DÜMF Mühendislik Dergisi

web: http://dergipark.gov.tr/dumf

Araştırma Makalesi / Research Article

RSA algoritmasının şifreleme hızını arttıran algoritmalar ve performansları

Tarık YERLİKAYA*

Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Trakya Üniveristesi,22000 Edirne
tarikyer@trakya.edu.tr, ORCID: 0000-0002-9888-0151, Tel: 0 (284) 226 12 17 (2215)

Canan ASLANYÜREK
Pınarhisar Meslek Yüksekokulu, Kırklareli Üniversitesi,39000 Kırklareli
c.aslanyurek@klu.edu.tr, ORCID: 0000-0001-7513-9953, Tel: 0 288 615 33 03(5825)

Geliş: 02.05.2019, Revizyon:18.07.2019, Kabul Tarihi: 08.08.2019

Öz

Gelişen teknoloji sayesinde ağda dolaşan veri miktarı her geçen gün biraz daha artmakta ve bu artış da verilerin korunması problemini beraberinde getirmektedir. İletişimde verilerin güvenliğinin sağlanması, doğru adrese iletilip iletilmediğinin anlaşılması için şifreleme ve dijital imza denilen çeşitli yöntemler kullanılmaktadır.
Bu çalışmada, simetrik ve asimetrik şifreleme olarak ikiye ayrılan bu şifreleme yöntemlerinden asimetrik şifreleme yöntemi ele alınmıştır. Asimetrik şifreleme algoritmaları arasında en yaygın olarak kullanılan RSA(Rivest, Shamir, Adelman) algoritması incelenmiştir. RSA algoritması kullanılarak veriler şifrelenirken birtakım problemler ortaya çıkmaktadır. Bu problemlerden en önemlisi hız problemidir. Çünkü RSA algoritmasında şifreleme ve deşifreleme (şifre çözme) işlemleri için kullanılan matematiksel işlemlerin yoğun olması ve güvenliği arttırmak adına çok büyük asal sayılar kullanılması sebebiyle verilerin şifrelenmesi ve şifrelerin çözülmesi daha yavaş gerçekleşmektedir. Bundan dolayı RSA algoritmasının şifreleme ve şifre çözme hızını arttırmak için çeşitli algoritmalar kullanılmaktadır. Yapılan çalışma ile RSA algoritması kullanılarak yapılan şifreleme işlemlerinde karşılaşılan hız problemi ele alındı. RSA algoritmasının hızını arttıran bazı algoritmalar java programlama dili kullanılarak uygulaması yapıldı. Bu algoritmaların şifreleme işlemi yapılırken çalışma performansları ölçülüp, RSA'nın hızını arttırmadaki etkileri karşılaştırılmıştır. Aynı zamanda RSA şifreleme algoritmasında kullanılan p ve q asal sayılarının boyutları 256 bit, 512 bit, 1024 bit ve 2048 bit şeklinde farklı uzunluklarda alınarak şifreleme hızına olan etkileri irdelenmiştir.

Anahtar Kelimeler *: Asimetrik Şifreleme, Rsa, Rsa Algoritmasında Hız,ECC, Simetrik Şifreleme*

* Yazışmaların yapılacağı yazar

DOI: 10.24012/dumf.559789

Giriş

Veri gizliliğinin insanlık için çok önemli olduğu geçmişten beri bilinmektedir. Bu yüzden de verilerin gizli kalması için geliştirilen birçok yöntem bulunmaktadır. Eskiden metindeki karakterlerin yerlerinin değiştirilmesi ve metnin şeklinin bozulması olarak gerçekleştirilen bu yöntemler, gelişen teknolojiyle daha fazla kullanıcının yararlanacağı bir şekilde bilgisayarlarla yapılan yöntemlere dönüşmüştür.

Geçmişte daha çok diplomasi ve askeri alanlarda çok önemsenen bilgi gizliliği, hayatın vazgeçilmez bir parçası haline gelen elektronik ticaretin kullanılmasıyla birlikte interaktif bankacılık işlemlerinin de çok yaygınlaşması ile insan hayatında çok büyük önem kazanmıştır.
Kriptoloji, Yunanca krypto's (saklı) ve logo's (kelime) sözcüklerinin birleşmesinden oluşmuş ve iletişimde gizlilik, veri bütünlüğü sağlayan şifreleme bilimi olarak değerlendirilmektedir (Rivest vd., 1978). Kriptoloji, kriptografi (şifreleme) ve kriptanaliz (şifre çözme) olmak üzere ikiye ayrılır (Yerlikaya, vd., 2007). Kriptografi, haberleşmenin güvenliğini kötü niyetli taraflara karşı korumak için yapılan çalışmalar bütünüdür (Yerlikaya, vd., 2013). Kriptanaliz var olan kriptosistemlerin analizini içeren metotlar bütünüdür. Karşı tarafa iletilmek istenen bilgi, açık mesaj (plain text) ve bu bilginin şifreli haline ise şifreli mesaj (cipher text-cryptography) olarak tanımlanır(Stallings, 1995).

