

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**RSA YÖNTEMİNDE GÜVENLİK AÇIĞI VE GÜÇLENDİRİLMİŞ RSA
YÖNTEMLERİNİN İNCELENMESİ İLE ÖNERİLEN BİR HİBRİT
ŞİFRELEME MODELİNİN AĞ GÜVENLİĞİ AÇISINDAN
DEĞERLENDİRİLMESİ**

MOHAMMAD RAHİQ BAİGZAD

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Dr. Öğr. Üyesi Tarık YERLİKAYA

EDİRNE-2021

MOHAMMAD RAHİQ BAİGZAD'ın hazırladığı “RSA Yönteminde Güvenlik Açığı ve Güçlendirilmiş RSA Yöntemlerinin İncelenmesi ile Önerilen Bir Hibrit Şifreleme Modelinin Ağ Güvenliği Açısından Değerlendirilmesi” başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Bilgisayar Mühendisliği** Anabilim Dalında bir **Yüksek lisans tezi** olarak kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad):

İmza

Dr.Öğr.Üyesi Tarık YERLİKAYA

.....

Dr.Öğr.Üyesi Cem TAŞKIN

.....

Dr.Öğr.Üyesi Hakan GENÇOĞLU

.....

Tez Savunma Tarihi: 17/05/2021

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

(Dr. Öğr. Üyesi Tarık YERLİKAYA)

İmza

Tez Danışmanı

.....

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

.....

Prof.Dr.Hüseyin Rıza Ferhat KARABULUT

Fen Bilimleri Enstitüsü Müdürü

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

..... /..... /

Mohammad Rahiq BAIGZAD

İmza

Yüksek Lisans Tezi

RSA Yönteminde Güvenlik Açığı ve Güçlendirilmiş RSA Yöntemlerinin İncelenmesi
ile Önerilen Bir Hibrit Şifreleme Modelinin Ağ Güvenliği Açısından Değerlendirilmesi

Trakya Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

RSA gibi bir asimetrik şifreleme yöntemi her zaman açık anahtar kullanılarak gizli anahtarı elde etmeyi amaçlar. Gerçek mesaj bütününü oluşturacak özellikle küçük değerlere sahip bir açık anahtar genele açık bir ağ kanalı üzerinden paylaşıldığında, saldırgan bu anahtarı kullanarak faktörleme işlemini kolaylıkla gerçekleştirir ve gizli anahtarı elde eder. Diğer yandan bir simetrik algoritması düşünüldüğünde, bu yapılar güvenilir ancak taraflar arasında özel anahtar değişimi istendiğinde bu işlemin bir genel ağ üzerinden gerçekleşmesi saldırganın özel anahtarı elde etmesini sağlar. Bu çalışmada temelde ağ güvenliği, RSA güvenliği ve esas olarak simetrik yapısına yatkın ancak asimetrik tekniğini de kullanan (özel anahtar değişimi süreci) yeni bir Hibrit Şifreleme Modelinin performans değerlendirmesi RSA ile kıyaslamalı bir şekilde ele alınmaktadır. Değerlendirmeler sonucundan, Önerilen Hibrit Şifreleme Modelinin hem güvenlik açısından hemde zaman ve alan karmaşıklığı açısından verimli olduğu söylenebilmektedir.

Yıl : 2021

Sayfa Sayısı : 74

Anahtar Kelimeler: Ağ Güvenliği, RSA Güvenliği, Güçlendirilmiş RSA Yöntemleri,
Önerilen Hibrit Modeli

Master Thesis

Evaluation of a Proposed Hybrid Encryption Model in Terms of Network Security with Examination of Vulnerability in RSA Method and Reinforced RSA Methods

Trakya University Institute of Natural Sciences

Computer Engineering Department

ABSTRACT

An asymmetric encryption method such as RSA always aims to obtain the private key using the public key. When a public key with small values that will constitute the real message is shared over a public network channel, the attacker can easily perform the factoring process by using this key and obtain the secret key. On the other hand, considering a symmetric algorithm, these structures are reliable, but when the private key exchange between the parties is requested, the process takes place over a public network allowing the attacker to obtain the private key. In this study basically, network security, RSA security and the performance evaluation of a new Hybrid Encryption Model, which is mainly prone to symmetrical structure but also uses asymmetric technique (private key exchange process) is discussed in a comparative way with RSA. As a result of the evaluations, it can be said that the Proposed Hybrid Encryption Model is efficient both in terms of security and in terms of time and space complexity.

Year : 2021

Number of Pages: 74

Keywords : Network Security, RSA Security, Reinforced RSA Methods, A Putative Hybrid Model

ÖNSÖZ

Aldığım yüksek lisans eğitim süresince, şahsımla değerli bilgilerini paylaşan saygı değer jüri hocalarıma, özellikle her kelimesinin hayatıma kattığı önemini unutmayacağım saygıdeğer danışman hocam **Dr. Öğr. Üyesi Tarık YERLİKAYA**'ya ve bu süreç boyunca biran da olsun yardımını esirgemeyen değerli aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT	v
ÖNSÖZ.....	vi
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	x
BÖLÜM 1.....	1
GİRİŞ VE AMAÇ	1
1.1. Araştırma Hipotezleri	3
1.2. Araştırma Modeli	4
BÖLÜM 2.....	5
AĞ GÜVENLİĞİNE İLİŞKİN KAVRAMSAL İNCELEMELER	5
2.1. İnternetin Kısa Tarihçesi	5
2.2. Bir Ağın Amacı	6
2.3. Ağ Güvenliği İhtiyacı	6
2.4. Ağ Güvenliği Tarihçesi ve Zaman Çizelgesi.....	7
2.5. Ağ Güvenliği	8
2.6. Güvenilir Sistemler ve Güvenlik Sistemleri.....	9
2.7. Bilgi Güvenliği Farkındalığı	10
2.8. İnternet Güvenliği Saldırısının Sınıflandırılması	11
2.8.1. Pasif Saldırı.....	11
2.8.2. Aktif Saldırı	12
2.9. İnternet Mimarisi ve Kırılgan Güvenlik Boyutları	14
2.9.1. İPv4 Mimarisi	15
2.9.2. İPv6 Mimarisi	15
2.10. Türkiye’de Gerçekleşen Ağ Saldırıları Örneği.....	16
2.10.1. Ankara’da Tapu Bigilerinin Sızdırılması	16
2.10.2. TEİAŞ Kurumuna Saldırı.....	16
2.10.3. HSBC Bankasına Saldırı	17
2.11. Ağ Güvenliğine Yönelik Tehditler ve Önlemler	17
2.11.1. Kötüçül Yazılımlar Saldırısı	18
2.11.2. Hizmet Dışı Bırakma Saldırıları (DoS/DDoS, Botnet).....	19

2.11.3.	<i>SQL Enjekte Saldırıları</i>	19
2.11.4.	<i>Sazan Avlama (Phishing) Saldırıları</i>	19
2.11.5.	<i>XSS Saldırıları</i>	20
2.11.6.	<i>Sosyal Mühendislik Saldırıları</i>	20
2.12.	Bazı Simetrik Şifreleme Algoritmaları	20
2.12.1.	<i>Veri Şifreleme Standardı (DES)</i>	20
2.12.2.	<i>Gelişmiş Şifreleme Standardı (AES)</i>	21
2.12.3.	<i>Blowfish</i>	21
2.12.4.	<i>Twofish</i>	21
2.13.	Bazı Asimetrik Şifreleme Algoritmaları.....	22
2.13.1.	<i>RSA Algoritması</i>	22
2.13.2.	<i>Diffie-Hellman Anahtar Değişimi</i>	22
2.13.3.	<i>Eliptik Eğri Şifrelemesi</i>	23
2.13.4.	<i>Dijital İmza Standardı</i>	23
2.14.	Karma İşlevleri	23
BÖLÜM 3		24
RSA YÖNTEMİNİN GÜVENLİK ZAFİYETİ VE GÜÇLENDİRİLMİŞ RSA YÖNTEMLERİNİN İNCELENMESİ		24
3.1.	RSA Algoritmasının Temel Güvenlik Zafiyeti	24
3.1.1.	<i>Klasik Faktörleme (GNFS)</i>	25
3.1.2.	<i>Kuantum Hesaplaması ve Klasik Hesaplaması</i>	26
3.1.3.	<i>Kuantum Algoritmalarına Karşın Savunmasız Kripto Sistemleri</i>	26
3.1.4.	<i>RSA Asimetrik Şifreleme Yönteminin SHOR Algoritması İle Kırılganlığı</i>	28
3.1.5.	<i>Güçlendirilmiş RSA Yöntemleri</i>	31
BÖLÜM 4		38
ÖNERİLEN HİBRİT ŞİFRELEME MODELİ		38
4.1.	Motivasyon	38
4.2.	Çalışma Mekanizması	40
4.2.1.	<i>Örneklendirme</i>	42
4.2.2.	<i>Süreç Diyagramı</i>	43
4.2.3.	<i>Kod Örneği (C#)</i>	44
4.2.4.	<i>Zaman ve Alan Karmaşıklığı</i>	45
BÖLÜM 5		54
SONUÇ VE TARTIŞMA		54
KAYNAKÇA		58
ÖZGEÇMİŞ		64

ŞEKİLLER DİZİNİ

Şekil 1.1. Önerilen Hibrit Şifreleme ve Asimetrik Şifreleme tercih modeli.	4
Şekil 2.1. Bir saldırının analiz şeması.	18
Şekil 4.1. Önerilen Hibrit Şifreleme modelinin süreç şeması.	43
Şekil 4.2. RSA modelinde $M=3$ gizli mesajı için şifreleme süresi.	46
Şekil 4.3. RSA modelinde $M=3$ mesajını şifrelemek için döngü sayısı.	46
Şekil 4.4. Önerilen Hibrit Modelinde $M=3$ gizli mesajı için şifreleme süresi.	46
Şekil 4.5. Önerilen Hibrit Modelinde $M=3$ mesajını şifrelemek için döngü sayısı.	46
Şekil 4.6. RSA modelinde $M=13$ gizli mesajı için şifreleme süresi.	47
Şekil 4.7. RSA modelinde $M=13$ mesajını şifrelemek için döngü sayısı.	47
Şekil 4.8. RSA modeli ve Önerilen Modelin şifreleme süresi karmaşıklığı.	49
Şekil 4.9. RSA modelinde şifreli mesaj=12 için deşifreleme süresi.	49
Şekil 4.10. RSA modelinde şifreli mesaj=12 çözmek için döngü sayısı.	49
Şekil 4.11. Önerilen Hibrit Modelinde şifreli mesaj=12 için deşifreleme süresi.	50
Şekil 4.12. Önerilen Hibrit Modelinde şifreli mesaj=12 çözmek için döngü sayısı.	50
Şekil 4.13. RSA modeli ve Önerilen Modelin deşifreleme süresi karmaşıklığı.	50

ÇİZELGELER DİZİNİ

Çizelge 2.1. 2013 yılı web uygulama saldırıları.	20
Çizelge 3.1. Kuantum bilgisayarlarının şifreleme yöntemleri üzerindeki etkisi.....	27
Çizelge 4.1. RSA modeli ve Önerilen Modelin şifreleme süresi karmaşıklığı.	48
Çizelge 4.2. RSA modeli ve Önerilen Modelin deşifreleme süresi karmaşıklığı.	51
Çizelge 4.3. Önerilen Hibrit şifreleme modelinin zaman ve alan karmaşıklığı.	53

BÖLÜM 1

GİRİŞ VE AMAÇ

Yaygın olarak kullanılan veri işleme çabalarından önce, bir organizasyon için değerli olduğu düşünülen bilgilerin güvenliği fiziksel yollarla sağlanmıştır (Stallings, 2006). İkinci dünya savaşı akımında ilk bilgisayarların yapımından yirmi yıl sonra ağ bağlantılarına yönelik birincil fikirler ortaya konulmaya başlamıştır. Bu dönemlerde bilgisayarların hem maliyetini hem de uzun bir mesafeden bağlantılı hale getirilmesi ilginç bir fikirdi (Bonaventure, 2011). Bulunduğumuz yüzyılda da gelişmelerin çoğunda bilgi teknolojisinin itici bir güç olduğu görülmektedir. Bu teknolojilerin bilgilerin toplanması, işlenmesi ve dağıtılması konusunda bir devrim başlattığı ortadadır. Bu devrimin yaşanması da, bilgi işleme gibi çabalar ve teknolojiler arası birleşmeler olmadan mümkün olamaz. Günümüzdeki telefonlar, radyo, televizyon ve bilgisayarlar bu gelişmeler için kullanılan nevi araçlardır. Yani farklı coğrafi konumlarda dağıtılmış bilgisayarlar veri ve bilgi alışverişine izin vermek için birbirlerine bağlanabilirler. Bu bağlantılar bilgisayar ağlarını oluşturmaktadır. Bilgisayar ağları, ağ kullanıcılarına bilgileri depolayabilmelerine, alabilmelerine ve paylaşabilmelerine izin veren aygıtlar topluluğudur (Tamimi & Khalifa, 2010). Ancak özel bilgi paylaşımı konusuna gelince benzer ağların her zaman açık bir pozisyonda olması bilgi güvenliğine ilişkin büyük endişe yaratmaktadır. Çünkü ağ ortamındaki güvenlik karmaşıklığını yönetmek için belli bir teknik bulunamamaktadır. Ayrıca bir ağın güvenliği hakkında düşünüldüğünde, tüm ağın güvenliğinin sağlanması üzerine düşünülmeli, ve sadece iletişim zincirinin her

iki ucundaki bilgisayarların güvenliği ile ilgilenmemelidir. Diğer yandan Bir düğümden diğer düğüme veri transferi gerçekleştirilirken iletişim kanalının pasif veya aktif saldırılara açık olmaması gerekmektedir. Zira ağ saldırganları iletişim kanalını hedef alarak verileri çalabilir, verileri deşifre ederek başka bir mesajı orijinal mesaj üzerine ekleyebilmektedir (Pawar & Anuradha, 2015). Buna bağlı olarak bilim adamları benzer saldırılara karşı koyabilecek ve ağ güvenliğini sağlayabilecek çeşitli protokol tabanlı kriptoloji tekniklerini, açık anahtarlı asimetrik kriptoloji tekniklerini ve gizli anahtar simetrik kriptoloji teknikleri gibi gizleme ve şifreleme yöntemlerini geliştirmiş/geliştirmektedirler. Ancak bu yöntemlerin çoğu aslında bilgiyi tamamen gizlemek için herhangi bir garanti sağlamamakta, sadece bilginin karmaşık hale getirilmesi üzerine odaklanmaktadır; örneğin, RSA gibi bazı asimetrik algoritmalar.

RSA algoritması sayı teorisi esasına dayanan gelişmiş açık anahtarlı şifreleme yapısına sahip bir kriptoloji yöntemidir. Bu algoritmanın temel güvenlik sorunu şifreleme için kullanılan asal sayılarının çarpanlara ayırma karmaşıklığına bağlı olduğu söz konusudur. Bu bağlamda zayıf bir anahtar üretme eylemi RSA algoritmasının saldırılara karşın savunmasız bir pozisyonda bırakacağı açıktır. Bu nedenle RSA algoritmasında bir $\text{mod } n$ işlemini gerçekleştirmek için büyük asal sayıların kullanılmasına dikkat edilmelidir (Steyn, 2012). RSA algoritmasını etkili kılan temel sebeplerden biri bir N büyük tam sayısını oluşturan p ve q çarpanlarının hesaplamalı olarak kolay bir şekilde faktörlere ayrılamaz olduğudur (Frenkel, 2013; Rouse, 2014). Örneğin, aralık 2009 yılında 232 ondalık sayı içeren 768 bitlik bir RSA modülü 13 araştırmacı tarafından binlerce paralel bilgisayar kullanılarak iki yıllık bir süre içerisinde çarpanlara ayrılmıştır. Bu işlem tek çekirdekli bir 2.2GHz'lık AMD işlemcisiyle yapıldığında 2000 senelik bir süre içerisinde gerçekleşmesi öngörülmüştür (Frenkel, 2013; Wikipedia, 2020). Federal Bilgi İşleme Standartları Yayını (FIPS PUB) 186-4, n modülünün uzunluğu için 1024 bit, 2048 bit ve 3072 bit olmak üzere üç seçenek belirtmiştir (Information Technology Laboratory [ITL], 2013). Çünkü RSA'nın güçlülüğü büyük asal sayı anahtarı boyutuna bağlıdır. Bu nedenle FIPS PUB 186-4 (ITL, 2013) ile çalışan çoğu donanımlar ve yazılımlar güvenliği sağlamak için 1024 bit uzunluğunun kullanımından artık 2048 bit'e geçmektedirler. Ancak hesaplama gücünün artması, daha verimli faktörlere ayırma algoritmaları ve kriptanaliz tekniklerinin gelişmesi, benzer anahtarları kırma yeteneğini de artacaktır (Rouse, 2014). Örneğin,

bazı kriptu adamları kuantum hesaplama bilimlerinin günümüzde kullanılan şifreleme yöntemleri üzerinde yıllar geçmeden etki yaratamayacağına ikna olmuş görünmekte, ve 2031 yılından öncesinde kuantum kriptolojisinde gerçek bir ilerleme kaydedilemeyeceğini beklemekteler (Leek, 2014). Oysaki kuantum mekaniğini kullanan Shor algoritmasının 2001 yılında büyük sayıları faktörleri ayırmak üzere kullanıldığında 15 sayısını çarpanlarına ayırabildiği MIT fizikçisi ve elektrik mühendisi Isaac Chuang tarafından ortaya konulmuştur. Isaac Chuang çabası henüz RSA şifrelemesini kırmak için gereken boyutta işlevsellik sağlayan bir kuantum bilgisayarının olamamasını belirtmekle birlikte, yine de bu deneyimin bir kuantum bilgisayarının şifreleme sistemleri için oluşturduğu tehdidi işaret ettiği ortadır (Nordrum, 2016).

Bu tezde, günümüzde yaygınca kullanılan asimetrik şifreleme yöntemlerinin güvenlik zafiyetlerinin RSA algoritması örnek alınarak değerlendirilmesi, daha sonra bilim insanlarının RSA güvenlik zafiyetlerine karşın geliştirdikleri güçlendirilmiş/değiştirilmiş RSA yöntemlerinin bu zafiyetleri hangi düzeyde gidermiş olduğunun incelenmesi ve elde edilen sonuçlarla Önerilen bir Hibrit Şifreleme modelinin güvenlik performansının aşağıdaki hipotezlere göre değerlendirilmesi amaçlanmıştır.

1.1. Araştırma Hipotezleri

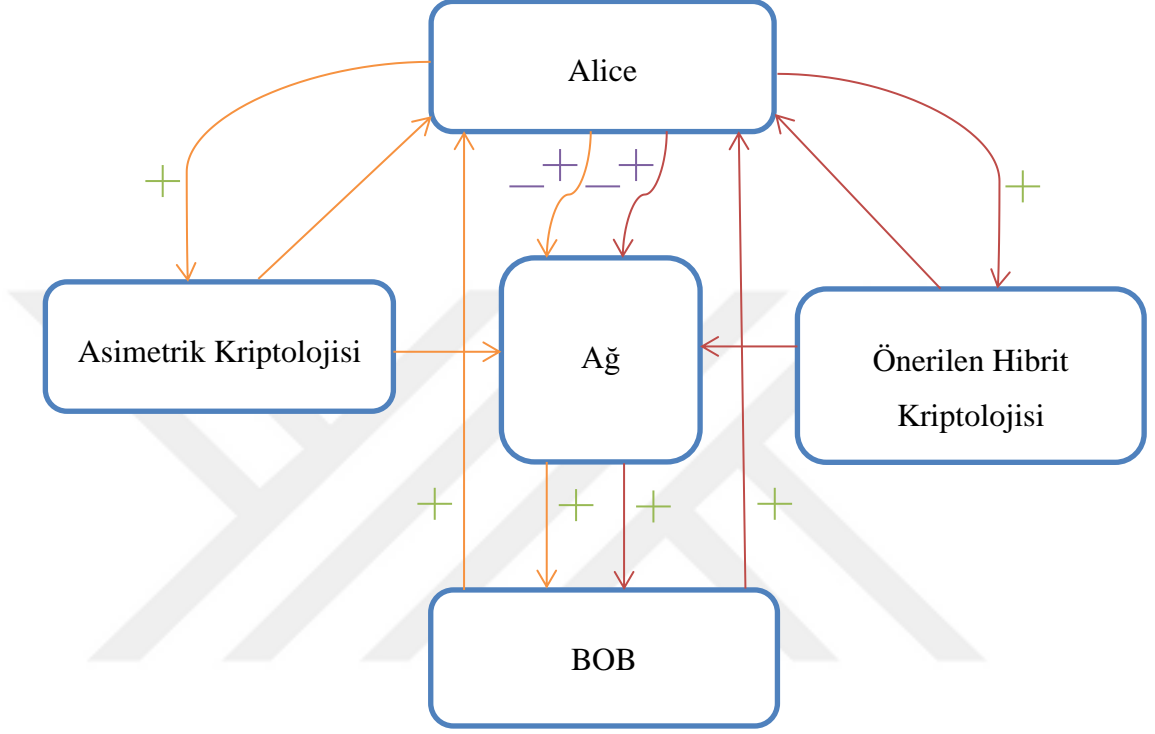
H_0 : Önerilen Hibrit Şifreleme yöntemi güvenilirirdir.

H_1 : Önerilen Hibrit Şifreleme yöntemi günümüzdeki Asimetrik şifreleme yöntemine kıyasla şifreleme ve şifre çözme işlemlerini en az zaman dilimi içerisinde gerçekleştirebilmektedir.

H_2 : Önerilen Hibrit Şifreleme yöntemi günümüzdeki Asimetrik şifreleme yöntemine kıyasla özel anahtar değişimini en az zaman dilimi içerisinde gerçekleştirebilmektedir.

1.2. Araştırma Modeli

Aşağıda, Alice ve Bob arasında güvenli bir veri transferi için Önerilen Hibrit Kriptolojisi veya Asimetrik Şifreleme Kriptolojisi tercih modeli verilmiştir.



