T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

İKİ UÇ SİSTEM ARASINDA SİMETRİK VE ASİMETRİK ŞİFRELEME ALGORİTMALARI

G191210373 – Aysun ÇAĞ YILMAZKULAŞ

Bölüm	**:**	**BİLGİSAYAR MÜHENDİSLİĞİ**
Danışman	**:**	**Prof. Dr. İbrahim ÖZÇELİK**

2020-2021 Güz Dönemi

İçindekiler

Şekiller Dizini

Tablolar Dizini

SİMETRİK VE ASİMETRİK ŞİFRELEME ALGORİTMALARI

Özet

Bilgisayar ağlarında haberleşme güvenliğini sağlamak için şifreleme kullanılmaktadır. Bu nedenle bilgisayarlarda ya da bilgisayar ağlarında şifrelemenin önemi günümüzde gün geçtikçe artmaktadır. Bu çalışmada simetrik şifreleme algoritmaları ve asimetrik şifreleme algoritmaları hakkında bilgi verilmektedir. Sonrasında şifrelemede kullanılan anahtar boyutlarının analizi gerçekleştirilmiştir. Aynı zamanda şifreleme algoritmalarının performans kriterleri de incelenmiştir.

Anahtar Kelimeler: şifreleme, açık anahtar, özel anahtar, kriptografi

SYMMETRIC AND ASYMMETRIC ENCRYPTION ALGORITHMS

Abstract

Encryption has been used for providing communication security in computer networks. Therefore, the importance of this in computers or computer networks is increasing day by day. In this study, information is given about symmetric encryption algorithms and asymmetric encryption algorithms. Afterwards, the key dimensions used in encryption were analyzed. At the same time, performance criteria of encryption algorithms are also examined.

Keywords: encryption, public key, private key, cryptography

1. Giriş

Son yıllarda internet kullanımının yaygınlaşması bir takım güvenlik sorunlarını da beraberinde getirmiştir. Bunun başlıca sebepleri, internet'in açık bir sistem olması ve üzerinde dolaşan verinin gasp edilmeye uygun olmasıdır. İnternette alınan ve gönderilen veri paketleri birçok halka açık ağlardan geçer, bu da bu paketlere herkes tarafından ulaşmayı mümkün kılmaktadır. Son derece gizli bilgilerin internette dolaşması, önemli bir kaygı halini almaktadır [1-3]. Bu tür bilgileri korumak mümkün olmadıkça, internette iş yapmak veya gizli, şahsi yazışmalarda bulunmak asla güvenli olmayacaktır. Bilgi güvenliği; başkası tarafından dinlenme, bilginin değiştirilmesi, kimlik taklidi gibi tehditlerin ortadan kaldırılması ile sağlanmaktadır. Bilgi güvenliği sağlamada kullanılan temel araç kriptografidir. Kriptografi bilgi güvenliğini inceleyen ve anlaşılabileni anlaşılamaz yapan bir bilim dalıdır. Gizlilik, güvenilirlik, veri bütünlüğü, kimlik doğrulama, özgünlük ve inkâr edilemezlik gibi konular kriptografinin önemli çalışma alanlarıdır.

2. Temel Kriptoloji Terimleri

Kriptografi: Okunabilir durumdaki bir verinin içerdiği bilginin istenmeyen taraflarca anlaşılamayacak bir hale dönüştürülmesinde kullanılan yöntemlerin tümü ile ilgili bilim dalıdır.

Kriptoloji: Kriptografik metodların matematiksel temelleri ile ilgilenen bir matematik dalıdır. Çeşitli iletilerin, yazıların belli bir sisteme göre şifrelenmesi, bu mesajların güvenlikli bir ortamda alıcıya iletilmesi ve iletilmiş mesajın deşifre edilmesi yöntemlerinin tümünü kapsamaktadır.

Kriptoanaliz: Kriptografik algoritmaların açıklarını bulup ortaya çıkarma işlemleridir.

Plaintext (Düz-metin): Şifrelenecek mesaj plaintext olarak adlandırılır.

Şifreleme (Encription): Alıcının haricinde kimsenin okuyamayacağı şekilde veriyi kodlama işlemidir.

Ciphertext (Şifreli-mesaj): Şifrelenmiş mesaj ciphertext olarak adlandırılır.

Şifre Çözme (Decyrption): Şifrelenmiş verinin çözülüp eski haline getirilme işlemidir.

Şifreleme Algoritması: Veriyi şifrelerken ve çözerken kullanılan matematiksel metodlardır.

Anahtar (Key): Şifreleme ve çözme işlemleri için kullanılmaktadır.

3. Şifreleme Algoritmaları

Geçmişte kullanılan kriptoloji algoritmaları, bu algoritmaların gizliliğine dayanmaktaydı. Günümüzde kullanılmakta olan güçlü şifreleme algoritmaları gizli değildir. Bu algoritmaların farklı uzunlukları ve yapılarındaki anahtarlarla güvenlikleri sağlanmaktadır. Anahtarlar, bütün modern algoritmalarda şifrelmeyi ve şifre çözmeyi kontrol için kullanılırlar.

Bir gönderici alıcıya açık ağlar üzerinden bir ileti göndermek istediğinde, açık ağlardan gönderilen iletiler üçüncü şahıslar tarafından dinlenme ve değiştirilme tehdidi altındadır. Burada söz konusu ileti *plaintext* adı da verilen düz metindir. Bu iletinin içeriğini saklamak üzere gerçekleştirilen gizleme işlemi de şifrelemedir (*encryption*). Bu işlemde düz metin anahtar kullanılarak şifreli metine dönüştürülmektedir. Bu şekilde bilginin içeriği başkalarının anlamayacağı hale gelmektedir. Bu bilgi, bir yere iletilmek amacıyla şifrelenen bir mesaj veya gizlenmek amacıyla şifrelenen bir bilgi olabilir. Şifrelenmiş bu ileti şifreli metindir (*ciphertext*). Şifreli metini düz metine geri çevirme işlemi ise şifre çözümüdür (*decrypt*). Şekil 1'de bu süreç gösterilmektedir.

