tabanın 15 defa kendisiyle çarpılmasını gerektiren döngü kurulur.. İkili üs alma metodunda 15 sayısının bit karşılığı olan 1111 daki bit karakterleri adedi kadar yani 4 defa işlem yapılır. Algoritma şöyledir:

1. Sonuç=1
2. Eğer son bit 1 ise sonuç = sonuç*taban mod n
3. Değilse taban = taban*taban mod n
4. Üssü bir bit sağa kaydır
5. 2. Adıma git

Bu işlem sonucunda tabanın 4 adımda 15. Kuvveti alınmış olur.

UYGULAMA

Çalışmamızda RSA algoritmasının Android ve Windows platformlarındaki performansı test edilmiştir. Android platformunun doğal dili olan java kullanılarak uygulama geliştirilmiş ve bu uygulama bir emulator tablet de ve gerçek tablet de çalıştırılmıştır. Windows platformu için c# dilinde uygulama yazılmış ve pc de test edilmiştir. Şifrelenecek veri olarak 65 basamaklı ve 260 basamaklı iki adet sayı seçilmiştir.

Uygulamanın PC versiyonu Windows 10 işletim sistemi, AMD Phenom II işlemci 8 GB ram donanımına sahip masaüstü bilgisayarda, mobil versiyonu Android 4.0.4 işletim sistemi NVIDIA Tegra 1Ghz dual core 1GB Ram Motorola Xoom tablet ve Android 4.1 Jelly Bean işletim sistemi Armeabi – v7a tek Çekirdekli İşlemci 1536GB Ram Emulator tablet ile yapılmıştır.

RSA algoritmasının gücü P ve Q sayılarının büyük seçilmesi işlemine dayanır. Çalışmamızda P ve Q sayıları 210 basamaklı sayılar olarak seçilmiştir. Bu sayılar internet kullanılarak bulunmuştur. [7]

P=449417999055441493994709297093108513015373787049558499205492347871729927573118262811508386655998299074566974373711472560655026288668094291699357843464363003144674940345912431129144354948751003607115263071543163

Q=643808006803554439230129854961492699151386107534013432918073439524138264842370630061369715394739134090922937332590384720397133335969549256322620979036686633213903952966175107096769180017646161851573147596390153 [7]

N=P*Q=2893389061935254999093170856551293001773658486700109961034756033503178071108094410504900624339669067240975947894640920418682347794608464677806000131150809116632025487984149797615133484802447858476276955511439647547230142540914027248277018746492731723054391029052435045639900211954700658629784297642678110394919124636699652538653628662441

530779411610353308007986488516578967427848289713639255595384651640941168379276739393