Şekil 1.1. Önerilen Hibrit Şifreleme ve Asimetrik Şifreleme tercih modeli.

BÖLÜM 2

AĞ GÜVENLİĞİNE İLİŞKİN KAVRAMSAL İNCELEMELER

2.1. İnternetin Kısa Tarihçesi

İnternet 1969 yılında, Savunma Bakanlığı (DOD) tarafından Gelişmiş Araştırma Projeleri Ajansı Ağı (ARPANET) araştırmaları kapsamında ortaya konuldu. ARPANET başlangıçta bilim insanlarının farklı konumlardaki uzak bilgisayarlara bağlanabilmeleri ve bilgileri paylaşımları amacıyla tasarlanmıştır. Örneğin, e-posta iletileri günümüzde de en popüler uygulama olarak tanımlanabilir. İnsanlar araştırma projelerinde işbirliği yapmak ve çeşitli konuları tartışabilmek için ARPANET'i kullandıklarından, bu teknoloji en verimli bir dijital postanesi haline dönüşmüştür.

1980'lerde Bob Kahn ve Vinton Cerf internetin ortak dili olan TCP/IP'yi oluşturdu. İlk kez ARPANET'i oluşturan ağların gevşek koleksiyonu bugünkü kullandığımız internet'i oluşturdu. Ayrıca 80'li yılların ortaları kişisel bilgisayar ve süper mini bilgisayarın varlığı ile büyük bir patlamaya neden oldu. Bu dönemlerde ucuz masaüstü makineleri ve güçlü ağ sunucuları birçok şirketlerin İnternet'e ilk kez bağlanmalarını sağladı. Daha sonra 1990'larda internet herkese açık bir şekilde sunulmaya başladı ve WWW terimi doğdu. Bu sırada Netscape ve Microsoft, internet için bir tarayıcı geliştirmek için aralarında rekabet ettiler (Elawad & Dawo, 2016).

2.2. Bir Ağın Amacı

Farklı bilgisayarları herhangi bir yerden herhangi bir başka bilgisayara bağlayabilme kabiliyeti büyük bir imkândır. Çoğu şirket, çevrimiçi, ticari sırlar, ürün geliştirme planları, pazarlama stratejileri, finansal analizleri vb. kaynaklar büyük miktarda gizli bilgiye sahiptirler. Bu bilgilerin çeşitli bilgi sızıntı gibi saldırılara maruz kalması ve böylece bilginin kötü niyetli kişiler tarafından okunur olması korkunç sonuçlara yol açması muhtemeldir. Örneğin virüsler, solucanlar ve diğer dijital zararlıları güvenliği bozabilir, değerli verileri yok edebilir ve karışıklığı temizlemek büyük zaman kayıplarına yol açabilir. Bu nedenle internet, alanlar-arası yönlendirme konusunda BGP (Sınır Ağ Geçidi Protokolü) protokolünü kullanır. Ancak BGP oturumları veri iletmek için TCP kullandığından, TCP tabanlı saldırılardaki artış, BGP güvenliği için nevi tehdit olduğu söylenmektedir. Geçmişte, internet topluluğu veri iletimlerinde SNMP (Basit Ağ Yönetimi Protokolleri) protokolünden kullanılmıştır (Sushila & Sunita, 2014).

2.3. Ağ Güvenliği İhtiyacı

Ağ modellerinin muhtemel saldırılara karşı güvenliğe ihtiyacı vardır. Ağ güvenliği iki temel güvenlik noktasına odaklanmaktadır.

- ✓ **Veri güvenliği:** bilgilerin yetkisiz erişim ve kayıplara karşı korunmasını sağlar.
- ✓ **Bilgisayar ağ güvenliği:** verileri bilgisayar korsanlarından korur. Burada ağ güvenliği yalnızca tek bir ağda değil, birden fazla ağlarda güvenlik sağlamak demektir.

Bir ağ güvenlik modeli, hem verileri hem de ağ sistemlerini korumak için uzman kişileri gerektirir. Ağ güvenliği uzmanları bilgisayar ağ güvenliğini koruyabilmek, ekonomik kayıpları önleyebilmek ve ağ tehditlerini teşhis edebilmek için güvenlik sistemlerinin tasarımı, korunması ve incelenmesi üzerinde odaklanırlar. Teknoloji yapılandırmaların gelişmesiyle ağ güvenliği analizi yöntemlerinde sürekli yenilikler olmuş ve ağ güvenliğine ilişkin model yapımları sürekli olgunlaşmıştır (Gupta, 2014).

2.4. Ağ Güvenliği Tarihçesi ve Zaman Çizelgesi

Son zamanlarda güvenliğe olan ilgi Kevin Mitnick'in işlediği suçtan kaynaklandı. Kevin Mitnick, ABD tarihinin bilgisayarla ilgili en büyük suçunu işledi ve kayıplar sonucu seksen milyon dolara çıktı. O zamandan sonra bilgi güvenliği hep gündeme geldi (Kartalopoulos, 2008).

Bilgilerin internet üzerinden paylaşılr olması nedeniyle bilgi güvenliğinin de sağlanması gerekmektedir. Geçmişte kullanılan internet protokolleri kendilerini güvence altına alabilmek için geliştirilmemişti, bir TCP/IP yığının, güvenlik protokolleri uygulanamazdı. Benzer yapılandırmalar interneti saldırılara açık kalmasına sebep oldu. Son dönemlerde internet mimarisindeki modern gelişmeler sayesinde ağ ortamı daha güvenilir hale gelmiştir (Elawad & Dawo, 2016). Bilgisayar ve ağ güvenliğinin meydana gelmesi için birçok olay rol almıştır. Güvenliğin zaman çizelgesi 1930'lara kadar uzayabilir. 1918'de Polonyalı bir kriptografist düz mesajları şifreli hale getirmek için bir muamma makinesi icat etti. 1930'da Alan Turing ismini taşıyan bir matematikçi bu bilmecenin kodunu kırdı. Bu nedenle ikinci dünya savaşı sırasında ağ bağlantılarının güvence altına almak zorunlu hale getirildi (Elawad & Dawo, 2016).

1960'lı yıllarda "Hacker" terimi birkaç Massachusetts Teknoloji Enstitüsü (MIT) öğrencisi tarafından tanımlandı. Savunma bakanlığı, elektronik veri ve bilgi alışverişi için bir kanal olarak popülerlik kazanan ARPANet'in kullanımını başlattı ve bugün internet olarak bilinen taşıyıcı ağı oluşmasına zemin açtı. 1970'lerde TelNet protokolü geliştirildi. Bu başlangıçta devlet müteahhitleri ve akademik araştırmacılarla sınırlı olan veri ağlarını kamu erişimine sundu (Kartalopoulos, 2008). 1980'lerde bilgisayarlarla ilgili bilgisayar suçları ortaya çıkmaya başladı. Örneğin; Yetkililer tarafından düzenlenen 414 çetesi, dokuz günlük bir çatlama sonucundan sonra gizli sistemlere girmeyi başardı. 1986 tarihli Bilgisayar Sahtekarlığı ve Kötüye Kullanımı Yasası, Ian Murphy'nin askeri bilgisayarlarından çaldığı bilgi suçu nedeniyle oluşturuldu. Yüksek Lisans öğrencisi Robert Morris, Morris solucanını denemek amacıyla internet'e bağlı 6.000'den fazla savunmasız bilgisayarları etkisiz hale getirdiği için mahkûm oldu. Morris, solucanının çoğaltılabileceğini düşünerek, bilgisayar kullanıcılarını ağ güvenliği sorunları konusunda uyarmak için Bilgisayar Acil Durum Müdahale Ekibi (CERT)

oluşturdu (Elawad & Dawo, 2016). 1990'larda internet halka açıldığında, güvenlik kaygısı muazzam derecede arttı.

Bugün dünya genelinde yaklaşık 950 milyon insan internete bağlı bulunmaktadır. Akımda olan günlerde güvenlik ihlali ile ilgili yaklaşık 225 büyük olay kaydedilmiştir. Bu güvenlik ihlallerinin büyük ölçüde parasal kayıplara yol açması öngörülmüştür. Bu nedenle, uygun bir güvenliğe yatırım yapmak için, büyük organizasyonlar ile günlük kullanıcılar arasında farkındalık oluşturulması önem arz etmektedir (Kartalopoulos, 2008).

2.5. Ağ Güvenliği

Sultana ve Fouzia (2015) yazarlığında yayınlanan makalede, sistem ve ağ teknolojisi, çok çeşitli uygulamalar için önemli bir terimdir. Buna bağlı olarak güvenlik, ağlar ve uygulamalar için pek çok önem arz etmektedir. Ağ güvenliği, gelişmekte olan ağlarda kritik bir gereklilik olmasına rağmen, kolayca uygulanabilecek önemli güvenlik yöntemleri konusunda eksikliklere maruz kalmaktadır.

Güvenlik geliştiricileri ile ağ geliştiricileri arasında bir “iletişim boşluğu” söz konusu mevcuttur. Ağ tasarımı, Açık Sistemler Arayüzü (OSI) modeline dayanan geliştirilmiş bir süreçtir. Bu, modülerlik, esneklik, kullanım kolaylığı ve protokollerin standardizasyonunu sunar. Farklı katmanların protokolleri, modüler gelişime izin veren yığınları oluşturmak için kolayca birleştirilebilir. Bu da geliştirme esnekliğine izin verir. Ağ tasarımının aksine, güvenli ağ tasarımı aynı avantajları içermez. Ağ güvenliği ele alındığında, tüm ağın güvenli olması önemlidir. Ağ güvenliği yalnızca iletişim zincirinin her iki ucundaki bilgisayarların güvenliği değildir. Veri aktarılırken iletişim kanalı saldırılara açık olmamalıdır. Çünkü olası bir saldırıda saldırgan iletişim kanalını hedef alarak, verileri çalabilir, şifreleri çözebilir ve yanlış bir mesaj ekleyebilir. Ağın güvenliği, bilgisayarların güvenliği ve iletinin şifrelenmesi kadar önemlidir.

Bir ağ güvenliğinin kriterleri şunlardır (Dowd & McHenry, 1998):

- ✓ *Erişim:* Yetkisi olan kullanıcılara belirli bir ağ ile iletişim kurma olanağı sağlar.
- ✓ *Gizlilik:* Ağdaki bilgiler gizli tutulmalı.

- ✓ *Kimlik doğrulama*: Ağ kullanıcılarının, önceden tanımlanmış olmasından emin olunmalı.
- ✓ *Dürüstlük*: İletinin kanal boyunca değiştirilmediğinden emin olunmalı.
- ✓ *Reddetmeme*: Kullanıcının mevcut ağı kullandığından emin olunmalı.

Etkili bir ağ güvenliği planı geliştirmek için; güvenlik sorunları, potansiyel saldırganlar, gerekli güvenlik seviyesi ve bir ağı savunmasız hale getiren faktörlerin incelenmesi gerekmektedir (Dowd & McHenry, 1998). Bilgisayarın ağdaki güvenlik açığını azaltmak için birçok ürün bulunmaktadır. Bu araçlar şifreleme, kimlik doğrulama mekanizmaları, izinsiz giriş algılama, güvenlik yönetimi ve güvenlik duvarlarıdır. “İntranetler” hem internete bağlılar hem de internetten makul bir şekilde korunmasını sağlarlar. Sonuçta internetin güvenlik konularını anlamak, internet erişimi ve internet güvenliği olan ağlar için yeni güvenlik yaklaşımların geliştirilmesine büyük ölçüde yardımcı olur. Bir saldırı tespit sistemi en sık kullanılan saldırı türlerine göre kurulur. Ağa izinsiz girişler aşağıdaki nedenlerden oluşur:

- Kaynakları gereksiz yere tüketmek
- Herhangi bir sistem kaynağının işlevine müdahale etmek
- Daha sonraki saldırılardan yararlanabilecek sistem bilgisi kazanmak

Güvenlik protokolleri bazen OSI ağ referans modeli katmanının bir parçası olarak görünür. Ağ tasarımını güvence altına almak için katmanlı bir yaklaşım kullanılarak güncel çalışmalar yapılır. Bu tür güvenlik yaklaşımı, yaygın güvenlik sorunlarından bazılarını ortadan kaldırarak etkili bir tasarım sağlar (Sultana & Fouzia, 2015).

2.6. Güvenilir Sistemler ve Güvenlik Sistemleri

Fırlar’a (2003) göre güvenilir sistem, yoğun trafikte bile beklenen tüm performansı sergileyen ve herhangi bir tıkanmaya maruz kalmayan güçlü sistemlerdir. Bu nedenle de cihazların dikkatlice seçilmiş olması ve en iyi şekilde konfigüre edilmesi gerekmektedir. Güvenilir sistemler, internete bağlanan kurumsal ağların oluşabilecek tehditlere karşı korunması, kurum bilgilerine izin verildiği ölçüde erişilebilmesi ve hata mevcut kurum erişimlerinin de kontrol altında alınması demektir. Bir kullanıcı ağı

İnternet'e bağlanırken koruma duvarı (fire wall) gibi güvenlik sistemlerine sahip değilse, olası saldırılara maruz kalabilir. Bu nedenle de, bir sisteme giriş yapıldığında kullanıcı adı ve şifre sorgulaması gibi kapıların kuruluşu önem arz etmektedir. Bir güvenlik düzeyi, özel bir verinin hangi düzeyde korunabileceğini ifade eder. Veriler farklı düzeylerde korunabilir. Bu düzeylerden en alt düzey veri kaydı düzeyidir. Bir veri-tabandaki kayıtlara erişmek için bilgileri belirli alanlarda şifrelemek mümkündür. Böylece yalnızca erişim hakkı olan veya şifre anahtarına sahip kullanıcılar bilgiye erişebilirler. Kayıt alanı düzeyinde sıkı güvenlik sağlamak için güçlü yöntemler kullanılmaktadır. Bu tür güvenlik düzeyleri genel olarak koruma duvarları tarafından da sağlanmaktadır (Fırlar, 2003, s. 12).

2.7. Bilgi Güvenliği Farkındalığı

Bilgi güvenliği farkındalığı kurum, kuruluş veya kişisel verilerin güvenliğine yönelik oluşabilecek aksamaların kavranması, bunlara karşı önlemler alınarak bilgi güvencesinin sağlanması demektir.

İşletme çalışanlarının bilgi güvenliği politikalarına uyum sağlamaları nevi sosyo-organizasyonel bir faaliyet olarak önem arz etmektedir (Boss & Kirsch, 2007; Siponen, Pahlila & Mahmood, 2007). Çünkü insani kaynaklar bilgi güvenliğinin sağlanması konusunda en çok etkilenen olarak bilinmektedirler (Mitnick & Simon, 2002; Warkentin & Willison, 2009). Örneğin internet, akıllı cihaz vb. gelişmiş kullanımlarda da bireyler ancak bilgi seviyesi kadar önlem alabilmekteler. Örneğin, E-devlet bünyesinde kullanılan ulusal uygulamalar da artık ağ tehditlerine karşı savunmasız hale geldiği görülmektedir.

Kurumların uzun yıllar boyunca çaba harcayarak sahip oldukları değerli bilgilere yönelik güvenliği sağlamak ve sunduğu hizmetlere süreklilik kazandırmak için gizlilik, bütünlük ve erişilebilirliğin sağlanması gereklidir (Bensghir, 2008). Ayrıca verilerin benzer tehditlere karşı güven altına kalması şu üç temel sürecin bütüncül bir şekilde uygulanması gerekmektedir (Öztemiz & Yılmaz, 2013). Bu süreçlerden ilki, planlama ve politikaları kapsayan yönetsel süreç, ikincisi, virüsten korunma gibi önlem süreci, üçüncüsü de kullanıcı eğitimi veya bilgi güvenliği farkındalığı sürecidir. Bir kuruluşta bilgi güvenliğine yönelik sistematik bir şekilde odaklanmadığı durumlarda kurumların ciddi maddi zararlara maruz kalması muhtemeldir (Cavusoglu & Raghunathan, 2004).

Kişisel ve kurumsal olarak güvenliğe iki ana yönden bakıldığında bilgi güvenliği farkındalığının, genel farkındalık ve bilgi güvenliği politikaları farkındalığı olarak ikiye ayırt edebilmekteyiz. Genel bilgi güvenliği farkındalığı bireylerin bilgi güvenliği konusunda temel bilgilere, oluşabilecek problemler ve bu problemlerin oluşturacağı zararlar hakkında fikir sahibi olmalarını gerektirir. Bilgi güvenliği politikası farkındalığı ise işletme çalışanlarının güvenlik konusunda politikaların bilmeleri ve politikalarda barındırılan gereken hedefi anlamaları demektir (Bulgurcu, Cavusoglu & Benbasat, 2010). Bilgi güvenliğini arttırmak amacıyla verilen eğitimlerde, kurumların çalışanların sorumluluklarını bilmeleri gerekmektedir (Şahinaslan, Kantürk, Şahinaslan & Borandağ, 2009).

Bilgi güvenliği farkındalığını yaratmak için değişiklik oluşturan sistematik bir döngü içerisinde çalışılmalıdır. Dijle'nin Türkiye'de eğitimli insanların bilişim suçlarına yönelik yaptığı bir araştırmada, kötü niyetli yazılımları kullanarak bilgiyi çalan kullanıcı sayısı %51,7 ve firmalar tarafından açık veren lisanssız yazılımların kullanımı sonucunda meydana gelen bilgi sızıntısı %75 olarak belirlenmiştir (Öğütçü, 2010).

2.8. İnternet Güvenliği Saldırısının Sınıflandırılması

İnternet'e güvenlik saldırıları Pasif Saldırılar ve Aktif Saldırılar olarak iki sınıfa ayrılmaktadır (Ahmad, Verma, Kumar & Shekhar, 2011).

2.8.1. Pasif Saldırı

Pasif saldırılarda saldırgan orijinal mesajın içeriğini değiştirmek istemez. Trafik analizi, kokuşma ve tuş kaydedicileri pasif saldırıları çeşitlerindendir (Simmonds, Sandilands & Van, 2004; Welch & Lathrop, 2003).

A. Müdahale

Müdahale saldırısı pasif saldırı türü olarak kullanıcıların izni veya bilgisi olmadan yapılan bir saldırı türüdür. Güvenlik ilkesinde gizlilik kurallarını ihlal eder, ve mesaj kaybına yol açması mümkündür. Müdahale tekniğinin alt başlıkları şunlardır:

a. Mesajın Yayınlanması

Bir kiřiye mesaj gönderdiğimizde, sadece o kiřinin mesajı okuyabilmesini isteriz. Bu durumda belirli bir güvenlik mekanizması kullanarak mesaj içeriğinin yayılmasını önleyebiliriz. Örneğın, bir algoritma kullanarak mesajı řifreleyebiliriz.

b. Trafik Analizi

Birçok mesaj tek bir kanaldan geçerse, saldırgan bilgisinde mesaj hakkında bazı detaylar oluşabilir.

c. Koklama

Koklama, verileri koklamak için kullanılan bir yöntemdir. Bu teknik gönderenin izni olmadan verileri bulmaya çalışır.

d. Keylogger

Bu program, tuřlamaları kaydeden, arka planda çalışan bir programdır. Tuř giriři yapıldıktan sonra bilgi geri alınmak üzere sisteme gizlenir veya saldırgana ham olarak sunulur. Saldırgan gereken bilgiyi bulmak ve sisteme ziyan vermek için tüm gizlenmiř giriř verilerini dikkatlice inceler.

2.8.2. Aktif Saldırı

Aktif saldırılar, orijinal iletide deęiřiklik getiren veya bazı yanlıř mesajlar yazdıran saldırı türüdür. Bu tür saldırılar karmaşıktır ve kolayca önlenemezler. Bu tip saldırılar Kesinti, Fabrikasyon ve Deęiřiklik saldırıları olarak üç türe ayrılmaktadır. Bu kategoriler altında DoS, DDoS, DRDoS, SQL Enjeksiyonu, Tekrar Saldırısı, Maskeleye, Orta Saldırı gibi saldırılar yer almaktadır (Welch & Lathrop, 2003; Simmonds vd., 2004).

A. Kesinti

Bu saldırıda bir yetkili başka bir varlık gibi davranır. Örneğın A, B ve C üç kullanıcı olsun. A kullanıcısı C kullanıcısı gözüyle B kullanıcısına bir mesaj gönderebilir. B kullanıcısı mesajın C kullanıcısından geldiğine inanır. Burada bir doęrulama kesintisi oluştuğunda kaynağın kullanılabilirliğı tehlikeye girebilir. Bu tip saldırılar dört tipte sınıflandırılır.

a. Hizmet Reddi (DoS)

İstekleri alan bir sistem başlatıcı ile bir geri dönüş iletişim oluşturmaya çalışırken (geçerli bir IP adresi kullanıyor olabilir veya olmayabilir) meşgule düşer ve sonraki kullanıcıların erişimleri bekleme durumunda kalır (Lee, Bu & Woo, 2009; Bicakci & Tavli, 2009; Houle, 2001).

b. Dağıtık Hizmet Reddi (DDoS)

DDoS saldırısı, güvenliği ihlal edilmiş çok sayıda sistemin tek bir hedefe saldırdığı ve böylece hedeflenen sistemin kullanıcılar için hizmet reddine neden olduğu saldırı türüdür. Hedef sisteme gelen mesajların akımı esasen sistemi kapanmaya zorlar, böylece kullanıcılar hizmet reddi sonuçlarına uğrarlar (Lee vd., 2009; Bicakci & Tavli, 2009; Houle, 2001).

c. Reflektörlü Dağıtılmış DoS (DRDoS)

Saldırganın daha etkili ve güvenli bir saldırı gerçekleştirmesine yardımcı olan bir reflektörden oluşur. Buda, tehlikenin artmasına, geri izlenme ve riskin azalmasına neden olur (Bicakci & Tavli, 2009; Houle, 2001).

d. SQL Enjeksiyon Saldırısı

SQL enjeksiyonu, bir uygulamanın veri tabanı katmanlarında oluşan bir güvenlik açığıdır. Bu eylem, SQL kodunun veri taban hizmetlerinde kullanılan etkileşimli web uygulamalarına geçirme eylemidir.