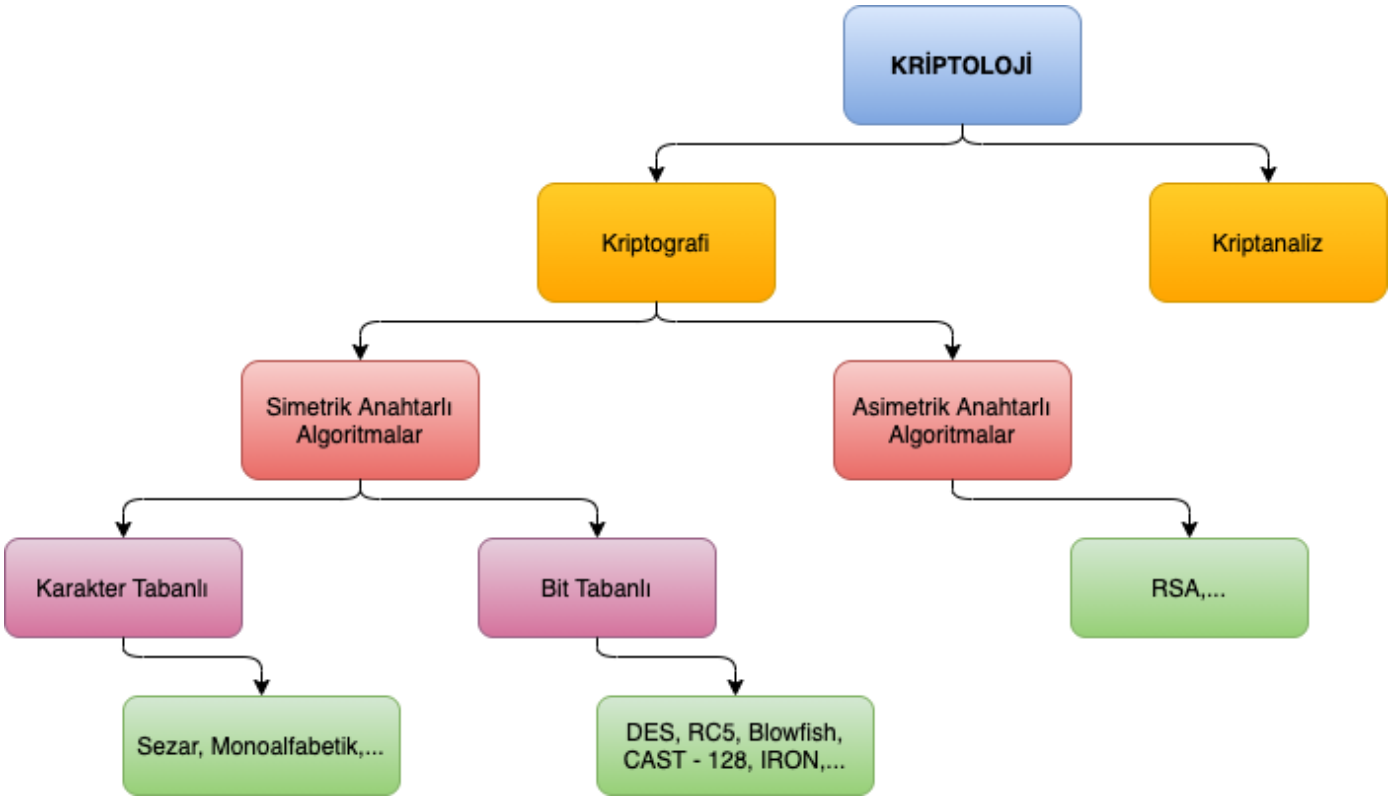
Şekil 3.1: Şifreleme ve şifreyi çözme işlemleri.

Anahtar kullanma yöntemlerine göre şifreleme algoritmaları genel olarak iki kategoriye ayrılmaktadır. Bunlar:

- Gizli anahtarlı (Simetrik) şifreleme algoritmaları

- Açık anahtarlı (Asimetrik) şifreleme algoritmaları

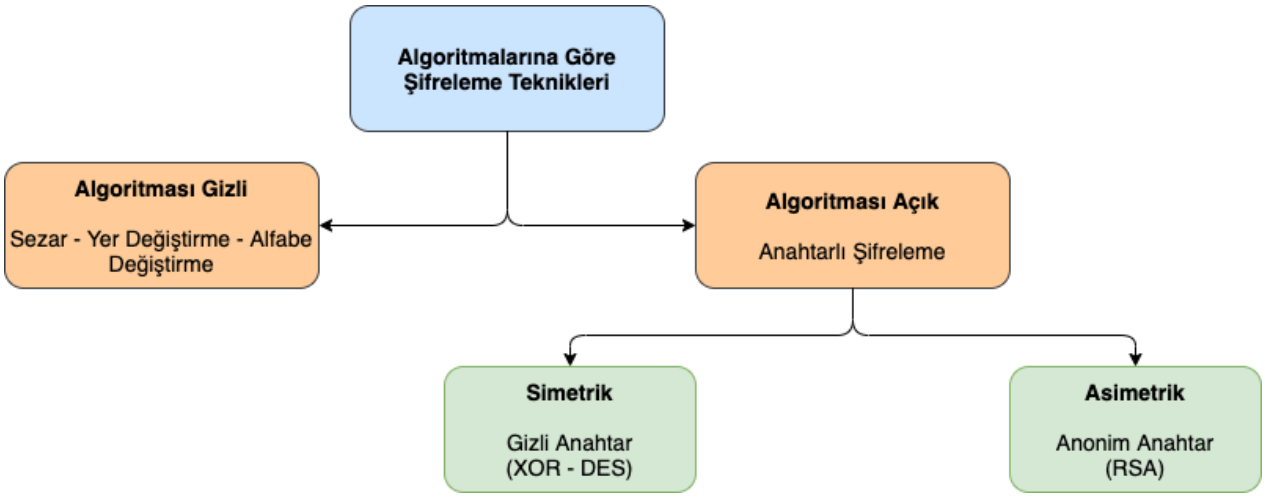
3.1. Algoritmaların Genel Sınıflandırması



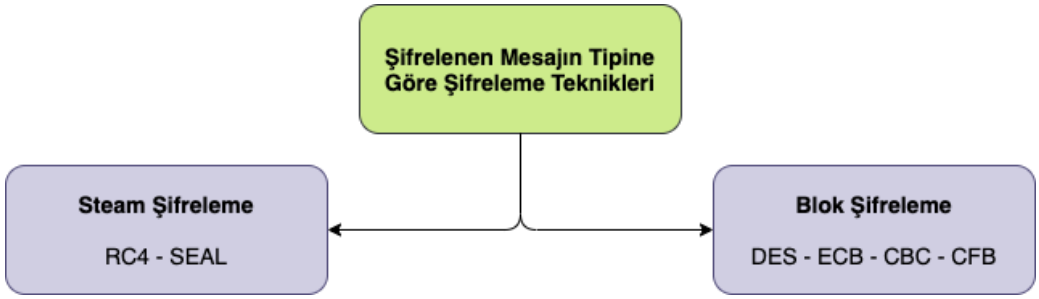
Şekil 3.2: Kriptoloji genel sınıflandırması.

Şifreleme algoritmaları sınıflandırılması aşağıdaki kriterlere göre ifade edilebilir:

- Algoritmanın Gizliliği / Açıklığı.
- Anahtar Sayısı.
- Şifrelenen Mesajın Tipi.



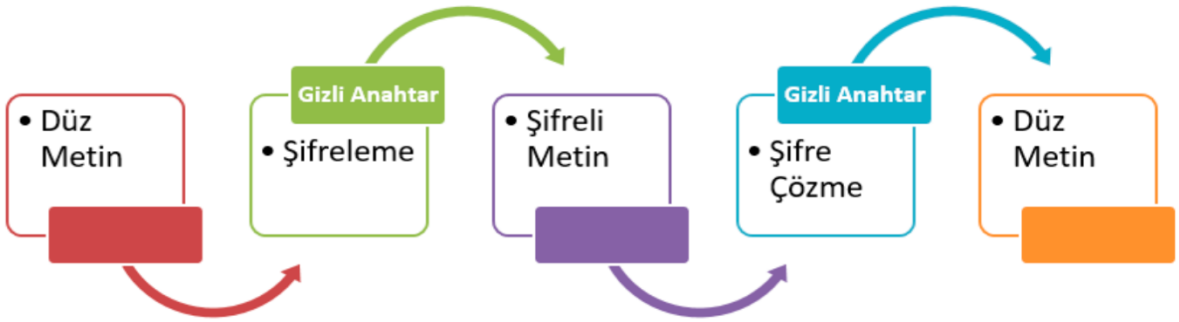
Şekil 3.3: Algoritmalarına Göre Şifreleme Sınıflandırması.



