ANAHTAR ANLAŞMAPROTOKOLLERİNİN İNCELEMESİ ÜZERİNE

Ömer KOCABIYIK

Yüksek Lisans Tezi

Matematik Anabilim Dalı

Dr. Öğr. Üyesi Abdullah ÇAĞMAN

AĞRI-2021

T.C.

AĞRI İBRAHİM ÇEÇEN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MATEMATİK ANABİLİM DALI

Ömer KOCABIYIK

ANAHTAR ANLAŞMA PROTOKOLLERİNİN İNCELEMESİ ÜZERİNE

YÜKSEK LİSANS TEZİ

TEZ YÖNETİCİSİ

Dr. Öğretim Üyesi Abdullah ÇAĞMAN

AĞRI – 2021

TEZ KABUL VE ONAYI

Ömer KOCABIYIK tarafından hazırlanan "ANAHTAR ANLAŞMA PROTOKOLLERİNİN İNCELEMESİ ÜZERİNE" adlı tez çalışması 08/07/2021 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Ağrı İbrahim Çeçen Üniversitesi Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Rıdvan ŞAHİN

Danışman

Dr. Öğr. Üyesi Abdullah ÇAĞMAN

Üye

Dr. Öğr. Üyesi Kadirhan POLAT

ONAY :

Bu tezin kabulü Enstitü Yönetim Kurulunun . . . . . tarih ve . . . . . sayılı kararı ile onaylanmıştır.

. . . /. . . /. . .

Prof. Dr. İbrahim HAN

Enstitü Müdürü

ÖZET

YÜKSEK LİSANS TEZİ

ANAHTAR ANLAŞMA PROTOKOLLERİNİN İNCELEMESİ ÜZERİNE

Tez Danışmanı: Dr. Öğr. Üyesi Abdullah ÇAĞMAN

2021, 62 sayfa

Jüri: Doç. Dr. Rıdvan ŞAHİN

Dr. Öğr. Üyesi Abdullah ÇAĞMAN

Dr. Öğr. Üyesi Kadirhan POLAT

Dijital teknolojinin hızla gelişimi faydaları ile beraber gizlilik ilkesini de ihlal eden bazı siber saldırıları beraberinde getirmektedir. Bu noktada iletişim konusunda güvenliğin sağlanması için şifreleme algoritmalarına ihtiyaç duyulmaktadır. Kriptografi bu ihtiyaçlara cevap verebilecek birçok şifreleme algoritmalarını geliştirmektedir. Bu çalışmada algoritmalar iki başlık altında incelenmeden önce şifreleme hakkında birinci ve ikinci bölümde belli terimler ve şifrelemede kullanılan bazı matematiksel ifadelere yer verilmiştir. Üçüncü bölümde simetrik ve asimetrik anahtarlı şifrelemeler detaylandırılarak ana konuya zemin hazırlanmıştır. Son bölümde tez başlığında belirtilen Anahtar Anlaşması Protokolleri ayrıntılı anlatılmıştır. Bahsedilen bazı protokoller örneklendirilmiştir.

2021, 62 sayfa

Anahtar Kelimeler: Kriptografi, Simetrik Anahtarlı Şifreleme, Asimetrik Anahtarlı Şifreleme, Anahtar Anlaşması Protokolleri, Diffie-Hellman.

ABSTRACT

MASTER'S THESIS

ON THE REVIEW OF KEY AGREEMENT PROTOCOLS

Advisor Of Thesis: Assist. Prof. Dr. Abdullah ÇAĞMAN

2021, 62 pages

Jüri: Assist. Dr. Rıdvan ŞAHİN

Assist. Prof. Dr. Abdullah ÇAĞMAN

Assist. Prof. Dr. Kadirhan POLAT

The rapid development of digital technology brings along some cyber attacks that violate the principle of confidentiality with its benefits. At this point, encryption algorithms are needed to ensure communication security. Cryptography develops many encryption algorithms that can meet these needs. In the first and second part of this study, before the algorithms are examined under two headings, certain terms about encryption and some mathematical expressions used in encryption are mentioned. In the third part, symmetrical and asymmetric keyed ciphers are elaborated and paved the way for on the main subject. Finally, Key Agreement Protocols mentioned in the thesis title are explained in detail. Implementations are applied with some mentioned protocols.

2021, 62 pages

Key Words: Cryptography, Symmetric Key Encryption, Asymmetric Key Encryption, Key Agreement Protocols, Diffie-Hellman.

TEŞEKKÜR

Yüksek lisans eğitimim boyunca, benden bilgi ve desteğini esirgemeyen, kıymetli danışmanım Sayın Dr. Öğr. Üyesi Abdullah ÇAĞMAN hocama teşekkürü bir borç bilirim.

Yine öğrenim hayatım boyunca maddi manevi desteklerini hep yanımda hissettiğim değerli ailelerimize, eşim Beyza Zuhal, oğlum Abdullah Taha ve kardeşlerime teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

ŞEKİLLER LİSTESİ ix

TABLOLAR LİSTESİ x

SİMGELER VE KISALTMALAR DİZİNİ xi

1. GİRİŞ 1

1.1. Şifreleme Terminolojisi 1

1.1.1. Kriptoloji 2

1.1.2. Kriptografi 3

1.2. Haberleşmede Güvenlik Prensipleri 4

1.2.1. Gizlilik 4

1.2.2. Bütünlük 5

1.2.3. Kimlik Doğrulama 5

1.2.4. İnkar Edilemezlik 5

1.2.5. Haberleşmede Süreklilik 5

2. ŞİFRELEMEDE MATEMATİKSEL KURAMLAR 6

2.1. Modüler Aritmetik 6

2.1.1. Modüler Aritmetikle İlgili Bazı Özellikler 6

2.2. Tek Yönlü Fonksiyon 9

2.3. Grup Teorisi 10

2.4. Ayrık Logaritma Problemi 10

2.5. Şifreleme ve Deşifreleme Fonksiyonları 11

3. ŞİFRELEME ALGORİTMALARI 14

3.1. Simetrik Anahtarlı Şifreleme Sistemleri 14

3.2. Asimetrik Anahtarlı Şifreleme Sistemleri 16

3.3. Açık Anahtar Altyapısı 17

3.3.1. Sertifika Otoritesi (CA) 18

3.3.2. Kayıt Otoritesi (RA) 19

3.3.3. Kök Sertifika Otoritesi (RCA) 20
3.4. Hash Fonksiyonları 21
3.5. Açık Anahtar Algoritmaları 24
3.5.1. Diffie-Hellman 24
3.5.2. RSA 27
3.5.3. ElGamal 29
3.5.4. Dijital İmza Algoritması (DSA) 33
4. ANAHTAR ANLAŞMASI PROTOKOLLERİ 35
4.1. Anahtar Anlaşma Protokollerinin Özellikleri ve Güvenlik İhtiyaçları 35
4.1.1. Bilinen Oturum Anahtarları (Known Session Keys) 35
4.1.2. İleriye Dönük Mükemmel Gizlilik (Perfect Forward Secrecy) . 36
4.1.3. Bilinmeyen Anahtar Paylaşımı (Unknown Key Share) 36
4.1.4. Anahtar Mutabakatında Kimliğe Bürünmeye Karşı Dayanım (Key-Compromise Impersonation) 36
4.1.5. Bilginin Yitirilmesi (Loss Of Information) 37
4.1.6. Mesaj Bağımsızlığı (Message Independence) 37
4.1.7. Anahtar Kontrolü (Key Control) 37
4.2. Temel (Anonim) Diffie-Hellman Anahtar Anlaşması 38
4.2.1. Ortadaki Adam Saldırısı 39
4.3. Sertifikalı Diffie-Hellman Anahtar Anlaşması 40
4.4. MQV Anahtar Anlaşma Protokolü 42
4.5. MTI Anahtar Anlaşma Protokolü 45
4.5.1. MTI A(0) Anahtar Anlaşması 46
4.5.2. MTI B(0) Anahtar Anlaşması 47
4.5.3. MTI C(0) Anahtar Anlaşması 48
4.5.4. MTI C(1) Anahtar Anlaşması 48
4.6. İstasyondan İstasyona Anahtar Anlaşma Protokolü (STS Protokolü) 49
4.6.1. Örtük Anahtar Kimlik Doğrulaması 50
4.6.2. Örtük Anahtar Onayı 50
4.6.3. Açık Anahtar Onayı 51
5. SONUÇ VE ÖNERİLER 54
KAYNAKLAR 56
ÖZGEÇMİŞ 62

ŞEKİLLER LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Kriptoloji 3
Şekil 1.2. Şifreleme Akış Şeması 4
Şekil 2.1. Basit bir haberleşme sistemi 12
Şekil 3.1. Simetrik Anahtarlı Şifreleme 14
Şekil 3.2. Asimetrik Anahtarlı Şifreleme 16
Şekil 3.3. Açık Anahtar Altyapısı Bileşenleri 19
Şekil 3.4. Kök Sertifika Otoritesi Oluşumu 21
Şekil 3.5. Hash Fonksiyonu ve Çıktıları 23
Şekil 3.6. Diffie-Hellman Anahtar Anlaşması 25
Şekil 4.1. Ortadaki Adam Saldırısı 39

TABLOLAR LİSTESİ

Tablo		Sayfa
Tablo 2.1.	Modüler aritmetik üzerinde yapılabilen işlemler	7
Tablo 3.1.	PKI'nin Veri Güvenliğine Katkıları	18
Tablo 3.2.	RSA Şifreleme Yöntemi	27
Tablo 4.1.	Temel Diffie-Hellman Anahtar Anlaşma Protokolü	38
Tablo 4.2.	Sertifikalı Diffie-Hellman Anahtar Anlaşma Protokolü	41
Tablo 4.3.	MQV Anahtar Anlaşma Protokolü	42
Tablo 4.4.	MTI A(0) Anahtar Anlaşma Protokolü	46
Tablo 4.5.	MTI B(0) Anahtar Anlaşma Protokolü	47
Tablo 4.6.	MTI C(0) Anahtar Anlaşma Protokolü	48
Tablo 4.7.	MTI C(1) Anahtar Anlaşma Protokolü	48
Tablo 4.8.	STS Anahtar Anlaşma Protokolü	51

SİMGELER VE KISALTMALAR DİZİNİ

$<$	Küçüktür
$>$	Büyüktür
$\leq$	Küçük veya eşittir
$\geq$	Büyük veya eşittir
$\in$	Elemanıdır
$\equiv$	Denktir
$\neq$	Eşit değildir
$\forall$	Her
$\mathbb{Z}$	Tam sayılar kümesi
φ	Phi
$\cup$	Birleşim
∞	Sonsuzluk
$\Longleftrightarrow$	Ancak ve Ancak
α	Alfa
$EBOB$	En Büyük Ortak Bölen
ord	Mertebe
DLP/ALP	Ayrık Logaritma Problemi
AES	İleri Şifreleme Standardı
DES	Veri Şifreleme Standardı
$IDEA$	Uluslararası Veri Şifreleme Algoritması
MAC	Mesaj Doğrulama Kodu
RSA	Rivest Shamir Aderman
$DH/DHAA$	Diffie-Hellman Anahtar Anlaşması
ECC	Eliptik Eğri Şifreleme
CPU	Merkezi İşlem Birimi
TLS	Taşıma Güvenliği Katmanı
STS	İstasyondan İstasyona Anahtar Anlaşması
KDF	Anahtar Üretme Fonksiyonu

MTI	Matsumoto Takashima Imai
DSA	Dijital İmza Algoritması
PKI	Açık Anahtar Altyapısı (Public Key Infrastructure)
CA	Sertifika Otoritesi (Certificate Authority)
RA	Kayıt Otoritesi (Registration Authority)
RCA	Kök Sertifika Otoritesi (Root Certificate Authority)
SHA	Güvenli Hash Algoritması
$MD4$	Mesaj Özetleme Algoritması 4

1. GİRİŞ

1.1. Şifreleme Terminolojisi

Geçmişte ve günümüzde insanlık; sırlarını saklamak, üçüncü kişilerden önemli gördükleri bilgileri gizlemek adına iletişim güvenliğine önem vermişlerdir. Bu güvenlik bazen çeşitli saldırılarla çözülmüş olsa da hala şifreleme tekniklerine gerek duyulmada ve türetimi devam etmektedir.

Gizli haberleşme, bundan yaklaşık 4000 yıl önce kullanılmaya başlanmış; özellikle diplomasi ve askeri servislerde yer almış ve II. Dünya Savaşı gibi modern savaşlarda önemli rol oynamıştır. İlk olarak Eski Mısırlıların tarihi anıtlarındaki hiyeroglif yazıların bazılarında görülen şifreleme daha sonra İbranililerin kutsal kitaplarındaki belirli kelimelerde görülmüştür (Çimen, 2007).