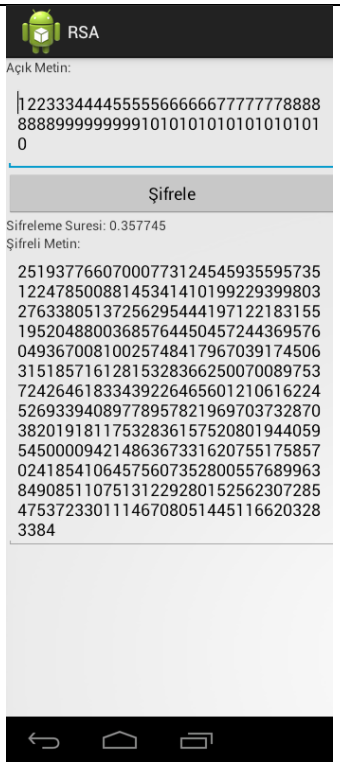
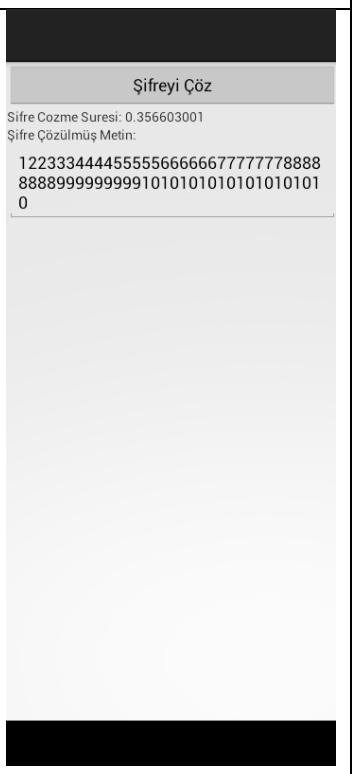
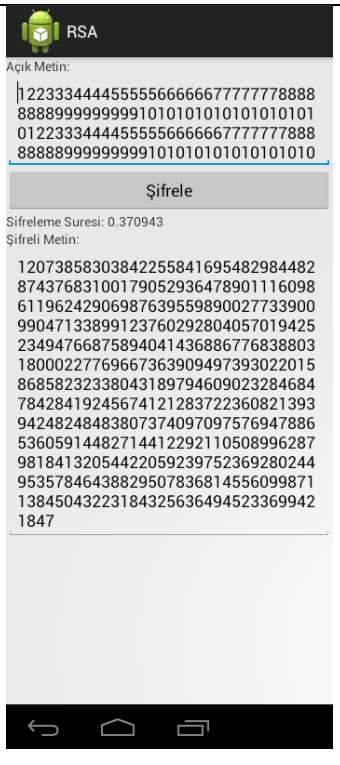
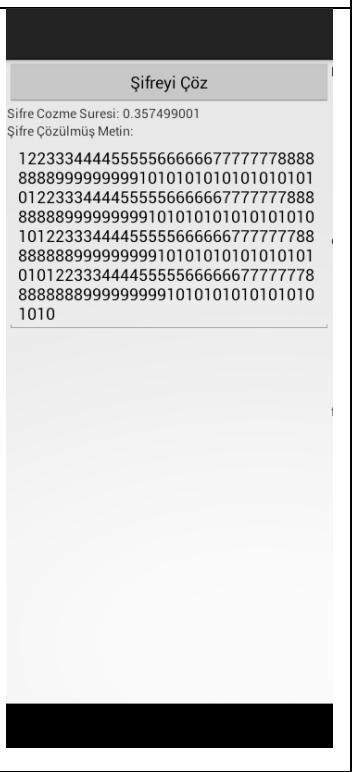
RSA algoritması için E ve D anahtarları Windows ortamında uygulama geliştirilerek üretilmiştir. İki farklı büyüklükte E anahtarı kullanılmıştır. E_1 420 basamaklı E_2 308 basamaklı olarak seçilmiş ve her iki anahtar için herbiri 420 basamaklı iki D sayısı hesaplanmıştır.

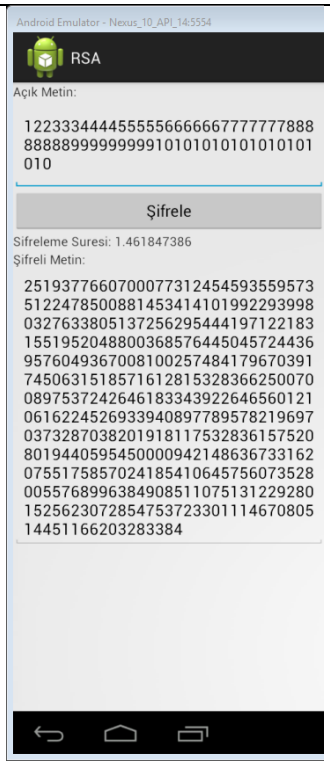
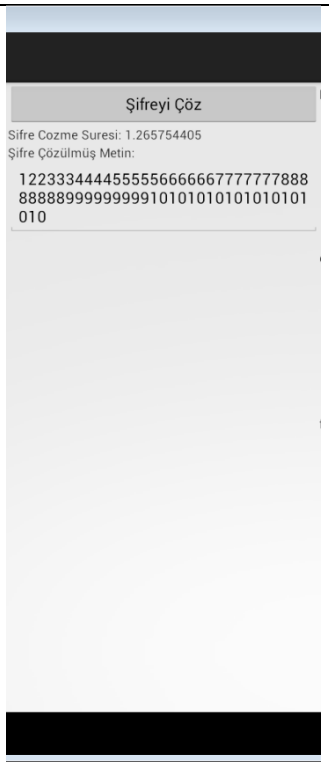
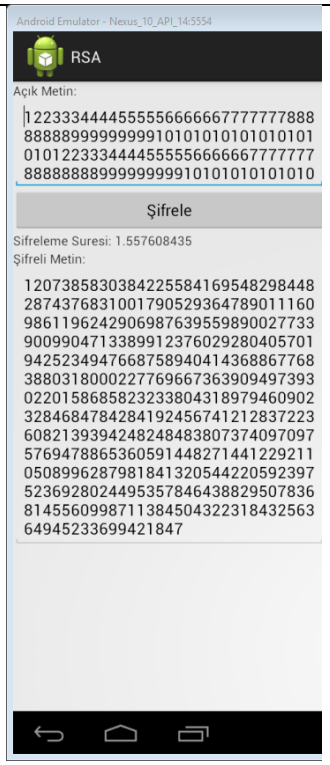
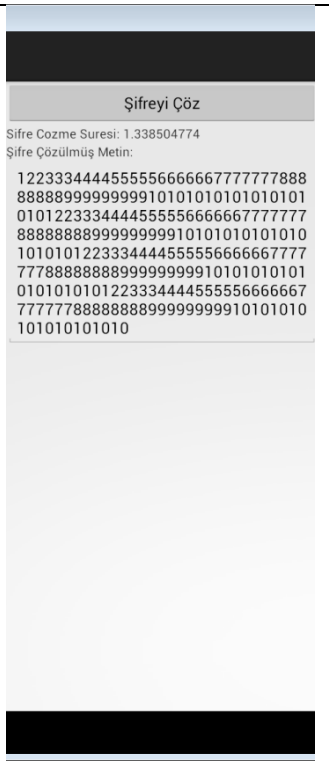
E_1=219100770047019029946923618920704654780753473413515819991871309638745318383257844367901619636661499361367622567440110817709893682197344237065412178825532003639041707530852179255612614538180946935055756586378210777259055692692918735485429836147390774001555113402470837863530244669842520322802040808293515399451846306437540155324649038786768125940836697831889235140434104142880088647385735930208366405965569285937616674153348639896427300375823168517