Şekil 3.4: Şifrelenen Mesajın Tipine Göre Şifreleme Sınıflandırması.

3.2. Simetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmaları şifreleme ve şifre çözme işlemleri için aynı anahtarı kullanma prensibine dayalı olarak çalışmaktadır. Bu durum veri şifreleme için matematiksel açıdan daha az problem çıkaran bir yaklaşımolup bu sebeple çok kullanılan bir yöntemdir. Bu tip algoritmalar kullanılırken, şifreleme işlemi gerçekleştirildikten sonra şifreli metin alıcıya gönderilirken bu metinle birlikte gizli anahtarı da alıcıya güvenli bir şekilde göndermek gerekmektedir. Simetrik şifreleme algoritmaları şifreleme ve şifre çözme işlemlerini çok hızlı bir şekilde gerçekleştirebilmektedir.



Şekil 3.5: Simetrik Şifreleme Algoritması Diyagramı.

Bu sistemin avantajı hızı, dezavantajı ise ortak anahtarların belirlenmesi ve taraflara iletilmesinde karşılaşılan problemlerdir.

Tablo 1’de çeşitli simetrik şifreleme algoritmaları hakkında bilgiler verilmiştir.

Algoritmanın Adı	Geliştiren	Tarihi	Tipi (Blok Uzunluğu)	Anahtar Uzunluğu	Döngü Sayısı	Çözülme Durumu	Kullanım Koşulları
DES (Data Encryption Standard)	IBM (ABD)	1977	Feistel Blok (64 bit)	56 bit (parity ile 64 bit)	16	Sağlam; 8 döngülü çeşidi çözülebiliyor; 16 zayıf anahtar	Serbest
IDEA (International Data Encryption Algorithm)	Lai-Massey, ETH Zurich (İsviçre)	1992	Blok (64 bit)	128 bit	8	Sağlam; 2^{51} zayıf anahtar	Ticari faaliyetler hariç serbest
RC2 (Rivest's Cipher veya Ron's Code2)	Rivest, RSA Data Security (ABD)	1992	Katar	2048 bite kadar	Bilinmiyor	Zayıflık bulunmadı	Algoritma RSA tarafından saklı tutuluyor
RC5 (Rivest's Cipher veya Ron's Code5)	Rivest, RSA Data Security (ABD)	1995	Blok (32, 64 veya 128 bit)	2048 bite kadar	255'e kadar	64 bit blok ve 12 döngü ile diferansiyel ve doğrusal şifre çözüme dayanıklı	Serbest
Blowfish	Bruce Schneier, Counterpane Systems (ABD)	1993	Feistel Blok (64 bit)	448 bite kadar	16	3 döngülü çeşidi diferansiyel şifre çözüme hassas	Ticari faaliyetler hariç serbest
FEAL (Fast Data Encipherment Algorithm)	Shimizu ve Miyaguchi (Japonya)	1988	Blok	FEAL-4 64 bit; FEAL-N 128 bit	FEAL-4 4 döngü; FEAL-N 31 döngü	Güvensiz; çeşitleri başarıyla çözümlendi	-
SAFER (Secure and Fast Encryption Routine)	Massey, Cylink Corporation (ABD)	1993	Blok (64 bit)	64 bit; 128 bit	10 döngüye kadar	İlk sürümlerinin anahtar açılımında zayıflıklar vardı	-
Skipjack (Clipper Chip)	NSA (ABD)	1993	Blok (64 bit)	80 bit	32	Algoritma gizli	Sadece özel entegre devre olarak bulunuyor
Lucifer	IBM (ABD)	1970?	Feistel Blok (64 bit)	128 bit	16	DES'in prototipi olduğundan zayıflıklar içermesi olasıdır	-
GOST 28147-89	I.A.Zabotin, G.P.Glazkov, V.B.Isaeva (Sovyetler Birliği)	1989	Feistel Blok (64 bit)	256 bit; 512 bit tanımlanabilir sübsitüsyon; 610 bit etken gizli bilgi	32	SSCB tarafından bütün gizlilik derecelerindeki bilgiler için uygun görülmüştür	Serbest
ASEKAL-21	Aselsan (Türkiye)	-	Doğrusal olmayan katar	57 bit ?	-	Ulusal olarak onaylanmış algoritma	Aselsan 2101, 2010 veri ve ses şifreleme birimlerinde kullanılıyor

Tablo 1: Bazı Simetrik Şifreleme Algoritmaları Bilgileri.

3.2.1. Bit Tabanlı Şifreleme Sistemleri

DES
(Data Encyripton Standard)

DES yapısı itibari ile bir blok şifreleme örneğidir. Basitçe, şifrelenecek olan açık metni parçalara bölerek (blok) her parçayı birbirinden bağımsız olarak şifreler, ardından şifrelenmiş metni açmak için de aynı işlemi bloklar üzerinde yapar. Bu blokların uzunluğu 64 bittir.

DES dünyada en yaygın kullanılan şifreleme algoritmalarından biridir ve IBM tarafından geliştirilmiştir. "Federal Register" tarafından 1975 yılında yayınlanmıştır. DES 64 bitlik veriyi 56 bitlik anahtar kullanarak şifreler. Ayrıca klasik Feistel Ağı'nı kullanarak temelde şifreleme işleminin deşifreleme işlemi ile aynı olmasını sağlar. Yayılma ve karıştırma teknikleri kullanılır. Anahtar uzunluğunun 56 bit olması DES'in en büyük dezavantajıdır. Bu algoritma günümüzde geliştirilen modern bilgisayarlar tarafından yapılan BruteForce saldırıları karşısında yetersiz kalmaktadır. Daha güvenli şifreleme ihtiyacından dolayı DES geliştirilerek, Triple- DES olarak değiştirilmiştir. Triple -DES algoritmasında geriye uyumluluğu da desteklemek amacıyla 2 adet 56 bitlik anahtar kullanılır.

Triple – DES
(Triple - Data Encyripton Standard)

1977'de IBM tarafından geliştirilip standart olarak kabul edilmiştir. Fakat 1997 yılında İsrail tarafından kırılmış bulunmaktadır. Bu şifreleme metodu çözülmüş olmasına rağmen günümüz bankacılık sektöründe halen kullanılmakta olan bir sistemdir. Triple-DES algoritması, DES algoritmasının şifreleme, deşifreleme, ardından tekrar şifreleme şeklinde

uygulanmasıdır. Standart DES 112 ya da 168 bitlik iki veya üç anahtar ile ardarda çalıştırılması ile oluşturulan bir şifreleme tekniğidir. Anahtar alanı 2^{112} veya 2^{168} sayısına ulaşınca bugün veya tahmin edilebilir bir gelecekte çözülmesi mümkün olmayan bir kod oluşmaktadır.