B. Fabrikasyon

Bu saldırıda kullanıcılar, uygun olmayan bazı erişim hizmetlerini kullanırlar. Bu saldırı, kimlik doğrulama mekanizmalarının yokluğunda daha etkilidir (Simmonds vd., 2004; Maiwald, 2001; Connolly, 2003).

a. Tekrar Saldırı

Tekrar saldırısı, geçerli bir veri iletiminin tekrarlandığı veya ertelendiği bir aktif saldırı biçimidir. Saldırgan, verileri yakalar ve yeniden yetkili kullanıma gönderir. Örneğin A kullanıcısı, bir tutarı C kullanıcısının banka hesabına aktarmak istiyor. Her iki A & C kullanıcılarının B bankası ile hesabı bir hesabı vardır. A kullanıcısı, B bankasına bir para transferi talep ederek elektronik mesaj gönderir. C kullanıcısı bu mesajı yakalayabilir ve B bankasına ikinci bir kopya gönderir, ancak B bankası bunun

yetkisiz bir mesaj olduğunu anlamayabilir. Böylece C kullanıcısı iki kez fon transferinden faydalanır. Bir tekrar saldırısı, zaman damgaları ve sürekli artan bir sıra numarasının değeri gibi önceki işlemde benzersiz bilgilerin dâhil edilmesini içeren güçlü dijital imzalar kullanılarak önlenabilir.

b. Maskeleye

Maskeli saldırı, bir sistemin kimliğini devraldığı bir saldırı türüdür. Bu yöntem saldırgan tarafından gizli bilgilere yasadışı bir şekilde erişmek için saldırganın yetkili kişi olarak davrandığı bir tekniktir (Maiwald, 2001).

C. Değişiklik

Modifikasyon, bütünlük prensibinin kaybına neden olan bir saldırı türüdür. Örneğin, bir kişi 100 liralık online işlem gerçekleştirirken bu bilgi saldırı sonucundan 1000 lira olarak yansıyabilir. Bu da bilgi bütünlüğünün etkilendiği anlamına gelmektedir (Connolly, 2003; Ouafi, Overbeck & Vaudenay, 2008; Simmonds vd., 2004).

a. Orta Saldırı Adamı

Bu teknik genel bir ağın kullanıcısı veya bir web sitesi arasında dolaşan bilgileri durdurmaya, okumaya ve değiştirmeye çalışan bir aktif saldırı türüdür. Saldırgan, kimlik çalma ve diğer dolandırıcılık işlemleri, yasa dışı bilgileri kullanarak gerçekleştirir (Ouafi vd., 2008).

2.9. İnternet Mimarisi ve Kırılgan Güvenlik Boyutları

İnternetteki güvenlik ihlallerinden kaynaklanan korku, kuruluşların korumalı özel ağları kullanmalarına zorlamaktadır. İnternet Mühendisliği Görev Gücü (IETF), İnternet Protokolü Paketi'nin çeşitli katmanlarında güvenlik mekanizmaları başlatmış, ve bu güvenlik mekanizmaları, ağ üzerinden aktarılan verileri mantıksal olarak korunmasını sağlar (Elawad & Dawo, 2016).

IP Güvenliği olarak da bilinen internet protokolünün güvenlik mimarisi, internet güvenliğinin standartlaştırılmış tarzıdır. Örneğin, IPsec, hem yeni nesil IP'yi (IPv6), hem de şimdiki sürümü (IPv4) destekler. IPv4 1980 yılında ARPANET'in NCP protokolünün yerini alacak şekilde tasarlanmıştır. Yirmi yıl sonrası IPv4 birçok sınırlık

gösterdiğinden IPv6 protokolü, IPv4'ten eksikleri tamamlamak amacıyla tasarlanmıştır. IPv6, IPv4 protokolünün bir üst kümesi değil, bunun yerine yeni bir tasarıma sahip bir yapıdır (Funmilola & Oluwafemi, 2015).

2.9.1. IPv4 Mimarisi

Bu protokolde oluşan sorunların tamamı güvenlik ile ilişkili değildir. IPv4 protokolü 32 bit genişliğinde bir adres mimarisine sahiptir. Bu da internete bağlanabilir maksimum bilgisayar sayısının iki milyar olduğunu göstermektedir. Buda yönlendirme tablosunun boyutu sürekli arttığı için yönlendirme konusunda ciddi anlamda sınırlılık yaratmaktadır. Örneğin, küresel yönlendirme tablolarının maksimum teorik boyutu 2,1 milyon giriş olduğu belirlenmiştir (Andress, 2005). Yönlendirme tablosundaki giriş sayısının azaltılmasına yönelik çeşitli yöntemler ele alınmış, ancak bu yöntemler kısa bir dönem için işe yarar, ve bu sorunu çözmek için köklü değişikliklerin yapılması gerekmektedir.

IPv4'ün küçük adres alanı kötü amaçlı kod dağıtımına yol açabilir. Ayrıca, IPv4 protokolünde, gömülü güvenlik sisteminin eksikliği birçok saldırıya yol açması mümkündür. IPv4'ün güvenliğini sağlayan birçok mekanizmalar mevcuttur, ancak bazen bunların kullanımı için herhangi bir gerekçe duyulmamaktadır. IPsec bu protokolü korumak için özel bir mekanizmadır. IPsec, kriptografi yöntemleri ile paket yüklerini korur, gizlilik, bütünlük ve kimlik doğrulama hizmeti sağlar (Funmilola & Oluwafemi, 2015).

2.9.2. IPv6 Mimarisi

IPv4'ün dikkat çekici büyümesi bazı temel sınırlılıklara yol açmıştır. IPv6 bu sorunları gidermek ve bazı gelişmiş hizmetler sağlamak için geliştirilmiştir. IP-NG olarak da adlandırılan IPv6, “yeni nesil” internet protokolüdür ve IPv4'ün devamıdır.

IPv6 hala geliştirme aşamasına olsa da, temel protokoller, kurallar ve biçimler yıllardır istikrarlı bir şekilde geniş destekler alıyor. Gerçek dünyadaki üretim dağıtımı (üretim ağı adreslerinin tahsisi ve atanması) birkaç yıldır devam etmekte ve IPv6 artık deneysel olarak kabul görmektedir.

IPv4 ile en çok tartışılan endişe, IPv4'ün gelecekteki ihtiyaçları karşılamak için yetersiz sayıda bireysel adres sağlamasıdır. Ağ Adres Çevirisi (NAT) gibi koruma, kurtarma ve diğer teknikler adres kullanım sayısına artırmış ve IPv4 adres havuzuna süreklilik sağlamıştır, ancak 32 bit adres adres alanı IPv4'ün gelecekte sınırlı kalmasının nedenidir. IPv6, adres alanlarındaki bit sayısını 32 bit'ten 128 bite genişleterek bu sınırı önemli ölçüde etkilemektedir.

IPv4 adres alanının parçalanmasının çoğu, IPv4 ağlarının yeniden numaralandırılmasındaki doğal zorluktan kaynaklanmaktadır. IPv6 bu sınırlamayı, dinamik ön numaralandırmaları, çoklu adreslere ve adres önekleri arasındaki geçişleri kolaylaştıran geçiş dönemlerine izin veren otomatik yapılandırma yöntemleri ve mekanizmalarıyla gidermektedir.

IPv6, IPv4'te bulunan güvenlik eksikliklerin çoğunu iyileştirir. Özellikle, IPv6, IPv4'e aktarılan IPSec (AH/ESP) gibi birçok gelişmiş güvenlik özelliği içerir. Taramaya direnç gibi çabalar yalnızca IPv6 adresleme teknikleri altına mümkündür. Örneğin, IPv6 adres alanının büyük boyutu tek başına kapsamlı bir güvenlik açığının taranması için önemli engeller oluşturur. Ayrıca adreslerin otomatik bir şekilde yapılandırılması gibi diğer IPv6 özellikleri, kötü niyetli bir saldırganın sistemlerin zayıflıklarının araştırmasını zorlaştırır. Bu faktörler rastgele veya sahte rastgele taramayı durduramaz, ancak belirli IPv6 ağlarının taranmasını engelleyebilir.

2.10. Türkiye’de Gerçekleşen Ağ Saldırıları Örneği

2.10.1. Ankara’da Tapu Bigilerinin Sızdırılması

Bu saldırı bir emlakçının bir Türk vatandaşına ilişkin tapu verilerini açıklaması sonucundan meydana gelmiş ve 1568000 kişiye ait tapu bilgilerinin çalınmasına neden olmuştur. Daha sonra Organize Suçlarla Mücadele ekibi bu bilgileri 500 TL karşılığında satan suçluları gözaltına almış ve verilerin içerden saldırıya maruz kaldığı düşünülerek benzer zafiyetlerin anlaşılması üzerinde odaklanılmıştır (Kızılkoyun, 2014).

2.10.2. TEİAŞ Kurumuna Saldırı

Bu saldırı 15 Kasım 2014’e TEİAŞ kurumuna ait bir müdür yardımcısının şifresi çalınarak sisteme giriş yapılarak gerçekleştirilmiştir. Daha sonra bu sistemin bir sosyal

mühendislik saldırısına maruz kaldığı ve donanım bakımından bazı zafiyetleri olduğu anlaşılmıştır (kozan, 2014).

2.10.3. HSBC Bankasına Saldırı

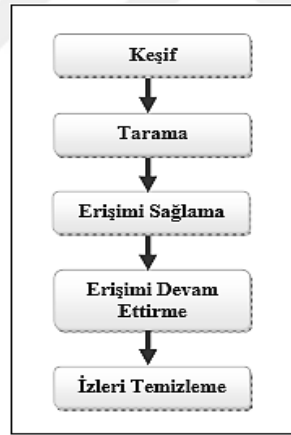
Bu saldırı sonucunda 13 Kasım 2014 tarihinde 2.7 milyon HSBC müşterilerinin kredi ve banka kartlarına ait bilgileri çalınmıştır. Bu saldırının üçüncü parti yazılımların kullanımından ortaya çıkmış olması anlaşılmıştır. Bu nedenle benzer saldırının bireysel kullanımlardan değil daha çok kurumsal politikalarındaki eksikliklerden kaynaklandığı söylenmektedir (HSBC, 2014).

2.11. Ağ Güvenliğine Yönelik Tehditler ve Önlemler

Ağ güvenliğine yönelik tehditler bazen bilişim kaynağını olumsuz yönden etkileyebilir. Bu tarz saldırılar ağ cihazlarına zarar vermek üzere yapılan saldırılardır. Yaşanabilecek benzer saldırılar ağ sistemlerinde bulunan güvenlik boşluklardan kaynaklanmaktadır. Bir ağ saldırısına uygulanan tekniklerin incelenmesi, saldırgan profilinin analiz edilmesi, benzer ağlarda güvenlik sağlamak için faydalı olduğu düşünülmektedir (Canbek & Sağıroğlu, 2007). Genel anlamda bir ağ ortamında gerçekleşen saldırı özeliği aşağıdadır (Allen, 2001):

- Engelleme (Interruption): Bilgi alışverişi esnasında ortaya çıkan duraklamalardır, Örneğin DoS, DDoS.
- Dinleme (Intercept): Gönderici ve alıcının kullandığı ağdaki iletişimin dinlenmesi ile istenen veriyi elde etmek mümkündür. Bu saldırı yetkisiz erişim olarak da bilinen pasif saldırı türüdür. Ağ koklaması (network sniffing) bu tür saldırılardandır.
- Değiştirme (Modification): Gönderici ve alıcı arasında gerçekleşen bilgi transferleri sürecinde verinin saldırgan tarafından modifikasyonu veya modifike edilmiş mesajın alıcıya gönderilmesidir. Bu tarz saldırı sonucunda veri bütünlüğü bozulabilir. Virüsler, solucanlar veya truva atları benzer saldırıların gerçekleşmesine yol açmaktadır.
- Üretim (Fabrication): Bu tür saldırı alıcıya sahte veri iletmekle gerçekleştirilir.

Bir saldırı analizi için genellikle Şekil 2.1'e (Can & Akbaş, 2014) gösterildiği üzere keşif, tarama, erişim sağlama, erişimi devam ettirme ve izleri temizleme gibi özelliklerin incelenmesi gerekmektedir (Burlu, 2013). Keşif sürecinde saldırı yapılacak hedefle ilgili araştırmalar başlatılır. Örneğin, e-posta'lerin ele geçirilmesi. Keşif süresince elde edilen bilgilerle tarama işlemi gerçekleştirir. Hangi ağ topolojisinin ele alınması, port taraması, saldırı açıklarının taranması, sunucuda kullanılan işletim sisteminin taranması bu aşamada gerçekleşmektedir. Erişimi sürecinde, saldırgan keşif sürecinde elde ettiği tarama bilgileri ile sisteme sızma işlemini gerçekleştirir. Örneğin, Phyton, Perl programlama dilleriyle yazılan zararlı kodlar sistemdeki boşluklar sayesinde basit kullanıcılara yönetici yetkisini kazandırabilmektedir. Erişimi devam sürecinde, sistem açıklığına süreklilik sağlanması gerekir. Çünkü saldırgan mevcut açığı kullanarak diğer ağ sistemlerine sızabilir. Bu teknikte saldırgan truva atları, kök kullanıcı takımlar, arka kapılar gibi zararlı yazılımları kullanarak sızma işlemini gerçekleştirebilir. İz temizleme aşamasında yapılan saldırıya ait izlerin ortadan kaldırılması üzerinde odaklanır.



Şekil 2.1. Bir saldırının analiz şeması.

Aşağıda birkaç güvenlik tehditleri ve alınabilir önlemler ayrıntılı bir şekilde ele alınmaktadır.

2.11.1. Kötücül Yazılımlar Saldırısı

Kötü amaçla geliştirilmiş yazılımlar sistemdeki bilgisayar donanımı ve dosyaları hedefleyen, sistem bilgilerini izinsizce başkalarına veya yabancı birine ileten yazılımlar türüdür. Var olan en yaygın kötü amaçlı yazılımlar virüsler, solucanlar, Truva atları,

arka kapılar, spam, rootkit, exploit, keylogger, görüntü yakalama sistemleri, tarayıcı soyma ve casus yazılımlarıdır (Çifci, 2012).

2.11.2. Hizmet Dışı Bırakma Saldırıları (DoS/DDoS, Botnet)

Bu tür saldırıda “erişilebilirlik” kriterleri hedef alınmaktadır. Benzer saldırı sonucunda yalnız bilgi, para veya zaman kaybı gibi kavramlar önem arz etmemekte, bazen bir hizmet için itibar kaybı da önemlidir (Zargar, Joshi & Tipper, 2013). Bu saldırıda hedef sisteme, dönüş yapamayacağı kadar istek yollayarak bant genişliği ve CPU hızının azaltılması ve tüketilmesi amaçlanır. DDoS saldırıları ise DoS saldırılarından farklı olarak botnet’leri kullanmaktadır. Botnet, bir yöneticinin tamamen kontrolü altına olan köle sistemlerdir. Köle sistemleri kullanarak farklı konumlardan tek bir sisteme istekler gönderilir. Kullanılan botnet’in büyüklüğü en iyi bağlantılı web sayfaları hizmetlerini de engelleyebilir (Canbek & Sağıroğlu, 2006). Örneğin, Apple Daily ve PopVote sitelerine bu saldırı 500 gbps trafik ve saniyede 250 milyon DNS sorgusu ile gerçekleştirilmiştir. Benzer saldırı ABD ve Avrupa’yı hedef almak üzere 400 Gbps’lik bir zirve gerçekleştirmiştir (Gilbert, 2014). Kurumlardaki güvenlik politikaları ve risk analizleri benzer saldırıların önlenmesini mümkün kılar. Örneğin, zaman ve kaynaklarının tükenmesine neden olan etkenlerin belirlenmesi ve gerçek DoS saldırıların test edilmesi.

2.11.3. SQL Enjekte Saldırıları

SQL enjeksiyonu bir web saldırganının veri tabanına erişmesini sağlar (Halfond, Viegas & Orso, 2006). Saldırı sonucundan bilgilerin ifşa edilmesi, satılması yada üzerinde dolandırıcılık yapılması mümkündür.

2.11.4. Sazan Avlama (Phishing) Saldırıları

Bu tür saldırı banka, telekomünikasyon şirketleri gibi resmi adreslerden gelen e-posta aracılığıyla kurumsal bilgileri hedef alan kimlik hırsızlığı yapan saldırılardır. Bu tip saldırıda kullanıcılar istemeden kişisel bilgileri kötü amaçlı saldırganlara paylaşmaktadır (Altundal, 2012). Benzer saldırılara maruz kalmamak için bilgi güvenliği farkındalığı seviyesinin artırılması gerekmektedir.

2.11.5. XSS Saldırıları

Bu tür saldırı veri girişleri esnasında javascript aracılığıyla yapılan saldırı türüdür (Demirez, 2011). XSS saldırısı hakerler tarafından kullanılan saldırı türüdür. XSS saldırısında kullanıcıdan kimlik doğrulama işlemi istenebilir (Klein, 2002). Çizelge 2.1’de web saldırıları örneği gösterilmiştir (Application Vulnerability Trends Report [AVTR], 2014).

Çizelge 2.1. 2013 yılı web uygulama saldırıları.

Saldırı Türü	Yüzde
Xss	25
Bilgi Sızdırma	23
Kimlik Doğrulama	15
Oturum Yönetimi	13
SQL Enjekte	7
CSRF	6
Diğer	11

2.11.6. Sosyal Mühendislik Saldırıları

Son günlerde daha çok kullanıcılara yönelik koruma sağlayan uygulamalar geliştirilmektedir. Bu hizmetlerin bir kısmı kullanıcılara bilgi güvenliği ile ilgili eğitim verilmesi ile gerçekleştirilmektedir. Bu sırada saldırganlar etkili yöntemlerden biri olan sosyal mühendislik kavramından yararlanmaktadırlar. Sosyal mühendislik saldırısında, saldırgan insan davranışlarını değerlendirir, kullanıcıların kişisel verileri kendiliğinden paylaşımlarını sağlar (Bircan, 2014; Mitnick & Simon, 2001). Benzer saldırıların gerçekleşmesini engellemek için güvenlik politikalarının düzenlenmesi, müdahale yöntemlerinin ortaya konulması gerekmektedir.

2.12. Bazı Simetrik Şifreleme Algoritmaları

2.12.1. Veri Şifreleme Standardı (DES)

DES, 64 bitlik bloklarda çalışan 56 bitlik bir anahtar kullanan standarttır. Davis tarafından tanıtıldığı gibi DES algoritması, sabit uzunluklu açık metin bit dizisini karmaşık işlemle aynı uzunlukta şifreli metin dizisine dönüştürür (Davis,1978). 3DES (Üçlü DES), DES’in geliştirilmiş halidir. 192 bit anahtar boyutuyla 64 bit blok boyutu

kullanır. Bu standart şifreleme ve ortalama güvenlik seviyesini artırmak için 3 kez uygulanır (Barker & Barker, 2012).

2.12.2. Gelişmiş Şifreleme Standardı (AES)

AES, ticari uygulamalar için DES'in yerini almayı amaçlayan bir blok şifreleme tekniğidir. 128 bit blok boyutu ve 128, 192 veya 256 bit anahtar boyutu kullanır. Şifrenin dâhili tur sayısı anahtar uzunluğunun bir fonksiyonudur. Örneğin, 28 bit anahtar için tur sayısı 10'dur. Selefı DES'in aksine, AES bir Feistel ağı kullanmaz. Feistel ağıları her yineleme başına tüm bloğu şifrelemezler, örneğin DES'te $64/2 = 32$ bit bir turda şifrelenir. Öte yandan AES, 128 bitin tümünü tek bir yinelemede şifreler (Rijmen & Daemen, 2001; Pub, 2001).

2.12.3. Blowfish

Schneier tarafından icat edilen 64 bitlik bir simetrik şifreleme tekniğidir; büyük veri önbelleklerine sahip 32-bit işlemciler için optimize edildiğinde, bir Pentium/PowerPCclass makinesinde DES'ten daha hızlıdır. Anahtar uzunlukları 32 ile 448 bit arasında değişebilir. Serbestçe temin edilebilen, DES veya IDEA'nın yerine geçmesi amaçlanan blowfish, çok sayıda ürünlerde kullanılmaktadır. 16 turlu bir Feistel şifresidir ve büyük anahtara bağımlı S-kutuları kullanır. Bu kutular sekiz bit girişi alırken 32 bit çıkış üretirler (Schneier, 1993).

2.12.4. Twofish

128, 192 veya 256-bit anahtar boyutu kullanan 128-bit blok şifreleme tekniğidir. Son derece güvenli ve son derece esnek bir şekilde tasarlanmıştır, ve büyük mikroişlemciler, 8 bitlik akıllı kartlı mikroişlemciler ve özel donanımlar için çok uygundur. Bu algoritma Bruce Schneier liderliğindeki bir ekip tarafından tasarlanmıştır. Twofish'in ayırt edici özellikleri, önceden hesaplanmış anahtara bağımlı S-kutularının kullanımı ve nispeten karmaşık bir anahtar zamanlamasıdır. N-bit anahtarının yarısı gerçek şifreleme anahtarı olarak kullanılır ve N-bit anahtarının diğer yarısı şifreleme algoritmasını (anahtara bağı Alanlar) değiştirmek için kullanılır. Twofish, DES gibi Feistel bir yapıya sahiptir (Schneier, 2005).