Kriptografi, simetrik ve asimetrik kriptografi olarak iki gruba ayrılır. 20. yy içinde yaygın olarak kullanılan simetrik kriptografide, şifreleme ve şifre açma işlemleri için gizli anahtar adı verilen tek anahtar kullanılmaktadır (Nagar, Alshamma, 2012). Bu yöntemi kullananlar anahtar paylaşımını güvenli bir ağda yapmalıdır. Simetrik şifreleme algoritmalarına örnek olarak, Lucifer algoritmasından faydalanarak geliştirilen ve ABD tarafından FIPS olarak kabul edilen DES algoritması ve 2000'lerde geliştirilen Rijndael algoritmasını temel alarak geliştirilen AES algoritması verilebilir (Verma, Garg, 2011). Simetrik şifreleme algoritmaları, şifreleme ve deşifreleme işlemlerini algoritma yapılarından kaynaklı olarak hızlı bir şekilde gerçekleştirebilmektedir. Bununla birlikte tek anahtar ile şifreleme ve deşifreleme işlemi gerçekleştirirken anahtar güvenliğinin de sağlanması gerekir. Ayrıca şifreleme algoritmaları sadece veri güvenliğini sağlamanın yanında başka güvenlik unsurlarını (kimlik doğrulama, bütünlük vs.)da sağlaması gerekir.

Açık anahtarlı kriptosistem olarak adlandırılan asimetrik kriptografide genel ve özel anahtar olarak adlandırılan iki farklı anahtar kullanılır. Asimetrik şifreleme algoritmaları çözülmesi zor matematik problemleri.(Asal çarpanlarına ayırma ve Ayrık Logaritma) üzerine kurulmuştur. Şifreleme işlemi yapılırken genel anahtar, deşifreleme işleminde ise özel anahtar kullanılır. Böylelikle anahtar güvenliği sağlanmış olur. Herkes tarafından bilinen genel anahtar ile şifrelenmiş olan metni açılamaz. Çünkü kullanılan anahtar uzunluğuna bağlı olarak genel anahtardan özel anahtarı elde etmek zordur.

Asimetrik şifreleme algoritmalarından en önemlisi, günümüz teknolojisinde de kullanılan ve 1977 yılında R. Rivest, A. Shamir ve L. Adleman tarafından oluşturulmuş olan RSA (Rivest, Shamir, Adleman) şifreleme algoritmasıdır. RSA algoritması asimetrik şifreleme algoritmalarına uygun biçimde geliştirilmiş ve e-imza ve kimlik doğrulama işlemlerinde kullanılmakla birlikte güvenli anahtar paylaşımı işlemlerinde kullanılmaktadır. (Yerlikaya, vd., 2005).

Bu çalışmada, öncelikle kullanılan temel tanım ve kavramlar anlatıldı. Diğer bölümlerde RSA kriptosistemi incelenerek RSA algoritmasının hızını artırmak için yapılmış olan algoritmalar incelenmiştir. Bu algoritmaların uygulaması java programlama dili kullanılarak yapılmış ve performansları değerlendirilmiştir.

Materyal Ve Metod

Asimetrik şifreleme

Genel anahtar şifreleme sistemi olarak da adlandırılan asimetrik şifreleme, ilk kez Whitfield Kiffie ve Matin Hellman tarafından sunuldu. Bu şifreleme sisteminin temel amacı şifreleme ve şifre çözme işleminin farklı anahtarları kullanmasıdır. Genel anahtar şifreleme, daha önce sözü edilen simetrik şifreleme sisteminde anahtar dağıtım konusundaki eksikliği giderir.

Asimetrik kriptosistemlerde açık (public) ve özel (private) anahtar olarak adlandırılan iki ayrı anahtar kullanılmaktadır. Böylece simetrik kriptosistemlerde yapılmak zorunda olan anahtar takas işlemine gerek duyulmaz (Çenesiz, Soğukpınar, 2000). Açık anahtarlı şifreleme algoritmalarında kullanılan özel anahtar, genel anahtar adı verilen diğer bir anahtar kullanılarak belirli matematiksel işlemler sonucunda elde edilir. Genel anahtardan özel anahtarı elde etmek zor olduğundan genel anahtarı ele geçiren biri mesajı deşifre edemez.

Yaygın bir şekilde kullanılan açık anahtarlı şifreleme algoritmaları;

- 1977, RSA
- 1985, El-Gamal
- 1986, Eliptik Eğri

Asimetrik şifreleme algoritmalarının genel olarak iki kullanım alanı vardır: Şifreleme ve dijital imza.

Eliptik Eğri şifreleme algoritması (EEC)

Eliptik Eğri şifreleme algoritması 1985 yılında RSA şifreleme algoritmasına alternatif olarak sunulan bir algoritmadır (Kurt, 2012). RSA algoritması ile şifreleme yapıldığında veri güvenliğini sağlamak için kullanılan asal sayıların büyüklerinin arttırılması RSA şifreleme algoritmasının şifreleme ve şifre açma hızını oldukça düşürdüğünden IEEE
P1363 standartlarını yerine getiren Eliptik Eğri algoritması sunulmuştur (Gupta vd., 2002). RSA şifreleme algoritması veri güvenliği için kullanılan büyük asal sayıların çarpanlarına ayrılması esasına dayanırken Eliptik Eğri algoritmasında ayrık logaritma problemi(ALP) kullanılmaktadır.