Ağ sistemlerinde kişisel veriler korunmakta, ülke çıkarları adına şifrelemede önemli adımlar atılmaktadır. Teknolojinin hayatımıza girmesiyle birlikte cep telefonlarından bankamatiklere, e-imzadan mail adreslerine kadar hayatımızın her anında yer alır vaziyete gelmiştir.

Yazımızın ana başlığını anlamamızı kolaylaştırmak için bilmemiz gereken bazı şifreleme terimleri karşımıza çıkmaktadır.

Şifreleme ile kimlik doğrulama veri emniyetini desteklemeyi sağlayan yekdiğeriyle ilintili teknolojilerdir. Tarafların dediklerinin doğruluğunun sağlanması kimlik doğrulamanın etkinliğinin sonucudur.

Şifreleme(Encryption), iletişim sırasında verinin güvenliğini sağlarken, değiştirilmesini de gizleyerek önlemeye çalşır.

Deşifreleme(Decryption) ise şifrelenmiş metnin tekrar anlaşılabilir hale getirilmesi işlemidir.

Kriptoloji, haberleşen iki ya da ikiden fazla tarafın veri transferini güvenilir bir şekilde yapmasına olanak veren ve prensipleri zor çözülebilen matematiksel tekniklere dayanan uygulamalar bütünüdür (Karahan, 2014).

Kriptanaliz, şifreyazım yöntemleri aracılığıyla oluşturulmuş bir şifreleme sisteminin analizi yapılarak zayıf ve etkili taraflarının ortaya konulmasını hedefleyen bilim dalıdır (Doğantimur, 2009).

- Anahtar(key), şifreleme ve deşifreleme için kullanılan kriptografi araçlarının genel ismidir.
- Plaintext, orjinal(yalın) metindir.
- Ciphertext, şifreli hale getirilmiş metin veya veridir.
- Sunucu(server), verileri saklayıp düzenleyip, dijital imzaları karşı tarafa ileten yetki sahibidir.
- İstemci(client), sunucu tarafından hizmet alarak bilgi paylaşımı yapılan kimsedir (Yıldırım, 2006).

Anahtar anlaşması protokolleri ilgilendiren konulara gelmeden evvel kriptoloji ve türevleri ile kriptosistemleri açıklayarak devam edilecektir.

1.1.1. Kriptoloji

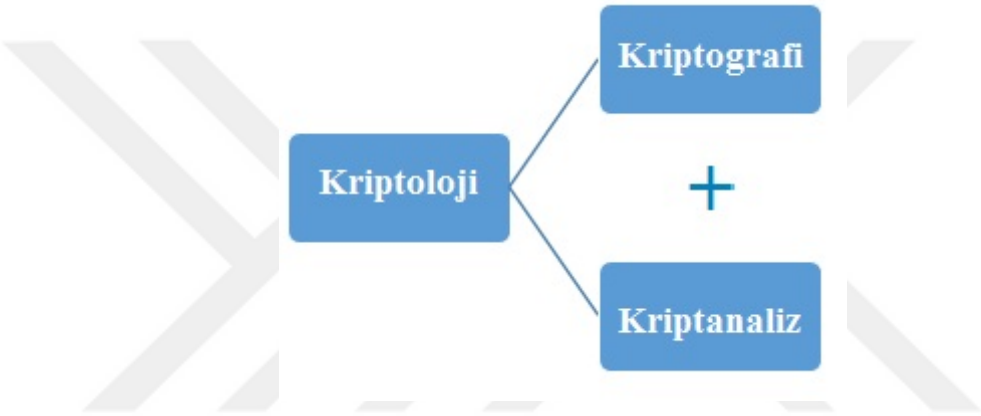
Tanım 1.1 *Gizlilik, veri bütünlüğü, varlık doğrulaması, veri aslının onaylanması gibi bilgi güvenliğiyle ilintili hususları matematiksel çalışmalar kullanarak ortaya koyan bilim dalına* ***Kriptoloji*** *Sifreleme bilimi) denir (Katz et al., 1996).*

Veriyi şifreleme ve çözümünde kullanılan matematiksel bilim olmakla beraber önemli bilgilerin korunması veya güvenli olmayan kanallar üzerinden verinin güvenli aktarımını sağlar. Bilim olarak kabülünden bu yana güvenli iletişimin incelenip çözüme kavuşturulması (kırılması) vazifesini yerine getirmektedir (Yıldırım, 2006).

Kriptanaliz, şifreli metinden düz metnin ya da anahtarın çözümsel tahminler, matematiksel yöntemlerin kullanımı, sezgi, keşif, sebat ve şans ile elde edilmesidir (Keleş, 2012).

Güçlü bir kriptografi, şifreli metnin düz metne dönüştürülmesi sürecinin uzunluğu ve harcanan kaynak nispetince anlaşılabilir. Şifreli bir metni çözecek uygun bir araç olmaksızın çözüme kavuşturmak güçtür. Bugün ise kuantum bilgisayarların fonksiyonel hacmi olmaz denileni olur hale getirip kolaylaştırıcı bir unsur olarak karşımıza çıkmaktadır (Mavroeidis et al., 2018). Bu da her algoritmanın elbette bir açığının ortaya çıkabilir oluşunu destekler.

Kriptoloji, kriptografi ve kriptanalizin bir araya gelmesiyle ortaya çıkar. Bunu 1.1 olarak gösterebiliriz.



Şekil 1.1. Kriptoloji

1.1.2. Kriptografi

Kriptografi, gizli anlamında *kryptos* ile yazmak kelimesine karşılık gelen *graphein* kelimelerinin birleştirilmesiyle ortaya çıkarılmıştır (List, 2007). Gizlilik, bütünlük, kimlik doğrulama, vb. veri güvenliğini; matematiksel işlevler kullanarak sağlamaya çalışır. Böylece veri iletimi sırasında karşılaşılabilecek ataklardan veriyi ve tarafları koruma amacını yerine getirmiş olur. Aslında kriptografi, açık bilginin öğrenmesini istemediğimiz üçüncü taraflarca okunamayacak duruma getirilmesi işlemleridir (Çimen, 2007). Özetle şifreli metni deşifreleme, deşifreli metni şifreli hale getirmek olarak da tanımlabilir. Şekil 1.2 'i inceleyiniz.

Şekil 1.2. Şifreleme Akış Şeması

Şekil 1.2'den hareketle; Şifreleme yapılmamış veriye kriptolojide "açık metin" (clear text ya da plain text) denmektedir. Açık metin, okunabilen bir yazı, bir bilgisayar programı ya da bir veri dosyasından ibaret olabilir.

Bir şifreleme algoritması yardımıyla, okunamayacak şekilde işlenmiş veriye ise "şifreli metin" (cipher text) denir.

Açık metinden şifreli metne geçmeye "şifreleme",

Şifreli metinden açık metne geçmeye ise "deşifreleme" ismi verilir.

Şifreleme ve deşifreleme işlevini yerine getiren bu tür bir sistem de "kriptosistem/kriptografi" olarak adlandırılır.

1.2. Haberleşmede Güvenlik Prensipleri

Bilgiye izin olmadan erişimin sağlanması, açığa çıkarılması, ortadan kaldırılması, içeriğinin değiştirilmesi, bozulmasından kısacası her türlü tehlikelere karşı koruma işlemine bilgi güvenliği ismi verilir. Üçüncü tarafların zararlı saldırı girişimlerine karşın güvenlik önlemleri de fazlacadır (Çubukçu, 2018).

Haberleşmede bilgi güvenliğinin sağlanabilmesi kullanıcılar ve sağlayıcılardan beklenenler vardır. Güvenliğin temel unsurları olan gizlilik, bütünlük, kimlik doğrulama, inkar edilemezlik ve haberleşmede süreklilik ile bu beklentiler karşılanabilir (İmamoğlu, 2011).

1.2.1. Gizlilik

İletilen mesajın muhtevasının gizli kalabilmesidir. Bu ilk prensipte dikkat edilecek husus mesajın tamamıyla saklanmasından ziyade yetki sahibi olmayanların eline geçmesini önlemek, önlemediği takdirde bundan haberdar olunabilmelidir.

1.2.2. Bütünlük

Mesajın iletimi esnasında içeriğinde yer alan bilgilerin herhangi bir değişikliğe uğramamasıdır. Mesajın değiştirilmesi, bozulması, yeni bilgiler eklenmesi ya da silinmesine karşın olduğu gibi kalabilmesinin sağlanması gerekir.

1.2.3. Kimlik Doğrulama

Mesajı ileten kişinin kimliğinin doğru olup olmadığından emin olunmasıdır. Kullanıcının yapması gereken ilk işlemlerden biri kendini kanıtlamasıdır. Kanıtlamanın sağlanması ile kullanıcının sisteme girişine müsaade edilir.

1.2.4. İnkar Edilemezlik

Mesaj göndericisi ya da alıcısının yaptığı bilgi alışverişini sonradan yadsımamasıdır. Bir işin gerçekleşmiş olmasının ispat edilmesi gereken adımlar olabilmektedir bu nedenle kimlik doğrulama ve bütünlük ön koşullarını sağlamalıdır.

1.2.5. Haberleşmede Süreklilik

Haberleşme sırasında iletişimin herhangi bir kopukluğa maruz kalmamasıdır. Bu prensibin devamlılığı açısından birbirinden farklı iletişim yolları denenmeli, iletişim kanallarının güvenliğine ve iletişim araçlarının bakımına önem verilmelidir.

2. ŞİFRELEMEDE MATEMATİKSEL KURAMLAR

Bu bölümde, kullanacağımız bazı matematiksel kavramlar, tanım ve teoremler verilirken Balcı (1985); Altındiş (1999); Şenay (2007) ve Çallıalp (2009)'den yararlanılmıştır.

2.1. Modüler Aritmetik

Açık anahtarlı şifrelerin oluşumu ve açığa çıkarılmasında modüler aritmetiğin destekleyici vazifesi bulunmaktadır.

Tanım 2.1 *$m \neq 0$ bir tamsayı olsun. $a, b \in \mathbb{Z}$ için, $a - b$ farkı m'nin bir k katı oluyorsa;*

$$a - b = k.m \tag{2.1}$$

ya da

$$a \equiv b \mod m \tag{2.2}$$

şeklinde gösterilebilir.

Burada $a \equiv b \mod m \iff m|a - b$ ile tanımlanır ve "a ve b, $\mod m$'ye göre *denk(kongrüent)*tir" denir.

Önerme 1 Yukarıdaki tanımda verilen "$\equiv$" bağıntısı $\mathbb{Z}$'de bir denklik bağıntısı belirtir (Hammack, 2016).

2.1.1. Modüler Aritmetikle İlgili Bazı Özellikler

Modüler aritmetik toplama, çıkarma, çarpma ve bölme gibi dört işlem gerektiren işlevler ile değişme, birleşme, geçişlilik, vb. gibi diğer özellikleri bünyesinde barındırır.

2.1.1.1. Toplama İşlemi

$$(a \mod m) + (b \mod m) = (a + b) \mod m$$

2.1.1.2. Çıkarma İşlemi

$$(a \mod m) - (b \mod m) = (a - b) \mod m$$

2.1.1.3. Çarpma İşlemi

$$(a \mod m) \times (b \mod m) = (a \times b) \mod m$$

2.1.1.4. Bölme İşlemi

$(a \mod m)/(b \mod m) = a \times b^{-1} \mod m$ (b nin tersi ile çarpma işleminden faydalanılabilir.)

2.1.1.5. Modüler Aritmetik Üzerinde Yapılabilen Diğer İşlemler

Başlık 2.1.1 de bahsedilen diğer özellikleri yakından incelemek için Tablo 2.1.1.5 e göz atınız:

Özellik	**Açıklama**
Değişme Kuralı	$(a + b) \mod n = (b + a) \mod n$ $(a \times b) \mod n = (b \times a) \mod n$
Birleşme Kuralı	$[(a + b) + c] \mod n = [a + (b + c)] \mod n$ $[(a \times b) \times c] \mod n = [a \times (b \times c)] \mod n$
Geçişlilik Kuralı	$a \equiv b \mod m$ ve $b \equiv c \mod m$ ise $a \equiv c \mod m$ vardır.
Dağılma Kuralı	$[a \times (b + c)] \mod n = [(a \times b) + (a \times c)] \mod n$
Etkisiz Eleman	$(0 + a) \mod n = a \mod n$ $(1 \times a) \mod n = a \mod n$
Toplamsal invers(-a)	$\forall\, a \in \mathbb{Z}_n$ için; öyle bir b vardır ki; $a + b = 0 \mod n$ 'dir.