2.13. Bazı Asimetrik Şifreleme Algoritmaları

Açık anahtarlı şifreleme, bir ortak anahtar ve bir özel anahtar olmak üzere farklı anahtarları kullanılarak şifreleme ve şifre çözme işlemini gerçekleştiren bir açık asimetrik şifreleme yöntemidir. Bu anahtarlar matematiksel olarak ilişkilidir, ancak bir anahtarın bilgisi birinin diğer anahtarı kolayca bulmasını izin vermez. A gönderici M düz mesajı şifrelemek için B alıcının genel anahtarını kullanır ve şifrelenmiş metni C alıcıya gönderir. C Alıcı şifreli metni çözmek ve M düz metine ulaşmak için elindeki özel anahtarı kullanır. Anahtar çifti gerekli olduğundan, bu yaklaşıma asimetrik şifreleme algoritması denir. Asimetrik şifreleme gizlilik, kimlik doğrulama veya her ikisi için tasarlanabilmektedir. Günümüzde yaygın olarak kullanılan açık anahtar şifreleme algoritmaları aşağıdadır (Kumar, 2015, s. 5-7):

2.13.1. RSA Algoritması

RSA algoritması ilk ve hala en yaygın olarak kullanılan açık şifreleme yöntemidir. Bu yöntem üç MIT matematikçisi Ronald Rivest, Adi Shamir ve Leonard Adleman tarafından ortaya konmuştur (Rivest, Shamir & Adleman, 1978). RSA bugün yüzlerce yazılım ürününde kullanılmaktadır ve anahtar değişimi, dijital imzalar veya küçük veri bloklarını şifrelemek için kullanılmaktadır. RSA, değişken boyutlu bir şifreleme bloğu ve değişken boyutlu bir anahtar kullanır. Anahtar çifti çok özel bir şekilde türetilir, yani özel kurallara göre seçilen iki asal sayının çarpımı neredeyse iki kat uzunluğunda olan bir n sayısını üretir. RSA anahtar üretimi, şifreleme ve şifre çözme olarak üç aşama ile gerçekleştirilir (bkz. Bölüm 3).

2.13.2. Diffie-Hellman Anahtar Değişimi

Basit bir açık anahtar algoritması Diffie-Hellman anahtar değişimidir. Bu protokol, iki kullanıcının ayrı logaritmalara dayalı bir ortak anahtar kullanarak gizli bir anahtar oluşturmasını sağlar. Bu yöntem yalnızca iki katılımcının orijinalliği korunabildiği zaman güvenlidir. DH, yalnızca gizli anahtar değişimi için kullanılır, kimlik doğrulama veya dijital imzalar için kullanılmaz (Diffie & Hellman, 1976).

2.13.3. Eliptik Eğri Şifrelemesi

Diffie-Hellman anahtar değişiminin analogudur. ECC, eliptik eğrilere dayanan bir ortak anahtar şifreleme algoritmasıdır. Eliptik eğri aritmetiği anahtar değişimi, şifreleme ve dijital imza dahil olmak üzere çeşitli eliptik eğri şifreleme (ECC) modellerini geliştirmek için kullanılabilir. Eliptik eğri aritmetiği, sonlu bir alan üzerinde tanımlanan bir eliptik eğri denklemini kullanır. Denklemdaki katsayılar ve değişkenler sonlu bir alanın öğeleridir (Koblitz, 1987; Miller, 1985).

2.13.4. Dijital İmza Standardı

Dijital imza standardı (DSS), güvenlik sağlama algoritmasını (SHA) kullanan bir NIST standardıdır (National Institute of Standards & Technology [NIST], 1993). Dijital imza, mesaj oluşturucusunun imza görevini gören kodun eklemesini sağlayan bir kimlik doğrulama mekanizmasıdır. Genellikle imza, mesajın karmasını alıp mesajı, oluşturucunun özel anahtarıyla şifreler.

2.14. Karma İşlevleri

İleti özetleri ve yerleşik şifreleme olarak da bilinen karma işlevler aslında anahtar kullanmayan algoritmalarıdır. Bir hash H fonksiyonu, M uzunluklu bir veri bloğunu giriş olarak kabul eder ve sabit boyutlu bir hash değeri $h=H(M)$ üretir. Genel olarak, bir hash fonksiyonunun temel amacı veri bütünlüğüdür (Lamberger, Mendel, Rechberger, Rijmen & Schl  ffer, 2009). Karma algoritması, iki giriş (zincirleme değişkeni adı verilen bir bit giriş ve bir bit blok) alan ve bir bit çıktı üreten bir sıkıştırma fonksiyonunun tekrarlı kullanılmasıdır. Zincirleme değişkeninin son değeri karma değerdir.

BÖLÜM 3

RSA YÖNTEMİNİN GÜVENLİK ZAFİYETİ VE GÜÇLENDİRİLMİŞ RSA YÖNTEMLERİNİN İNCELENMESİ

3.1. RSA Algoritmasının Temel Güvenlik Zafiyeti

Bazın insanlar gönderdikleri bilgileri başkaları tarafından çalınmasını, okunmasını veya değiştirilmesini istemeyebilirler; ancak tam o sırada davetsiz birinin bu bilgilere ulaşip düşündüğü hedefi uygulayabilir olduğu da mümkündür. Davetsiz olan bu bireyler başka bir deyişle saldırganlar olarak da bilinmektedirler. Saldırganlar klasik bir sistemle üzerinde yıllarca çalışılarak edinilebilen şifrelenmiş mesajları daha hızlı bir sistem veya algoritma kullanarak kısa bir süre içerisinde elde edebilirler.

RSA algoritmasında, özel anahtarın sadece genel anahtar üzerinden türetilabilir olması hem avantaj hemde bir nevi güvenlik problemini doğmasına neden olmaktadır. RSA algoritmasında verilen bir mesajın üssü alınarak faktörleri bilinmeyen bir bileşik N sayısına modüle edilir ($C=M^e \bmod N$), ve gönderici modüle edilmiş bu mesajı (C =şifrelenmiş mesaj) rahatlıkla genel kanal üzerinden alıcıya gönderebilir. Ancak alıcı C mesajını oluşturan e ve N parçaları olmadan bu mesajın içeriğine ulaşamaz. Bu durum göndericinin bir C gizli mesajını iletirken (e,N) açık anahtarlarının da iletmesini gerektirir. Ancak N sayısının bazın daha küçük bir sayı olması N sayısını oluşturan p ve q faktörlerini rahatlıkla bulunmasını sağlar, ve böylece RSA algoritmasının kırılganlığına yol açması mümkündür.

Diğer yandan, RSA algoritmasının küçük sayılarla işlem yapıldığı zaman kırılabilirlik sağladığı, bir büyük N tam sayısının p ve q faktörlerine ayıramaz olduğunu kanıtlayamaz. Bu bağlamda son yüzyılda matematikçiler ve bilim insanları tarafından asal sayılarının çarpanlara ayırma konusunda birçok çaba sarf edilmiştir. Ancak geliştirilen çoğu faktörleme algoritmaları klasik bilgisayarlar üzerinde çalıştığı için faktörleme işlemini süper polinom bir sürede gerçekleştirebilmektedir. Bu nedenle 1994 yılında Peter Shor isminde bir matematikçi büyük asal sayılarının bir kuantum bilgisayarı kullanıldığı zaman etkili ve hızlı bir şekilde polinomel bir sürede faktörlere ayrılabilir olduğunu ortaya koymuştur. Bu bölümde, RSA algoritmasının kırılabilir yönünü ve bazı araştırmacılar tarafından önerilen güçlendirilmiş RSA yöntemlerini araştırmadan önce faktörleme işlemini bir süper polinom sürede gerçekleştiren GNFS yönteminin kısaca açıklanması ve faktörleme işlemini bir polinomel sürede gerçekleştiren kuantum hesaplama mekaniğinin biraz daha detaylı incelenmesi hedeflenmektedir.

3.1.1. Klasik Faktörleme (GNFS)

1970 yılında 20 basamaklı sayıları çarpanlara ayırmak neredeyse imkansızdı, 1980 yılında, Brillhart-Morrison'un faktörleme algoritması 50 basamaklı sayıları çarpanlara ayırabildi, 1990 yılında, quadratic sieve faktörleme algoritması 116 basamaklı ve 1994 yılında da 129 basamaklı sayıları başarıyla çarpanlara ayırabildiğini ortaya koydu (Pomerance, 1996). Ancak John Pollard 1996 yılında quadratic sieve faktörleme algoritmasının aldığı %15 zaman diliminde 130 basamaklı bir RSA sayısını faktörlere ayırarak quadratic sieve faktörleme algoritmasının yerini almayı başarmıştır.

2014 yılında Shah Muhammad, Firoz, Biprodip ve Syed Tauhid yazarlığında yayınlanan bir makalede GNFS ile Shor algoritması arasındaki faktörleme performansı 100 basamaklı bir sayı örnek alınarak değerlendirilmiştir. Değerlendirme sonucundan GNFS algoritması ister bağımsız ister dağıtılmış çoklu klasik bilgisayarlarla faktörleme işlemini bir 1000 basamaklı sayı üzerinde uygulandığında çarpanlara ayıramaz olduğu söylenmektedir, oysaki kuantum bilgisayarlarının bu işlemi saniyeler içinde yapabildiği ve böylece kriptoloji yapılarının güvenliği ile ilgili endişe yaratacağı vurgulanmıştır (Hamdi, Zuhori, Mahmud & Pal, 2014).

BNFS algoritması için bir n sayısının süper polinomel faktörleme zamanı $O(e^{[(\log n)^{1/3} (\log(\log n))^{2/3}]})$ dır (Pomerance, 1996). Oysaki Shor algoritmasının bir kuantum bilgisayar üzerinde uygulandığında n sayısı için polinomel faktörleme zamanı $O((\log n)^2 (\log \log n) (\log \log \log n))$ olduğu söylenebilmektedir (Shor, 1994).

3.1.2. Kuantum Hesaplaması ve Klasik Hesaplaması

Kuantum bilgisayarları kavramı ilk defa 1982 yılında Richard Feynman tarafından ortaya atıldı. Kuantum mekaniğinin, mikroskobik fiziksel olaylarla ilgili olduğu belirtilmiştir. Bir kuantum bilgisayarda bitler yerine kubitleri kullanılır (Nielsen & Chuang, 2011). Kubitler yalnızca 0 ve 1 değil aynı zamanda her ikisini de paralel bir şekilde temsil edebilen süperpozisyon parçacıklarıdır. Süperpozisyondaki bir kubit üzerinde bir işlem yapıldığında hem 0 hemde 1 bitleri etkilenebilir. Kuantum hesaplamada bir başka olayda da kuantum dolanıklığıdır. İki kubit dolaşık olduğunda kuantum durumları da dört farklı duruma sahip tek bir nesne olarak çalışırlar. İki kubit durumundan biri değişirse, dolaşık kubit de değişir (Jozsa, 1997). Bir kuantum bilgisayar n -kübiti, 2^n şeklinde işleyebilmektedir.

Evrensel ve evrensel olmayan iki tür kuantum bilgisayarları bulunmaktadır. Evrensel kuantum bilgisayarları farklı işlemleri yapmak için geliştirilmiş, ancak Evrensel olmayan kuantum bilgisayarları belirli bir amaç için kullanılmaktadır (örneğin, makine öğrenimi algoritmalarının optimizasyonu). Araştırmacılar kuantum bilgisayarlarının üstünlüğünü kuantum paralelliğine bağlı olduğunu belirtmekteler.

3.1.3. Kuantum Algoritmalarına Karşın Savunmasız Kripto Sistemleri

Kriptoloji kavramı günümüzdeki elektronik iletişim sistemleri üzerinde önemli bir rol oynamaktadır. Örneğin, e-postalar, şifreler, finansal işlemleri ve hatta oylama sistemleri bile gizlilik ve bütünlük gibi güvenlik yapılarına ihtiyaç duymaktalar (Campagna & Xing, 2015). Şifreleme, yalnızca anahtarları birbirine paylaşmış tarafların şifrelenmiş mesajı açabilir olmasını sağlar. Kuantum bilgisayarları klasik bilgisayarların yapamadığı yapabildiği için gizli anahtarları hızlı bir şekilde tarayabilir veya bir dinleyicinin gönderici ve alıcı arasındaki iletişim kanalının kesmesine izin verir (Buchanan & Woodward, 2016). NIST'tin sunduğu Çizelge 3.1'e, kuantum

hesaplamanın mevcut şifreleme şemaları üzerindeki etkisi gösterilmiştir (Chen Vd., 2016).

Çizelge 3.1. Kuantum bilgisayarlarının şifreleme yöntemleri üzerindeki etkisi.

Kriptografik Algoritmaları	Türü	Amaç	Quantum Bilgisayar Etkisi
AES-256	Simetrik anahtarlı	Şifreleme	Güvenli
SHA-256, SHA-3	--	Hash fonksiyonları	Güvenli
RSA	Açık anahtarlı	İmzalar, anahtar üretimi	Pek güvenli değil
ECDSA,ECDH	Açık anahtarlı	İmzalar, anahtar değişimi	Pek güvenli değil
DSA	Açık anahtarlı	İmzalar, anahtar değişimi	Pek güvenli değil

A. Asimetrik Kripto Sistemlerinde SHOR Algoritması

1994'te matematikçi Peter Shor, "Kuantum Hesaplama için Algoritmalar: Ayrık Logaritmalar ve Çarpanlara Ayırma" (Shor, 1994) adlı makalesinde, büyük tam sayıları bir kuantum bilgisayarı ile çarpanlara ayırabilir olduğunu kanıtlamıştır. Bu algoritmanın büyük asal sayıları çarpanlara nasıl ayırdığı aşağıdaki örnekte gösterilmiştir. Örneğin, 15 sayısının asal çarpanlarını bulalım. Bunu yapmak için 4-kübitlik bir ikili kaydına ihtiyacımız vardır. 4 kübitlik bir kayıt, geleneksel bir bilgisayarın normal 4 biti olarak düşünebiliriz. Burada 15 sayısı ikili bitler olarak 1111 olsun, algoritma aşağıdaki gibi çalışmaktadır:

- ✚ Çarpanlara ayırmak istediğimiz asal sayı $N = 15$ olsun.
- ✚ $1 < x < N$ rastgele bir sayı olsun, örneğin $x=2$.
- ✚ N 'yi oluşturan bir olası P faktörünü bulmak için $x^r \bmod N$ ilişkisinden x ile N arasındaki r periyodunun bulunması gerekmektedir. Burada $2^{r=0} \bmod_{15}=1$, $2^{r=1} \bmod_{15}=2$, $2^{r=2} \bmod_{15}=4$, $2^{r=3} \bmod_{15}=8$ işleminden kalan 1,2,4,8 sayıları, sonradan gelen periyod aramalarında da 1,2,4,8 olarak tekrarlanmaktadır (örneği; $2^{r=4} \bmod_{15}=1$, $2^{r=5} \bmod_{15}=2$, $2^{r=6} \bmod_{15}=4$, $2^{r=7} \bmod_{15}=8$). Buda x 'ten N 'ye ulaşmak için tekrarlanacak toplam periyod sayısının 4 olduğunu göstermektedir ($2^{r=4} \bmod_{15}=1$ ilişkisinden $r=4$ 'tür). Burada $N=15$ için bir olası asal sayı faktörü $P=x^{r/2}-1=3$ 'tür.

Örnekte açıklandığı üzere, Shor algoritması büyük asal sayıların çarpanlarını kolaylıkla bulabilmektedir. Buda bazı açık anahtarlı şifreleme yöntemlerinin N gibi anahtar bütününe açık konumdan paylaşımları durumunda saldırganın $f(x_1)$ ve $f(x_2)$ çarpanlarını bularak gizli anahtara ulaşmasını sağlar. Aşağıda bir kuantum hesaplama algoritmasının RSA gibi bir asimetrik şifreleme yöntemini nasıl kırabileceği SHOR algoritması ile alınarak değerlendirilmektedir.

3.1.4. RSA Asimetrik Şifreleme Yönteminin SHOR Algoritması İle Kırılabilirliği

RSA kriptolojisinde güvenli bir şifreleme ve deşifreleme eylemini gerçekleştirmek için kullanılan yöntemlerden biri büyük asal sayılarla işlem yapmaktır. Klasik bir bilgisayarla p ve q gibi iki asal sayıyı birbirine çarparak daha büyük bir N asal sayısını elde etmek mümkündür. Ancak bir klasik bilgisayarla büyük N sayısını oluşturan p ve q asal faktörlerini bulmak gerçekten zordur. Aynı zamanda RSA algoritması şifrelenmiş bir mesajı deşifre etmek için bu faktörleri (p ve q) ve bu faktörlerden üretilen diğer parametreleri doğrudan (örneğin; alıcıya gönderilmiş e ve N açık anahtarı) veya dolaylı bir şekilde (örneğin; $\phi(N)$ 'ilişkisinden e parametresinin üretilmesi) nevi anahtarlar olarak kullanılmaktadır. Buda RSA algoritmasını kırabilmek için kesin bir rota olduğunu ve bir şekilde p ve q faktörlerini elde ederek deşifreleme işleminin gerçekleştirilebilir olduğunu göstermektedir. Örneğin; N=35 asal sayısının p ve q faktörlerini bulalım. Normalde 35 sayısının asal faktörleri kolayca $N=p=5*q=7$ şeklindedir. Bu işlem ilk bakışta kolay gelse de klasik bir bilgisayar bu işlemi yapabilmek için 35'den küçük tüm olası sayıları değerlendirmek zorundadır. Böyle bir hesaplama daha büyük N sayılara gelince daha da zorlaşacak ve işlem başına gereğinden fazla zaman kaybına neden olacaktır. Bu bağlamda büyük N sayılarını kolaylıkla p ve q faktörlerine ayırabilmek için bazı stratejik metotlar geliştirilmiş bulunmaktadır. Bunlardan biri Euler stratejisidir. Euler çalışmalarında, asal sayılar, bağdaşık asal sayılar ve modüler aritmetiği gibi birçok temel kavramlar üzerinde odaklanmıştır. Burada RSA algoritmasında N sayısının faktörlerini bulmak için daha çok modüler aritmetiğinden destek alınır. Modüler aritmetiği bir N sayısını oluşturan periyodu bulmak için de kullanılabilir ($x^{r=0,1,2,\dots,n} \bmod N=1$), ancak N sayısı daha büyük sayılar olduğunda klasik bilgisayarla p ve q çarpanlarını çok az bir sürede bulabilmek neredeyse mümkün görülmemekte, ve böylece RSA algoritmasına güvenlilik

kazandırmaktadır. Bu bölümde daha büyük asal sayıları çok az bir zaman dilimi içerisinde daha etkili bir şekilde faktörlere ayırabilen SHOR algoritmasına değinerek, RSA algoritmasının kırılma yönünü değerlendirmekteyiz.

RSA algoritmasının anahtar üretme, şifreleme ve deşifreleme süreci:

- ❖ p ve q için iki asal sayı seç
- ❖ $N = p * q$
- ❖ $\phi(N) = (p-1) * (q-1)$
- ❖ $1 < e < \phi(N)$ olmak üzere $\phi(N)$ ile aralarında asal olan bir e sayısı seç
- ❖ $e*d \bmod \phi(N) \equiv 1$
- ❖ Genel anahtar (e, N)
- ❖ Özel anahtar (d, N)
- ❖ Şifreli metin $C \equiv M^e \bmod N$
- ❖ Deşifreli metin $M \equiv C^d \bmod N$

Burada (e, N, C) parametreleri açık kanal üzerinden alıcıya gönderilirken N sayısı saldırganın eline geçtiğinde saldırgan bir seri kuantum hesaplamalarını Shor algoritması sayesinde gerçekleştirerek N üzerinden p ve q faktörlerini aşağıdaki gibi elde edebilir.

Bir asal N sayısının asal p ve q faktörlerini bulmak için kullanılan SHOR yöntemi 5 aşamada gerçekleştirilir (Lomonaco, 2000; Vazirani, 2012). Bunlardan yalnızca ikinci aşamada bir kuantum bilgisayarının kullanılması gerekmekte, diğer aşamalar klasik bir bilgisayarla da yapılabilmektedir.

Adım 1: $x < N$ olmak üzere rastgele pozitif bir tam sayı seçilir. İlk önce x ve N sayılarının aralarında eş asal (co-prime) olup olmadığını bakmak için Öklid algoritması veya herhangi bir klasik hesaplama tekniği kullanılabilir, eş asallık koşulu $\gcd(x, N) = 1$ dir. Eğer $\gcd(x, N) \neq 1$ ise N 'nin bir faktörünü bulduk demektir ve işlem durdurulur. Eğer $\gcd(x, N) = 1$ ise adım 2'ye geçilir.

Adım 2: Fonksiyonun bilinmeyen r periyodunu bulmak için bir kuantum bilgisayar kullanılır, $f_x(r) = x^{r=0,1,2,\dots} \bmod N = 1$.

Adım 3: Eğer r tekil bir tam sayı ise, Adım 1'e geçilir ve x için başka bir rastgele sayı seçilir. Eğer r çift sayı ise Adım 4'de geçilir.

Adım 4: $(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1 = 0 \bmod N$

Eğer $(x^{r/2} + 1) = 0 \pmod{N}$ ise Adım 1'e geçilir. Eğer $(x^{r/2} + 1) \neq 0$ ise Adım 5'e geçilir.

Adım 5: p ve q faktörünü bulmak için Öklid algoritması kullanarak $p = \gcd(x^{r/2} - 1, N)$ hesaplanır; $q = N/p$.

Yukarıda Adım 2'de x ile N arasındaki r periyodunu bulmak için bir kuantum bilgisayarlarının kullanılması önerilmektedir. Buda SHOR algoritmasının bir N sayısının faktörlerini bulabilmek için aslında çarpanlara ayırma işleminden daha ziyade sıra bulma (order finding) işlemine odaklı olduğunu göstermektedir. İşte bu nedenle bir klasik bilgisayarının özellikle daha büyük sayılara gelince benzer sıraları bulabilmesi zordur, oysaki kuantum bilgisayarlar bu işlemi tek deneme ile yapabilmektedir.

RSA algoritmasını kırabilmek için bazen N sayısını faktörlere ayırma gibi işlemler gerekemeyebilir. Örneğin; 2009 yılında Sattar J yazarlığında yayınlanan "RSA Şemasını Kırarak İçin Etkin Yöntem" makalede, N sayısını p ve q gibi asal faktörlere ayırmaksızın RSA algoritmasının farklı bir yöntemle de kırılabilir olduğu gösterilmiştir. Bu yöntem aşağıdaki adımlarla çalışmaktadır (Aboud, 2009).

Adım 1: Bir açık anahtar içeriği (e,n) olsun.

Adım 2: n sayısı ikili bitlere değiştirilir.

Adım 3: b ikili bitler miktarı olmak üzere $d = b/4$ olsun.