NESSIE (New European Schemes for Signature, Integrity and Encryption) tarafından yayımlanan bir rapora göre Eliptik Eğri şifreleme algoritmasında daha küçük uzunlukta sayılar kullanılarak yapılan şifreleme sisteminin RSA şifreleme algoritmasına eş değer olduğu görülmüştür (NESSIE Consortium, 2003). Aynı zamanda bu raporda Eliptik Şifre algoritmasının RSA kriptosistemine göre daha hızlı şifreleme yaptığı gözlemlenmiştir.

RSA şifreleme algoritması

RSA şifreleme sistemi asimetrik şifreleme algoritmalarının en bilinenidir (Bellare, Rogaway, 1994). RSA, MIT çalışanı olan Ron Rivest, Adi Shamir ve Len Adleman tarafından 1977 yılında bulunmuş ve bu üç kişinin soyadlarının baş harfleri birleştirilerek isimlendirilmiştir (Okumuş, 2012). Martin Gardener, 1977 yılında yazılan bir makalede RSA'nın buluşundan bahsetmiştir. Bu yöntem basit bir matematiksel hesaba dayandırılarak geliştirilmiştir (Yıldırım, 2014). Daha sonra asimetrik kriptosistemlere uygun olacak şekilde geliştirilmiştir. Hem mesaj şifreleme hem de dijital imza işlemlerinde güvenli bir şekilde kullanılır (Koç, 1994). S/MIME, PEM, MOSS ve PGP gibi gizli elektronik haberleşme protokolleri RSA şifreleme algoritmasını kullanır (Yerlikaya, vd., 2005). RSA şifreleme algoritması şifrelemenin yanında imzalama için de kullanılan algoritmadır (Stallings, 2003).

RSA kriptosisteminde kullanılan genel ve özel anahtar mesajı almak isteyen tarafından üretilir. Genel anahtar herkesin kullanımına açıktır. Alıcıya mesaj göndermek isteyen taraf bu genel anahtar ile mesajı şifreler ve karşıya gönderir. Alıcı gelen mesajı daha önceden ürettiği gizli anahtar ile açar ve orijinal mesajı elde eder.

RSA algoritmasının çalışma yapısı

RSA kriptosisteminde şifrelenecek olan mesaj öncelikle [0, N-1] arasındaki pozitif tamsayı

bloklar haline dönüştürülür (Yerlikaya, vd., 2005). Aşağıda RSA algoritmasının çalışma yapısı anlatılmıştır (Jahan, vd., 2015).

Anahtar Oluşturma Algoritması:

- Yaklaşık olarak eşit bit uzunluğuna sahip, birbirinden bağımsız p ve q gibi iki büyük asal sayı seçilir.
- Bu asal sayıların çarpımı N = p.q ve bu sayıların bir eksiklerinin çarpımı Ø(N) = (p-1) . (q-1) değeri hesaplanır.
- 1 < e < Ø(N) ve Ø(N) ile aralarında asal olan bir e sayısı seçilir.
- Genişletilmiş Öklid algoritması ile seçilen bu e sayısının mod Ø(N)'e göre tersi hesaplanır; sonuç d gibi bir tamsayıdır. d ≡ e-1 mod Ø(N)
- Alıcı tarafın genel anahtarı e, N, özel anahtarı ise d, N olur.
-

RSA kriptosistemindeki bu anahtar üretim algoritmasında oluşan N sayısına modülüs, e herkes tarafından bilinen genel anahtar ve d ise özel anahtar olarak adlandırılır. p ve q sayılarının çarpımı sonucu elde edilen N sayısı hem şifreleme hem de şifre açma işlemlerinde kullanılır.

Şifreleme İşlemi:

- Gönderici, veriyi alacak tarafın açık anahtarı olan e, N' yi bulur.
- Şifrelenecek olan metin, M ∈ [N – 1] olmak şartıyla bir tamsayıya dönüştürülür.
- Şifrelenecek metnin e'ninci kuvveti alınıp ve mod N'ye göre karşılığı hesaplandığında şifreli metin oluşturulmuş olur. C ≡ M^e mod N
- Gönderici şifreli metni (C) alıcıya gönderir.

Şifre Açma İşlemi:

Alıcı taraf açık metni elde edebilmek için M ≡ C^d mod N işlemini gerçekleştirir. Şifre açma işleminin doğrulaması aşağıda gösterilmiştir:

- M^(Ø(N)) ≡ 1 mod N
- e. d ≡ 1 mod Ø(N)
- e. d ≡ k. Ø(N) + 1
- C^d ≡ M^(e.d) ≡
- M^(k.Ø(N)+1) ≡ (M^(Ø(N)))k . M
- ≡ M mod N

RSA şifreleme algoritmasında anahtar oluşturulurken asal sayılar kullanılmasının amacı, asal sayıların çarpanlarına ayrılmasının diğer sayılara göre daha zor olmasıdır. Böylece üretilen anahtarın elde edilmesi daha zor olmaktadır.