Tablo 2.1. Modüler aritmetik üzerinde yapılabilen işlemler

Tanım 2.2 *$\mathbb{Z}$'deki "$\equiv$" denklik bağıntısının ifade ettiği denklik sınıflarına m modülüne göre kalan sınıfları denir ve kalan sınıfları kümesi $\mathbb{Z}_m$ ile gösterilir. Ayrıca $\mathbb{Z}_m$'de her sınıftan bir ve yalnız bir eleman seçilerek oluşturulduğu için tam temsilciler sistemi adı da verilmektedir. $a \in \mathbb{Z}$ olmak üzere $\overline{a} = \{x \in \mathbb{Z} : m|a-x\}$ a'nın denklik sınıfını ifade eder.*

Tanım 2.3 $\mathbb{Z}_m$'de kendileri $\overline{0}$ dan farklı olduğu bilinen kalan sınıflarının çarpımları $\overline{0}$ oluyorsa buna *sıfır bölen sınıfları* adı verilir.

Örnek vermek gerekirse, $\mathbb{Z}_{12} = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}, \overline{7}, \overline{8}, \overline{9}, \overline{10}, \overline{11}\}$ olup $\mathbb{Z}_{12}$'nin sıfır bölenleri $\{\overline{2}, \overline{3}, \overline{4}, \overline{6}, \overline{8}, \overline{9}, \overline{10}\}$ olur.

Önerme 2 $\mathbb{Z}_m$'de $\oplus$ işleminin $\forall \overline{a}, \overline{b}, \overline{c} \in \mathbb{Z}_m$ için sırasıyla *değişme, birleşme, etkisiz eleman* ve *ters eleman* özellikleri vardır:

- $\overline{a} \oplus \overline{b} = \overline{b} \oplus \overline{a}$
- $\overline{a} \oplus (\overline{b} \oplus \overline{c}) = (\overline{a} \oplus \overline{b}) \oplus \overline{c}$
- $\overline{a} \oplus \overline{0} = \overline{0} \oplus \overline{a} = \overline{a}$
- $\exists \overline{x} \in \mathbb{Z}_m$ için $\overline{a} \oplus \overline{x} = \overline{x} \oplus \overline{a} = \overline{0}$ bulunabilir.

Benzer özellikler $\mathbb{Z}_m$'de tanımlı $\otimes$ işlemi için de geçerlidir.

Önerme 3 $\mathbb{Z}_m$'de $\otimes$ işleminin $\forall \overline{a}, \overline{b}, \overline{c} \in \mathbb{Z}_m$ için sırasıyla *değişme, birleşme, birim eleman* ve *yutan eleman* özellikleri vardır:

- $\overline{a} \otimes \overline{b} = \overline{b} \otimes \overline{a}$
- $\overline{a} \otimes (\overline{b} \otimes \overline{c}) = (\overline{a} \otimes \overline{b}) \otimes \overline{c}$
- $\overline{a} \otimes \overline{1} = \overline{1} \otimes \overline{a} = \overline{a}$
- $\overline{a} \otimes \overline{0} = \overline{0} \otimes \overline{a} = \overline{0}$ bulunabilir.

Tanım 2.4 $\overline{a} \in \mathbb{Z}_m$ olmak üzere $(a, m) = 1$ oluyorsa $\overline{a}$'ya bir *asal kalan sınıfı* adı verilir ve tüm asal kalan sınıfları için $\mathbb{Z}_m^*$ ile gösterilir. Tam temsilciler sisteminden birine ise *indirgenmiş tam temsilciler sistemi* adı verilir. Ayrıca m modülü asal ise $\mathbb{Z}_m^*$ = $\mathbb{Z}_m \backslash \{\overline{0}\}$'dır.

Teorem 2.1 (Euler Totient Fonksiyonu) $n \in \mathbb{Z}^+$ *olmak üzere Euler Totient fonksiyonu* $\varphi(n)$*,* n *den küçük-eşit* $(1 \leq a \leq n)$ *ve* n *ile aralarında asal* $(a, n) = 1$ *olan bütün pozitif tam sayıların sayısı olarak ifade edilir.*

- p asal sayı olmak üzere $\varphi(p) = p - 1$'dir.
- p, q asal sayılar ve $n = p \times q$ ise $\varphi(n) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1)$'dir.

$\mathbb{Z}_m^*$'nin eleman sayısı $\varphi(n)$ Euler fonksiyonu sayesinde elde edilir.

Teorem 2.2 (Euler Teoremi) $n \in \mathbb{Z}^+$, $a \in \mathbb{Z}$ *ve* $\gcd(a, n) = 1$ *ise* $a^{\phi(n)} \equiv 1 \mod n$ *dir.*

Teorem 2.3 (Fermat'ın Küçük Teoremi) p *bir asal sayı ve* a*' da* $modulo$ p*' de sıfırdan farklı herhangi bir tamsayı olsun. Böylece* $a^{p-1} \equiv 1 \mod p$ *yi sağlar (Fraleigh, 2003).*

2.2. Tek Yönlü Fonksiyon

$$F : X \to Y$$

$$F : x \to f(x) = y$$

ancak ve ancak şu aşağıdaki şartları taşıması halinde tek yönlü bir fonksiyon olarak kabul edilir:

- $f(x)$ tüm x değerleri için hesaplaması kolay ve çözümlenebilir durumda iken,
- Tersi fonksiyonu için verilen bir y değeri için x değerini bulmak (gizli anahtar elde yok ise) zorlaşır.

Örnek vermek gerekirse, $g^k \mod m \equiv x$ bir modüler üs alma işleminde kolayca işlem yapılabilir iken x sayısı için m'den k'yı elde etmek ayrık logaritma probleminin hesaplamasına girer ve ki bu da çözüm açısından epey vakit almaktadır (Tuncal, 2008).

2.3. Grup Teorisi

Tanım 2.5 *Herhangi bir G grubu için, grup elemanların sayısı G grubunun* ***mertebesi*** *olarak isimlendirilir ve* ***ord(G) = |G|*** *sembolleriyle ifade edilir.*

Lagrange Teoremi gereğince H grubu G grubunun bir alt grubu varsayılırsa $|H|$'nin değeri $|G|$'nin değerini böler. Bu şekilde eğer G grubunun mertebesinin bir asal sayı olduğu bilinirse G' nin alt grubunun tek ve kendisi olduğu söylenir. Böyle hallerde G grubu çarpmalı olacak şekilde yazılabilir.

G grubu çarpmalı olarak yazılırsa ve $g \in G$ olmak üzere g sayısı G grubunun mertebesini belirtiyorsa; g sayısı $i \in N \cup \{\infty\}$ ve $g^i = 1$ şartlarını sağlayan i'nin en küçük değeridir. Burada $\forall j, l \in Z$:

$$g^i = g^j \iff j \equiv l \mod ord(g)$$

dir.

Tanım 2.6 *G bir grup ve $a_1, ...a_n \in G$ olacak şekildeki tüm alt gruplar g elemanının üssü kabul edilir ve $< g >$ ifadesiyle gösterilirler. Öyle ki $G =< a_1, ...a_n >$ ise $a_1, ...a_n$, G'nin* ***üreteçleri****dir. Aynı şekilde $< g >= G$ ise g sayısı G grubunun* ***üreteci*** *denir. Bir üretecin varlığı tüm gruplara* ***devirli grup*** *(cyclic group) adını verir.*

G grubunun mertebesi p asal sayısı olduğu biliniyorsa gruptaki 1 dışındaki tüm sayılar G grubunun üreteci olur. Yani bu durumda $< g >$ nin mertebesi 1 veya p sayısından başkası değildir.

2.4. Ayrık Logaritma Problemi

Tanım 2.7 $\mathbb{Z}_p^*$ grubu için g nin bir ilkel kök ve $y \in G$ sıfırdan farklı bir elemanı iken, $1 \leq x \leq p-1$ aralığındaki x tamsayısı için

$$g^x = y$$

eşitliğinde x değerinin bulunmasına **ayrık logaritma problemi** denir. Buradaki x tamsayısına g tabanındaki y elemanının ayrık logaritması denilir ve $\log_g(y)$ ile ifade edilir (Turp, 2019).

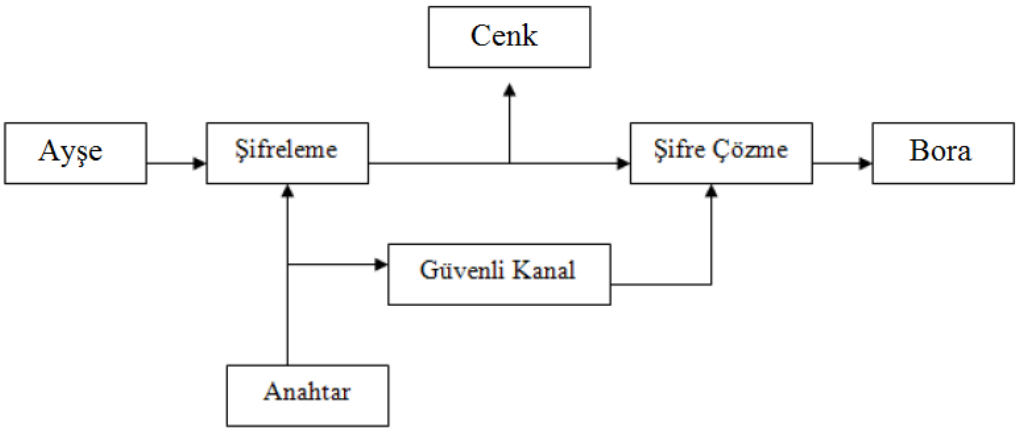
Şifreleme algoritmalarının kullanımına bakıldığında pek çoğunun ayrık logaritma probleminin çözümünün zorluğundan yararlandığı görülmektedir. Diffie-Hellman ve varyantları ile ElGamal şifrelemelerinde açıkca gözümüze çarpmaktadır. Açık anahtarlı şifrelemede G sonlu devirli grubunun üreteci g olmak üzere: $\mathbb{Z}_p^*$ çarpımsal grubu, $GF(2^n)$ sonlu cismi ve *Eliptik Eğriler* gibi farklı kullanım alanlarında da üs almalarla ve ayrık logaritmalarla karşılaşılmaktadır (Katz et al., 1996).

Üstelleştirmenin ayrık logaritma problemine nispeten daha basit çözümünün olduğu varsayılır. Herhangi bir $b! = 0$ için p nin bir asal sayı olduğunu söyler isek g varlığından bahsedilmesi gerekir. g nin art arda verilen üsleri $\mod p$ ile grup haline gelir ki; $g \mod p, g^2 \mod p, ..., g^{p-1} \mod p$ nin değerleri aynı çıkmaz ve 1 ile $p-1$ aralığında değerler verir. Böylece g ye **ilkel (primitif) kök** ismi verilir. Çözümün kolaylığı ise daha zor hale gelir.

2.5. Şifreleme ve Deşifreleme Fonksiyonları

Kriptografinin hedefi, tarafların emniyetli iletişimini yerine getirirken üçüncü zararlı kişilerin müdahil olmamasını sağlamaktır. Bunu yerine getirmek üzere şifreleme sistemleri durmadan geliştirilmeye devam etmektedir. Kriptosistemler şifreleme fonksiyonları tarafından üretilmektedir. Daha anlaşılır olması için iletişimdeki tarafları göndericiyi Ayşe, alıcıyı Bora ve iletimi aksatmayı hedefleyen üçüncü zararlı kişiyi

de Cenk diye isimlendirelim. Ayşe'den Bora'ya gönderilmek istenen mesajda bir bilgi, sayısal bir veri ya da herhangi bir şey yazılı olabilir. Ayşe kendi anahtarıyla açık metni şifreleyerek iletişim ağı üstünden Bora'ya şifrelediği mesajı gönderir. Cenk ağ üzerindeki şifreli metni gözler fakat açık metin şifreli olduğu için mesaj içeriğinde ne yazdığını göremez. Bora ise şifreli mesajı, kendi deşifreleme anahtarıyla çözerek açık metne ulaşır.