TWOFISH

1993 yılında yayınlanan bu algoritma Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall ve Niels Ferguson tarafından geliştirilmiş bir simetrik blok şifreleme algoritmasıdır. AES kadar hızlıdır. Ayrıca DES gibi Feistel yapısını kullanır. DES'den farklarından biri anahtar kullanılarak oluşturulan değişken S-box (Substitution box – Değiştirme kutuları)' lara sahip olmasıdır. Ayrıca 128 bitlik düz metni 32 bitlik parçalara ayırarak işlemlerin çoğunu 32 bitlik değerler üzerinde gerçekleştirir. AES'den farklı olarak eklenen 2 adet 1 bitlik rotasyon, şifreleme ve deşifreleme algoritmalarını birbirinden farklı yapmış, bu ise uygulama maliyetini arttırmış, aynı zamanda yazılım uygulamalarını %5 yavaşlatmıştır.

IRON

IRON da diğer iki algoritma gibi Feistel yapısını kullanır. Bu algoritma 64 bitlik veri bloklarını 128 bitlik anahtarla şifrelemede kullanılır. Döngü (round) sayısı 16 ile 32 arasındadır. Alt anahtarlar döngü sayısına bağlıdır. Alt anahtarların sayısı döngü sayısına eşittir. Bu nedenden ötürü algoritma anahtar bağımlıdır.

IRON algoritmasının var olan algoritmalardan farkı da budur. Bu algoritmanın avantajı bitler yerine 16- tabanındaki (hex) sayılar kullanmasıdır, dezavantajı ise yazılım için tasarlanmış olmasıdır.

AES
(The Advanced Encryption Standard)

AES, John Daemen ve Vincent Rijmen tarafından Rijndael adıyla geliştirilmiş, ardından 2002 yılında standart haline gelmiştir. AES uzunluğu 128 bitte sabit olan blok ve uzunluğu 128, 192 ya da 256 bit olan anahtarları kullanır. Kullanılan tekniklerden bazıları baytların yer değiştirmesi, 4x4' lük matrisler üzerine yayılmış metin parçalarının satırlarına uygulanan kaydırma işlemleridir. 2010 yılı itibariyle en popüler simetrik algoritmalardan biridir. Eğer bilgisayar 1 saniyede DES'i kırabilseydi, 128 bit AES anahtarı 149 trilyon yıl sonra kırılabilir.

3.2.2. RC4 Şifreleme

RC4 algoritması şifrelenecek veriyi akan bir bit dizisi olarak algılamaktadır. Belirlenen anahtar ile veriyi şifreleyen bir algoritmadır. RC4′ün başlıca özellikleri aşağıda belirtilmiştir:

- Genellikle hız gerektiren uygulamalarda kullanılır.
- Şifreleme hızı yüksektir ve MB/sn seviyesindedir.
- Güvenliği rastgele bir anahtar kullanımına bağlıdır.
- Anahtar uzunluğu değişkendir.
- 128 bitlik bir RC4 şifrelemesi sağlam bir şifreleme olarak kabul edilir.
- Bankacılık ve Dökümantasyon (PDF) şifrelemelerinde yaygın olarak kullanılır.

3.2.3. MD5 Algoritması (Message-Digest Algorithm 5)

MD5 (Message-Digest algorithm 5) Ron Rivest tarafından 1991 yılında geliştirilmiş tek yönlü bir şifreleme algoritmasıdır. Veri bütünlüğünü test etmek için kullanılan bu şifreleme algoritması girdinin büyüklüğünden bağımsız olarak 128-bit'lik bir çıktı üretir. Girdideki en ufak bir bit değişikliği bile çıktının tamamen değişmesine sebep olabilir. MD5'ın en sık kullanım alanı, bir verinin (dosyanın) doğru transfer edilip edilmediği veya değiştirilip değiştirilmediğinin kontrol edilmesidir.

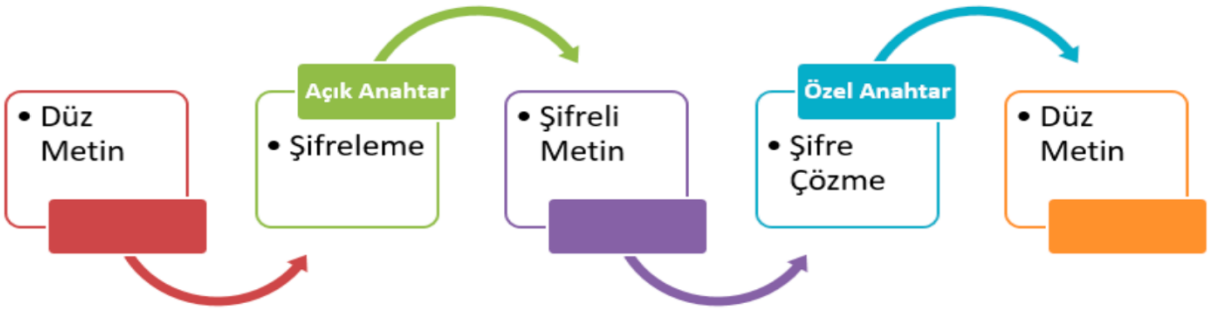
3.2.4. SHA Ailesi (Secure Hash Algorithm – Güvenli Özetleme Algoritması)

SHA (Secure Hash Algorithm – Güvenli Özetleme Algoritması), Amerika'nın ulusal güvenlik kurumu olan NSA tarafından tasarlanmıştır.

SHA-1, uzunluğu en fazla 264 bit olan mesajları girdi olarak kullanır ve 160 bitlik mesaj özeti üretir. Bu işlem sırasında, ilk önce mesajı 512 bitlik bloklara ayırır ve gerekirse son bloğun uzunluğunu 512 bite tamamlar. SHA-1 çalışma prensibi olarak R. Rivest tarafından tasarlanan MD5 özet fonksiyonuna benzer.160 bitlik mesaj özeti üreten SHA-1 çakışmalara karşı 80 bitlik güvenlik sağlar.

3.3. Açık Anahtar (Asimetrik) Şifreleme Algoritmaları

Açık anahtarlı şifreleme algoritmaları simetrik şifreleme algoritmalarından radikal bir farklılık göstermektedir. Bu tip şifreleme algoritmaları açık (public) ve özel (private) anahtar olmak üzere iki ayrı anahtar kullanmaktadır.



Şekil 3.6: Asimetrik Şifreleme Algoritması Diyagramı.

Asimetrik algoritmalar da denilen açık anahtarlı algoritmalarda şifreleme için kullanılan anahtar ile şifre çözme için kullanılan anahtar birbirinden farklıdır. Anahtar çiftlerini üreten algoritmaların matematiksel özelliklerinden dolayı açık-özel anahtar çiftleri her kişi için farklıdır, diğer bir deyişle her kullanıcının açık-özel anahtar çifti yalnızca o kullanıcıya özeldir. Ayrıca şifre çözüm anahtarı (en azından makul bir zaman dilimi içerisinde) şifre anahtarından hesaplanamaz. Bu algoritmalara açık anahtarlı algoritmalar denmesinin sebebi şifre anahtarının halka (kamuya/genel kullanıma) açık olmasıdır. Bir yabancı bir iletiyi şifrelemek için şifreleme anahtarını kullanabilir, ancak sadece ilgili şifre çözüm anahtarına sahip bir kişi iletinin şifresini çözebilir. Bu sistemde, şifre anahtarına genellikle açık anahtar adı verilmektedir. Şifre çözüm anahtarı da genellikle özel anahtar olarak adlandırılmaktadır. Özel anahtar kimi zaman gizli anahtar olarak da adlandırılır, ancak simetrik algoritmalarla karışmaması için bu terim genelde kullanılmamaktadır.