Örnek:

Anahtar üretimi:

- İki asal sayı seçilsin p = 13, q = 11
- N = p.q değeri N = 13.11 =143
- (N) = (p-1) . (q-1) = 12.10 =120
- (N) ile aralarında asal ve 1 < e < (N) koşulunu sağlayan bir e sayısı seçelim. 1 < e < 120 , e= 7 olsun.
- 1 < d < (N) ve e.d 1 mod (N) koşulunu sağlayan d = 103"tür.
- Alıcı tarafın genel anahtarı e = 7, N = 143, gizli anahtarı d = 103, N =143 olur.

Şifreleme işlemi:

- Alıcı taraf genel anahtar olan e=7, N=143 sayılarını gönderici tarafa herkes tarafından görülebilecek şekilde gönderir.
- Göndericinin açık metni M=9 olsun.
- Gönderici taraf C ≡ 97 mod 143 koşulunu sağlayan C = 48 sayısını bulur.
- Gönderici taraf C = 48 sayısını herkes tarafından ulaşılabilecek bir kanal ile karşı tarafa gönderir.

Şifre Açma işlemi:

- Alıcı taraf M ≡ 48103(mod 143) koşulunu sağlayan M sayısını bulup açık metni elde eder.

RSA algoritmasında hız

RSA şifreleme algoritmasının güvenilirliğini arttırmak için çok büyük asal sayılar kullanılır. Şifreleme ve deşifreleme işlemi bu çok büyük asal sayılar kullanılarak gerçekleştirildiğinden dolayı sonuca ulaşma süresi artmaktadır. Bu yüzden çok büyük mesajların RSA ile şifrelenmesi çok uygun görülmez (Montgomery, 1985; Scheinder, 1996; Stallings, 1998). RSA

şifre açma süresi büyük şifre çözme üssüne sahip olduğundan şifreleme süresinden daha uzundur

Şifreleme ve şifre açma sürelerini kısaltmak için çeşitli algoritmalar geliştirilmiştir. Çok büyük asal sayılar kullanılarak işlem yapılan RSA için Montgomery tarafından 1985 yılında Montgomery algoritması oluşturulmuştur. Bu algoritmanın hızını düşüren diğer bir sebep kullanılan anahtar boyutlarının büyük olmasıdır. Bunu iyileştirmek için 1990 yılında Wiener tarafından Rebalanced CRT-RSA ve 2003 yılında da Paixo ve Alison tarafından RPrime RSA algoritmaları üretilmiştir. Ayrıca RSA algoritmasının şifre açma hızını yaklaşık 4 kat oranında arttıran CRT-RSA algoritması Quisquater ve Couvreur tarafından 1982 yılında sunulmuştur. Daha sonra Collins vd. 1997 yılında CRT-RSA algoritmasını kullanarak şifre açma kısmında MultiPrime-RSA algoritmasını sunmuşlardır.

RSA şifreleme hızını arttıran algoritmalar

RSA şifreleme algoritmasının hızını arttıran algoritmalardan bazıları;

Montgomery Modüler Çarpım Algoritması

Peter Montgomery tarafından tasarlanan bir algoritmadır. Bu algoritma birtakım donanım elemanları kullanılarak bölme işlemi gerçekleştirmeden modüler kalan değerini daha hızlı bir şekilde hesaplamaktadır (Bayram, Örs, 2010; Koltuksuz, 1998).k sayısı n bitlik bir tamsayı olmak şartıyla; a ve b asal sayılarının N modülüsüne göre çarpımı hesaplanır. R^{-1} değişkeni R'nin mod N'ye göre tersidir.

Girdi: a =$(a_{k-1}\ldots..a_0)_2$, b= $(b_{k-1}\ldots..b_0)_2$,
n_{k-1}= 1 olmak şartıyla; N = $(n_{k-1}\ldots..n_0)_2$, $R=2^k$
Çıktı: MONT(a,b) = a . b . R^{-1} mod N

1. R ← 2^k
2. N ' ← - N-1 mod R
3. P ← a . b
4. U ← (P mod R). N ' mod R
5. C ← (P + U. N) / R
6. if (C > N) then C ← C – N
7. return (C)

Hızlı Mod Alma Algoritması

RSA şifreleme algoritmasında şifreleme işlemi yapılırken M açık metin, e genel anahtar ve N değeri modülüs olmak koşuluyla;
Girdi: M,e,N
Çıktı: M^e mod N
e binary formata dönüştürülür.

1. e_n= $\sum_{i=0}^{n-1} e(0,i)$ 2i,……,
2. K=1, P=M
3. for i=0 to e do
4. P= P^2 mod N
5. if e_i=1 then
6. P= K.P mod N
7. P= P^2 mod N
8. Return P
   End
   End

Binary Modüler Üs Alma Algoritması (Koç, 1994)

RSA şifreleme algoritmasında şifrelenecek metin olan M, genel anahtar e ve modülüs değeri n olmak şartıyla;

Girdi: M,e,n , e=($e_{k-1}e_{k-2}\ldots.e_1e_0$)
Çıktı: C=M^e mod n

1. if e_{k-1}=1 then C=M else C=1
2. for i=k-2 downto 0
   2a. C=C.C (mod n)
   2b. İf e_i=1 then C= C.M (mod n)
3. return C

Araştırma Bulguları

Aşağıda bahsedilen uygulamalar işlemci türü AMD A6, işlemci hızı 1.50GHz, sabit diski 500 GB, Bellek 6 GB, işletim sistemi Windows 10 olan bir bilgisayarda gerçekleştirilmiştir. Yapılan şifreleme uygulamaları, java programlama dili ile standart kriptoloji kütüphaneleri kullanılmadan NetBeans IDE 8.2 kullanılarak yapılmıştır.