Şekil 2.1. Basit bir haberleşme sistemi

Şifreleme algoritmalarında kullanımı tercih edilen kavramlardan bazıları şifrelemenin beşlisi olarak tanıtılır:

- P; açık metnin bir sonlu kümesidir.
- C; gizlenmiş (şifreli) metnin bir sonlu kümesidir.
- K; (Anahtar uzayı), olması beklenen anahtarların bir sonlu kümesidir.
- $E = E_k : k \in K$ şifreleme algoritması ve $D = D_k : k \in K$ deşifreleme algoritmasının bir kümesidir.
- Her $k \in K$ olmak üzere bir $E_k \in E$ şifreleme fonksiyonu ve buradan hareketle bir $D_k \in D$ deşifreleme fonksiyonu vardır. $E_k : P \to C$ ile $D_k : C \to P$ fonksiyonları, her $x \in P$ için,

$$D_k \quad E_k \quad x = x \quad (2.5)$$

eşitliğini yerine getirir (Stinson, 1995). Sondaki (2.5) özelliği; E_k ile şifrelenmiş açık metnin D_k ile çözülerek orijinal açık metne ulaşılabileceğini göstermektedir.

Kriptosistem beşlisi yardımıyla iki taraf arasındaki iletişimin nasıl olduğunu görelim:

Ayşe'den Bora'ya açık bir kanal üzerinden gönderilmek istenen bir mesaj olduğunu varsaylaım. Öncelikle rastgele bir $k \in K$ anahtarı seçimi yapılır. Bu anahtar daha önce ieltişimin tarafları aynı yerde bulunurken gerçekleştirilirmiş veya emniyeti sağlanmış bir ağ üstünden karar verilmiş ve birbirlerine gönderilmiş olabilir.

İletilmek istenen metin ilk olarak sayısal bir hale getirilir ve bir sayı dizisi oluşturulur. Böylece P, C ve K'yı doğal sayılarının alt kümelerini almak genelliği etkilemeyecektir. Oluşturulmak sayı dizisinin $n \geq 1$ ve $1 \leq i \leq n$ olmak üzere $x_i \in P$ için,

$x = x_1 x_2 \dots x_n$ şeklinde oluşunu kabul edelim. Sonra Ayşe $1 \leq i \leq n$ için $y_i = E_k \quad x_i$ ile her bir x_i açık metnine denk gelen bir y_i değeri hesaplar ve $y = y_1 y_2 \dots y_n$ sayı dizisini oluşturur. Böylece şifreli metnin sayısal olarak karşılığı elde edilmiş olur.

Bora $y = y_1 y_2 \dots y_n$ şifrelmiş mesajını alarak $x_i = D_k \quad y_i$ deşifreleme fonksiyonuyla orjinal $x = x_1 x_2 \dots x_n$ açık metnine denk gelen sayısal ifadelere ulaşır. Bilinmelidir ki şifrenin çözünü ile orjinal metne ulaşılmak isteniyorsa hem D_k hem de E_k nin $1 - 1$ fonksiyon olması gerekmektedir.

$P = C$ eşitliği var ise, şifreleme fonksiyonun permütasyon olduğu belirtilir. Şifreli metinler ve şifresi çözülmüş metinlerin kümelerinin aynı olması demek şifreleme fonksiyonun küme elemanlarını yeni bir şekilde sıralaması manasına gelir (Külen, 2013).

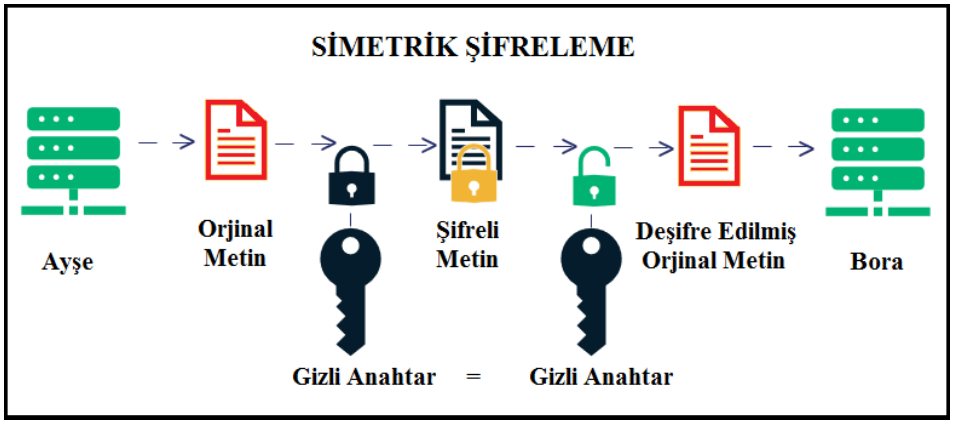
3. ŞİFRELEME ALGORİTMALARI

Tüm modern algoritmalar şifreleme ve deşifreleme işlemleri için bir anahtar kullanırlar ve şifreli mesaj yalnızca kullanılan anahtarın şifreleme anahtarıyla uyumuyla çözülebilir. Şifreleme süresince anahtarlı ya da anahtarsız olmak üzere iki farklı yöntem kullanılabilir. Özet (Hash) fonksiyonları, sıkıştırma fonksiyonları anahtarsız yöntemlere örnek olarak gösterilebilir. Genel şifreleme teknikleri olan anahtarlı kriptosistemler ise iki ana başlık altında incelenebilir (Schneier, 1996; Stallings, 2006):

- Simetrik anahtarlı şifreleme (veya gizli-anahtarlı şifreleme)
- Asimetrik şifreleme (veya açık-anahtarlı şifreleme)

3.1. Simetrik Anahtarlı Şifreleme Sistemleri

Simetrik şifreleme algoritmaları, şifreleme ve deşifreleme işlemleri için tek bir anahtar kullanan algoritmalardır. Bu anahtarlar haberleşen taraflar dışından gizlenir. Yalnızca mesajı şifreleyen ve çözmesi istenen kişi tarafınca bilinirler. Gönderici şifreli metin ile birlikte, tarafların daha önce anlaştığı anahtarı alıcıya gönderir. Gönderilen bu anahtarla deşifreleme sağlanır.



Şekil 3.1. Simetrik Anahtarlı Şifreleme

Mesajı şifreleyen ve deşifreleyen kişilerin anahtarları sadece o taraflarca bilinmesi gerekir. Şifreleme anahtarının bilinmesi bilginin elde edilmesini kolaylaştırır. Ancak problem, anahtarın üçüncü casus şahıslara yakalanmadan taşınımıdır.

Gizli anahtarlı şifreleme sistemleri geleneksel ve modern olmak üzere iki grupta incelenir. Geleneksel yöntemli şifrelemeye karakter tabanlı, modern yöntemlere ise bit tabanlı şifrelemeler ismi verilmektedir. Klasik (tarihsel) şifrelemelere Sezar, Affine, Vigenere, Vernam, Yerine Koyma, Hill , vb.; modern blok şifrelemelere ise AES, DES, 3DES, RC4V, IDEA, IRON ve Blowfish gibi örnekler verilebilir (Yeşilbaş, 2016).

Simetrik şifrelemenin en kolay yöntemlerinden biri Sezar şifresi olarak bilinen yerine koyma şifrelemesidir. Bu şifrelemede bilgi, diğerinin yerine konulur. Bu tür algoritmalar alfabe kaydırma işlemini temel alır ancak bugün yeterince kolay kalan bir algoritma olarak görülür. Simetrik anahtarlı sistemlerin avantaj ve dezavantajları ise şu şekilde söylenebilir:

Avantajları;

- Gayet hızlıdırlar. Şifrelenmiş verinin başkaca bir yere iletilmediği zamanlarda kullanışlı olabilirler.
- Kullanımı anlaşılırdır ve donanımlara genellikle uyum sağlarlar.
- Gönderici ve alıcı ortak bir anahtar üzerinde anlaşmaya varırlar ve bu anahtar kendilerinde saklı tutulmalıdır, mutabık oldukları anahtar kısa olduğu takdirde başarılı iletim daha kolay sağlanır.

Dezavantajları;

- İletişim kurmaya çalışan taraflar farklı yerlerde ise simetrik anahtarın gönderimi esnasında kanalın güvenli oluşu garanti edilmelidir.
- Saldırgan, anahtarı gözleyebilir hatta değiştirebilir.
- Anahtar dağıtma problemi vardır.
- Şifreli metnin gönderilmesi esnasında şifrenin gizli tutulması maliyeti artıran bir uygulamadır.

3.2. Asimetrik Anahtarlı Şifreleme Sistemleri

Asimetrik şifreleme, simetrik şifreleme algoritmalarından radikal bir şekilde ayrılmaktadır. Bu fark asimetrik şifreleme sistemlerinde açık (public) ve özel (private) olmak üzere iki ayrı anahtar kullanımından kaynaklanır.

Açık anahtarlı sistemler olarak da bilinen asimetrik şifrelemeli algoritmalarda metnin şifrelemesi için tercih edilen anahtarla şifrelenmiş metni çözmek üzere tercih edilen anahtar birbirinden farklıdır. Anahtarlar algoritmaların farklı yapıdaki matematiksel kullanımından dolayı her taraf için kamu-kendine has çiftleri olarak farklıdır, yani her bir tarafın açık-özel anahtarları yalnızca taraflarca biriciktir. Ayrıca deşifreleme anahtarı diğer (özel) şifre anahtarından elde edilemez. Bu şifreleme türünün açık anahtarlı olarak adlandırılmasının nedeni şifreleme anahtarının halka (kamuya/genel kullanıma) açık olmasıdır.

İki tarafın iletişimi esnasında üçüncü şahsın casus olduğunu varsayarsak metni şifrelemek için paylaşılan genel/kamu/açık anahtarı şifreleme anahtarı olarak kullanabilir, fakat yalnızca şifreyi çözen anahtara sahip olan kişi metni deşifre edebilir. Özel anahtar bazı zamanlarda gizli anahtar olarak da isimlendirilir.

Şekil 3.2. Asimetrik Anahtarlı Şifreleme

Bir tarafın karşı tarafın genel anahtarıyla şifrelediği metin, ancak karşı tarafa ait özel anahtarla açılabilmektedir. Yine herhangi bir tarafın özel anahtarıyla belgelediği sayısal imzanın doğruluğuna, ancak o kullanıcının genel anahtarını kullanarak erişim sağlanabilir. Açık anahtar, e-kimlik bilgileri ve diğer kişisel belgelerle beraber saklanır. Herkes birbirinin açık anahtarını, e-kimliklere erişim sağlamak amacıyla istediği zaman elde edebilir (Kodaz ve Botsalı, 2010).

Bazı açık anahtarlı algoritmalara örnek vermek gerekirse: Diffie-Hellman, RSA, ElGamal, ECC (Eliptik Eğri Sistemleri) ve DSA'dır.

Avantajları;

- İnkar edilememezlik, kimlik sınaması ve gizliliği sağlar.
- Anahtar dağıtımı simetriğe göre daha basittir ve algoritmada anahtar fazlalığa sebebiyet vermez.
- Güvenlik daha üst düzeydir çünkü anahtarın açığa çıkma veya ele geçirilme riski daha azdır.

Dezavantajları;

- Simetrik şifrelemeyle kıyaslandığında çok daha yavaş olduğu söylenir. Bu da kullanılan anahtarların büyük sayılar ve zor matematiksel işlemlerden seçilerek yapılması kaynaklıdır.
- Donanımların CPU harcamalarını arttırıcı nedeni olur.

Diffie-Hellman, RSA, ElGamal ve DSA'yı içine alan asimetrik anahtarlı şifrelemeleri ve ilgili terimleri Başlık 3.5 de detaylı incelenecektir.

3.3. Açık Anahtar Altyapısı

Haberleşmenin emniyeti ve veri birikimi maksadıyla asimetrik şifrelemeyi temel alarak anahtar oluşturma, sertifika ve anahtar idaresini sağlama, servis ve ifa görevlerinin bir arada bulunmasına Açık Anahtar Altyapısı (Public Key Infrastructure) adı verilmektedir.

İngiliz İstihbaratında görevli James Ellis ve Clifford Cocks isimli kriptanalistler şifreleme ve anahtar dağıtımındaki sorunların çözümü için yapmış oldukları çalışmalarına 1970'lerin başlarında başlasalar da teşkilat yapısı dolayısıyla 1990'ların sonlarına değin çalışmalarını kamuoyuna açamamışlardır (Singh, 2000; Wayner, 1997). Diffie ve Hellman tarafından ortaya konulan asimetrik şifreleme (Diffie and Hellman, 1976) ile devrim niteliğinde bir adım atılırken bu çalışmaların hemen ardından Rivest, Shamir ve Adleman'ın algoritması (Rivest et al., 1978) ile güvenli iletişim için açık anahtar altyapısının kökleri atılmaya başlanmıştır.