Adım 4: $ed \equiv 1 + k(n - s + 1) \pmod{2^b}$ bulunur.

Adım 5: k, 1'den e'ye $p^2 - s * p + n \equiv 0 \pmod{2^b}$ koşulu doğru olduğu sürece tekrarlanır.

a. $ed \equiv 1 + k(n - s + 1) \pmod{2^d}$ hesaplanır.

b. $p^2 - s * p + n \equiv 0 \pmod{2^d}$ hesaplanır.

Adım 6: $p_0 \equiv p \pmod{2^d}$ hesaplanır.

Adım 7: $q_0 * p_0 \equiv n \pmod{2^d}$ Baghdad ters alma metodu ile hesaplanır.

Adım 8: $\theta(n)$ aşağıdaki süreçle hesaplanır:

- $n \equiv (2^d * x + p_0) * (2^d * y + q_0)$ hesaplanır.
- $p = (2^d * x + p_0)$
- $q = (2^d * x + q_0)$
- $\theta(n) = (p-1) * (q-1)$

Adım 9: $d = e^{-1} \pmod{\phi(n)}$ hesaplanır.

Bu yöntem, daha az bir zaman karmaşıklığına sahip verimli bir performans sergilediği için günümüzdeki saldırı algoritmalarından daha hızlı olduğu düşünülmektedir (Aboud, 2009).

3.1.5. Güçlendirilmiş RSA Yöntemleri

Yukarıdaki bilgilerden istinaden RSA algoritmasının güvenliği genel olarak açık bir ağ konumunda paylaşılan e , N ve C parametrelerine bağlı olduğu görülmektedir. Ek olarak aynı anahtarla farklı girdileri farklı zaman ölçüsünde deşifre edilmesi de RSA algoritmasının kırılabilirliğine sebep olabilir. Örneğin bir saldırgan, kurban farklı kripto metinleri işlerken geçen süreyi kaydedebilir ve anahtara ilişkin bilgi elde etmesi mümkündür. Örneğin, kurban $M = c^d \pmod{n}$ denkleminde aynı d 'yi kullanarak farklı c 'yi hesaplarken saldırgan bu süreyi zamanlandırabilir, ve d 'yi hesaplayarak m 'yi elde etmesi mümkündür. Bu bölümde RSA'nın güvenlik açısından nasıl güçlendirilebileceğine yönelik yapılan araştırmalar üzerinde odaklanmıştır.

RSA algoritmasının güvenliğini güçlendirmek amacıyla araştırmacılar tarafından birçok yöntemler geliştirilmiş/geliştirilmektedir. Bu tekniklerden en önemlisi de RSA'nın asimetrik yapısını değiştirmeden (alıcı anahtarının güvenli bir şekilde kanal yolundan temin edilmesi) şifreleme ve deşifreleme işlemini gerçekleştiren yöntemlerdir. Bu bağlamda 2013 yılında Rohit , Kaushal, Saurabh ve Vincent yazarlığında yayınlanan makalede, RSA yapısı üzerinde 2 katman daha eklenerek açık anahtar oluşturan parametreden (e, N) N sayısının çıkartılması üzerinde odaklanmıştır. Böylece matematiksel olarak birinin N sayısının çarpanlarına ayırarak p ve q faktörlerine ulaşması mümkün görülmemekte, ve RSA algoritması bir derece daha güvenilir hale gelebilmektedir (Ambedkar, Gupta, Gautam & Bedi, 2011; Sun, Wu, Ting & Hinek, 2007). Algoritmanın çalışma süreci üç aşamadan oluşmaktadır:

a. Anahtar üretimi

- A ve B iki farklı rastgele asal sayı olsun.
- $N = A * B$ hesaplanır, N bit cinsinden anahtar uzunluğudur.

- $\Phi(N) = (A - 1) * (B - 1)$ hesaplanır, burada ϕ Euler'in sağlam fonksiyonudur.
- k_1 aşağıdaki koşullar altında hesaplanır:
 - $\sqrt{N} < k_1 < \phi(N)$
 - $\text{GCD}(k_1, \phi(N)) = 1$, burada k_1 ve $\phi(N)$ aralarında eş asal olmalıdır.
 - k_1 kısa bir bit uzunluğuna ve küçük bir Hamming ağırlığına sahip olmalıdır.
- N 'yi değiştirmek için X hesaplanır:

$A > B$ olduğunda,

 - $N - A < X < N$
 - $\text{GCD}(X, N) = 1$

$A < B$ olduğunda,

 - $N - B < X < N$
 - $\text{GCD}(X, N) = 1$
- $k_2, k_1 * k_2 \text{ Mod}(X) = 1$ şeklinde hesaplanır.

- Şimdi açık anahtar (k_1, X) ve özel anahtar ise (k_2, X) şeklindedir.

b. Şifreleme

Şifreleme işlemini gerçekleştirmek için açık anahtar (k_1, X) kullanılır. Şifreli mesaj $C = \text{AÇIK MESAJ}^{k_1} \text{ Mod } X$ şeklindedir.

c. Deşifreleme

Şifreleme işlemini gerçekleştirmek için özel anahtar (k_2, X) kullanılır. Deşifreli mesaj $D = \sqrt[k_2]{C} \text{ Mod } X$ şeklindedir.

Standart RSA algoritmasında bir mesajın N değerinden küçük olması, ve bu yöntemde de bir mesajın N değerinin karekökünden küçük olması, her iki yöntemin de (hem standart RSA ve hem değiştirilmiş RSA) kısıtlı kalmasına neden olmaktadır. Bunun nedeni de bir sayının modu alındığında, elde edilen cevabın her zaman modun sayısına göre daha küçük olmasıdır. Bunun için değiştirilmiş RSA tekniğinde başka bir katman ekleyerek bu sınırlılığı kaldırabiliriz. Örneğin, Adım 5'i tekrarlayarak X 'i

hesapladıktan sonra k_2 'yi elde edebiliriz. Ancak bu sefer elde edilen cevabın karekökünü değil dördüncü kökünü hesaplamalıyız. Değiştirilmiş RSA algoritmasının RSA algoritmasına göre kıyasla daha güvenilir olduğu ancak zaman karmaşıklığının RSA algoritmasına göre kıyasla biraz fazla olduğu vurgulanmıştır (Minni, Sultania, Mishra & Vincent, 2013).

Dijital imza güvenliğini artırmak için bir diğer çalışmada Jaju ve Chowhan (2015) standart RSA algoritmasındaki p ve q asal sayıları yerine p, q ve r şeklinde üç asal sayısı kullanarak genel ve özel anahtarları için n yerine X değerini açık ve gizli anahtarlar parametresi olarak kullanmıştır. Değiştirilmiş RSA algoritması aşağıdaki süreç içerisinde şifreleme ve deşifreleme işlemini gerçekleştirmektedir.

- p, q ve r şeklinde üç rastgele asal sayı seçilir.
- $n = p * q * r$ hesaplanır. n uzunluğu bit cinsinden ifade edilir.
- $\phi(n) = (p - 1) * (q - 1) * (r - 1)$ hesaplanır.
- e değeri aşağıdaki gibi hesaplanır:
 - $n < e < \phi(n)$
 - $\text{GCD}(\phi(n), e) = 1$ olmak üzere, e ve $\phi(n)$ aralarında eş asaldır.
 - e kısa bit uzunluğu ve küçük hamming ağırlığındadır.
- n değeri yerine X değerini kullanmak için:
 - Eğer $p > q$ ise, X değeri, $n - p < X < n$ ve $\text{GDC}(X, n) = 1$ şeklindedir.
 - Eğer $p < q$ ise, X değeri, $n - q < X < n$ ve $\text{GDC}(X, n) = 1$ şeklindedir
- $d, d = e^{-1} \pmod{\phi(n)}$ şeklinde hesaplanır:
- Açık anahtar (e, X) şeklindedir.
- Özel anahtar (d, X) şeklindedir.
- M mesajı $M < n$ şeklindedir.
- Gizli mesaj $C = M^e \pmod{X}$ şeklindedir.
- Deşifrelenmiş mesaj $M = C^d \pmod{X}$ şeklindedir.

Değiştirilmiş RSA algoritmasının standart RSA algoritmasına göre avantajları; bir n sayısının tanımlanması için p, q ve r faktörlerinin her üçünü de elde edilmesini gerektirmektedir. Buda faktörlere ayırma işlemini gerçekleştirirken daha fazla zaman karmaşıklığına neden olarak n sayısının elde edilmesini zorlaştırır. Diğer yandan n sayısı yerine X değerinin bir açık anahtar parçası olarak kullanılması n sayısının gizli değerlerini doğrudan faktörlere ayırma işlemini engellemektedir. Değiştirilmiş RSA algoritmasında şifreleme ve deşifreleme işlemleri standart RSA'ya göre daha fazla zaman karmaşıklığına neden olduğu vurgulanmıştır (Jaju & Chowhan, 2015).

Mathur, Gupta, Goar ve Kuri (2017) yaptıkları bir çalışmada, üssel güçleri (Exponential Powers), N asal sayıları, çoklu açık anahtarları ve K-NN algoritmasını kullanarak standart RSA algoritmasının güvenliğini artırılması üzerinde odaklanılmıştır. Değiştirilmiş RSA algoritmasının aynı zamanda alıcı ve gönderici arasında bir doğrulama sistemini kullandığı belirtilmiştir. Zaman karmaşıklığının standart RSA algoritmasına göre bu algoritmada daha fazla olması değiştirilmiş algoritmanın sınırlıklarındandır (Iswari, 2016).

Bu algoritma aşağıdaki gibi çalışmaktadır:

a. Anahtar üretimi

- A, B, C ve D olmak üzere dört asal sayı seçilir.
- $L = A * B * C * D$ hesaplanır.
- $\phi(L) = (A - 1) * (B - 1) * (C - 1) * (D - 1)$ şeklinde hesaplanır.
- J açık anahtar, $\text{GCD}(J, \phi(L)) = 1$ olmak üzere hesaplanır.
- K gizli anahtar, $K * J \bmod \phi(L) = 1$ olarak hesaplanır.
- N ve O olmak üzere iki rastgele sayı seçilir. $O, \phi(L)$ 'ye göre asal olmamalıdır.
- $Q = PJ$ olmak üzere iki P ve Q sayısı seçilir.

b. Şifreleme

- Şifrelenmesi gereken mesajı ASCII değerleri alınır.
- $E = (\text{ASCII değeri } Q/P) K \bmod L$ olmak üzere, E her bir ASCII değeri için hesaplanır.

- $R1 = (\text{mesaj})^K \bmod L$, gizlenecek mesaj.
- Eğer ASCII değerleri ve R1 aynı ise, K- En yakın komşu algoritması (K-NN) ile alınır.
- $R2 = (\text{message} * N^{R1}) \bmod L$ şeklindedir.
- $H(m)^Y = (R2^O * E^{R1}) \bmod L$ doğrulama işlemi.

c. Deşifreleme

- $(\text{mesaj}) = R1^J \bmod L$
- $H(m)^Y \bmod L$ doğrulama işlemi.

Bir diğer makalede İslam ve arkadaşları değiştirilmiş RSA (MRSA) ismini taşıyan yeni bir yöntem tanılandırmaktalar (İslam, İslam, & Shabnam, 2018). MRSA yönteminde özel ve açık anahtarlar her biri üç bileşenden oluşmaktadır. Bu bileşenlerden biri w,x,y ve z gibi dört asal sayıdan oluşan N bileşenidir. Açık anahtar bileşenlerinden (e,f,N), e ve f bileşeni rastgele olarak seçilmektedir. Bu iki bileşenin rastgele seçilmesi N sayısının faktörlere ayırma işlemi ile birlikte anahtar üretimi sürecini daha da karmaşık yapar. MRSA algoritmasının bir diğer avantajı ise N bileşeninin hem açık anahtar ve hem gizli anahtar parametrelerinde kullanılmasıdır. Bu durum bir saldırganın yalnız N bileşenini elde ederek sistemi kırabilmesini zorlaştırır. Ayrıca gizli anahtar bileşenleri d,g ve N şeklindedir. MRSA algoritmasının çalışma mekanizması aşağıda gösterilmiştir.

a. Anahtar üretimi

- w, x, y ve z olmak üzere dört büyük asal sayı seçilir.
- Açık anahtarın e,f ve N bileşenleri rastgele seçilir.
- Ardından özel anahtarın üssü d,g ve N hesaplanır

- Prosedür:

- $N = w * x * y * z$ hesaplanır.
- $\Phi(N) = (w - 1) * (x - 1) * (y - 1) * (z - 1)$ hesaplanır.
- $1 < e < \Phi(N)$ ve $\gcd(e, \Phi(N)) = 1$ rastgele seçilir.

- $1 < f < \Phi(N)$ ve $\gcd(f, \Phi(N)) = 1$ rastgele seçilir.
- $d * e \equiv 1 \pmod{\Phi(N)}$ olmak üzere d rastgele hesaplanır.
- $f * g \equiv 1 \pmod{\Phi(N)}$ olmak üzere g rastgele hesaplanır.

Burada açık anahtar ve gizli anahtar sırayla e, f ve d, g olmak üzere rastgele bileşenlerden oluştuğu ortadadır, buda algoritmanın okunamaz olmasını sağlar (Islam, Islam & Shabnam, 2018).

b. Şifreleme

- Burada mesaj M ($< N$) olmak üzere şifrelenmiş mesaj, $X \leftarrow (M^e \pmod N)^f \pmod N$ şeklindedir.

c. Deşifreleme

- Deşifrelenmiş mesaj $Y \leftarrow (X^g \pmod N)^d \pmod N$ şeklindedir.

Mod uzunluğunun artması, MRSA bileşenlerinin faktörlere ayrılması konusunda karmaşıklığa neden olur, ve böylece özel anahtarın uzunluğunu artırarak anahtarın kolayca tespit edilmesini zorlaştırır. MRSA algoritmasının zaman karmaşıklığı anahtar üretimi sürecinde geleneksel RSA'ya göre kıyasla yüksek olduğu düşünülmektedir. Anahtar üretimi için gereken sürenin artması bir saldırganın sistemi kırması için gereken sürenin artması demektir.

RSA algoritmasının açıklarını hedef alan bazı saldırılara karşın Goel, çift mod yapısına tabi iki özel anahtar kullanarak şifreleme işlemini gerçekleştiren, çift mod yapısına tabi iki açık anahtar kullanarak deşifreleme işlemini gerçekleştiren ve mod değerlerini üretmek için ikiden fazla büyük asal sayı kullanan RSA tabanlı yeni bir algoritmayı tanılandırmıştır. Çift mod şeması iki büyük açık anahtar ve iki özel anahtarla, bu anahtarların faktörlere ayırma işlemini zorlaştırarak özel anahtarın kolaylıkla elde edilmesini engeller. Algoritmanın çalışma süreci aşağıdadır (Goel, 2017).

a. Anahtar üretimi

p_1 ve p_2 , q_1 ve q_2 olmak üzere dört rastgele asal sayı seçilir.

$n_1 = p_1 \times p_2$ ve $n_2 = q_1 \times q_2$ hesaplanır.

$\phi_1 = \text{LCM}((p_1 - 1), (p_2 - 1))$ ve $\phi_2 = \text{LCM}((q_1 - 1), (q_2 - 1))$ olarak hesaplanır.

$1 < e_1 < \phi_1$ ve $1 < e_2$ şeklinde e_1 ve e_2 tam sayıları seçilir.

$e_1 \times d_1 \bmod \phi_1 = 1$ ve $e_2 \times d_2 \bmod \phi_2 = 1$ olacak şekilde d_1 ve d_2 gizli üsler hesaplanır.

Açık anahtar (n_1, n_2, e_1, e_2) , gizli anahtar (n_1, n_2, d_1, d_2) şeklindedir.

$d_1, d_2, p_1, p_2 \dots q_1, q_2 \dots \phi_1$ ve ϕ_2 gizli tutulmalıdır.

b. Şifreleme

Açık mesaj m ($0 \leq m < n_1 < n_2$) şeklinde pozitif tam sayı olarak gösterilir.

Şifreli mesaj $c = (m^{e_1} \bmod n_1)^{e_2} \bmod n_2$ şeklindedir.

c. Deşifreleme

C_2 ara değeri tersine çevrilebilen şifreli mesaj olmak üzere $C_2^{d_2} \bmod n_2 = (C_1^{e_2} \bmod n_2)^{d_2} \bmod n_2 = C_1$ şeklindedir.

C_1 tersine çevrilebilen $C_1^{d_1} \bmod n_1 = (m^{e_1} \bmod n_1)^{d_1} \bmod n_1 = m$ şeklindedir.

İkili özel anahtarların kullanılmasının nedeni, saldırganın bu anahtarlardan birini tespit ettiği halde, ikinci anahtarın hala güvende olması şifrelenmiş metne ulaşmayı imkansız kılar (Goel, 2017).

BÖLÜM 4

ÖNERİLEN HİBRİT ŞİFRELEME MODELİ

4.1. Motivasyon

Bölüm 3’te tartışıldığı üzere RSA gibi asimetrik şifreleme yöntemleri günümüzde hernekadar yaygın olarak kullanılmaya tercih edilmiş/ediliyor olsa da, güvenlik açısından zafiyetler gösterdiği ortadadır. Bu zafiyetlerin çoğu genel bakımdan benzer şifreleme tekniklerin açık anahtarlı yapılar kullandığından kaynaklanmaktadır. Ağda paylaşılan bir kriptosisteminde açık anahtar yapısı kullanıldığı zaman, saldırganın gizli anahtarı elde etmesi, özellikle paralel işlem yapan ve kuantum mantığına dayanan bir sistemle çok daha kolay olabilecektir. Örneğin, önceki bölümlerde de değinildiği üzere RSA yapısı bir açık anahtar bileşenini (e, N, C) kanaldan paylaştığı durumlarda saldırgan örneğin N değerini bazın büyük sayı olsa da bir kuantum hesaplama işlemini gerçekleştirerek p ve q faktörlerine ayırabilmesi neredeyse saniyelere içerisinde mümkün görülmektedir. Diğer yandan, RSA yönteminde kullanılan bazı büyük sayılar büyük maliyet gerektiren ve her bireyin sahip olamayacağı kuantum sistemlerine karşın güvensiz kalsa da, en azından her bireyin sahip olabileceği klasik bilgisayarlara karşın güvenliğinin sağlanması yine de büyük sayıların kullanılması sayesinde. Ayrıca RSA’nın güvenliği daha büyük sayılara dayanırken işte bu durum yanı sıra bellek kullanımı açısından fazla maliyete yol açmaktadır.

Kolay faktörleyebileme gibi sorunları gidermek için araştırmacılar ve diğer bilim insanları birçok farklı değiştirilmiş güvenilebilir RSA yapılarını önermiş bulunmaktalar. Ancak bu yöntemlerin de bazıları N değerinin sadece karmaşık hale getirilmesi üzerinde odaklı olduğu görülmekte ve bu durumun kuantum hesaplama gibi hızla işlem yapan bir mekanikle kolayca faktörlere ayrılabilmesi muhtemeldir. Benzer durum bir simetrik kriptosistemi üzerinden değerlendirildiğinde bu yapılar özel anahtar paylaşmadıkları sürece güvenlidir. Diğer yandan, her iki yapıyı (asimetrik + simetrik) aynı zamanda kullanan bir hibrit kriptolojisi ele alındığında bu yapıda temel olarak bir mesaj öncelikle göndericinin özel anahtarı ile şifrelenir, daha sonra şifreleme işlemini gerçekleştiren gönderici özel anahtarı alıcının genel anahtarıyla şifrelenir, ve son olarak şifrelenmiş mesaj ve şifrelenmiş anahtar bileşenleri halinde alıcıya sunulur. Alıcı kendine ait özel anahtarla şifrelenmiş anahtarı deşifre eder ve göndericinin özel anahtarını elde etmiş olur. Böylece alıcı göndericiye ait özel anahtar bilgisiyle şifrelenmiş mesajı deşifre ederek gerçek mesaja ulaşabilir.

Yukarıdaki bilgilerden anlaşıldığı üzere bir hibrit yönteminde RSA tekniği kullanıldığında özel anahtar değerinin klasik simetrik yönteminde olduğu gibi kanal dışında seçilmiş olmasına gerek kalmaz; çünkü RSA yönteminde kullanılan açık anahtar özelliği hibrit şifreleme sürecinde kullanıldığında hem gönderici tarafındaki özel anahtar şifreler hemde alıcı tarafına ait özel anahtar değerini belirler, böylece göndericiye ait şifrelenmiş özel anahtar, alıcının genel anahtar bilgisiyle üretilen alıcı özel anahtarıyla elde edilir, ve gerçek bilgiye ulaşılır. Ancak, şifreleme ve deşifreleme işlemini gerçekleştirmek için taraflar arasında bilinmesi zorunlu olan bir özel anahtar değerinin sürekli olarak bir RSA açık anahtar yöntemiyle belirlenmesi, benzer hibrit yapılarının hem güvenlik açısından hemde hız açısından nevi dezavantaj gösterdiğini ortaya koymaktadır (örneğin RSA+AES gibi hibrit yapılarının güvenlik açısından dezavantaj gösterdiğinin sebebi, kullanılan bir RSA açık anahtar içeriğinin saldırgan tarafından faktörlere ayrılabilme ihtimalidir).