RSA algoritmasının 256, 512, 1024, 2048 bit uzunluğunda sayılar kullanılarak özel anahtar ve şifreleme işlemlerinin ortalama süresi saniye cinsinden aşağıdaki tabloda yer almaktadır. Açık anahtar=3649134810816461 olarak alınmıştır. Şifrelenen mesaj=123620 olarak belirlenmiştir.

Tablo 1. Standart rsa uygulaması performansı

p-q Sayılarının uzunluğu(bit)	Açık Anahtar	Özel Anahtar Bulma Süresi(sn)	Şifreleme Süresi(sn)
256	3649134810816461	0,0085	0,0104
512	3649134810816461	0,0094	0,0209
1024	3649134810816461	0,0289	0,0587
2048	3649134810816461	0,0578	0,0685

Hızlı mod alma algoritması kullanılarak yapılan RSA uygulamasının çalışma performansı aşağıda tablo 2'de verilmiştir.
Açık anahtar=3649134810816461 olarak alınmıştır. Şifrelenen mesaj=123620 olarak belirlenmiştir.

Tablo 2. Hızlı mod alma algoritması kullanılarak rsa uygulaması performansı

p-q Sayılarının uzunluğu(bit)	Açık Anahtar	Özel Anahtar Bulma Süresi(sn)	Şifreleme Süresi(sn)
256	3649134810816461	0,0059	0,0042
512	3649134810816461	0,0011	0,0084
1024	3649134810816461	0,0167	0,0201

Binary modüler üs alma algoritması kullanılarak yapılan RSA uygulamasının çalışma performansı aşağıdaki tabloda yer almaktadır.
Açık anahtar=3649134810816461 olarak alınmıştır. Şifrelenen mesaj=123620 olarak belirlenmiştir.

Tablo 3. Binary modüler üs algoritması kullanılan rsa uygulaması

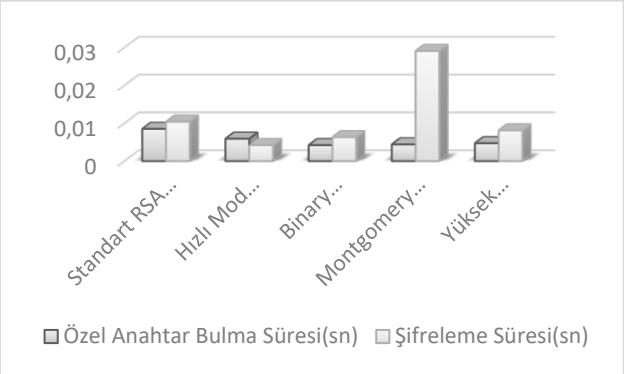
p-q Sayılarının uzunluğu(bit)	Açık Anahtar	Özel anahtar Bulma Süresi(sn)	Şifreleme Süresi(sn)
256	3649134810816461	0,0042	0,0063
512	3649134810816461	0,0175	0,0086
1024	3649134810816461	0,0182	0,0262
2048	3649134810816461	0,0546	0,0326

Montgomery Çarpım Algoritması kullanılarak yapılan RSA uygulamasının çalışma performansı Tablo 4'te yer almaktadır.

Tablo 4. Montgomery çarpım algoritması ile rsa çalışma performansı

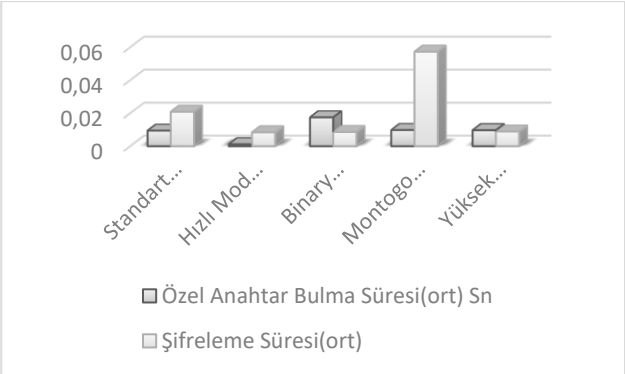
p-q Sayılarının uzunluğu(bit)	Açık Anahtar	Özel Anahtar Bulma Süresi(sn)	Şifreleme Süresi(sn)
256	3649134810816461	0,0044	0,0291
512	3649134810816461	0,0098	0,0573
1024	3649134810816461	0,0131	0,1085
2048	3649134810816461	0,0221	0,2122

Tablo 1, Tablo 2, Tablo 3 ve Tablo 4'teki performanslar göz önüne alınarak aşağıdaki grafikler elde edilmiştir.