Diffie-Hellman öncülüğündeki asimetrik şifreleme algoritmaları geçmişten günümüze Secure Sockets Layer (Güvenli Soket Katmanı-SSL)/Transport Layer Security (Taşıma Güvenliği Katmanı-TLS), Secure Shell (Güvenli Kabuk-SSH), Internet Protocol Security (İnternet Protokolü Güvenliği-IPSec), Public Key Infrastructure (Açık Anahtar Altyapısı-PKI) gibi birçok protokolde kullanılmaktadır.

DH temelli modern şifreleme algoritmalarının önemli problemlerinden sayılan herkese açık anahtarların kimlik doğrulamalı dağıtımının gerçekleştirilmesi PKI'nın amaç ve ilgi alanına girmektedir. Yani açık anahtarların herkesçe bilinmesine karşın iletişimin taraflarına ulaştırılarak anahtarlar-taraflar ilişkisinin emniyetli şekilde bağlanımı gerçekleştirilmelidir (Hunt, 2001).

PKI, haberleşmede güvenlik prensiplerini Tablo 3.1'de belirtilen çözüm yolları ile yerine getirir (Erol, 2006):

Haberleşmede Güvenlik Prensipleri	**Güvenliği Yerine Getiren Çözümler**
Gizlilik	Veri Şifrelenme Algoritmaları
Bütünlük	Dijital İmza, Sertifika ve E-Kimlik
Kimlik Doğrulama	Hash(Özet) Fonksiyonları ve Dijital İmza
İnkar Edilemezlik	Dijital İmza ve İşlem Kaydı
Haberleşmede Süreklilik	Veri Yedekleme ve düzenekleri, Sistemin bakımı

Tablo 3.1. PKI'nin Veri Güvenliğine Katkıları

Açık anahtar altyapısı yardımıyla, tarafların gerçekleştirdiği veri transferi sırasında arka tarafta haberleri olmadan güvenli işlemler gerçekleşmektedir. PKI'nın bu sistemlerde bulundurduğu temel öğeler şu şekildedir:

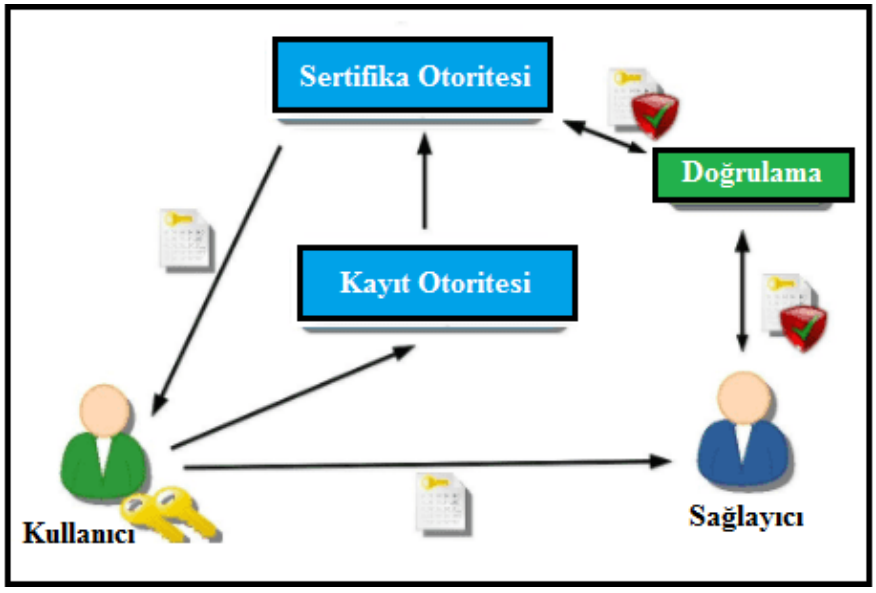
3.3.1. Sertifika Otoritesi (CA)

İletişimde yer alan tarafın kimliği ve açık anahtarı arasında bir ilişki olup olmadığını teyit etmeye yarayan dijital dökümana **sertifika** adı verilir (Taş ve Akın, 2002). Sertifikaların bir arada bulunmasının ve dağıtımının güvenliği teyit edilmiş üçüncü taraflarca gerçekleştirilmesi esastır. Üçüncü taraf yani güvenilir makam pozisyonundaki sertifika otoriteleri tarafların kimliği ve açık anahtarı arasındaki bağlantıya onay verir.

CA tüm bunlara ek olarak sertifika türetme, türetilen sertifikaların bilgilerini ve açık anahtarı muhafaza etme, Sertifika İptal Listesinin (SİL) nin tertibi ve istekte bulunan taraflara bu listelerin gönderimi sorumluluğuna sahiptir.

3.3.2. Kayıt Otoritesi (RA)

Sertifika Otoritesince gerçekleştirilen birtakım idari işlemler Kayıt Otoritesince de yapılabilmektedir. Örnek vermek gerekirse; sertifika üretimi yapmak üzere sondaki kullanıcının kendine ait bilgilerini doğrulaması istenmektedir. İşlemin daha kolay hale gelebilmesi adına kullanıcı, doğrulama yapmayı başka başka yerlerde inşa edilmiş kayıt makamlarınca gerçekleştirmektense Kayıt Otoritesi üstünden tek noktadan gerçekleştirir.



Şekil 3.3. Açık Anahtar Altyapısı Bileşenleri

Kayıt Otoritesi (RA), sertifikalara erişim talebinde bulunanların kimliklerini tasdikleyip kayıt altına alarak dijital imzasını kullanarak Sertifika Otoritesine sevk etme işlevine sahiptir. Bu sayede CA sertifika talebinin güvenilir kaynaktan olup olmadığını fark edebilir. Kayıt Otoritesi farklı CA'lara da aynı şekilde hizmet edebilir bu yüzden gizli anahtarını iyi muhafaza edebilmelidir (Durak, 2007).

Kayıt Otoritesinin Sertifika Otoritesi ile gerçekleştirmiş olduğu etkinlikler iki türlü olmaktadır (Sağır, 2014):

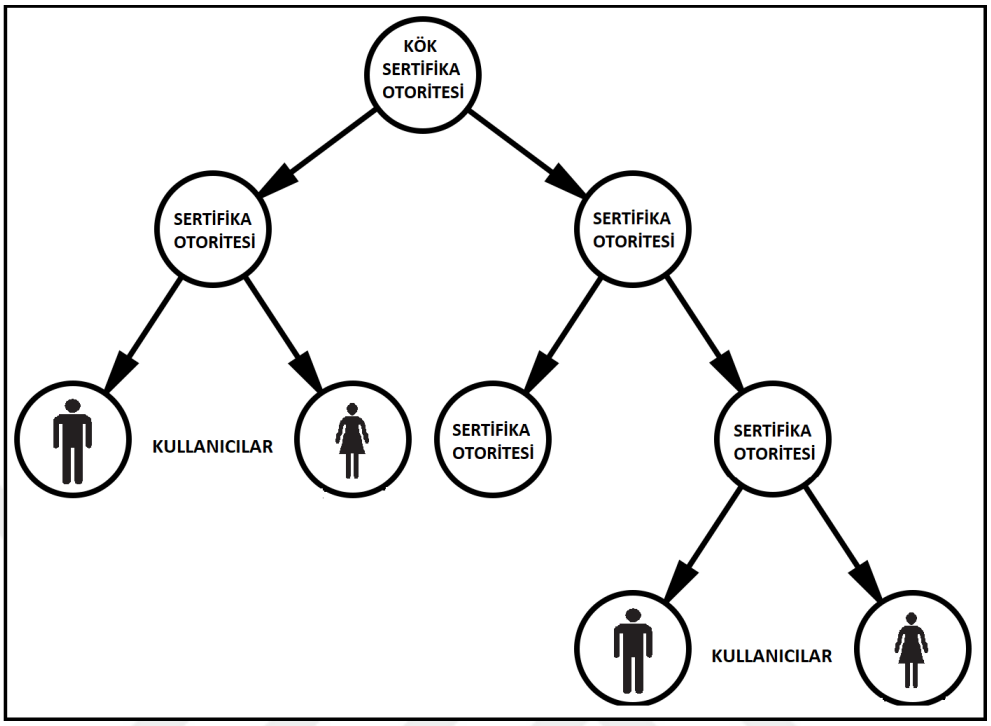
1. RA, sertifika talebinde bulunmadan evvel gerekli bilgileri bir araya getirip tasdikler. Genellikle kişisel başvurularda tercih edilen yöntemlerden biridir. Başvuruyu yapan kimliğini bir doküman ile RA'ya kanıtlamış olur. Toplanan bilgiler ışığında sertifika talebi Sertifika Otoritesine nakledilmiş olunur.
2. Kişilerce elektronik ortamda sertifika talebinde bulunulur (e-posta adresinin talepte bulunana ait olup olmayışı örnek verilebilir). CA talebi alarak bilgi tasdiki amacıyla bunu RA'ya havale eder. RA tarafından olumsuzluk olmadığına ve bilgini doğru olduğuna dair iznin verilmesiyle sertifika talebi CA tarafından yanıtlanır.

3.3.3. Kök Sertifika Otoritesi (RCA)

Kullanıcı kimliklerinin beyanı gibi sertifika beyanları için kök sertifikalara gereksinim duyulmaktadır. Sertifikaların emniyeti için başka bir sertifikaya, onun da diğer sertifikayla ilişki içinde olarak kök sertifikaya kadar uzanmasına sertifika zinciri denir. Kök sertifika ise sertifika makamının sayısal sertifikasıdır ve ara sertifikaları imzalayarak emniyetlerini sağlar. Ara sertifikalar da diğer sertifikaları veya sondaki kullanıcının sertifikasını imzalayarak emniyeti sağlar (Zuccherato, 2003).

Kök Sertifika Otoritesi, PKI etkinliklerinin düzenli ve emniyetli olabilmesi adına sistemin en üzerindeki yer alan; bütün elemanların dijital imzasına inandıkları yetkilidir. CA'lar sertifikaların yüklenimi esnasında RCA'ya güvenerek işlemlerini gerçekleştirirler. RCA biriminin sıralanımı ülkelere göre çeşitlilik arz edebilir. RCA ve altında CA'lar yer alabileceği gibi CA'lara sertifika temin eden farklı CA'lar da

bu oluşumda yer alabilir. Ancak bilinmesi gereken oluşumun en üstünde RCA'nın olduğudur (Sağıroğlu ve Alkan, 2005).



Şekil 3.4. Kök Sertifika Otoritesi Oluşumu

Kök Sertifika Otoritesinin gizli anahtarı PKI için en değerli olandır. Gizli anahtarın açığa çıkması veya yetki sahibi olmayanlarca edinilmesi bütün iletişim altyapısının güvenliğini riske sokar. Bu sebeple gizli anahtarı saklamak adına yüksek emniyetli bölgelere ihtiyaç duyulur. Fiziki müdahalelere karşı dayanıklı, şifreleme anahtarlarını koruma odaklı çalışan güvenli bir şifreleme işlemcisi olan Donanımsal Güvenlik Modulü (Hardware Security Module) sayesinde saklanım gerçekleşir (Yıldırım vd., 2005).

3.4. Hash Fonksiyonları

Veri bütünlüğünü koruma ilkesini gözeterek farklı uzunluktaki bir veri girdisini uzunluğu sabit olan benzersiz bir veri çıktısı haline getiren sayısal fonksiyonlardır. Hash fonksiyonları ile türetilen verilere hash değeri ya da özet değeri ismi verilir. Hash değerlerinin benzersiz olmasından kasıt ise hesaplama yapıldıktan sonra veri üzerinde oynama yapılır ve yine hash fonksiyonu hesaplanırsa daha

önceki değerle uyumsuzluk ortaya çıkar. Bu da veri üzerinde değişiklik yapıldığının göstergesidir ve veri bütünlüğü prensibinin delinmesi manasına gelir (Ural ve Örenç, 2019; Okuyucu, 2020).