Bu çalışmada sözü geçen önerilen hibrit şifreleme modelinde, hem daha çok zaman karmaşıklığına neden olan hemde alıcıdan gelen bir genel anahtarla güvensizce özel anahtar belirleme işlemini gerçekleştirmeksizin, **özel anahtar değerinin taraflar arasında kanal dışında seçilmesi varsayılmıştır**. Gönderici bir mesajı şifrelemek için öncelikle rastgele bir n sayısı seçer ve **mesajı** bu sayı ile şifreler (**mesaj $\cdot n$**); daha sonra

n sayısını gönderici ve alıcı arasında bilinen (her şeyden önce gönderici ve alıcı aralarında kanal dışında bir **P** özel anahtarı seçmelidir) bir **özel anahtarla** şifreler (**n+P**); son olarak da alıcıya **şifreli mesaj (mesaj*n)** ve **genel anahtar (n+P)** olmak üzere iki bileşen gönderilir. Alıcı daha önce gönderici ile anlaştığı bir P özel anahtar bilgisine sahip olduğu için **genel anahtar (n+P) bileşeninden özel anahtar değerini çıkartma yapar ((n+P) – P)**; böylece gönderici tarafında bilinen **n** değeri elde edilir ve **şifreli mesaj/n** yaparak gerçek mesaja ulaşılır. Şifreleme prosedürü örneğin RSA+AES hibrit modelinde olduğu gibi kanal yolundan alıcıdan gelen bir genel anahtar değerine bağlı değildir (buda algoritma hızını artırır), ve şifreleme ve genel anahtar üretimi sadece gönderici tarafında bilinen değerlerle (örneğin **şifreli mesaj=n*M** ve **genel anahtar=n+P**; burada P özel anahtar değeri kanal dışında seçilmiş olduğundan sadece alıcı ve gönderici arasında bilinen bir değerdir. Yani özel anahtar değeri genele açık kanal yolundan geçen bir değerle belirlenmediği için saldırganın özel anahtar bilgisini analiz edebilmesi zordur) gerçekleştirilebilmektedir. Ayrıca bu model bir hibrit şifreleme yönteminin tersine (normalde bir hibrit yapısında alıcı genel anahtarı göndericiye her işlem başına sadece bir kez göndermektedir, çünkü alıcı tarafından farklı genel anahtar üretilerek alıcı özel anahtarı değerinin belirlenmesi ve buna göre göndericiye her seferinde yeni genel anahtar değerinin iletilmesi fazla maliyete neden olmaktadır) yüksek maliyet gerektirmeden gönderici tarafından seçilerek alıcıya gönderilen rastgele bir **+R** sayısı ile birden çok defalarda özel anahtar değiştirme işlemini gerçekleştirebilmektedir (büyük sayılar konusunda daha avantajlıdır). Diğer yandan bu yöntem büyük hacimli verileri çok az ve sabit maliyetli zaman ve alan karmaşıklığıyla şifreleyebilmektedir. Hedeflenen modelin küçük işletmelerde (daha çok yerel ağ kullanan ve fiziksel teması yakın ağsal ortam) kullanıldığında hem güvenlik açısından hemde hız açısından daha çok verimlilik sağladığı öngörülmektedir.

4.2. Çalışma Mekanizması

Bu bölümde, simetrik ve asimetrik kriptoloji özelliğinden faydalanan bir önerilen şifreleme modelinin güvenliği, zaman ve alan karmaşıklığı değerlendirilmektedir. Bu modelde ilk önce taraflar arasında bilinen bir özel anahtar seçilmeli, ve bu özel anahtar kanal üzerinden paylaşılmamalıdır. Sonraki adımda, deşifreleme işlemini gerçekleştirmek için bir genel anahtar parametresi oluşturulmalıdır. Bu modelde genel

anahtar, taraflar arasında anlaşmalı olarak seçilmiş bir **P** özel anahtar sayısı ve bir **M** mesajını şifrelemek için kullanılan **n** sayısı parçasından oluşmaktadır. Ayrıca aynı kullanıcılar farklı bir özel anahtarla işlem yapmak istediklerinde, önceden seçilmiş (ilk belirlenen özel anahtar) bir **P** özel anahtarını kanal üzerinden paylaşılabilen bir $+R$ sayısı ile değiştirilebilmektedirler. Ele aldığımız önerilen hibrit krypto modeli aşağıdaki adımlarla çalışmaktadır.

Adım 1:

Gizli anahtar seçimi:

P gönderici ve alıcı için bilinen ve kanal dışında seçilmiş herhangi bir gizli anahtar sayısı olsun.

Genel anahtar üretimi:

Gönderici için bilinen **P** özel anahtarı üzerinden genel anahtar değeri $k=P+n$ şeklindedir.

Şifreleme işlemi:

M bir mesaj ve **n** bu mesajı gizleyecek herhangi bir sayı olsun. **C** şifreli mesaj $C=M*n$ şeklindedir.

Deşifreleme işlemi:

Alıcı göndericinin sahip olduğu **P** bilgisine sahiptir. Bu nedenle göndericinin sadece **k** ve $+R$ değerlerini alıcıya göndermesi yeterlidir. **D** deşifrelenmiş mesaj şu şekildedir.

P alıcı için önceden bilinen gizli anahtar olsun.

$D=C/k-P$ şeklindedir.

Adım 2:

Özel anahtar değişim süreci:

Özel anahtarı değiştirmek için örneğin $+R$ göndericiden alıcıya gönderilen herhangi bir sayı olsun, yenilenmiş gizli anahtar $P=P+R$ şeklindedir.

Adım 3:

- *Özel anahtar değiştikten sonra...*

Genel anahtar üretimi:

Genel anahtar değeri $k=P+n$ şeklindedir (burada P değeri değişen yeni değerdir).

Deşifreleme işlemi:

$D= C/k-P$ şeklindedir.

4.2.1. Örneklendirme

Adım 1:

Gizli anahtar seçimi:

$P=12$ gönderici ve alıcı arasında anlaşmalı olarak seçilmiş herhangi bir gizli sayı olsun.

Genel anahtar üretimi:

Genel anahtar değeri $k=12+3=15$ şeklindedir.

Şifreleme işlemi:

$M=7$ bir mesaj ve $n=3$ bu mesajı gizleyecek bir sayı olsun. C şifreli mesaj $C=7*3=21$ şeklindedir.

Deşifreleme işlemi:

Alıcı ve gönderici genel kanalı kullanmadan önce aralarında bir özel $P= 12$ anahtarı üzerinde anlaşmaktalar.

Deşifreli mesaj $D= 21/15-12=7$ şeklindedir.

Adım 2:

Özel anahtar değişimi süreci:

$+R = -9$ göndericiden alıcıya gönderilen bir sayı olsun, yenilenmiş gizli anahtar değeri $P= (P=12)-9=3$ şeklindedir.

Adım 3:

- *Özel anahtar değiştikten sonra...*

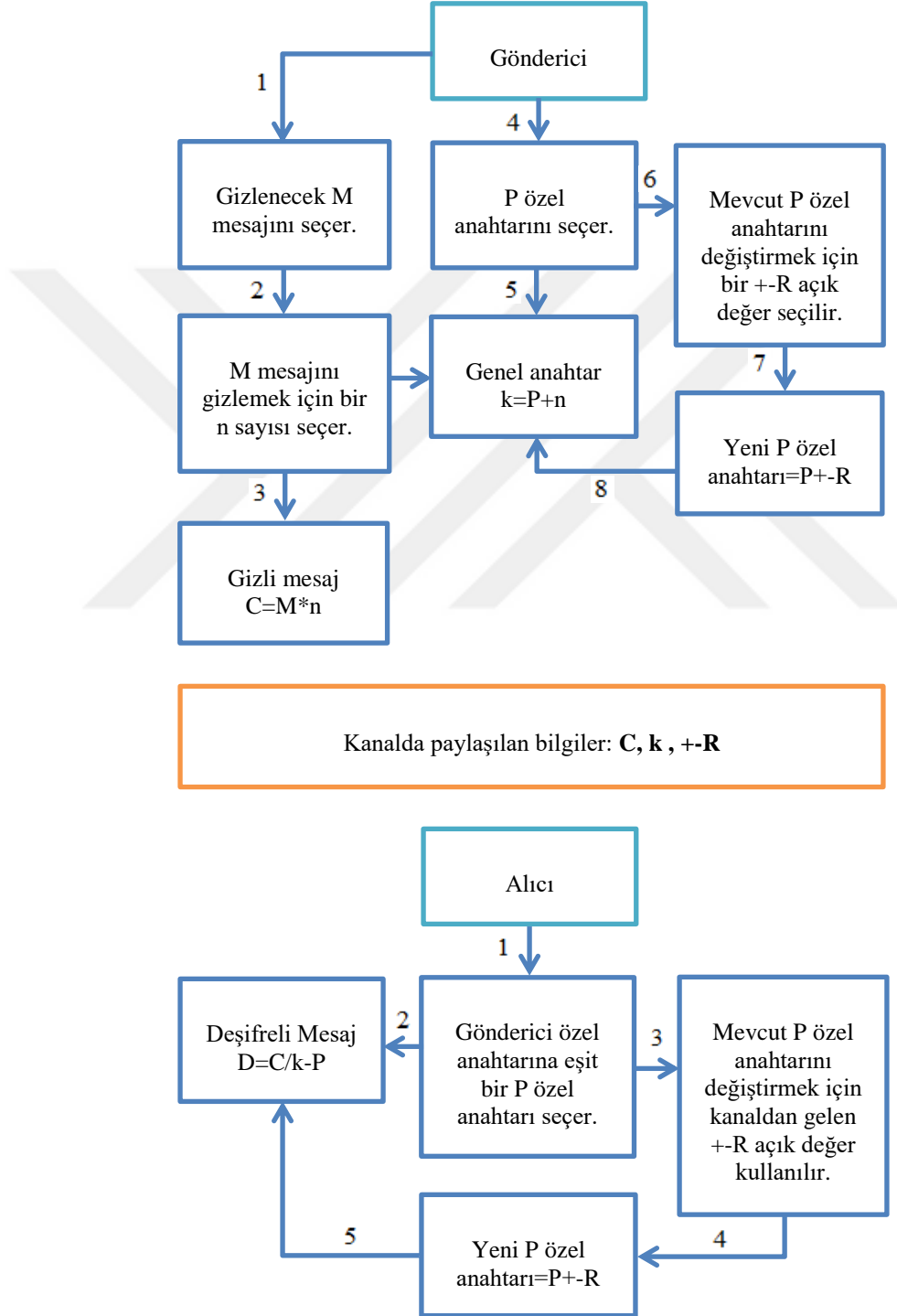
Genel anahtar üretimi:

Genel anahtar değeri $k=(P=3)+(n=3)=6$ şeklindedir.

Deşifreleme işlemi:

$D = 21/6 - 3 = 7$ şeklindedir.

4.2.2. Süreç Diyagramı



Şekil 4.1. Önerilen Hibrit Şifreleme modelinin süreç şeması.

4.2.3. Kod Örneği (C#)

```
using System;
using System.Diagnostics;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Numerics;
namespace model_cryptography
{
    public partial class Sender_Form : Form
    {
        public Sender_Form()
        {
            InitializeComponent();
        }
        public void getdefault()
        {
            P.Text = Properties.Settings.Default.privatekey;
        }
        public void savedefault()
        {
            Properties.Settings.Default.privatekey = P.Text;
            Properties.Settings.Default.Save();
        }
        private void encryption_btn_Click(object sender, EventArgs e)
        {
            Stopwatch enc_time = new Stopwatch();
            encryption_runtime.Text = "";
            enc_time.Reset();
            enc_time.Start();
            C.Text = (BigInteger.Parse(M.Text) * BigInteger.Parse(n.Text)).ToString();
            enc_time.Stop();
            encryption_runtime.Text = enc_time.Elapsed.ToString();
        }
        private void privatekey_select_btn_Click(object sender, EventArgs e)
        {
            savedefault();
        }
        private void Sender_Form_Load(object sender, EventArgs e)
        {
            getdefault();
            _getdefault();
        }
        private void publickey_gen_btn_Click(object sender, EventArgs e)
        {
            k.Text = (BigInteger.Parse(P.Text) + BigInteger.Parse(n.Text)).ToString();
        }
        private void privatekey_change_btn_Click(object sender, EventArgs e)
        {
            P.Text = (BigInteger.Parse(P.Text) + BigInteger.Parse(R.Text)).ToString();
        }
        public void _getdefault()
        {
            _P.Text = Properties.Settings.Default._privatekey;
        }
        public void _savedefault()
        {
            Properties.Settings.Default._privatekey = _P.Text;
            Properties.Settings.Default.Save();
        }
        private void privatekey__select_btn_Click(object sender, EventArgs e)
        {
            _savedefault();
        }
        private void send_Click(object sender, EventArgs e)
        {

```

```

{
    _C.Text = C.Text;
    _k.Text = k.Text;
    _R.Text = R.Text;
}
private void decryption_btn_Click(object sender, EventArgs e)
{
    Stopwatch dec_time = new Stopwatch();
    decryption_runtime.Text = "";
    dec_time.Reset();
    dec_time.Start();
    D.Text = (BigInteger.Parse(_C.Text) / (BigInteger.Parse(_k.Text) - BigInteger.Parse(_P.Text))).ToString();
    dec_time.Stop();
    decryption_runtime.Text += dec_time.Elapsed;
}
private void privatekey__change_btn_Click(object sender, EventArgs e)
{
    _P.Text = (BigInteger.Parse(_P.Text) + BigInteger.Parse(_R.Text)).ToString();
}
}
}

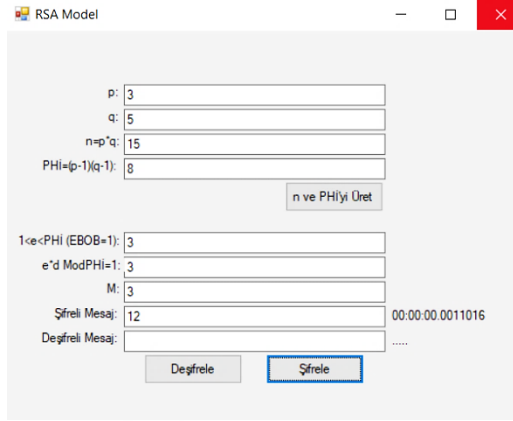
```

4.2.4. Zaman ve Alan Karmaşıklığı

Bir önceki bölümlerde asimetrik şifreleme yöntemlerinin güvenlik açıkları ve bu açıkları gizleyebilir birçok yöntemlerden bahsedilmiştir. Sonuç olarak, Örneğin, RSA gibi açık anahtarlı şifreleme yönteminin güvenliğinin tamamen büyük sayılara bağlı olduğu ve bu sayıların yanı sıra hızla işlem yapan bir mekanikle faktörlere ayrılabilme ve kırılabilirliğe neden olabileceği örneklendirilmiştir. Ayrıca, büyük sayıların kullanılması bu yöntemin bellek kaplama ve zaman karmaşıklığının artırması konusunda da dezavantaj gösterdiği anlamına gelmektedir. Diğer bakımdan RSA yönteminin güvenlik kırılabilirliği sadece N sayısının faktörlere ayrılabilir gibi örneklerle kalmaksızın, bazın kanaldan gönderilen e parametresinin de saldırgan hedefine rastladığı Bölüm 3'te tartışılmıştır.

Bu bölümde, önerilen modelin şifreleme ve deşifreleme işlemlerinde getirdiği uygulama zaman karmaşıklığı ve bellek kullanımı maliyetlerinin analizi RSA yöntemi ile birlikte uygulamalı olarak kıyaslanmaktadır. Daha verimli sonuçlara erişebilmek için algoritmanın uygulama zamanı ve uygulama alanı ölçü kabul edilerek C# programlama dili ile değerlendirilmekte, hem hedeflenen önerilen modelde hemde RSA modelinde gizlenecek sayılar, özel ve genel anahtar değerleri eşit olarak seçilmektedir. Geliştirme Platformu olarak Visual Studio 2019 sürüm 16.9.3 ve programlama dili olarak da C# programlama dilinden yararlanmaktadır. Uygulama Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz 2.81GHz 16GB RAM 64-bit Windows10 işletim sistemi üzerinde gerçekleştirilmektedir.

Şifreleme Süreci: $M=3$ sayısı gizlenecek bir mesaj olsun, RSA modelinde bir $M=3$ mesajını gizlemek için gereken uygulama zamanı:

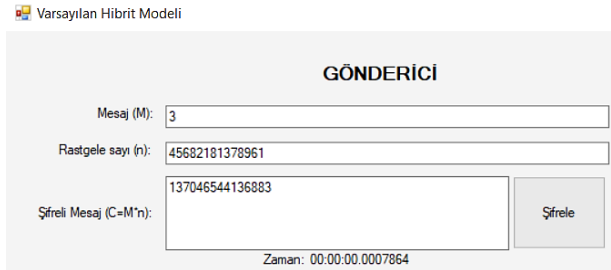


Şekil 4.2. RSA modelinde $M=3$ gizli mesajı için şifreleme süresi.

```
C:\Users\Person\Desktop\RSA_Model\RSA_Model\bin\Debug\RSA_Model.exe
Döngüsel adım 1: 3
Döngüsel adım 2: 9
Döngüsel adım 3: 27
Şifrelenmiş Mesaj: 12
```

Şekil 4.3. RSA modelinde $M=3$ mesajını şifrelemek için döngü sayısı.

Yukarıda, Şekil 4.2 ve 4.3'ten istinaden RSA modelinde bir $M=3$ sayısını şifrelemek için minimum bir üssel $M^e \text{Mod}_n$ işlemi gerçekleştiğinde şifrelenmiş mesajdan (şifrelenmiş mesaj=12) elde edilen toplam şifreleme süresi 0011016_{ms} olarak bulunmuştur. Aynı $M=3$ sayısı büyük bir sayı ile çarpılarak önerilen hibrit modeli üzerinden şifrelendiğinde aşağıdaki performansı ortaya koymaktadır:

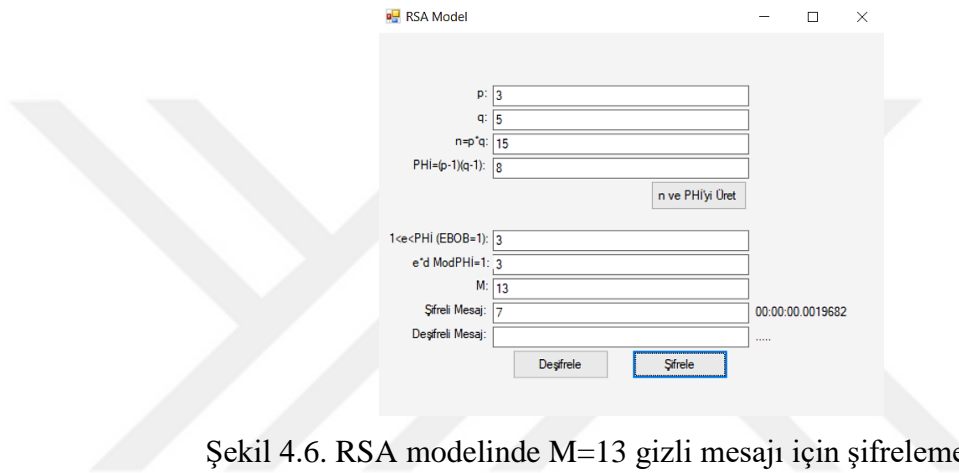


Şekil 4.4. Önerilen Hibrit Modelinde $M=3$ gizli mesajı için şifreleme süresi.

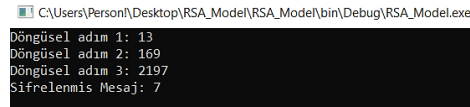
```
C:\Users\Person\Desktop\model_cryptography\model_cryptography\bin\Debug\model_cryptography.exe
Şifrelenmiş Mesaj: 12
```

Şekil 4.5. Önerilen Hibrit Modelinde $M=3$ mesajını şifrelemek için döngü sayısı.

Yukarıda, Şekil 4.4 ve 4.5'ten istinaden önerilen hibrit modelinde bir $M=3$ sayısını şifrelemek için $M*n$ işlemi gerçekleştiğinde şifrelenmiş mesajdan (şifrelenmiş mesaj=137046544136883) elde edilen toplam şifreleme süresi 0007864_{ms} olarak bulunmuştur. Buda önerilen modelin şifreleme sürecinde herhangi bir üssel işlem gerçekleştirmediğinden ve büyük sayılarla (n) işlem gördüğünde de (çarpma) RSA modeline göre kıyasla daha az zaman maliyetine neden olduğunu göstermektedir. RSA algoritması Şekil 4.3'te olduğu gibi üssel bir işlemle şifreleme işlemini gerçekleştirdiği için büyük sayılara gelince daha çok zaman karmaşıklığına yol açmaktadır. Örneğin:



Şekil 4.6. RSA modelinde $M=13$ gizli mesajı için şifreleme süresi.



Şekil 4.7. RSA modelinde $M=13$ mesajını şifrelemek için döngü sayısı.

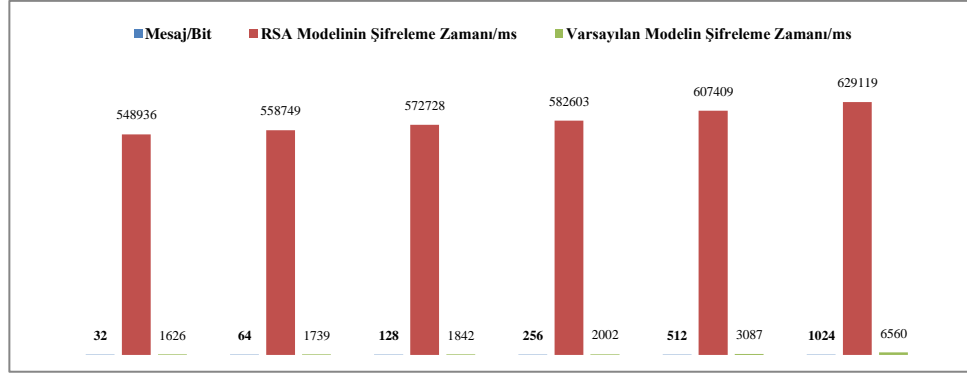
Şekil 4.6 ve 4.7'ye görüldüğü üzere RSA algoritmasında birbirinden büyük M sayıları (Şekil 4.2'de $M=3=2\text{bit}$ ve Şekil 4.6'de $M=13=4\text{bit}$) bir $M^e \text{Mod}_n$ üssel işlemine tabi tutulduğunda daha çok zaman karmaşıklığı farkı yaratabilmektedir.

Aşağıda Çizelge 4.1 ve Şekil 4.8'de görüleceği üzere 32bit, 64bit, 128bit, 256bit, 512bit ve 1024bit açık sayıları üzerinde şifreleme işlemi gerçekleştirilmiş, RSA ile önerilen hibrit modelinin uygulamadaki şifreleme zamanı karmaşıklığı değerlendirilmiştir.