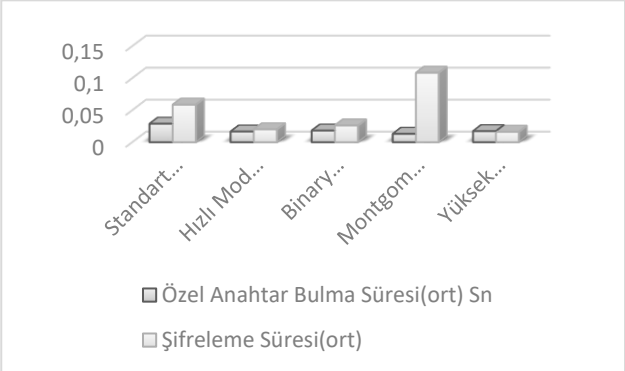
Şekil 1. 256 bit uzunluğunda p ve q – e = 3649134810816461

Şekil 1'deki veriler 256 bit uzunluğunda p, q sayıları ile e= 3649134810816461 açık anahtarı kullanılarak elde edilmiştir. Şekilde görüldüğü gibi özel anahtar Binary Modüler Üs Alma algoritması kullanıldığında en kısa sürede bulunmuştur. Şifreleme hızlarına bakıldığında ortalama olarak 0,004 saniye ile en hızlısı Hızlı Mod Alma algoritması, en yavaşı Montgomery Modüler Çarpım algoritmasıdır. Montgomery Modüler Çarpım algoritması çok işlem içerdiğinden sadece yazılımda iyi performans göstermemektedir. Gerekli donanım desteği yapıldığında başarılı bir performans gösterdiği yapılan araştırmalar neticesinde elde edilmiştir. Montgomery algoritması dışındaki diğer algoritmaların performanslarının standart RSA algoritmasından daha iyi olduğu tespit edilmiştir.



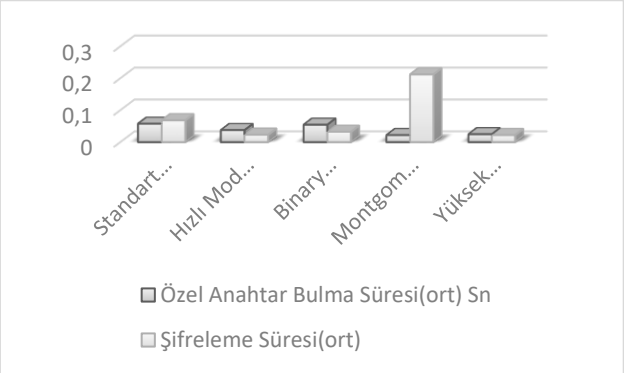
Şekil 1. 512 bit uzunluğunda p ve q – e=3649134810816461

Şekil 2'deki veriler 512 bit uzunluğunda p, q sayıları ile e= 3649134810816461 açık anahtarı kullanılarak elde edilmiştir. Şekil incelendiğinde en iyi performans Hızlı Mod Alma algoritması kullanıldığında elde edilmiştir. Hızlı Mod alma algoritması kullanıldığında şifreleme süresinin ortalama olarak yaklaşık 0,008 saniye olduğu görülmüştür. 256 bitlik p ve q kullanıldığında 0,004 saniye olan şifreleme süresinin 2 kat arttığı gözlemlenmiştir. p ve q sayıların uzunluğunun artması şifreleme ve özel anahtarı bulma süresini de arttırmıştır. 512 bitlik sayılar kullanıldığında Hızlı Mod Alma algoritması kullanıldığında şifreleme süresinin Standart RSA algoritmasına göre yaklaşık olarak 2,4 kat daha hızlı olduğu görülmektedir.



Şekil 2. 1024 bit uzunluğunda p ve q – e=3649134810816461

Şekil 3'deki veriler 1024 bit uzunluğunda p, q sayıları ile e= 3649134810816461 açık anahtarı kullanılarak elde edilmiştir. Şekil 3'e bakıldığı zaman p ve q sayılarının uzunluğunun 1024 bit olması ile şifreleme performansı Hızlı Mod Alma algoritması kullanıldığında en iyidir. Hızlı Mod Alma algoritmasının Şekil 2'deki performansıyla Şekil 3'teki performanslar kıyaslanırsa yaklaşık olarak 2,5 kat oranında bir düşüş gözlenmektedir. Montgomery algoritmasının performansının yine çok düşük olduğu görülmektedir. Hızlı Mod Alma algoritmasından sonra en iyi performans Binary Modüler Üs Alma algoritmasındadır.



Şekil 3. 2048 bit uzunluğunda p ve q – e=3649134810816461

Şekil 4'deki veriler 2048 bit uzunluğunda p, q sayıları ile e= 3649134810816461 açık anahtarı kullanılarak elde edilmiştir. 2048 bit uzunluğundaki p ve q sayıları kullanıldığında şifreleme performansları en iyiden başlayarak sıralanırsa Hızlı Mod Alma algoritması, Binary Modüler Üs Alma Algoritması, Standart RSA ve Montgomery Modüler Çarpım algoritmasıdır. Hızlı Mod Alma algoritması ortalama olarak 0,02 saniyede şifreleme işlemini gerçekleştirmiştir. Standart RSA'dan yaklaşık olarak 3 kat daha hızlıdır. Binary Modüler Üs Alma algoritması ise ortalama 0,03 saniyede şifreleme yapmış ve Standart RSA algoritmasından yaklaşık olarak 2 kat daha hızlı olduğu görülmüştür.