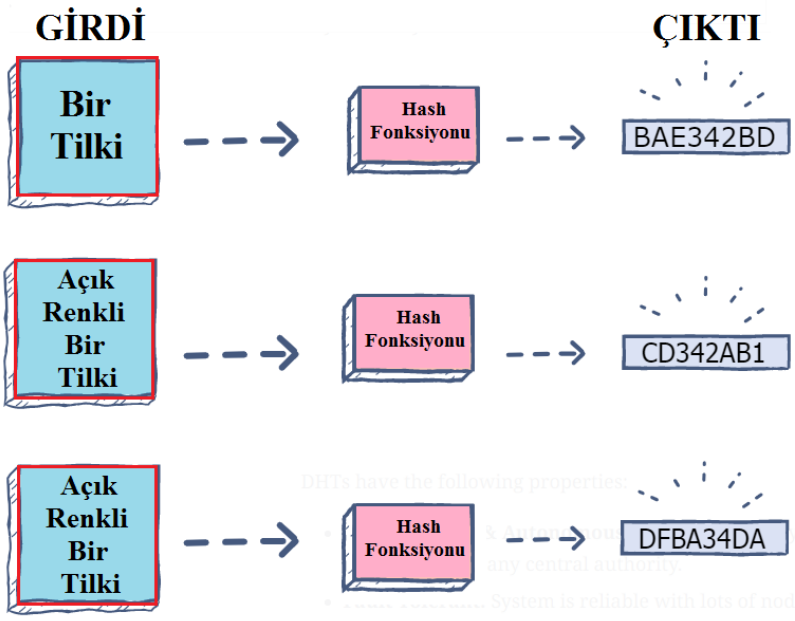
Hash fonksiyonları çalışma prensibi gereğince girdi verisi ne kadar büyük olursa olsun küçük boyutlu çıktılar vermeye göre tasarlanmıştır. Böylece imzalama ve parola kaydı tutma açısından da büyük kolaylıklar sağlamaktadır.

Dosyaların incelenmesi ve işleme tabi tutulması sonrasında imza atılması süre ve büyük boyutlarla uğraşma problemini beraberinde getirmektedir. Bu yüzden verinin özeti alınarak buna imza atılır. Doğrulama yapan şahıs da özet ve imzalı veriyi karşılaştırır. Aynı şekilde parolaların sistem üzerinde kayıtlarının düz metin şekilde tutulması yerine daha küçük boyutlu hash değerleri tutularak kullanıcı parolası sisteme girildiğinde parola özeti ve tutulan özet karşılaştırılarak parola gizliliği ve emniyeti de sağlanmış olur (Çakmak, 2018).

Hash fonksiyonlarının çeşitli özellikleri aşağıdaki gibi sıralanabilir:

- Hash fonksiyonları tek yönlü olarak çalışmaktadır. Böylece özetlenmiş veriden orjinal veriye ulaşılamaz.
- Hash fonksiyonları anahtarsız işlemlerdir.
- Simetrik veya asimetrik şifreleme sınıflandırmalarına dahil edilmezler.
- Bloklar uzadıkça bütünlüğün de güvenliği o denli artar.

Hash fonksiyonlarının farklı sonuçlar vermesine karşın düşük de olsa aynı çıktıyı verme ihtimali vardır. İki ayrı verinin hash fonksiyonu sonucunda aynı hash değerini vermesine çakışma (collision) denilir ve hash fonksiyonları için arzu edilen bir durum değildir. Bu yüzden fonksiyonun çakışmaya karşı dirençli olması beklenir (Doğanaksoy vd., 2008).
Eşit olmayan M_1 ile M_2 mesajlarının hash fonksiyonları için $Hash(M_1) \neq Hash(M_2)$ çakışmayı önler. Şekil 3.5 de farklı ve aynı veri girdilerinin hash fonksiyonu sonucunda çakışmaya karşı birbirinden farklı çıktılar verdiği görülmektedir:



Şekil 3.5. Hash Fonksiyonu ve Çıktıları

SHA, MD, RIPEMD, MAC, HAVAL ve WHIRPOOL algoritmaları hash fonksiyonları arasında en fazla tercih edilendirler. Günümüzde güvenliği zayıfladığı için uzun yıllar kullanılan MD5 ve SHA-1 hash algoritmalarından vazgeçilerek SHA-2 ile SHA-3 ailesi tercih edilmektedir.

MD hash algoritmaları serisi 128 bitlik hash değerleri ile en fazla tercih edilen algoritmalardandır. Ron Rivest tarafından kullanıma sürülmüştür (Rivest and Dusse, 1992).

SHA hash algoritmaları serisi 160 bitlik hash değerleri üretir. NSA (Ulusal Güvenlik Ajansı) tarafından geliştirilip kullanılmaya başlanmıştır (Eastlake and Jones, 2001).

RIPEMD hash algoritması 160 bitlik hash değerleri üretir. Avrupa Birliğince tercih edilmektedir (Stallings, 2006).

Anahtarsız algoritmalardan farklı olarak MAC hash algoritması vardır ki oluşum ve doğrulama gerçekleştirmek için bir anahtara ihtiyaç duyar. Veri bütünlüğü ve doğrulaması ile tek yönlü çalışan, veri alışverişini sağlayan bir anahtara dayalı HMAC varyasyonuna da sahiptir (Bellare et al., 1996).

Kriptografik hash fonksiyonları günümüzde Blockchain uygulamaları, kripto paralarda sıkça kullanılmaktadır. Bitcoin kripto para biriminde SHA-256 hash fonksiyonu tercih edilmektedir.

3.5. Açık Anahtar Algoritmaları

Asimetrik (açık anahtarlı) şifreleme algoritmalarının içeriğine bakıldığında anahtarların birbirleri ile matematik temelli bağlantılarının olduğu görülür. Bu kriptosistemlerin dayandığı işlemler başlıca şunlardır (Bozkurt, 2005):

Tamsayıların çarpanlarına ayrılmasını baz alan: RSA, Rabin-Williams kriptosistemleri,

Ayrık logaritma problemi ve imzalamayı baz alan: Diffie Hellman Anahtar Anlaşması, ElGamal, Dijital İmza Algoritması (DSA) kriptosistemleri,

Sadece imzalamayı baz alan: Schnorr, Nyberg-Rueppel yöntemleri ve yukarıda belirtilen işlemlerin hepsine kendi yapısını entegre edebilen Eliptik Eğriler.

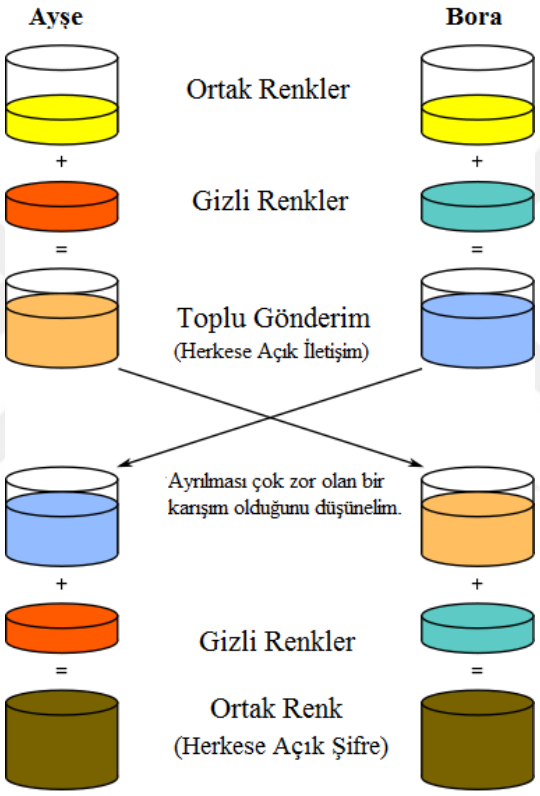
Ortak bir anahtarın (oturum anahtarı) elde edilmesi adına anahtar değişimi (key exchange) / anahtar anlaşması (key agreement) gerçekleştiren Diffie-Hellman öncülüğünde hem asimetrik kriptosistemlere hem de tezin temelini oluşturan anahtar anlaşma protokollerine buradan başlanacaktır:

3.5.1. Diffie-Hellman

1976 senesinde Whitfield Diffie ve Martin Hellman tarafından yayınlanan "New Directions in Cryptography" isimli makalede açık anahtarlı kriptosistemden ilk kez bahsedilirken bu sisteme verilen ilk örnek de kendi isimlerini verdikleri bu anahtar degiş tokuşu/anlaşması olmuştur (Diffie and Hellman, 1976). İki tarafın(daha önce iletişim kurmamış taraflar da olabilir) karşılıklı ve ortaklaşa, güvensiz bir kanal üzerinden, ortak gizli bir anahtar elde etmelerine imkan sağlayan bir süreç ve sistemdir.

Diffie-Hellman ortak gizli anahtar oluşturma algoritması ayrık logaritma problemi zorluğuna dayalı bir sistem üzerine inşa edilmiş olup emniyeti ise çok büyük asal sayıları tercih etmekten geçen açık anahtarlı sistemin ilk yapı taşıdır.

Diffie–Hellman algoritması sayesinde simetrik şifreleme sistemleri için problem durumunda olan anahtar dağıtımı ve gizli anahtara sahip çıkma sorununa çare olmuştur denilebilir. DH algoritması bu raddede gizli anahtarlı şifreleme için ortak gizli anahtarın tayin edilmesinde kullanılmaktadır. Diffie-Hellman algoritması renkler yardımı ile şu şekilde açıklanabilir (Karataş, 2019):



Şekil 3.6. Diffie-Hellman Anahtar Anlaşması

Sarının ilk ve herkese açık ortak bir renk olarak seçimi ile birlikte; Ayşe gizli renk olarak turuncuyu, Bora ise maviyi seçmiştir. Her iki taraf da kendi karışımlarını meydana getirip renkleri takas ederler. Değiş tokuşla gelen ilk karışım gizli renklerin tekrar eklenmesi yardımıyla renk değişimine uğrar ve ikisinde de aynı olur. Artık şifrelemede kullanılabilecek ortak anahtar elde edilmiştir denilir.

Renklerle açıklanılmaya çalışılan sistemin çalışma prensibi ise şu şekilde verilmiştir:

p, yeterince büyük bir asal sayı ve g, mertebesi q olan $\mathbb{Z}_p^*$'da bir primitif(ilkel) kök asal sayısı olmak üzere, $\mathbb{Z}_p^*$'da ayrık logaritma probleminin (DLP) çözümü mümkün görünmesin. p ve g de herkesçe bilinsin.
Haberleşen taraflardan A ve B, ortak anahtar oluşturabilmek üzere aşağıdaki adımları izlerler:

* A, $0 \leq a \leq p-2$ eşitsizliğini sağlayan rastgele bir a sayısını belirler.
  $c = g^a \mod p$'yi hesaplayıp B'ye gönderir.
* B, $0 \leq b \leq p-2$ eşitsizliğini sağlayan rastgele bir b sayısını belirler.
  $d = g^b \mod p$'yi hesaplayıp A'ya gönderir.

A, ortak anahtar olan **K** yı şu şekilde hesaplar: $\mathbf{K} = d^a = (g^b)^a \mod p$.

B, ortak anahtar olan **K** yı şu şekilde hesaplar: $\mathbf{K} = c^b = (g^a)^b \mod p$.

Böylece A ve B tarafları, **K** ortak anahtarı üzerinde anlaşmış olurlar. Oluşturulan ortak gizli anahtar simetrik şifrelemede kullanılmak üzere türetilmiş olabilir.

A ve B kişilerinin güvenli mesaj trafiği için p asal sayısını takriben 1000 bit ve asal g sayısını yaklaşık p/2 olarak atamanın daha elverişli olduğu göz önüne alınmalıdır (Turp, 2019).

Örnek 3.5.1 A ve B kişilerinin $p = 23$ asal sayısını ve $g = 5$ ilkel kökünü kullandıklarını göz önüne alarak;
A, gizli anahtar olarak $a = 2$'yi ve B, gizli anahtar olarak $b = 6$'yı seçiyor.

- A kişisi $c = g^a \mod p \equiv 5^2 \mod 23$'ü hesaplar ve $c = 2$'yi B kişisine gönderir.
- B kişisi $d = g^b \mod p \equiv 5^6 \mod 23$'ü hesaplar ve $d = 8$'i A kişisine gönderir.

A kişisi **K** ortak anahtarını $(g^b)^a \mod p \equiv 8^2 \mod 23 = 18$,
B kişisi **K** ortak anahtarını $(g^a)^b \mod p \equiv 2^6 \mod 23 = 18$
Hem A hem de B, **K** değerini 18 olarak hesaplarlar. $\mathbf{K} = 18$ değeri sadece A ve B'nin bildiği gizli anahtardır.

3.5.2. RSA

Diffie, Hellman ve Merkel'in önerilerine karşılık vermek amacıyla 1978 yılında Ron Rivest, Adi Shamir ve Leonard Adleman (Rivest et al., 1978) işbirliği ile geliştirilmiş olan RSA, kullanım oranı en yüksek olan asimetrik şifrelemedir. Güvenliği sağlama ve anahtar oluşturma algoritması olarak da ün kazanmıştır. Bu güvenliği sağlarken tamsayılarda çarpanlara kolay ayrılamama ilkesini benimser.