Bir kullanıcının açık anahtarıyla şifrelenen bir mesajı, yalnız ve ancak ona ait özel anahtar çözebilmektedir. Aynı şekilde, herhangi bir kullanıcının özel anahtarıyla attığı sayısal imzanın doğrulanabilmesi, yalnızca o kullanıcının açık anahtarını kullanarak mümkün olabilmektedir. Açık anahtar kamuya açıktır, elektronik kimlik belgelerinin içinde diğer kişisel bilgilerle birlikte tutulur ve herkes birbirinin açık anahtarını e-kimliklerine ulaşmak suretiyle istediği zaman elde edebilir.

3.3.1. RSA

Dünyada en yaygın biçimde kullanılan asimetrik algoritma, ismini mucitlerinin baş harflerinden almıştır(**R**onald L.Rivest, Adi **S**hamir ve Leonard **A**dleman).

Büyük sayıların modüler aritmetiğine dayalı çok basit bir prensibi vardır. Anahtarlar, iki büyük asal sayıdan üretilir. Dolayısıyla, algoritmanın güvenliği büyük sayı üretme problemine dayalıdır.

Şifreleme :	**$y = x^e \pmod n$**
Şifre Çözme :	**$x = y^d \pmod n$**
	$n = p*q$
x : Açık Metin	**e : Açık Anahtar**
y : Şifreli Metin	**d : Gizli Anahtar**
n : Açık Modül	**p ,q : Gizli Asal Sayılar**

Şekil 3.7: RSA Şifreleme ve Çözme.

	RSA Anahtar Elde Etme	Örnek
1	İki asal sayı seçilir. (p ve q)	p=3 \ q=11
2	Açık Modül (Public Moduls) hesaplanır.	n=p*q=33
3	Anahtar üretimi için ara bir değişken hesaplanır. (z)	z=(p-1)*(q-1)=2*10=20
4	Açık anahtar "e" aşağıdaki şekilde hesaplanır.	
	e < z ve gcd(z,e)=1 , yani z ve e'nin en büyük ortak böleni 1'dir. Bu özelliği sağlayan birden fazla sayı olabileceğinden biri seçilir.	e=7
5	Gizli anahtar d hesaplanır. (d*e) mod = 1	d=3

Şekil 3.8: RSA Anahtar Elde Etme Örneği.

	RSA Kullanım Örneği	
1	Açık metin 4 olsun.	x = 4
2	Şifrelenir.	$y = 4^7 \pmod{33} = 16$
3	Şifre çözülür.	$x = 16^3 \pmod{33} = 4$

Şekil 3.9: RSA Kullanım Örneği.

3.3.2. RSA Algoritması Neden Önemlidir?

Çeşitli problemlere uygulanmaya başlanan RSA algoritması, birçok gerçek dünya problemlerinde ve mühendislik alanında kullanılmaktadır. Özellikle son yıllarda RSA algoritması ile ilgili çok sayıda çalışma bulunmaktadır.

- Lee ve Chang (1998) şifreleme işlemlerinin hızlı gerçekleşmesini sağlamak için küçük ortak anahtarın dinamik olarak büyütülmesini sağlayan bir yöntem önermişlerdir.
- Somani vd. (2010) Bulut bilişimde verilerin güvenliğini geliştirmek için RSA algoritmasını ve dijital imza algoritmasını önermişlerdir.
- Dubey vd. (2012) Bulut kullanıcısı ile bulut sağlayıcısı için MD5 ve RSA algoritmasını kullanarak verilerin toplanması ile paylaşılması için güvenli bir sistem tasarladıklarını böylece güvenilir hesaplamaların yapılabileceğini söylemişlerdir.
- Patidar ve Bhartiya (2013) RSA algoritmasının hızlandırılması için modifiye edilmiş RSA algoritmasını önermişlerdir. Önerilmiş olan algoritmada iki asal sayı yerine üç asal sayı kullanarak hız ve güvenlik üzerine çalıştıklarını söylemişler ve orijinal RSA ile kıyaslamışlardır. Kıyaslama sonuçlarına göre önerilen algoritmanın daha güvenilir olduğunu söylemişlerdir.
- Ayele ve Sreenivasarao (2013) RSA için iki ortak anahtar içeren bir yöntem önermişlerdir. İki ortak anahtarın ayrı olarak gönderilmesi saldırganın anahtar hakkında fazla bilgi sahibi olmamasına ve metinin şifresinin çözülmemesine sebep olduğunu söylemişlerdir. Önerilen RSA'nın daha az hız ile yüksek güvenlik sağladığını söylemişlerdir.
- Jaiswal vd. (2014) Ağ üzerinden veri alışverişi sırasında hesaplanma zamanın hızlanması ve güvenirliğinin daha iyi olması için modifiye edilmiş olan RSA algoritmasını

önermişlerdir. Orijinal RSA ile önerilen RSA ’nın karşılaştırılması sonucunda önerilmiş olan RSA’nın daha iyi güvenlik sağladığı ve işlem hızının arttırıldığı sonucuna varmışlardır.

- Thangavel vd. (2015) ESRKGS adı verilen modifiye edilmiş ve geliştirilmiş bir RSA anahtar üretim algoritması önermişlerdir. Önerilmiş olan algoritmada iki asal sayı yerine dört büyük asal sayı kullanmışlardır. Yapmış oldukları deney sonuçlarına göre önermiş oldukları algoritmanın yüksek oranda güvenli ve kolay kırılamayacağını kanıtladıklarını söylemişlerdir.
- Çavuşoğlu vd. (2017) RGN ve RSA algoritmasının birlikte kullanılmasıyla oluşan kaos tabanlı hibrit RSA (CRSA) şifreleme algoritması tasarlamışlardır. Oluşturulan bu algoritmayla metin ve görüntü şifrelemesinin yapıldığını ifade etmişlerdir. Yaptıkları güvenlik analiz sonuçlarının klasik RSA ile karşılaştırılması sonucunda önerdikleri algoritmanın daha iyi sonuçlar verdiğini görmüşlerdir.
- El Makkaoui vd. (2017) bulut bilişiminde veri gizliğinin sağlanması ve şifre çözme işlemini hızlandırmak için hızlı bulut-RSA’yı önermişlerdir. Elde ettikleri simülasyon sonuçlarına göre hızlı-RSA’nın çalışma zamanında iyi bir performans sergileyerek öngörülen güvenlik seviyesini sağladığını belirtmişlerdir.
- Stergiou vd. (2018) IoT ve cloud teknolojilerinin entegrasyonunda AES ve RSA algoritmalarını kullanmışlardır. Şifreleme işlemlerinde RSA algoritmasının kullanılmasıyla IoT’un işlevinde daha yüksek düzeyde iletişim güvenliğinin sağlanabileceği sonucuna varabildiklerini belirtmişlerdir.
- Liu vd. (2018) RSA algoritmasını kullanarak güvenli ve sağlam bir dijital görüntü şeması filigran modeli önermişlerdir. Gizli verilerin güvenliğini garanti altına almak için

asimetrik şifreleme algoritmalarından biri olan RSA algoritmasını kullanmışlardır. Yapmış oldukları deney sonuçlarına göre önerdikleri yöntemin daha iyi sağlamlık, daha az şifreleme süresi ve büyük veri gömme kapasitesine sahip olan diğer yaklaşımlara göre daha iyi performans göstermiş olduğunu aktarmışlardır.