Sonuç ve Öneriler

Bu çalışmada, asimetrik şifreleme algoritmaları arasında en çok kullanılan RSA şifreleme algoritması ve performansı incelenmiştir. Asimetrik şifreleme algoritmalarında şifreleme ve şifre çözme iki anahtar kullanır. Ayrıca şifreleme ve deşifreleme işlemlerinde çok büyük asal sayılar kullanılır. Bu da asimetrik şifreleme

algoritmalarının güvenliğini arttırmaktadır. Fakat, kullanılan bu büyük sayılar işlemlerin daha yavaş gerçekleşmesine sebep olmaktadır.
RSA şifreleme algoritması da hem dijital imza hem de şifrelemede güvenli olduğu sebebiyle çok yaygın kullanılmaktadır. Ancak, kullanılan sayıların büyüklüğü hız problemini ortaya çıkarmaktadır. Bu sebepten dolayı RSA algoritmasında şifreleme ve deşifreleme hızını arttıran algoritmalar geliştirilmiştir. Şifreleme hızını arttıran algoritmaların bazıları incelendi ve java programlama dili kullanılarak uygulamaları yapıldı. Bu algoritmaların RSA kriptosistemin kullanılmasıyla RSA'nın çalışma performansı ölçüldü.

Yapılan performans ölçümleriyle, en hızlı şifrelemenin Hızlı Mod Alma algoritması kullanıldığında gerçekleştiği görülmüştür. Standart RSA algoritmasından ortalama olarak 2,5 katı daha hızlı olduğu tespit edilmiştir. Üstelik kullanılan sayılarının büyüklüğü arttıkça aradaki hız farkının arttığı gözlemlenmiştir. Binary Modüler Üs alma algoritması ise Standart RSA algoritmasına göre ortalama olarak 2 kat daha hızlıdır. Montgomery Modüler Çarpım algoritmasının sadece yazılımda çok başarılı olmadığı görülmüştür. Fakat, gerekli donanım desteği yapıldığında Standart RSA algoritmasına göre yaklaşık olarak 2 kat daha hızlı çalıştığı yapılan araştırmalar neticesinde elde edilmiştir.

Kaynaklar

Aksuoğlu, A. (2010). RSA Algoritmasının İyileştirilmesi için Yeni Bir Yaklaşım. *Yüksek Lisans Tezi*, Anadolu Üniversitesi.

Bayam, K. A., & Örs, B. (2010). Differential power analysis resistant hardware implementation of the RSA cryptosystem. *Turkish Journal of Electrical Engineering & Computer Sciences*, **18**(1), 129-140.

Bellare, M. & Rogaway, P. (1994). Optimal Asymmetric Encryption-How to encrypt with RSA. *Advances in Cryptology-CRYPTO'94*.

Çenesiz, F. & Soğukpınar, D. (2000). Kurumsal Ağ Güvenliğinde Sayısal İmza Kullanımı: Tasarım ve Uygulama. *5.Bilgisayar Ağları Sempozyumu BAS2000*, Ankara, 198-207

Gupta, V., Gupta, S., Chang, S., & Stebila, D. (2002, September). Performance analysis of elliptic curve cryptography for SSL. *In Proceedings of the 1st ACM workshop on Wireless security* (pp. 87-94). ACM.

Jahan, I., Asif, M., & Rozario, L. J. (2015). Improved RSA cryptosystem based on the study of number theory and public key cryptosystems. *American Journal of Engineering Research (AJER)*, **4**(1), 143-149.

Koç, Ç.K. (1994). High-Speed RSA Implementation. *Technical report. RSA Laboratories TR201*
Koltuksuz, A., Güvenlik, E. T., & Denetimi, Ö. (1998). Doğruluk, Bütünlük ve Sayısal İmza. *4. Türkiye İnternet Konferansı,* İstanbul-Türkiye.

Kurt, M. (2012). Eliptik Eğri Şifreleme Algoritmasının Uygulaması Ve Analizi. *Yüksek Lisans Tezi*, Trakya Üniversitesi/Fen Bilimleri Enstitüsü, Edirne.

Montgomery, P. L. (1985). Modular multiplication without trial division. *Mathematics of computation*, **44**(170), 519-521.

Nagar, S. A., & Alshamma, S. (2012, March). High speed implementation of RSA algorithm with modified keys exchange. *In 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)* (pp.639-642). IEEE.

NESSIE Consortium, (2003). NESSIE Security Report, *Technical report NESSIE*.

Okumuş, İ. (2012). RSA Kritosisteminin Hızını Etkileyen Faktörler. *Doktora Tezi*, Atatürk Üniversitesi.

R. L. Rivest, A. Shamir and L. Adleman.(1978). "A method for obtaining digital signatures and public-key cryptosystems" *Communications of the ACM*, vol. 21, pp. 120-126.

Schneider B. (1996). *Applied Cryptography*.New York.

Stallings, W. (1995). *Network and internetwork security: principles and practice* (Vol. 1). Englewood Cliffs: Prentice Hall.