RSA açık anahtarlı kriptosisteminin çalışması üç aşamada gerçekleşmektedir:

A Kişisi	**B Kişisi**
Anahtarın Üretilmesi	
p ve q gibi iki büyük asal sayı seçilir. $n = p \cdot q$ dersek, $\phi(n) = (p-1) \cdot (q-1)$ ile hesaplanır. 1 ile n arasında $ebob(e, \phi(n)) = 1$ şartını sağlayan bir e sayısı seçilmelidir. $e \cdot d \equiv 1 \pmod{\phi(n)}$ için d bulunur. Genel anahtar $n = p \cdot q$ ve e paylaşılır.	
Şifreleme Aşaması	
	Plaintext (açık metin) seçilir: m. A kişisinin açık anahtarı olan (n, e) ile $c \equiv m^e \pmod{n}$ hesaplanır. Ciphertext (şifreli metin) olan c, A kişisine gönderilir.
Şifreli Metnin Çözümü	
Gizli anahtar d yardımıyla m mesajı çözülür: $m \equiv c^d \pmod{n}$ değeri hesaplanır.	

Tablo 3.2. RSA Şifreleme Yöntemi

RSA'nın emniyetini sağlanması, n değerinin yeterince büyük olmasına bağlıdır ve bu da çoğunlukla n'nin bit sayısıyla ölçümü ile ifade edilir. Otoritelerce n için önerilen değer 1024 bit olmasıdır. p ve q ise aynı bit uzunluğuna sahip olmalıdır, bu sebeple 1024 bitlik bir RSA şifrelemesi için p ve q nun takriben 512 bit olması gerektiği belirtilir.

Simetrik şifreleme için bir ortak anahtar oluşturma gayesi gütmediği için anahtar değiş tokuşuna dahil edilemez ancak TLS (Transport Layer Security)'nin ilk

versiyonlarında anahtar değişimine müsaade ederken ileriye dönük gizlilik (forward secrecy) sağlamadığından dolayı bu konuda yerini Diffie-Hellman'a bırakmıştır.

Gerçek bir şifreleme yaparken yeteri kadar büyük p ve q asal sayıları tercih edilmesine karşın anlaşılması adına aşağıda bir örnek verilmiştir.

Örnek 3.5.2 Ayşe anahtarın üretilmesi aşamasında sırasıyla,
$p = 11$ ve $q = 17$ asal sayılarını seçer.
$n = p \cdot q = 11 \cdot 17 = 187$ değerini hesaplar.
$\phi(n) = (p-1) \cdot (q-1) = 10 \cdot 16 = 160$ değerini hesaplar.
$d \cdot e \equiv 1 \pmod{\phi(n)} \equiv d \cdot 7 \equiv 1 \pmod{\phi(187)}$ için
$d = 23 \quad (23 \cdot 7 = 161 \equiv 1 \pmod{160})$ değerini hesaplar.
$1 < e < \phi(n)$ ve e ile $\phi(n)$ aralarında asal olacak şekilde bir $e = 7$ sayısını seçer.
$(n, e) = (187, 7)$ açık anahtarlarını paylaşır. $d = 23$ gizli anahtarını saklar.

Bora da Ayşe'nin açık anahtarını kullanarak m metnini şu şekilde şifreler:
$m = 12$ yi açık metin olarak seçmiştir.
$c \equiv m^e \pmod{n} \equiv 12^7 \pmod{n} \equiv 177$ hesaplar.
$c = 177$ şifreli metnini paylaşır.

Ayşe C şifreli metnini çözmek üzere bir d gizli anahtarını kullanır:
$d = 23$ gizli anahtarı ile $m \equiv c^d \pmod{n} \equiv 177^{23} \pmod{187}$ hesaplar,
$m = 12$ gizlenmiş metnine ulaşmış olur.

3.5.2.1. RSA İmza Mekanizması

RSA açık anahtarlı şifreleme sistemi dijital imzalamalar yapmak ve doğrulamak üzere tasarlanmıştır. (n, e) göndericinin genel anahtarları ve d özel anahtarı olmak üzere mesajın imzalanabilmesi için m açık metninin $\{0, 1, ..., n-1\}$ aralığında seçilmesi istenmektedir. Böylece imzalama yapılabilir.

- **İmzalama**

  Ayşe, Bora'ya m açık mesajını imzalayarak göndermek isterse gizli anahtarını kullanarak bunu gerçekleştirmelidir:

$$\alpha = m^d \pmod{n}$$

  (m, α) imzalanmış mesajı Bora'ya ulaştırılır.

- **İmza Doğrulama**

  Bora, Ayşe'den edindiği (m, α) ile imzalanmış mesajı doğrular:

$$m = \alpha^e \pmod{n}$$

  hesaplanarak m açık metnine ulaşılırsa imza doğrulaması gerçekletirilmiş olur.

3.5.3. ElGamal

ElGamal açık anahtar algoritması 1985 senesinde kriptosisteme kendi adını veren Mısırlı bilim insanı Taher ElGamal tarafından literatüre kazandırılmıştır (ElGamal, 1985). Anahtar değiş-tokuşu modunda Diffie-Hellman Anahtar Anlaşması pratiğini baz almaktadır. Sistemin güvenliği Ayrık Logaritma Problemi (DLP) ile Diffie-Hellman probleminin basit bir hesaplanışının olmayışı esasıyla sağlanır. Metnin şifrelenmesi ve dijital imzalamasını gerçekleştirebilmektedir. Ancak sorun olarak görülebilecek özelliklerinden birisi de şifrelenen mesajın uzunluğunun iki katına çıkmasıdır. Bu da dar bant genişliğinde uzun açık metinlerin şifrelemesinin zor hale gelmesi demektir.

Geçmişten şu ana kadar sistem üzerine bilinen başarılı herhangi bir atak gerçekleştirilememiştir. Bu yönde algoritmaların özelliği açık anahtardan tersinir şekilde gizli anahtarın bulunamayışına dayandığı için oldukça güvenli sayılmaktadır.

ElGamal asimetrik şifreleme sistemi üç aşamadan meydana gelmektedir (Kahate, 2013):

- Anahtarın üretilmesi
- Şifreleme aşaması
- Deşifreleme aşaması

3.5.3.1. Anahtarın üretilmesi

Ayşe, ALP ye göre çözümü zor olan olması gerektiğini düşünerek:

1. Büyük bir p asal sayısını seçer ve $\mod p$ ye göre çarpımsal grup olan $\mathbb{Z}_p^*$ 'i tanımlar. Bu grupta ilkel (primitif) kök olarak $g \in \mathbb{Z}_p^*$ sayısı seçilir.
2. $1 \leq x \leq p-2$ olacak şekilde rassal bir $x \in \mathbb{Z}$ i belirler. $g^x \mod p$ değerini hesaplar.
3. DHAA'da olduğu gibi x değerini kendi gizli anahtarı olarak saklar ve (p, g, g^x) açık anahtarlarını yayınlar.

3.5.3.2. Şifreleme aşaması

Bora, Ayşe'nin açık anahtarlarını kullanarak m açık metnini şu şekilde şifreler:

1. Açık metni niteleyen m değeri $\{0, 1, ..., p-1\}$ aralığında bir tamsayıdır.
2. $1 \leq k \leq p-2$ olacak şekilde bir $k \in \mathbb{Z}$ belirlenir. k değeri *geçici anahtar* olarak ifade edilebilir.
3. $c_1 = g^k \mod p$ ve $c_2 = m \cdot (g^x)^k \mod p$ değerlerini hesaplar.
4. $C = (c_1, c_2)$ ise m açık metninin şifrelenmiş hali olarak yayınlanır.

3.5.3.3. Deşifreleme aşaması

1. x gizli anahtarı ile $c_1^{-x} = g^{-xk} \mod p$ değerini hesaplar.

2. $c_1^{-x} \cdot c_2$ ile m açık metnine ulaşılır:

$$c_1^{-x} \cdot c_2 \equiv g^{-xk} \cdot m \cdot g^{xk} \equiv m \mod p$$

Gerçek bir şifreleme yaparken yeteri kadar büyük sayılar tercih edilmesine karşın anlaşılması adına aşağıda bir örnek verilmiştir.

Örnek 3.5.3 A şahsının $p = 167$ asal sayısını ve $\mathbb{Z}_{167}^*$ da $g = 2$ üretecini seçtiğini farz edelim. A kendi seçtiği gizli anahtarı $x = 25$ değeri ile,

$$g^x \mod p \equiv 2^{25} \mod 167 \equiv 124$$

hesaplar. Bu şekilde gizli anahtar $x = 25$ saklanır ve açık anahtar değerleri ($p = 167$, $g = 2$, $g^x = 124$) B ile paylaşılır.

B şahsı ise A'ya $m = 85$ mesajını göndermek istemektedir. Mesajını şifrelemek üzere rastgele seçtiği $k = 34$ değeri ile $C = (c_1, c_2)$ i hesaplar.

$$c_1 = g^k \mod p \equiv 2^{34} \mod 167 \equiv 28$$

ve

$$c_2 = m \cdot (g^x)^k \mod p \equiv (2^{25})^3 4 \mod 167 \equiv 58$$

değerlerini bulur. $C = (c_1 = 28, c_2 = 58)$ şifreli metni A ile paylaşılır.

B'den gelen şifreli mesaj A'nın gizli anahtarı sayesinde şu şekilde çözülebilir:

$$c_1^{-x} \equiv 28^{-25} \mod 167 \equiv (\tfrac{1}{28})^{25} \mod 167 \equiv (\tfrac{1+167}{28})^{25} \mod 167$$

$c_1^{-x} \equiv 6^{25} \mod 167 \equiv 108$ ve $c_1^{-x} \cdot c_2 = g^{-xk} \cdot m \cdot g^{xk}$

$m \equiv 108 \cdot 58 \mod 167 \equiv 85$ ile gizlenmiş metne ulaşılmış olur.

3.5.3.4. ElGamal İmza Mekanizması

RSA açık anahtarlı şifrelemesinde gerçekleştirilen imzalamada olduğu gibi gönderilen mesajın doğru kişiden gelip gelmediği ElGamal kriptosisteminde de sağlanabilmektedir. Ayşe'nin açık anahtarının (p, g, g^x) ve gizli anahtarının x olduğunu varsayalım. m açık metninin $\{0, 1, ..., n-1\}$ aralığında seçilmesi istenmektedir. Eğer bu aralıkta değilse hash (özet) fonksiyonu yardımıyla m nin $\mathbb{Z}_p$ nin bir elemanı olması gerekliliği yerine getirilir.

- **İmzalama**

  Ayşe, Bora'ya m açık mesajını imzalayarak göndermek isterse şu şekilde gerçekleştirmelidir:

  1. $1 \leq k \leq p-2$ ve $EBOB(k, p-1) = 1$ olmak üzere rastgele bir k tamsayısı seçilir. Ayrıca imzalama için r ve s değerleri hesaplanır.
  2. $r = g^k \mod p$ ve $m = x \cdot r + k \cdot s \mod (p-1)$ ve buradan $s = k^{-1}(m - r \cdot x) \mod (p-1)$ eşitlikleri inşa edilir.
  3. (m, r, s) imzalanmış mesajı Bora'ya ulaştırılır.

- **İmza Doğrulama**

  Bora, Ayşe'den edindiği (m, r, s) ile imzalanmış mesajı doğrular:

  1. $1 \leq r \leq p-1$ aralığında olup olmadığına bakılır. Bu aralıkta değilse imza geri çevrilir.
  2. $v_1 = g^m \pmod p$ ve $v_2 = (g^x)^r \cdot r^s \pmod p$ hesaplanır.
  3. $(v_1 = v_2)$ eşitliğin olması durumunda imza onaylanırken eşitliğin olmaması durumunda imzaya onay verilmez.

3.5.4. Dijital İmza Algoritması (DSA)

ABD yönetiminin NIST (Ulusal Standartlar ve Teknoloji Enstitüsü) tarafından 1991 senesinde ilk defa bir hükümetce tavsiye edilerek yürürlüğe girmiş olan ayrık logaritma problemine dayanan matematiksel imzalama algoritmasıdır. Dijital imzaların türetilmesi maksadıyla dizayn edilmiştir (FIPS, 2013). Asimetrik bir algoritma olmasına karşın diğer açık anahtarlı kriptosistemler gibi şifreleme yapma amacı gütmemektedir. Verilerin hepsini imzalamak yerine özetlerini imzalayarak kısa boyutlu sayısal imzaların oluşumunu sağlamaktadır. Seçilen anahtarın boyutları değiştikçe imza boyutları da doğru orantılı olarak değişir.