Çizelge 4.1. RSA modeli ve Önerilen Modelin şifreleme süresi karmaşıklığı.

RSA Asal p	RSA Asal q	RSA'nın e, ve Önerilen Modelin n Değeri / 1394 Bit	RSA'nın n Değeri / 1394 Bit	Açık Mesaj	Açık Mesaj/Bit	RSA Modeli (şifreleme zamanı/ms)	Önerilen Model (şifreleme zamanı/ms)
449417999055441493994709297093108513015373787049558499205492347871729927573118262811508386655998299074566974373711472560655026286868094291699357843464363003144674940345912431129144354948751003607115263071543163	643808006803554439230129854961492699151386107534013432918073439524138264842370630061369715394739134090922937332590384720397133335969549256322620979036686633213903952966175107096769180017646161851573147596390153	2191007700470190299469236189207046547807534734135158199918713096387453183832578443679016196366614993613676225674401108177098932420514855096875088931556678287531788602201771689135818489580909391445481640700996397328010731387216496283556076595798470956550527572577859656543150376036960635758078365336518657391482441313369097854802020516044387832110784785042930655040959873483654544388484756545632816163725513303088751836125	28933890619352549990931708565512930017736584867001099610347560335031780711080944105049006243396690672409759478946409204186823477946084646778060001311508091166320254879841497976151533484802447858476276955511439647547230142540914027248277018746492731723054391029052435045639900211954700658629784297642678110394919124636996525386362866244153077941161035330800798648851657896742784828971363925559538465164094116837927673939	423499 0790	32	548936	1626
				124389 457391 635768 30	64	558749	1739
				24687913 20060931 30734065 89107203 6790281	128	572728	1842
				6098873485 3896467722 4898273584 0269840568 7463133552 0648737880 2675311312 3469490	256	582603	2002
				685874631335906 4873788026753913 1543004538084218 6707806450047878 0814756869770884 773890230746814 5656760870745450 0303007901001404 4538981344607792 4024903477	512	607409	3087
				99072476873426415681145858426 31543566580065574328687414098 68549290643269828442803359791 65031546294962609905572604301 68308375379905276258145054596 65833147503759435608587025808 81401676550540986590760756120 05778457516517521977988266121 68321570449597384334018067093 06255002504068469987070177783 413362779395721913	1024	629119	6560

Yukarıda Çizelge 4.1'den istinaden Önerilen Modelin şifreleme süresi RSA modeline göre kıyasla daha az olduğu görülmektedir. Ancak önerilen modelde $C=M*n$ ilişkisine bakıldığında herhangi bir modülü işlemi gerçekleşmediği için elde edilen şifrelenmiş mesaj değeri daha çok bit uzunluklu farkı yaratmaktadır (bkz; Çizelge 4.2). Bu nedenle, önerilen modelde şifreleme süresi az olsa da elde edilen şifrelenmiş veri büyüklüğüne göre şifreleme işleminde bellek kaydı zamanının daha çok fark yaratması mümkündür.



Şekil 4.8. RSA modeli ve Önerilen Modelin şifreleme süresi karmaşıklığı.

Şifre Çözme Süreci: Yukarıda bir mesajı şifrelerken uygulama zamanı önerilen hibrit modeli ve RSA modeline göre değerlendirilmiştir. Aşağıda deşifreleme sürecinde gereken uygulama zamanı ele alınmaktadır.

p: 3
 q: 5
 n=p*q: 15
 PHI=(p-1)*(q-1): 8
 n ve PHI'yi Üret
 1<e<PHI (EBOB=1): 3
 e*d ModPHI=1: 3
 M: 3
 Şifreli Mesaj: 12
 Deşifreli Mesaj: 3
 00:00:00.0013824
 Deşifrele
 Şifrele

Şekil 4.9. RSA modelinde şifreli mesaj=12 için deşifreleme süresi.

```

C:\Users\Person\Desktop\RSA_Model\RSA_Model\bin\Debug\RSA_Model.exe
Döngüsel adım 1: 12
Döngüsel adım 2: 144
Döngüsel adım 3: 1728
Desifrelenmiş Mesaj: 3
  
```

Şekil 4.10. RSA modelinde şifreli mesaj=12 çözmek için döngü sayısı.

Şekil 4.9 ve 4.10'de, RSA modelinde $M=3$ gizlenecek mesajdan üretilen şifrelenmiş mesaj=12 deşifre edilirken gereken uygulama zamanı şifreleme sürecinden kıyasla daha fazla olduğu görülmektedir (bkz; Şekil 4.2 ve 4.9). Çünkü RSA modelinde d özel anahtarı veya şifrelenmiş mesaj sayısı büyük olduğunda $C^d \text{Mod}_n$ ilişkisinde üssel maliyet de o miktarda artacaktır (burada, şifrelenmiş mesaj=12 için deşifreleme sürecindeki uygulama zamanı 0013824_{ms}). Bu durum önerilen model üzerinde uygulandığında aşağıdaki sonucu ortaya koymaktadır.

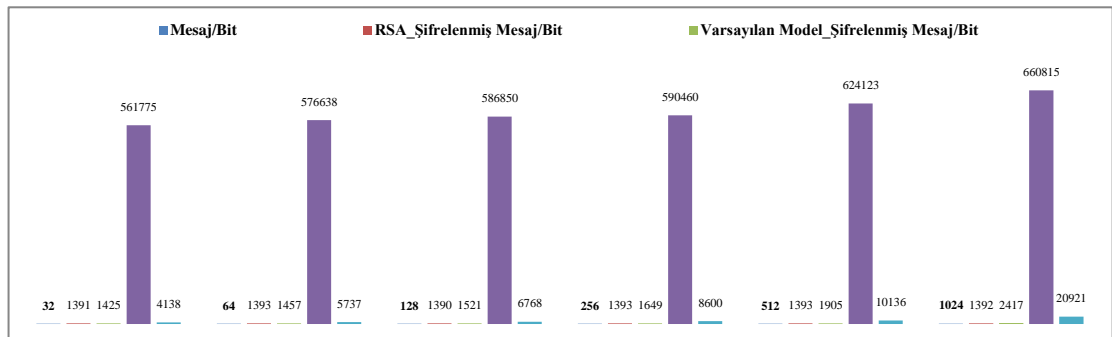
Şekil 4.11. Önerilen Hibrit Modelinde şifreli mesaj=12 için deşifreleme süresi.

C:\Users\Person\Desktop\model_cryptography\model_cryptography\bin\Debi
Desifrelenmiş Mesaj: 3

Şekil 4.12. Önerilen Hibrit Modelinde şifreli mesaj=12 çözmek için döngü sayısı.

Şekil 4.12’de P özel anahtar ve k genel anahtar olmak üzere $D=C/k-P$ ilişkisinden deşifreleme süresi 0010125ms olarak bulunmuştur. Buda önerilen modelde özel anahtar sayısı büyük olduğu durumlarda büyük genel anahtar sayılarını üreterek deşifreleme süresinin şifreleme süresinden kıyasla artacağını göstermektedir (Şekil 4.4 ve 4.11).

Aşağıda Çizelge 4.2 ve Şekil 4.13’te görüldüğü gibi 32bit, 64bit, 128bit, 256bit, 512bit ve 1024bit açık sayılarının şifrelenmesi sonucundan elde edilen 1391bit, 1393bit, 1390bit, 1393bit, 1393bit ve 1392bit RSA şifrelenmiş değerleri 1425bit, 1457bit, 1521bit, 1649bit, 1905bit ve 2417bit Önerilen Model şifrelenmiş değerleri deşifre edilirken uygulamadaki şifre çözme zaman karmaşıklığı kıyaslanmıştır.



Şekil 4.13. RSA modeli ve Önerilen Modelin deşifreleme süresi karmaşıklığı.

Yukarıda Şekil 4.13’te görüldüğü üzere Önerilen Modelinin uygulamada deşifreleme süresi RSA modeline göre kıyasla daha az olduğu söylenebilmektedir.

Ayrıca, Çizelge 4.2’den istinaden uygulamada Önerilen Modelin deşifreleme zamanı şifreleme zamanından kıyasla daha yüksektir. Deşifreleme zamanının yüksek olmasının sebebi, $D=C/k-P$ ilişkisinden k açık anahtar değerinin özellikle daha büyük sayılarla deşifreleme sürecinde yer almasıdır.

Çizelge 4.2. RSA modeli ve Önerilen Modelin deşifreleme süresi karmaşıklığı.

RSA'nın d ve Önerilen Modelin P Gizli Anahtarı / 1394 Bit	Önerilen Modelin k Açık Anahtar Değeri / 1395 Bit	Açık Mesaj/Bit	RSA _ Şifrelenmiş Mesaj										RSA _ Şifrelenmiş Mesaj/Bit	Önerilen Model _ Şifrelenmiş Mesaj	Önerilen Model _ Şifrelenmiş Mesaj / Bit	RSA Modelinin (deşifreleme zamanı/ms)	Önerilen Modelin (deşifreleme zamanı/ms)																																															
2756102912320424246703858095385034664529965413962635548207773243210721029271194094398402536097499688621973443276654121788255252036390417075308321792556120144538189946935055755865782107772590556926929187354842983614739077400135511340247083786353024646698425203228024408029351539451846306457540155242464903878676812594083669783188923514043410414288008684758573593020830664059625933168517		32	20967348380475337470210009 7129918397510304630377787 94119031739684208206359983 28801641869030011221759227 092568457509146908954092 527200416690908891913153 52204132792068207927139910 420418387392068207927139910 304454545071912365486643 987938534341034889348747621 042690234223240255342012 30825038844169619880919667 802384382161961988028119 9525287530654910718025246 60881413871486101715744770 76820991392590601371101127 82315544333731818989731513										1393	210	1391	92788974323103345877995571 49626539533066204291667493 5918487287118532465080714 66109331673078734459614298 27633773050231895807667719 71281839345055940557347291 0690906662514768231729343 62486862232596119535482808 886507300546402578549692 92128317410771970739744595 0829220372805080101564992 028065452944176471609356 340883363285696961261702 970017723983428941078701 32136962842211026879126026 147191751068092660334257	1425	561775	4138																																													
49471106137630345461423219874555001426053127553142174812648633959817421310373438077418732464114680435649680450552296634525456905271721836107241128044329135980715523292549996060597353499483724108280585805695689484639127232047475807849744337734032489527758088006462608535003147909882425376688976514779350122577774648331210975118482195623366487690316390971168239889620113674422373219786164803621940003464575004442		128	15157741108146599877559395 195661228485389647722489 8277490971741475520648737 7880267531312469494900935 9988186615994320516594162 4785361837030387582301982 91711045126727495382982302 37923476646712450438208774 91441499761892612892542649 585310925020295830576627 73291683576090980561166288 1756196633812119544976992 1864297954814365757851481 09513231913067717878599 657826369526392615514051 14718794952843966015514051 47591973525476933626991395 07779325768428764182932296										1390	155	1393	54091407930170666936609504 1705179304501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 84589359667167513766411 7923390304701125	1457	6056	1457	27253825900238260016250492 6900051778297696286121782 3922394563865634164200827 7936496909043204983963584 6584979639894675783896823 7709143955208666450891759 75079487060010772474605118 82651425113984489009217026 9114206778041051626661631 0389363564730505044341680 339911390092865160347109 73480204870245967884215355 336881339887823047587453 70369920090751733078690028 5914302657400166479786256 4703758225385006983750	1457	576638	5737																																									
2756102912320424246703858095385034664529965413962635548207773243210721029271194094398402536097499688621973443276654121788255252036390417075308321792556120144538189946935055755865782107772590556926929187354842983614739077400135511340247083786353024646698425203228024408029351539451846306457540155242464903878676812594083669783188923514043410414288008684758573593020830664059625933168517		256	15227552913895235861482729 7015882281134539850506968 045405191736137483390317 5989430893803418671079055 988186615994320516594162 4785361837030387582301982 91711045126727495382982302 37923476646712450438208774 91441499761892612892542649 585310925020295830576627 73291683576090980561166288 1756196633812119544976992 1864297954814365757851481 09513231913067717878599 657826369526392615514051 14718794952843966015514051 47591973525476933626991395 0779325768428764182932296										1393	8237	1393	1336267877068218478798009 79148136701790556715486999 4694906271115298827073789 894623271115298827073789 61205075339514676480990 912407784596110462638024 3750712311262575858747982 22124372125208814553463735 20416931843399189564089433 1340184105857441671579878 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 038657813652408154098548 1529294730013687476961574 4544194607427595331866468 07721482808486514847411 19598506387816931439661045 9285414703993453308278866 243718590935707742039352 8228925487815926221245256 56625818817184711686243981 0486810367744540826601963 8459359667167513766411 7923390304701125	1649	8237	1649	54091407930170666936609504 1705107934501326637697723 1176191045254026924488018 4219447181730481849816818 077809164616986904572083 8536720527585887363642 23975412659926103451490776 71231185345836408039006 2133778828748152667914892 07721482808486514847411 03865781365240815409

Alan Karmaşıklığı Analizi: Teoride, bir algoritma yürütülürken, n büyüklüğündeki üye sayısı için genel bir ölçekleme de zaman ve bellek kullanım ölçüsüdür. Bir algoritmanın en kötü durumu (Worst Case) veya ortalama çalışma süresi ve bellek kullanımı genellikle giriş uzunluğunun bir fonksiyonu olarak büyük O ile ifade edilir. Bir fonksiyonun büyük O notasyonu cinsinden tanımlanması genellikle, fonksiyonun büyüme sınırı üzerinden yapılmaktadır (Devi, Selvam & Rajagopalan, 2011, s. 844).

Bu bölümde hedeflenen önerilen modelin ortalama alan karmaşıklığı ve zaman karmaşıklığı analizi RSA modeline kıyasla yapılmaktadır. Bu bağlamda öncelikle bir RSA modelinin şifreleme sürecinin uygulamada kapladığı bellek aşağıda verilen kod üzerinden değerlendirilmektedir.

```
BigInteger increment = 1;
BigInteger pow=1;
while (increment <= e) // seviye 1
{
    pow *= M; // seviye 2
    Console.WriteLine("Döngüsel adım "+ increment + ": " + pow);
    increment ++;
}
Console.WriteLine("Şifrelenmiş Mesaj: "+pow % n);
```

Yukarıdaki verilen kod'ten anlaşılabacağı üzere RSA algoritmasının şifreleme sürecinde bir M mesajını şifrelemek için bir üssel işlem gerçekleştirmesi gerekmektedir, bu üssel işlem (seviye 2) $M^e \text{Mod}_n$ ilişkisinden e 'ye kadar devam ederek (seviye 1) her bir değişen durum için bir birim bellek kapladığı görülmektedir. Bu nedenle RSA algoritmasının şifrelemede alan ve zaman karmaşıklığı bilinmeyen e 'döngüsüne göre $O(n)$ olarak ifade edilmektedir. Bu durumun aynısı $C^d \text{Mod}_n$ ilişkisinden maksimum deşifreleme sürecinde de geçerlidir. Bu durum önerilen hibrit modelinde ele alındığında aşağıdaki sonucu ortaya koymaktadır.

```
Console.WriteLine("Şifrelenmiş Mesaj: "+ M * n);
```

Verilen kod'ten anlaşılabacağı üzere önerilen modelin şifreleme sürecinde $M*n$ ilişkisinden bir M mesajını şifrelemek için herhangi bir üssel işlem gerçekleştirmediği görülmektedir. Bu nedenle önerilen hibrit modeli bilinen bir $M*n$ işlemi için tek bellek birimi kaplayarak $O(1)$ sabit fonksiyonlu alan ve zaman karmaşıklığını ortaya koymaktadır. Ayrıca önerilen modelin deşifreleme sürecinde $C/k-P$ ilişkisinden herhangi bir üssel işlem gerçekleştirmediği görülmekte, ve deşifreleme alan ve zaman karmaşıklığını $O(1)$ olduğunu göstermektedir.

Çizelge 4.3. Önerilen Hibrit şifreleme modelinin zaman ve alan karmaşıklığı.

Fonksiyonlar	Uygulama Süreci	Zaman Karmaşıklığı	Bellek Kullanımı/Değişken
Sabit Değerler	$M, n, C = M * n, P, ++R,$ $k = P + n, D = C / k - P, P = P + -R,$ $k = P + n, D = C / k - P$	$O(1)$	$O(1)$
Sabit Koşullar	$++R$	$O(1)$	$O(1)$
Toplam Zaman Karmaşıklığı ve Bellek Kullanımı		$O(1) + O(1) = O(1)$	$O(1) + O(1) = O(1)$

Yukarıdaki Çizelge 4.3'te görüldüğü üzere, simetrik ve asimetrik özelliğinden faydalanan bir hibrit şifreleme yapısı önerilen modele göre temel olarak $O(1)$ sabit bir zaman ve alan karmaşıklığına sahiptir. Bu modelin özellikle modülü tabanı altında şifreleme ve deşifreleme işlemini gerçekleştiren bir asimetrik algoritmasından kıyasla daha az bir zaman ve alan maliyetine neden olduğu söylenebilmektedir.

BÖLÜM 5

SONUÇ VE TARTIŞMA

Ağ güvenliği, izinsizce veya ağın savunmasız olduğu durumlarda gerçekleşmekte olan ağ saldırılarını izlemek, önlemek ve bunlara yanıt verebilecek geliştirilmiş teknikler ve güvenlik politikalarını kapsayan bir terimdir. Ağ güvenliği, donanım ve yazılım teknolojileri için tüm potansiyel tehditlere yanıt vermek üzere tasarlanmıştır. Bu çalışmada, ağ güvenliğinin tanımına ilişkin temel kavramlar ele alınmıştır. Taramalar kapsamında; güvenilir sistemler ve güvenli sistemleri, kurumsal ve kişisel bakımdan ağ güvenliği farkındalığı, ağ güvenliğine neden ihtiyaç duyulduğu, genel olarak bir ağın amacı, güvenliğin boyutları, internet mimarisi, ağ güvenliğini tehlikeye atan saldırılar ve bu saldırıya karşı önlem yöntemleri gibi yaklaşımlara değinilmiştir. İnceleme sonucundan genel anlamda bir ağ ortamında gerçekleşebilecek saldırı çeşitlerine karşı her ne kadar uygun özelliklere sahip savunma yöntemler, teknikler ve algoritmalar geliştirilmiş olsa da, günümüzde teknoloji gelişmeleri ile birlikte kötü amaçlı bireylerin, robotların veya sanal yapıların saldırı becerileri de o kadar gelişmektedir. Bu durum özellikle asimetrik kriptoloji sistemlerine daha çok geçerlidir. Bir asimetrik şifreleme sistemi her ne olursa olsun bir mesajı tamamıyla gizleyerek kötü amaçlı bireylerin orijinal mesajın mahiyetine ulaşamamalarını garanti edemez. Örneğin, RSA algoritması bir açık anahtarlı şifreleme yöntemi olarak bir mesajı şifreledikten sonra alıcının bu mesajı açabilmesi için gönderici tarafından şifreleme sürecinde kullanılan genel anahtar açık bir pozisyondan elde etmesi gerekir.

Bu durum bir gizli mesajı okuyabilmek için alıcının gizli anahtarının ancak göndericinin açık anahtarıyla birlikte iş görebilir olduğunu ortaya koymaktadır. Hal böyle olunca bir saldırgan kanal boyunca paylaşılan açık anahtar elde ederek bu anahtar üzerinde farklı hesaplamalı işlemlerini gerçekleştirebilir ve böylece orijinal mesaja ulaşabilmektedir. Ancak RSA güvenliğinin özellikle büyük sayılar uzunluğuna bağlı olması bir saldırganın klasik bir bilgisayarla sanıldığı kadar kolay bir şekilde benzer büyük genel anahtar sayılarını (N) analiz edip gizli anahtar elde ederek gerçek bilgiye ulaşmasını zorlaştırmaktadır. Çünkü saldırgan gerçek bilgiye ulaşmak için genel anahtar oluşturan bileşenleri (N ve e) faktörlerine veya çarpanlarına ayırmak zorundadır. Bu bağlamda araştırmacılar büyük N asal sayılarını çarpanlarına ayırabilmek için çeşitli çalışmalar yapmaktadırlar. Ancak bu çalışmaların çoğu büyük sayıları faktörlere ayırma konusunda yetersiz kalmakta, ve bir geleneksel bilgisayar sisteminin benzer işlemi yapabilmesinin yıllar alacağı söylenmektedir. İşte bu nedenle 1994 yılında Peter Shor isminde bir matematikçi bir kuantum mekaniğinden yararlanan bir kuantum bilgisayarının SHOR algoritmasını kullanarak paralel olarak büyük sayıları etkili ve hızlı bir şekilde faktörlere ayırabildiğini ortaya koymuştur. Bu bağlamda 2001 yılında ilk deneme 15 sayısının faktörlere ayırma çabasıydı. Ancak daha büyük sayılara gelince bir kuantum bilgisayarının hala yeterli olup olmadığı henüz belirgin değildir. Bu nedenle bölüm üçte de örneklendirildiği gibi araştırmacılar son yıllarda N büyük sayılarını faktörlere ayırma gibi çabalara pek çok ilgi göstermemekte, ve RSA algoritmasında genel anahtar bileşenini oluşturan diğer parametreleri (örneğin, e) hedef alarak klasik bir bilgisayarla bu algoritmayı kırmayı amaçlamaktadırlar. Bu durum RSA gibi bir açık anahtar şifreleme yönteminin güvenliğini yalnız N sayısına bağlı olmadığını ima etmektedir. Bu sebeple araştırmacıların bir diğer kısmı ise RSA'nın güvenlik açıklarını kapatmak amacıyla çeşitli güçlendirme yöntemlerini geliştirmiş/geliştirmektedirler. Fakat bu tekniklerin çoğu, zaman ve alan maliyetine sebep olduğu için bir yandan geliştirilmiş RSA yapılarının dezavantajını göstermektedir. Ayrıca değiştirilmiş ve geliştirilmiş RSA yapılarının hala sadece karmaşık bir yapı sunduğu bu algoritmanın henüz güvende olmadığını ve matematiksel işlemlerle çözülebilir olduğunu göstermekle birlikte, RSA ile ilgili aşağıdaki güvenlik sorunları göz önünde bulundurulmasını gerektirmektedir:

- Saldırgan açık anahtarı kullanarak rastgele seçilen mesajları şifreleyebilir, ve kanal boyunca paylaşılan gizli mesajla karşılaştırarak şifrelenmiş mesaj hakkında bilgi elde edebilir.
- Klasik bilgisayarlarla bazın büyük N sayıları faktörlere ayırmak zor olabilir ve yerine kuantum bilgisayarları gibi paralel işlem yapabilecek teknolojilerden kullanılabilir, ancak son dönemlerde araştırmacılar tarafından sunulan yöntemlerde N sayılarını faktörlere ayırmaksızın da p ve q parametrelerini bulmak mümkün görülmektedir.
- RSA algoritmasında defalarca aynı anahtarla işlem yapılırken, ve alıcı da farklı şifrelenmiş mesajları aynı d gizli değeri ile açmaya çalışırken, saldırgan bu süreyi değerlendirebilir ve d ile ilgi bilgi elde ederek M mesajına ulaşması mümkündür.
- Standart RSA algoritmasında gönderici ve alıcı aralarında ek olarak herhangi bir doğrulama sistemi kullanılmamaktadır.
- RSA algoritmasının hesaplamalı bir yapıya sahip olması bu algoritmanın kuantum mekaniğini kullanan hesaplama sistemlerine karşın dayanıksız olduğunu göstermektedir.
- Aynı uzunlukta seçilen p ve q çarpanları hernekadar bilinmeyen n olarak gönderiliyor olsa da, saldırganın elinde geçtiğinde n sayısının $p=q$ uzunluğundaki sayılardan oluştuğu anlaşılabacaktır. Buda bir saldırganın faktörleme sürecinde n sayısının periyodunu ararken doğrudan p ve q uzunluğundaki sayıların aramasını sağlayabilir.