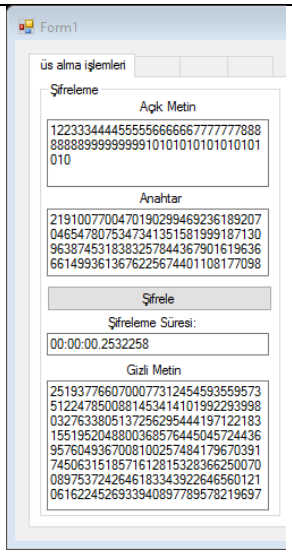
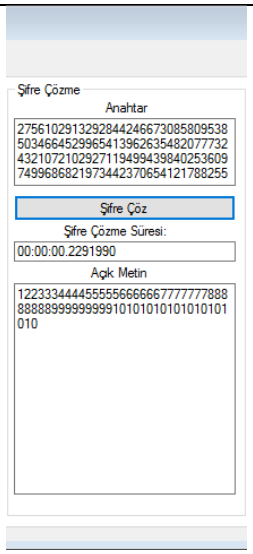
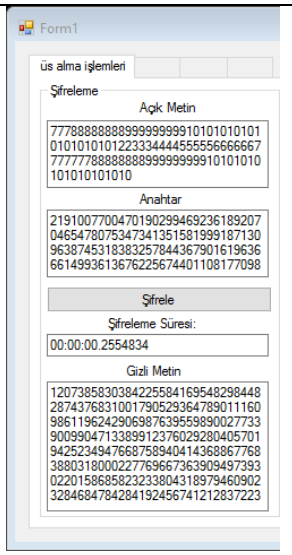
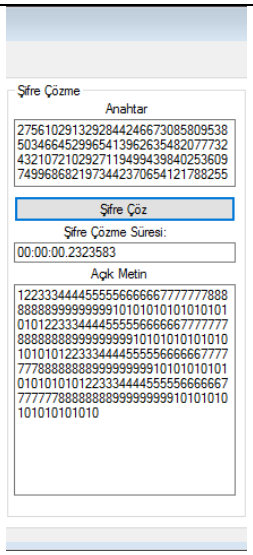
E_2=315708373089097488425021449668536410244502647288180957489215817834474129205029080507103654722281168713194040426421514457423857113014817847656441966584545024462891225816173193483793563318500081792590289497518616067319348379356331850008179259028949751861606