- Taha vd. (2018) mobil bulut bilişim siteminde veri güvenliğinin sağlanması için hibrit RSA algoritmasını önermişlerdir. Sonuçlara göre veri güvenliğinin arttığını ve veriyi şifrelemek için harcanan sürenin azaldığını söylemişlerdir.
- Subhashini ve Srivaramangai (2018) tarafından bulut bilişim ile ilgili yaptıkları çalışmada bulutta bulunan verilerin güvenli bir şekilde korunmadığı taktirde verilerin risk altında olabileceğini söylemişlerdir. Bundan dolayı bulut sistemde güvenliği üst düzeyde tutmak için çeşitli kriptografik algoritmaların olduğunu söylemişler ve bu algoritmaların iyileştirilmesi ile ilgili genel bir bakış sunarak tartışma gerçekleştirmişlerdir.
- Palathingal vd. (2018) bulut sistemlerinde verilerin güvenirliğinin sağlanması için steganografi yöntemini kullanmışlardır. Sistemdeki verilerin daha güvenli olması için RSA algoritmasını diğer algoritmalarla entegre ettiklerini ve bulut bilişim sisteminde verilerin güvenliği için güçlü bir yapı olduğunu vurgulamışlardır.

RSA algoritması ile literatürde yer alan AES ve DES algoritmalarının bazı özelliklerinin karşılaştırılması Tablo 2'de verilmiştir.

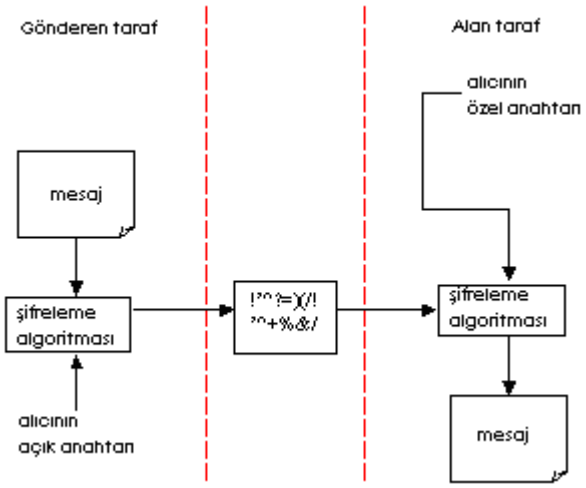
Özellikler	AES	DES	RSA
Tarihi	1997	1977	1978
Anahtar uzunluğu	128, 192, 256 bit	56 bit	>1024 bit
Blok uzunluğu	128 bit	64 bit	En az 512 bit
Şifreleme ve şifre çözme anahtarı	Aynı	Aynı	Farklı
Algoritma türü	Simetrik şifreleme algoritması	Simetrik şifreleme algoritması	Asimetrik şifreleme algoritması
Şifreleme hızı	Hızlı	Orta	Yavaş
Şifre çözme hızı	Hızlı	Orta	Yavaş
Güç tüketimi	Düşük	Düşük	Yüksek
Güvenirliği	Güvenli	Yeteri kadar güvenli değil	Güvenli
Kullanılan anahtar	Şifreleme ve şifreyi çözmek için kullanılan anahtar aynı	Şifreleme ve şifreyi çözmek için kullanılan anahtar aynı	Şifreleme ve şifre çözmek için kullanılan anahtar farklı
Devir sayısı	10, 12, 14	16	1
Başlama hızı	Hızlı	Hızlı	Hızlı
Truva atı	Kanıtlanmamış	Yoktur	Yoktur

Tablo 2: AES, DES ve RSA Algoritmalarının Özelliklerinin Karşılaştırılması.

4. Şifreleme İşlemi

Şifreleme açık ağlardan gönderilen bilginin dış kullanıcılar tarafından görülmesinin istenmediği zaman yapılmaktadır. Bu işlem için çift anahtarlı bir şifreleme algoritması kullanılabilir. Buna göre, mesajı gönderen taraf, gönderilen bilginin sayısal içeriğini, mesajı alacak tarafın açık anahtarını, sayısal şifrelemede kullanmaktadır. Mesajı alan taraf da, şifreli mesajı çözmek için şifreli mesajın sayısal içeriği ve kendisinin özel anahtarına gereksinim duymaktadır.

Burada dikkat edilecek olursa, şifreli mesajın üçüncü taraflar tarafından dinlenebilmesi ancak "özel anahtara" sahip olmaları ya da şifreli mesajı matematiksel yollarla deşifre etmeye çalışmaları ile mümkün olabilmektedir. "Güvenlik açısından iyi bir şifreleme" algoritması, özel anahtar olmadan şifreli mesajı deşifre etmeye imkân tanımayan bir algoritmadır.



Şekil 4.1: Şifreli mesaj gönderilmesi ve alınması.

5. Simetrik ve Asimetrik Şifrelemenin Avantaj ve Dezavantajları

Açık anahtarlı şifrelemenin öncelikli avantajı daha fazla güvenlik sağlamasıdır. Bunun sebebi gizli anahtarın herhangi bir şekilde taşınması durumunun söz konusu olmamasıdır. Aksine gizli anahtarlı yapılarda gizli anahtarın el ile ya da iletişim kanalları üzerinden iletilmesi söz konusudur. Bu da gizli anahtarın istenmeyen kişilerce elde edilme olasılığını arttırmaktadır.

Reddedilemez sayısal imzalar oluşturabilmesi açık anahtarlı yapıların diğer bir önemli avantajıdır. Gizli anahtarlı yapılar kullanılarak yapılan kimlik denetiminde gizli bir bilginin paylaşılması ve bazı durumlarda üçüncü bir kişiye güven duyulması gerekliliği vardır. Bu durumda taraflardan biri, anahtarın diğerlerince kötü niyetle kullanıldığını iddia edebilmektedir. Ancak açık anahtarlı yapılarda böyle bir durum söz konusu değildir. Bu özellik reddedilemezlik olarak adlandırılmaktadır.

Algoritmalar farklı şifreleme ve şifre çözme anahtarı kullanırlar. Yani, şifreleme anahtarı buna karşılık gelen şifre çözme anahtarından farklı olabilir.

Açık anahtarlı yapıları kullanmanın dezavantajlarından biri de şifreleme hızıdır. Çoğu gizli anahtarlı yapı açık anahtarlı yapılara göre daha hızlıdır. İki yapıyı birlikte kullanmak en güvenli ve en hızlı yöntemdir.

Açık anahtarlı şifrelemede, onay kurumuna yapılan başarılı bir saldırı sonucu, herhangi bir kullanıcının açık anahtarı yerine istenilen açık anahtar koyularak, bu kullanıcıya gönderilen mesajlar elde edilebilmekte, mesajlar değiştirilerek kullanıcıya, kullanıcının kendi açık anahtarıyla şifrelenerek gönderilebilmektedir.