Gelecek çalışmalarda tüm kriptο güvenlik sınırlıkların göz önünde bulundurulması ve benzer konuları daha etkili bir şekilde araştırılması yanında, araştırmacıların, hesaplamalı kriptο yapılarını hızlı bir şekilde tehdit ettiği düşünülen kuantum mekaniği konusunda daha çok yatkın olmaları gerekmektedir. Ayrıca üzerinde RSA algoritması gibi açık anahtarlı şifreleme yöntemleri kullanan sistemlerin güvenli bir OSI tabanlı protokolünü ve uygun bir kimlik doğrulama sisteminin kullandığından emin olunmalıdır. Diğer yandan şifrelenmiş bir mesajın güvenli bir bulut ortamında saklanması, normal bireylerin kolaylıkla gerçek bilgiye ulaşamamalarını sağladığı düşünülmektedir. Ancak alıcı gerçek mesaja ulaşmak için bulutta konumlandırılmış bir şifreli mesajı her seferinde indirmek zorunda kalacaktır. Bu durum yine de bir saldırganın bulut ile alıcı arasındaki bilgi geçişlerini fark ederek gerçek bilgiye ulaşmasını mümkün kılması muhtemeldir. Bu nedenle bulutta yerleştirilen bir gizli

mesajı okuyabilmek için alıcının bu mesajı indirmeden şifreli mesajı nasıl okuyabileceğinin araştırılması daha sonraki çalışmalar için önerilmektedir.

Araştırma bulgularından, simetrik algoritmasında tarafların özel anahtarı kanal yolundan paylaşmadıkları sürece benzer yapıların her zaman asimetrik şifreleme yöntemlerinden kıyasla daha güvenilir bir şifreleme hizmeti sunduğu söylenebilmektedir. Fakat farklı bir özel anahtar kullanıldığı durumda tarafların özellikle uzak mesafeden bir araya gelerek yeni bir özel anahtar değeri üzerinde anlaşmaları zordur, veya tarafların uzak konumlardan bu anlaşmayı internet üzerinden yapmaları özel anahtarın genele açık bir şekilde paylaşmaları anlamına gelmektedir. Bu nedenle dördüncü bölümde taraflar arasında ilk seferlik kanal dışında, sonradan açık ağ üzerinden de farklı bir özel anahtar üzerinde güvenli ve hızlı bir şekilde anlaşılabilen bir önerilen hibrit şifreleme modeli üzerinde odaklanılmıştır. Bu modelde ilk defa kanal dışında bir özel anahtar üzerinde anlaşıldıktan sonra, sonraki defalarda mevcut özel anahtarın kanaldan paylaşılabilen bir $+R$ sayısı üzerinden de değiştirilebilir olduğu gösterilmiştir. Böylece mevcut özel anahtar değerini saldırganla paylaşmadan hem gönderici hemde alıcı tarafında güvenli bir şekilde değiştirilmesi mümkündür. Bu bağlamda H_0 hipotezi doğrudur. Ayrıca bu modelin uygulamada herhangi bir modülü ve üssel işleme tabi olmadığı ve zaman ve alan karmaşıklığının $O(1)$ sabit değeri ortaya koyduğu modelin daha verimli bir performans sergilediğini göstermektedir. Bu bağlamda H_1 hipotezi doğrudur. Çizelge 4.3'te özel anahtar değişimi için kullanılan bir $+R$ sabit koşulunda zaman ve alan karmaşıklığı $O(1)$ sabit bir değer olarak elde edilmiştir. Bu durum ele aldığımız önerilen hibrit modelinin özel anahtar değişimi maliyetinin diğer hibrit şifreleme yöntemleri ve asimetrik yöntemlerinden kıyasla daha az olduğunu göstermektedir. Bu bağlamda hipotez H_2 doğrudur. Kısaca, Önerilmiş Hibrit Modelinin uygulamada RSA standardına göre zaman ve bellek kullanımının kıyasla daha az ve sabit bir değere $O(1)$ sahip olduğu söylenebilmektedir. Bu durum Önerilen Modelin hız açısından ve dolayısıyla da güvenlik açısından verimlilik sağladığını ortaya koymaktadır. Ayrıca algoritma üs-alma gibi katmanlı bir işlem gerçekleştirmediğinden özellikle büyük sayılara gelince belli bir saldırma noktası bırakmamakta ve saldırgan kanaldan geçen veriyi elde etse de gerçek bilgiyi tahmin etmesi zorlaşacaktır.

KAYNAKÇA

- Aboud, S. J. & AL-Fayoumi, M. A. (2020). Efficient method for breaking RSA scheme. *Ubiquitous Computing and Communication Journal*, 4(2), 15-20.
- Aboud, S. J. (2009). Efficient method for breaking RSA scheme. *Ubiquitous Computing and Communication Journal*, 1-5.
- Ahmad, K., Verma, S., Kumar, N. & Shekhar, J. (2011). Classification of internet security attacks. In *Proceeding of the 5th National Conference INDIACom-2011Bharti Vidyapeeth's Institute of Computer Applications and Management*, 0973-7529.
- Allen, J. H. (2001). Cert system and network security practices. In *Proceedings of the Fifth National Colloquium for Information Systems Security Education*, George Mason University, Fairfax, VA USA, 22-24.
- Altundal, Ö. F. (2020, 18 Aralık). DdoS nedir? ne değildir?. <http://www.siberguvenlik.org.tr/makaleler/ddos-nedir-ne-degildir> adresinden erişildi.
- Ambedkar, B. R., Gupta, A., Gautam, P. & Bedi, S. S. (2011). An efficient method to factorize the RSA public key encryption. In *2011 International Conference on Communication Systems and Network Technologies*, 108-111.
- Andress, J. (2005). IPv6: The next internet protocol. *Login*, 30(2), 21-28.
- Application Vulnerability Trends Report (2014). 5 Ağustos 2020 tarihinde <https://www.trustwave.com/Resources/Library/Documents/CenzicApplication-Vulnerability-Trends-2014/> adresinden erişildi.
- Barker, W. C. & Barker, E. (2012). SP 800-67: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.
- Bensghir, T. K. (2008). Kurumsal bilgi güvenliği yönetim süreci. 15 Ekim 2020 tarihinde <https://www.erzincan.edu.tr/userfiles/file/stratejfdb/guvenlik.ppt> adresinden erişildi.
- Bicakci, K. & Tavli, B. (2009). Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks. *Computer Standards & Interfaces*, 31(5), 931-941.
- Bircan, C. (2014). Sosyal mühendislik saldırıları. 9 Temmuz 2020 tarihinde <https://www.bilgiguvenligi.gov.tr/sosyal-muhendislik/sosyalmuhendislik-saldirilari-3.html> adresinden erişildi.

- Bonaventure, O. (2011). *Computer Networking: Principles, Protocols and Practice*. Textbook Equity Edition.
- Boss, S. R. & Kirsch, L. J. (2007). The last line of defense: Motivating employees to follow corporate security guideliness. *In Proceedings of the 28th International Conference on Information Systems*, 103.
- Buchanan, W. & Woodward, A. (2016). Will quantum computers be the end of public key encryption. *Journal of Cyber Security Technology*, 1(1), 1-22.
- Bulgurcu, B., Cavusoglu, H. & Benbasat, I. (2010). Information security policy compliance: An empirical study of rationality-based beliefs and information security awareness. *MIS quarterly*, 34(3), 523-548.
- Campagna, M. & Xing, C. (2015). Quantum safe cryptography and security: an introduction, benefits, enablers and challenges, *ETSI*, 8.
- Can, Ö. & Akbaş, M. (2014). Kurumsal ağ ve sistem güvenliği politikalarının önemi ve bir durum çalışması. *TÜBAV Bilim Dergisi*, 7(2), 16-31.
- Canbek, G. & Sağiroğlu, Ş. (2006). Bilgi, bilgi güvenliği ve süreçleri üzerine bir inceleme. *Gazi Üniversitesi Politeknik Dergisi*, 9(3).
- Canbek, G. & Sağiroğlu, Ş. (2007). Bilgisayar sistemlerine yapılan saldırılar ve türleri: Bir inceleme. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 23(1), 1-12.
- Cavusoglu, H. & Raghunathan, S. (2004). Economics of IT security management: four improvements to current security practices. *Communications of the Association for Information Systems*, 14(1), 65-75.
- Chen, L., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., Perlner, R. & Smith-Tone, D. (2016). NIST: Report on post-quantum cryptography. *NIST*.
- Connolly, K. J. (2001). *Law of Internet Security and Privacy*. Panel Publishers.
- Çifci, H. (2012). *Her Yönüyle Siber Savaş*. Tübitak Popüler Bilim Kitapları. Ankara
- Davis, R. (1978). The data encryption standard in perspective. *IEEE Communications Society Magazine*, 16(6), 5-9.
- Demirez, K. (2011). *Linux Backtrack 5*. Türkiye: Nirvana Yayınları.
- Devi, S. G., Selvam, K. & Rajagopalan, S. P. (2011). An abstract to calculate big o factors of time and space complexity of machine code. *Chennai and Dr.MGR University Second International Conference on Sustainable Energy and Intelligent System (SEISCON 2011)*, 844-847.
- Diffie, W. & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644-654.
- Dowd, P. W. & McHenry, J. T. (1998). Network security: It's time to take it seriously. *Computer*, 31(9), 24-28.

- E. Frenkel. (2013). 8 Ağustos 2020 tarihinde http://www.slate.com/articles/health_and_science/science/2013/06/online_credit_card_security_the_rsa_algorithm_prime_numbers_and_pierre_fermat.html adresinden erişildi.
- Elawad, El M. M. O. & El Dawo, H. (2016). Overview: Importance of network security and future issues. *IOSR Journal of Computer Engineering*, 18(4), 78-87.
- Fırlar, T. (2003). Ağ güvenliği. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 7(1), 9-16.
- Funmilola, A. & Oluwafemi, A. (2015). Review of computer network security system. *Network and Complex Systems*, 5(5), 40-46.
- Gilbert, D. (2014). İnternet: Largest ever DDoS cyber attack hits US and European victims. 14 Eylül 2020 tarihinde <http://www.ibtimes.co.uk/largest-ever-ddos-cyberattack-hits-us-european-victims-1435973> adresinden erişildi.
- Goel, A. (2017). Encryption algorithm using dual modulus. In *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 1-4.
- Gupta, A. (2014). Computer network security issues. *International Research Journal of Management Science & Technology*, 5(9), 58-62.
- Halfond, W. G., Viegas, J. & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In *Proceedings of The IEEE International Symposium On Secure Software Engineering*, 1, 13-15.
- Hamdi, S. M., Zuhori, S. T., Mahmud, F. & Pal, B. (2014). A Compare between Shor's quantum factoring algorithm and General Number Field Sieve. In *2014 International Conference on Electrical Engineering and Information & Communication Technology*, 1-6.
- Houle, K. J. (2001). Trends in Denial of Service Attack Technology, Whitepaper.
- HSBC. (2014). Türkiye'ye Siber Saldırı Şoku. 25 Aralık 2020 tarihinde <http://www.milliyet.com.tr/hsbc-turkiye-ye-siber-saldiri-bilisim1969049/> adresinden erişildi.
- Information Technology Laboratory. (2013). Digital Signature Standard (DSS). *National Institute of Standards and Technology*. 15 Ocak 2021 tarihinde Digital Signature Standard (DSS) (nist.gov) adresinden erişildi.
- Islam, M. A., Islam, M. A., Islam, N. & Shabnam, B. (2018). A modified and secured RSA public key cryptosystem based on “n” prime numbers. *Journal of Computer and Communications*, 6(3), 78.
- Iswari, N. M. S. (2016). Key generation algorithm design combination of RSA and ElGamal algorithm. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1-5.
- Jaju, S. A. & Chowhan, S. S. (2015). A Modified RSA algorithm to enhance security for digital signature. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, 1-5.

- Jozsa, R. (Ed.) (1997). *Entanglement and Quantum Computation*. Oxford University Press.
- K. Burlu (2013). *Bilişimin Karanlık Yüzü* (4. Baskı). Nirvana Yayınları.
- Kartalopoulos, S. V. (2008). Differentiating data security and network security. *In 2008 IEEE International Conference on Communications*, 1469-1473.
- Kızılkoyun, F. (2014). 26 Aralık 2020 tarihinde <http://www.hurriyet.com.tr/gundem/27662013.asp> adresinden erişildi.
- Klein, A. (2002). Cross site scripting explained. 18 Aralık 2020 tarihinde Applied Cryptography Group | Stanford University adresinden erişildi.
- Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177), 203-209.
- Kozan, Ü. (2014). 26 Aralık 2020 tarihinde Enerji Bakanlığı: Borçlar silinmedi - Son Dakika Haber ([hurriyet.com.tr](http://www.hurriyet.com.tr)) adresinden erişildi.
- Kumar, R. (2016). Internet security mechanism. *International Research Journal of Management Science & Technology*, 7(12), 15-19.
- Kumar, S. N. (2015). Review on network security and cryptography. *International Transaction of Electrical and Computer Engineers System*, 3(1), 1-11.
- Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V. & Schl  ffer, M. (2009). Rebound distinguishers: Results on the full Whirlpool compression function. *In International Conference on the Theory and Application of Cryptology and Information Security*, 126-143.
- Lee, P. P., Bu, T. & Woo, T. (2009). On the detection of signaling DoS attacks on 3G/WiMax wireless networks. *Computer Networks*, 53(15), 2601-2616.
- Leek, T. (2014). Questions. 17 Ocak 2021 tarihinde <https://security.stackexchange.com/questions/49280/cryptography-behind-chip-based-credit-cards-smart-cards> adresinden erişildi.
- Lomonaco, S. J. (2000). A lecture on shor's quantum factoring algorithm. 20 Ocak 2020 tarihinde (PDF) Shor's Quantum Factoring Algorithm (researchgate.net) adresinden erişildi.
- Mathur, S., Gupta, D., Goar, V. & Kuri, M. (2017). Analysis and design of enhanced RSA algorithm to improve the security. *In 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICIT)*, 1-5.
- Miller, V. S. (1985). Use of elliptic curves in cryptography. *In Conference on the Theory and Application of Cryptographic Techniques*, 417-426.
- Minni, R., Sultania, K., Mishra, S. & Vincent, D. R. (2013). An algorithm to enhance security in RSA. *In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 1-4.

- Mitnick, K. D. & Simon, W. L. (2001). *The Art Of Deception: Controlling the Human Element of Security*.
- Mitnick, K. D. & Simon, W. L. (2002). *The Art Of Deception: Controlling the Human Element of Security*. Indianapolis: Wiley Publishing.
- National Institute of Standards & Technology (1993). Technology administration: Secure hash standard. *US Department of Commerce, Technology Administration, National Institute of Standards and Technology*, 180(1).
- Nielsen, M. A. & Chuang, I. L. (2011). *Quantum Computation and Quantum Information* (10. Baskı). New York: Cambridge University Press.
- Nordrum, A. (2016). Tech-Talk: IEEE spectrum. 14 Ocak 2021 tarihinde <http://spectrum.ieee.org/techtalk/computing/hardware/encryptionbusting-quantum-computerpractices-factoring-in-scalable-fiveatom-experiment> adresinden erişildi.
- Ouafi, K., Overbeck, R. & Vaudenay, S. (2008). On the security of HB# against a man-in-the-middle attack. *In International Conference on the Theory and Application of Cryptology and Information Security*, 108-124.
- Öğütçü, G. (2010). *E-dönüşüm Sürecinde Kişisel Bilişim Güvenliği Davranışı ve Farkındalığın Analizi*. Yüksek lisans tezi. Başkent Üniversitesi/Fen Bilimleri Enstitüsü. Ankara.
- Öztemiz, S. & Yılmaz, B. (2013). Bilgi merkezlerinde bilgi güvenliği farkındalığı: Ankara'daki üniversite kütüphaneleri örneği. *Bilgi Dünyası*, 14(1), 87-100.
- Pawar, M. V. & Anuradha, J. (2015). Network security and types of attacks in network. *Procedia Computer Science*, 48, 503-506.
- Pomerance, C. (1996). A tale of two sieves. *Notices of the AMS*, 43(12), 1473-1485.
- Pub, N. F. (2001). 197: Advanced encryption standard (AES). *Federal Information Processing Standards Publication*, 197(441), 0311.
- Rijmen, V. & Daemen, J. (2001). Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, 19-22.
- Rivest, R. L. (1992). The MD5 message-digest algorithm. *Massachusetts Institute of Technology Laboratory for Computer Science*, 1-21.
- Rivest, R. L., Shamir, A. & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126.
- Rouse, M. (2014). Network Security. 15 Ağustos 2020 tarihinde <http://searchsecurity.techtarget.com/definition/RSA> adresinden erişildi.
- Schneier, B. (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). *In International Workshop on Fast Software Encryption*, 191-204.
- Schneier, B. (2005). Twofish cryptanalysis rumors. *Schneier on Security*.

- Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. *In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, USA: IEEE Computer Society*, 124-134.
- Simmonds, A., Sandilands, P. & Van Ekert, L. (2004). An ontology for network security attacks. *In Asian Applied Computing Conference*, 317-323.
- Siponen, M. T., Pahlila, S. & Mahmood, A. (2007). Employees' adherence to information security policies: An empirical study. *In IFIP International Information Security Conference*, Springer, Boston, MA, 133-144.
- Stallings, W. (2006). *Cryptography and Network Security* (4. Baskı). India: Pearson Education.
- Stallings, W. (2010). *Cryptography and Network Security: Principles and Practice* (5. Baskı). United States: Prentice Hall Press.
- Steyn, B. (2012). 5 Ocak 2021 tarihinde <http://doctrina.org/How-RSA-Works-WithExamples.html> adresinden erişildi.
- Sultana, S. M. & Fouzia, M. F. (2015). A survey on network security mechanism. *International Journal of Engineering Research and Applications*, 1-7.
- Sun, H. M., Wu, M. E., Ting, W. C. & Hinek, M. J. (2007). Dual RSA and its security analysis. *IEEE Transactions on Information Theory*, 53(8), 2922-2933.
- Sushila & Sunita. (2014). Network security. *International Research Journal of Management Science & Technology*, 5(12), 4-7.
- Şahinaslan, E., Kantürk, A., Şahinaslan, Ö. & Borandağ, E. (2009). Kurumlarda bilgi güvenliği farkındalığı, önemi ve oluşturma yöntemleri. *Akademik Bilişim*, 9, 11-13.
- Tamimi, A. A. & Khalifa, J. (2010). *Computer Networks and Communications*. Al-Zaytoonah University: Jordan.
- Vazirani, U. (2012). Quantum Mechanics and Quantum Computation”, Lecture notes on the online course. 17 Ocak 2020 Free Online Course: Quantum Mechanics and Quantum Computation from edX | Class Central adresinden erişildi.
- Warkentin, M. & Willison, R. (2009). Behavioral and policy issues in information systems security: The insider threat. *European Journal of Information Systems*, 18(2), 101-105.
- Welch, D. & Lathrop, S. (2003). Wireless security threat taxonomy. *In IEEE Systems, Man and Cybernetics Society Information Assurance Workshop*, 76-83.
- Wikipedia. (30 aralık 2020). RSA numbers. https://en.wikipedia.org/wiki/RSA_numbers adresinden erişildi.
- Zargar, S. T., Joshi, J. & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys & Tutorials*, 15(4), 2046-2069.

ÖZGEÇMİŞ

Adı Soyadı : Mohammad Rahiq BAİGZAD

EĞİTİM DURUMU

Lise : Arabhane Lisesi, Maimanah, Faryab, Afganistan
Lisans : Balkh Üniversitesi, Bilgisayar Bilimleri Fakültesi, Mazar-e Şerif, Balkh, Afganistan
Yüksek lisans : Kastamonu Üniversitesi, İşletme Anabilim Dalı, Kastamonu, Türkiye

DENEYİM

ACTED : Ulusal Sosyal Yardımlaşma Organizasyonu
DAACAR : Ulusal Hizmet Ulaştırma Organizasyonu
Mehraban TV : Yerel TV

DİL BİLGİSİ

Özbekçe : Ana Dili
Farsça : İleri Seviye
Derice : İleri Seviye
Peştuca : İleri Seviye
İngilizce : İleri Seviye
Türkçe : İleri Seviye