Stallings W. (1998). *Cryptography and Network Security: Principles and Practice*. Prentice Hall. ISBN 0-13-869017-0.

Stallings, W. *"Cryptography and network security* vol. 2" prentice hall, 2003.

Verma, S., & Garg, D. (2011). Improvement in RSA cryptosystem. *Journal of Advances in Info*
Yerlikaya, T. (2006). Yeni Şifreleme Algoritmalarının Analizi. *Doktora Tezi*, Trakya Üniversitesi.

Yerlikaya, T., Buluş, E., Ve Arda, D. (2005). Asimetrik Kriptosistemler ve Uygulamaları. *II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi,* İstanbul

Yerlikaya, T. (2006). Yeni Şifreleme Algoritmalarının Analizi. *Doktora Tezi,* Trakya Üniversitesi.

Yerlikaya, T., Buluş, E., & Buluş, H. N. (2007). RSA şifreleme algoritmasının pollard RHO yöntemi ile kriptanalizi. *IX. Akademik Bilişim'07. rmation Technology,* **2**(3), 146-151

Yerlikaya, T., Gençoğlu, H., Emir, M. K., Çankaya, M., & Buluş, E. (2013). Rsa Şifreleme Algoritması Ve Aritmetik Modül Uygulaması. *İstanbul Aydın Üniversitesi Dergisi,* **3**(9), 95-104.

Yıldırım, H. M. (2014). Bilgi Güvenliği ve Kriptoloji. Uluslararası Adli Bilişim Sempozyum. http://hmurat.bilkent.edu.tr/kripto 01062014.pdf

Yıldırım, K. (2006). Veri Şifrelemesinde Simetrik Ve Asimetrik Anahtarlama Algoritmalarının Uygulanması (Hybrid Şifreleme). *Yüksek Lisans Tezi,* Kocaeli Üniversitesi/Fen Bilimleri Enstitüsü, Kocaeli.

Algorithms and performances increasing the encoding speed of rsa algorithm

Extended Abstract

Thanks to the developing technology, the amount of data circulating in the network is increasing day by day and this increase brings with it the problem of data protection. In order to ensure the security of the data in communication and to understand whether it is transmitted to the correct address, a variety of methods called encryption and digital signatures are used. As the telecommunication network has grown explosively and the internet grows rapidly, information security becomes more and more significant.

In this study, asymmetric and symmetric encryption is divided into two of these encryption methods asymmetric encryption method is discussed. The RSA (Rivest, Shamir, Adelman) algorithm, which is used as the most broadcast among asymmetric encryption algorithms, was investigated. This algorithm relies on the difficulty of factoring large numbers which has seriously affected its performance and so restricts its use in wider applications.

There are some problems when encrypting data using RSA algorithm. The most important of these problems is the speed problem. This paper aims to speed up the implementation of the RSA algorithm Although RSA algorithm is very secure, it is rarely used in smart card, due to its long computation time
In the RSA algorithm, encryption and decryption of data is slower because the mathematical operations used for encryption and decryption(decryption) operations are intensive and because of the use of very large prime numbers to increase security. Therefore, a variety of algorithms are used to increase the encryption and decryption speed of the RSA algorithm. The effects of these algorithms in increasing the speed of RSA were compared with the working performances of these algorithms during the encryption process.

In this study, firstly, algorithms developed to increase the encryption rate of RSA cryptosystem were investigated. Later, the standard RSA algorithm was implemented using the Java programming language. As a result of this application, the RSA algorithm's private key discovery and encryption time were calculated on average. After the performance of the standard RSA algorithm was measured, .the application of the RSA encryption algorithm was performed separately using the Fast Modular Exponentiation, Binary Modular Exponentiation, and Montgomery Modular Multiplication algorithm which increased encryption speed. The performance of RSA cryptosystem was measured for each algorithm used as a result of the applications. The measured performances were first compared with the operational performance of the Standard RSA algorithm. According to the data obtained, the fastest encryption was performed when using the Fast Modular Exponentiation Algorithm. When using the Fast Modular Exponentiation algorithm, encryption speed was approximately 2.5 times faster than the standard RSA algorithm.

Encryption time of Binary Modular Exponent Algorithm Compared to with Standard RSA, this algorithm is 2 times faster than the Standard RSA. The Montgomery Modular Multiplication algorithm, which is one of the algorithms that increase the RSA encryption speed, is not only performing well in the software. As a result of the researches done, it is understood that this algorithm performs the encryption process approximately 2 times faster than the standard RSA algorithm when the necessary hardware support is provided.

When using 256-bit p and q numbers, the fastest encryption was performed by Binary Modular Exponent Algorithm, and as the number of these numbers increased, the performance of this algorithm decreased and the Fast Mode Acquisition algorith performed better. When the algorithms are compared among themselves, the algorithm that makes encryption the fastest has been found to be the Fast modular Exponentiation algorithm.

When these applications were performed, the length of the public key used in the RSA encryption algorithm was taken as 3649134810816461 and the message that was encrypted was determined as 123620.In addition, it has been seen that increasing the length of p and q numbers increases the encryption time of RSA encryption algorithm.

***Keywords**: Asymmetric Encryption, Rsa, Speed of Rsa*