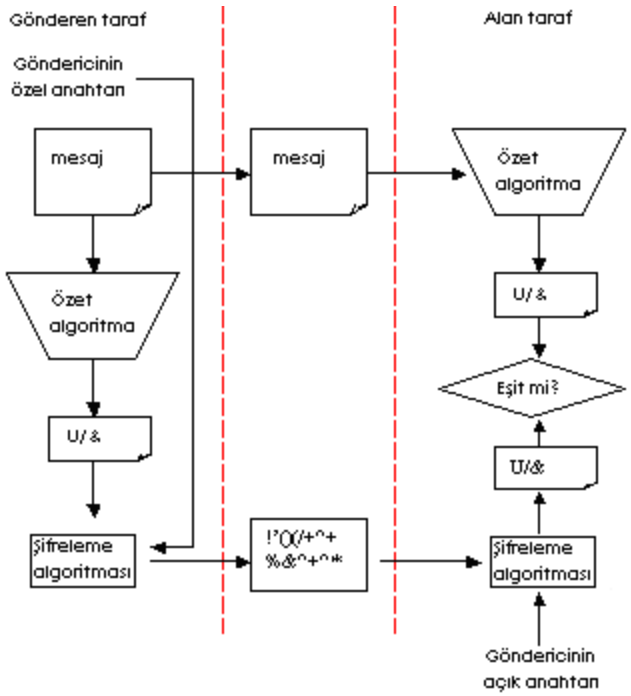
Bazı durumlarda gizli anahtarlı şifreleme tek başına yeterlidir. Gönderici ve alıcı kişiler yüz yüze görüşerek anahtar üzerinde anlaşabilirler. Bütün anahtarları bilen ve yöneten bir otorite bulunduğu durumlarda, açık anahtarlı şifreleme önemini yitirmektedir. Ancak kullanıcı sayısı arttığında bu da ayrı bir problem olabilmektedir.

Tek kullanıcının bulunduğu bir ortamda açık anahtarlı yapılar çok anlamlı değildir. Örneğin kişisel dosyalarınızı şifreli saklamak isterseniz, istediğiniz herhangi bir gizli anahtar algoritmasıyla kendi kişisel şifrenizi anahtar olarak kullanarak şifreleme yapabilirsiniz. Genel olarak açık anahtarlı yapılar, çok kullanıcılı açık ortamlar için idealdir.

Açık anahtarlı yapılar, gizli anahtarlı yapıların yerine geçmeye aday değil, daha çok onları daha güvenli hale getirecek tamamlayıcı unsurlardır. Örneğin, gizli anahtarları açık ağlar üzerinden taşımak için açık anahtarlı şifreleme kullanılır.

6. Sayısal İmza

Sayısal imza elektronik mesaja eklenmiş bilgidir. Çift anahtarlı bir şifreleme algoritmasıyla hazırlanan sayısal imza, hem gönderilen bilginin sayısal içeriğinin değiştirilmediğinin hem de gönderen tarafın kimliğinin ispatlanması için kullanılır ve gönderilecek mesajdan üretilen "mesaj özetinin" sayısal içeriği, gönderen tarafın kendi özel anahtarına bağlı olarak oluşturulur. Sayısal imzanın doğruluğunu kanıtlamak için mesajı alan taraf, kendisine gelen mesajın ve sayısal imzanın sayısal içeriği ile gönderen tarafın açık anahtarını kullanmaktadır. Şekil 6.1.'de bu durum gösterilmiştir.



Şekil 6.1 Sayısal imzalı mesaj gönderilmesi ve alınması.

"Mesaj özeti", gönderilecek mesajdan matematiksel yollarla üretilen sabit uzunlukta sayısal bilgidir. Bu işlem "hash" fonksiyonu olarak bilinir. Hash fonksiyonu bir kaç özelliği sağlar:

- Mesaj özeti anlamsız bir bilgidir.
- Hash fonksiyonu geri dönüşümü olmayan bir fonksiyondur. Diğer bir deyişle, herhangi bir mesajın özetine bakarak mesajın kendisini elde etmek mümkün değildir.
- Aynı özeti veren herhangi iki farklı mesaj bulmak da mümkün değildir.

Böylelikle, her mesajın farklı bir özeti olması ve dolayısıyla mesajda yapılacak en ufak bir değişikliğin imzayı geçersiz kılması sağlanmış olur. Sayısal imzalamada son adım, mesaj özetinin gönderen tarafın özel anahtarıyla şifrelenmesidir. Sayısal imza mesaja eklenir ve mesaj ile birlikte alıcıya gönderilir. Alıcının imzanın geçerliliğini kontrol etmesi iki adımda gerçekleşmektedir. Alıcı sayısal imzayı karşı tarafın açık anahtarı ile çözerek varsayılan mesaj özetini elde eder. Diğer yanda mesajın tekrar özetini çıkarır. Son olarak bu iki özeti karşılaştırır. Bu özetlerin tıpatıp aynı olması, imzanın doğruluğunu gösterir.

Açık-anahtar şifreleme için pek çok algoritma bulunmaktadır. En yaygın olan iki tanesi RSA (Ron Rivest, Adi Shamir, Leonard Adleman) algoritması ve DSA'dır (Digital Signature Algorithm - Dijital İmza Algortiması). RSA, pek çok uygulamada kullanılan bir algoritmadır. Mesajları şifrelemek için kullanılabileceği gibi dijital imzalarda da kullanılabilmektedir. DSA, sadece dijital imza kullanımı içindir. Mesajları şifrelemek için kullanılmamaktadır.

7. Şifrelemede Kullanılan Anahtar Boyutları

40 bitlik bir anahtar için $n=2^{40}$ veya n=1 099 511 627 776 (bir trilyon doksan dokuz milyar beş yüz on bir milyon altı yüz yirmi yedi bin yedi yüz yetmiş altı) olası anahtar söz konusudur. 1995'de yapılan bir yarışmada RC4 algoritması ile 40 bitlik bir anahtarla şifrelenmiş internet üzerinden yapılan bir kredi kartı işlemi, elinde sadece mütevazi bir bilgisayar laboratuarı olan bir öğrenci tarafından 3 buçuk saatte çözülmüştür.

Anahtarın deneme-yanılma yöntemiyle bulunmasını engellemek için, bugünkü süper bilgisayarlardan milyonlarca kat daha hızlı çalışan bir bilgisayarla bile milyarlarca yıl sürmesi için, kullanılan anahtarların uzunluğunun mümkün olduğunca büyük olması gerekmektedir.

Tablo 3'te farklı anahtar boyları için, saniyede bir milyon, bir milyar ve bir trilyon şifre deneyebilen bilgisayarlar için anahtar çözme süreleri verilmiştir. Tablo 4'te ise asimetrik şifreleme algoritması olan RSA şifreleme algoritması için kullanılan anahtar çiftlerinin farklı boyutlardaki oluşma süreleri ve şifreleme süreleri verilmiştir.