D_1=275610291329284424667308580953850346645299654139626354820777324321072102927119499439840253609749968

D_2=206482482124087453104124624891275575533123318626838700517237795580730787812058749176171053305919677893709948022295929390104800354482269174207994755508533397088920623999900066062268802345827968136886267519465886243413404622461586461980999632804471440168292933215929685080891895514393918731525078914675680151100713910241777283863021899292128291196252206821874403960889807145789250565190299813745951901757616181042652186335

		Şifreleme	Şifre Çözme
Motorola XOOM	65 Basamak - 213 Bit		
	260 Basamak - 861 Bit		

Emülator	65 Basamak - 213 Bit		
	260 Basamak - 861 Bit		

PC	65 Basamak - 213 Bit		
	260 Basamak - 861 Bit		

BIGINTEGER SINIFI

Biginteger sınıfı Microsoft Windows işletim sistemine .NET framework 4 ile dahil olmuş bir sınıftır. Çok büyük sayıları depolamaya ve onlarla işlem yapmaya yarar. Üst sınırı yoktur. Sayılar birer nesne gibi kabul edilir. Integer sayılar ile yapılabilen işlemler Biginteger sınıfı içinde özel metotlarla tanımlanmıştır. Ayrıca math sınıfındaki pow alma vb. metotlar da sınıfın içinde mevcuttur. Üst sınırının olmaması RSA algoritması için kullanılabilmesini kolaylaştırır. Ayrıca aynı sınıf java dilinde de mevcuttur. [5,6]

SONUÇLAR

	Platform	Açık Metin Uzunluğu	Şifreleme Süresi (sn)	Şifre Çözme Süresi (sn)
E1 =420 BASAMAK - 1395 BİT	Masa Üstü PC	260-BASAMAK 861-BİT	0,30	0,27
		65-BASAMAK 213-BİT	0,25	0,26
E2 =308 BASAMAK - 1023 BİT		260-BASAMAK 861-BİT	0,21	0,28
		65-BASAMAK 213-BİT	0,19	0,28
E1 =420 BASAMAK - 1395 BİT	Motorola Xoom	260-BASAMAK 861-BİT	0,37	0,35
		65-BASAMAK 213-BİT	0,35	0,35
E2 =308 BASAMAK - 1023 BİT		260-BASAMAK 861-BİT	0,28	0,35
		65-BASAMAK 213-BİT	0,21	0,35
E1 =420 BASAMAK - 1395 BİT	Emulator	260-BASAMAK 861-BİT	1,55	1,33
		65-BASAMAK 213-BİT	1,46	1,26
E2 =308 BASAMAK - 1023 BİT		260-BASAMAK 861-BİT	1,14	1,39
		65-BASAMAK 213-BİT	1,01	1,37

Elde edilen sonuçlara göre RSA algoritması mobil cihazlarda kabul edilebilir sürelerde şifreleme işlemlerini gerçekleştirebilmektedir. Günümüzde mobil cihazların donanımları kişisel bilgisayarların seviyesine geldiği düşünülecek olursa RSA algoritmasının sağladığı güvenlik uygulamalarda tercih edilebilir.

KAYNAKLAR

1. ÖZÇELİK M. A., KARABULUT M., SUBAŞI A., Bluetooth üzerinden güvenli veri iletimi. 2 ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, 29 Kasım - 01 Aralık 2012
2. ÇAKMAK A. ADALI E. , Mesajların Şifrelenmesinde Yeni Bir Yöntem ve Android Uygulaması. TÜRKİYE BİLİŞİM VAKFI BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ DERGİSİ ISSN 1305-899,1 Yıl 2013 Sayı 7
3. YERLİKAYA T., GENÇOĞLU H., EMİR M. K., ÇANKAYA M., BULUŞ E. RSA Şifreleme Algoritması Ve Aritmetik Modül Uygulaması. İstanbul Aydın Üniversitesi Dergisi Yıl 3 Sayı 9, Sayfa (95 - 104), 2011
4. UÇAN O. N., YERLİKAYA T., GENÇOĞLU H., GÜVENLİ HABERLEŞME TEKNİKLERİ. İstanbul Aydın Üniversitesi Dergisi Yıl 3 Sayı 12, Sayfa (69 - 82), 2011.
5. MSDN BigInteger Structure https://msdn.microsoft.com/tr-tr/library/system.numerics.biginteger(v=vs.110).aspx

6. Java™ Platform, Standard Edition 7, Class BigInteger
https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html

7. University of Tennessee at Martin Primes Page
https://primes.utm.edu/lists/small/small3.html