Anahtar Uzunluğu (n)	**Olası değer sayısı (2^n)**	**10^6 şifre/s hızında ortalama çözme süresi**	**10^9 şifre/s hızında ortalama çözme süresi**	**10^{12} şifre/s hızında ortalama çözme süresi**
32 bit	$\sim 4 \times 10^9$	36 dak	2.16 s	2.16 ms
40 bit	$\sim 10^{12}$	6 gün	9 dak	1 s
56 bit	$\sim 7.2 \times 10^{16}$	1142 yıl	1 yıl 2 ay	10 saat
64 bit	1.8×10^{19}	292 000 yıl	292 yıl	3.5 ay
128 bit	1.7×10^{38}	5.4×10^{24} yıl	5.4×10^{21} yıl	5.4×10^{18} yıl

Tablo 3: Farklı anahtar boyutları için anahtar çözme süreleri

Bit sayısı	Anahtar oluşturma süresi (sn)	Şifreleme (sn)
64	0.021	0.011
128	0.026	0.013
256	0.083	0.015
512	0.307	0.018
1024	2.985	0.106
2048	50.432	0.766
4096	798.625	18.687

Tablo 4: RSA algoritmasında farklı bit uzunluklarında anahtar oluşturma ve şifreleme süreleri.

8. Simetrik ve Asimetrik Şifreleme Algoritmalarının Genel Özellikleri

Simetrik ve asimetrik şifreleme algoritmalarının bazı önemli özellikleri Tablo 5'te özetlenmiştir. Tablo 6'da ise iki algoritmanın özellikleri karşılaştırılmıştır. İki algoritmayı birbirinden ayırmak için, simetrik şifrelemede kullanılan anahtarı gizli anahtar (secret key) olarak, asimetrik şifrelemede kullanılan anahtarları ise, genel anahtar (public key) ve özel anahtar (private key) olarak adlandırılmaktadır. Özel anahtar daima gizli tutulur fakat simetrik şifrelemede kullanılan anahtarla karıştırılmaması için gizli anahtar'dan ziyade özel anahtar olarak adlandırılır.

Simetrik şifreleme algoritmaları	Asimetrik şifreleme algoritmaları
Aynı algoritma ve aynı şifreleme anahtarı hem şifreleme hem de şifre çözmede kullanılır.	*Şifreleme ve şifre çözmek için bir algoritma fakat şifreleme ve şifre çözme için farklı anahtarlar kullanılır*
Gönderici ve alıcı aynı algoritmayı ve aynı anahtarı kullanır.	*Gönderici alıcının açık anahtarını bilmelidir. Gönderici ile alıcının anahtar çiftleri birbirinden farklıdır.*
Şifreleme için kullanılan algoritma gizli tutulmalı	*İki anahtardan biri gizli tutulmalı diğeri erişime açık olmalıdır.*
Algoritma bilgisi ve şifreli metin örnekleri anahtarı belirlemede yeterli olmamalı	*Algoritma bilgisi, anahtarlardan birinin ve şifreli metin örnekleri,diğer anahtarı belirlemede yeterli olmamalı*

Tablo 5: Simetrik ve asimetrik şifreleme algoritmalarının genel özellikleri.

Özellik	Simetrik şifreleme algoritmaları	Asimetrik şifreleme algoritmaları
Gizlilik	*Sağlamaktadır*	*Sağlamaktadır*
Bütünlük	-	*Sağlamaktadır*
Kimlik doğrulama	-	*Sağlamaktadır*
Inkar edilememezlik	-	*Sağlamaktadır*
Performans	*Hızlı*	*Yavaş*
Güvenlik	*Anahtar uzunluğuna bağlı*	*Anahtar uzunluğuna bağlı*

Tablo 6: Simetrik ve asimetrik şifreleme algoritmalarının özelliklerinin karşılaştırılması

9. Sonuç

Bu çalışmada simetrik ve asimetrik şifreleme algoritmaları incelenmiştir. Simetrik şifreleme algoritmaları tek bir anahtar kullanarak şifreleme ve şifre çözme işlemlerini geçekleştirmektedir. Bu algoritmalarda metin şifrelendikten sonra alıcıya bu şifreli metin gönderilirken, alıcıya gizli anahtarın da güvenli bir şekilde iletilmesi gerekmektedir. Bu durum simetrik şifreleme algoritmalarının en büyük dezavantajıdır. Asimetrik şifreleme algoritmalarında bu problem söz konusu değildir. Asimetrik şifreleme algoritmaları sayesinde alıcı ve gönderici taraflar kendilerine ait gizli anahtar oluşturabilirler ve verilerini bu anahtarla şifreleyebilirler. Asimetrik şifreleme algoritmaları çözülmesi zor matematiksel hesaplamalar üzerine kurulmuş algoritmalardır.

Asimetrik şifreleme algoritmalarının da dezavantajları bulunmaktadır. Bu algoritmaların güvenliğini sağlayabilmek için çok büyük asal sayılar kullanılmaktadır. Bu da zaman açısından çok büyük problemleri beraberinde getirmektedir.Asimetrik şifreleme algoritmalarını kullanan sistemler simetrik şifreleme algoritmalarını kullanan sistemlere göre çok daha yavaştır. Ayrıca asimetrik şifreleme algoritmalarının çok büyük sayılar kullanmasından dolayı donanımsal yapılara uyum sağlaması çok zor olmaktadır. Bundan dolayı sistemlerin hem simetrik hem de asimetrik şifreleme algoritmalarını birlikte kullanarak, simetrik şifreleme algoritmalarının dezavantajı olan gizli anahtar güvenliği problemini ve asimetrik şifreleme algoritmalarının hız problemini ortadan kaldırabilmektedir.

Bir şifreleme algoritmasının performansı şu kriterlere göre belirlenebilir:

- Sistemin kırılabilme süresinin uzunluğu,
- Şifreleme ve çözme işlemlerine harcanan süre,
- Şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarı,
- Algoritmanın kurulacak sisteme uygunluğu.

Şifreleme bilimi hızla gelişen bir bilim dalıdır. Eski algoritmaların dezavantajlarını ortadan kaldıracak yeni şifreleme algoritmaları geliştirilmektedir. Sonuç olarak, asimetrik şifreleme algoritmalarında ki hızlı gelişim sayesinde dezavantajları ortadan kaldırabilirse günümüz teknolojisinde simetrik şifreleme algoritmalarının yerini alacağını göstermektedir.

Kaynaklar

[1] Kodaz H. Veri İletiminde Güvenlik İçin Şifreleme, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2002.

[2] Krishnamurthy M, Seagren ES, Alder R, Bayles AW, Burke J, Carter S, Faskha E. Basics of Cryptography and Enryption, How to Cheat at Securing Linux, 2008, 249- 270.

[3] Stapko T. Security Protocols and Algorithms, Practical Embedded Security, 2008, 49-66.

[4] Aslan G.B. Sayısal İmza Sistemlerinin İncelenmesi, İTÜ, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 1999.

[5] Ham L, Ren J. Efficient identity-based RSA multisignatures, Computer & Security, Volume 27, Issues 1-2, March 2008, 12-15.

[6] Shao Z. Batch verifying multiple DSA-type digital signatures, Computer Networks, 2001, Volume 37, Issues 3-4, 383-389.

[7] Herranz J. Identity-based ring signatures from RSA, Theoretical Computer Science, Volume 389, Issues 1-2, 10 December 2007, 100-117.