DSA algoritması üç aşamadan oluşmaktadır (Kırımlı, 2007; Kaçmaz vd., 2016):

- Anahtarın üretilmesi
- İmza üretim aşaması
- İmzanın doğrulanması

3.5.4.1. Anahtarın üretilmesi

1. Öncelikle güvenli sayılan MD, SHA, vb. hash fonksiyonlarından biri seçilir.
2. Yeteri kadar büyük olmak üzere asal bir q sayısı seçilir.
3. $p-1 \pmod{q} \equiv 0$ denkliğini veren asal bir p sayısı seçilir.
4. $1 < g < p$ aralığı ve $g^q \mod p \equiv 1$ de mertebesi g olan bir ilkel kök seçilir. $1 < h < p-1$ aralığında keyfi bir h değeri için $g = h^{(p-1)/q} \mod p$ yazılır.
5. $0 < x < q$ aralığında gizli kalmak koşuluyla keyfi bir x sayısı belirlenir.
6. $y = g^x \mod p$ değeri hesaplanır.

Ayşe'nin açık anahtarları (p, q, g, y) ve gizli anahtarı x olur.

3.5.4.2. İmza üretim aşaması

1. m iletilmek istenen açık metin ve $Hash(m) = h$ mesajın özeti olarak ifade edilir.
2. $0 < k < q$ aralığında keyfi bir gizli k sayısı seçilir.
3. $r = (g^k \mod p) \mod q$ ve $s = (k^{-1}(Hash(m) + x \cdot r) \mod q)$ değerini hesap eder.

- $r = 0$ ve $s = 0$ olduğu durumlarda k değeri yinelenerek işlem tekrar edilir.
- (r, s) metnin imzalarıdır.

Ayşe açık metni olan m ve metnin dijital imzası olan $(m, (r, s))$ yi Bora'ya gönderir.

3.5.4.3. İmzanın doğrulanması

Bora, Ayşe'den aldığı $(m, (r, s))$ iletisinin gerçekten ondan geldiğini doğrulamak üzere açık anahtarlar (p, q, g, y) yi kullanarak sırasıyla şu hesaplamaları gerçekleştirir:

1. r ve s değerleri $0 < r < q$ ve $0 < s < q$ aralığında değilse imza geri çevrilir.
2. $s \cdot w = 1 \mod q$ yani $w = s^{-1} \mod q$ için w değerini hesap eder.
3. $u_1 = Hash(m) \cdot w \mod q$ yani $u_1 = h \cdot w \mod q$ değerini hesap eder.
4. $u_2 = r \cdot w \mod q$ değerini hesap eder.
5. $v = (g^{u_1} \cdot y^{u_2} \mod p) \mod q$ değerini hesap eder.

Eğer $v = r$ ise imza doğrulaması gerçekleşir.

4. ANAHTAR ANLAŞMASI PROTOKOLLERİ

Son yıllarda üzerinde çalışılan asimetrik şifreleme (public key) algoritmaları arasında bir sunucudan yardım almadan anahtar oluşturulması, anahtar taşınımından daha popüler hale gelmiştir. Anahtar anlaşması da bu noktada adından anlaşılacağı gibi bir oturum anahtarı oluşturabilmek için tarafların işbirliği yaptığı bir süreçtir.

Anahtar anlaşması protokolü/değişimi veya mekanizması, ortak bir sırrın güvenli aktarımı adına iki veya daha fazla tarafça, her biri kendi mesajını şifrelemek için bilgilerinin bir ürünü olarak kusursuz şekilde anahtar türettiği ve tarafların ortaya çıkan değeri önceden belirleyemeyeceği önemli bir kuruluş tekniğidir (Katz et al., 1996).

Anahtar anlaşmalarının anahtar taşınımından daha kayda değer olduğu ve anahtar taşımaya göre daha yüksek-kaliteli rastgele sayı anahtarları üretebileceği yönünde sezgisel bir algı vardır. Rastgelelik kaynağının yetersizliği tüm sistemin güvenliğini etkileyebilmektedir (Özkaynak, 2016). Ayrıca Diffie-Hellman protokolüne bağlı anahtar anlaşmaları ile şifrelemede ileri yönde gizlilik (forward secrecy) elde edilebilir.

4.1. Anahtar Anlaşma Protokollerinin Özellikleri ve Güvenlik İhtiyaçları

Eskiden beridir anahtar anlaşmalarında bazı eksikliklerin yer aldığı söylenmiştir. Sıkı çalışılarak ortaya konan birçok algoritma dahi bazen deşifreleme tekniklerine gerek kalmadan çözülebilmiştir. Anahtar anlaşma protokollerinin saldırılara en az seviyede maruz kalması adına aşağıdaki özelliklerin bilinmesi ve güvenlik ihtiyaçlarının yerine getirilmesi istenilmektedir:

4.1.1. Bilinen Oturum Anahtarları (Known Session Keys)

Taraflar arasındaki protokolün her bir eylemi sonunda benzersiz bir gizli anahtarın ortaya konulması sağlanmalıdır (Blake-Wilson and Menezes, 1998). Bu oturum anahtarının açığa çıkması durumunda inaktif saldırgan, ilerideki oturum anahtarlarını öğrenir veya benzeşimlerini oluşturmaya kalkarsa bilinen anahtar emniyetinin korumasızlığından söz edilir (Katz et al., 1996). O yüzden bir anahtarın açıklığı öteki oturum anahtarlarına tehdit unsuru olmamalıdır (Vesterås, 2006).

4.1.2. İleriye Dönük Mükemmel Gizlilik (Perfect Forward Secrecy)

Şifrelemede kullanılmak üzere önceden belirlenen gizli anahtarın açığa çıkması halinde geçmişe dönük oturum anahtarının gizliliği tehlikeye girmemiş olur (Katz et al., 1996). Böylece olası kaydedilmiş eski konuşmaların içeriğine de ulaşılamaz. Zamanla gizli anahtarın açığa çıkma ihtimaline karşı gizli anahtarın değişimi aşamasında oturum anahtarının da kendinden evvelkilerden farklı olmasına özen gösterilerek lalettayin edilmesi gerekir (Sağlam, 2002; Çakmak, 2018).

4.1.3. Bilinmeyen Anahtar Paylaşımı (Unknown Key Share)

Oturum anahtarını oluşturan taraflardan birine kendi bilgisi dahilinde olmadan düşman tarafından bir başkasının anahtarını kabullendirerek arzu edilmeyen ortak anahtarın oluşumuna sebebiyet verilmemelidir (Blake-Wilson and Menezes, 1999; Altuncu, 2019).

4.1.4. Anahtar Mutabakatında Kimliğe Bürünmeye Karşı Dayanım (Key-Compromise Impersonation)

Protokolü oluşturan taraflardan birinin gizli anahtarının ortaya çıktığı farzedilirse saldırgan bu anahtar yardımıyla ifşa olan tarafın kimliğine bürünebilir. Ancak özel

anahtarın yitirilişi saldırganın diğer taraflar ile iletişim kurmasına ve onların yerine geçmesine imkan tanımamalıdır (Blake-Wilson et al., 1997).

4.1.5. Bilginin Yitirilmesi (Loss Of Information)

Protokolün çalışması esnasında ortak anahtar oluşturma gayreti içinde iken herhangi bir bilginin düşman eliyle yakalanılması istenilmemektedir ve bu gibi durumlarda protokol emniyeti zaafa uğramamalıdır.

4.1.6. Mesaj Bağımsızlığı (Message Independence)

Taraflar arasında yürütülen protokolün özgün akışları birbiriyle ilişkili değildir (Jirwan et al., 2013).

4.1.7. Anahtar Kontrolü (Key Control)

Oturum anahtarının güncelliğinin korunabilmesi için protokolün taraflarının kendi hevesledikleri değeri birbirlerine dayatmamaları istenir (Boyd et al., 2003).

4.2. Temel (Anonim) Diffie-Hellman Anahtar Anlaşması

Temel Diffie-Hellman algoritmasında taraflar açık anahtarlarını birbirlerine kimlik doğrulaması olmaksızın göndermektedirler (Kaplan, 2000). Anahtar emniyeti açısından en zayıf olan seçenek olarak değerlendirilir. Anonim Diffie-Hellman dışındaki anahtar alışverişlerinde mesaj sertifikasyonu gereklidir. Sertifikalı Diffie-Hellman kullanıldığında ise sertifika iletisi, sunucunun anahtar alışverişinde kullanılan iletisi gibi davranır (Kaçmaz vd., 2016).

Protokolün çalışması aşağıdaki gibidir:

Ayşe		**Bora**
$x \in_r \mathbb{Z}_p^*$		$y \in_r \mathbb{Z}_p^*$
$X \equiv g^x \mod p$		$Y \equiv g^y \mod p$
	$\xrightarrow{X = g^x}$	
	$\xleftarrow{Y = g^y}$	
$\mathbf{K} \equiv Y^x = (g^y)^x \mod p =$	g^{xy}	$= \mathbf{K} \equiv X^y = (g^x)^y \mod p$

Tablo 4.1. Temel Diffie-Hellman Anahtar Anlaşma Protokolü

1. p, yeterince büyük bir asal sayı ve g, $\mathbb{Z}_p^*$'da bir primitif(ilkel) kök asal sayısı olan üreteç olarak seçilir.
2. Ayşe rassal olarak belirlediği x gizli anahtarı ile $X \equiv g^x \mod p$ yi hesap eder ve Bora ile bunu paylaşır.
3. Bora rassal olarak belirlediği y gizli anahtarı ile $Y \equiv g^y \mod p$ yi hesap eder ve Ayşe ile bunu paylaşır.
4. Ayşe, Bora'dan elde ettiği açık anahtarı kendi gizli anahtarı ile üstelleştirir ve $\mathbf{K} \equiv Y^x = (g^y)^x \mod p$ ortak anahtarını oluşturur.
5. Bora da Ayşe'den elde ettiği açık anahtarı kendi gizli anahtarı ile üstelleştirir ve $\mathbf{K} \equiv X^y = (g^x)^y \mod p$ ortak anahtarını oluşturur.

Anahtar anlaşma protokolü ayrık logaritma probleminin zor çözümünü dayanak noktası olarak benimsemesi tarafların iletişiminin üst düzeyde olmasını sağlıyor

gibi gözükse de sertifikasyon, dijital imza, vb. yönlerden eksikliği ortadaki adam saldırısına karşı savunmasız kalınabileceğinin göstergesidir.

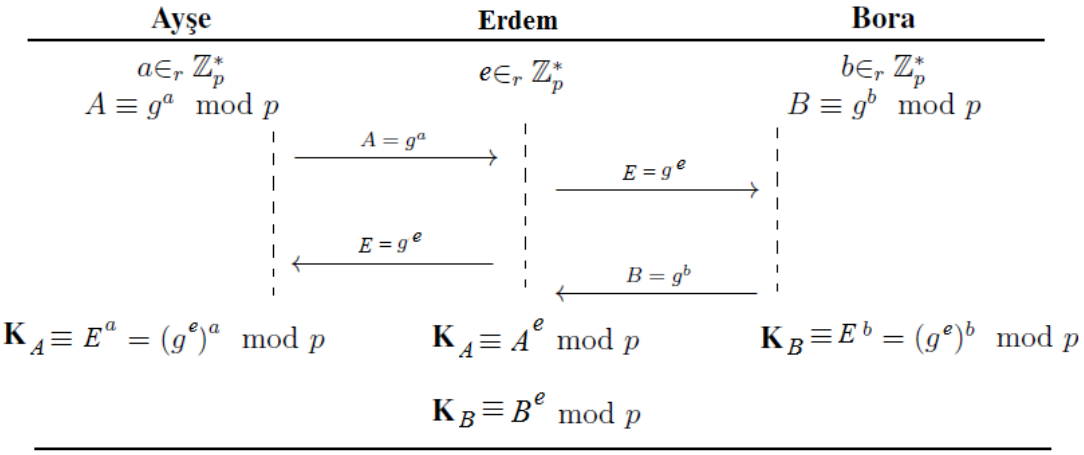
Diffie-Hellman protokolü statik ve geçici iki versiyona sahiptir. İlkinde taraflar statik genel anahtarları, diğerinde ise geçici genel anahtarları birbirleri arasında degiştirirler (Al Sultan et al., 2003).

4.2.1. Ortadaki Adam Saldırısı

Bir düşmanın iletişim halinde olan ya da yeni bir iletişime başlayacak olan tarafların ortasına kendini konumlandırarak izlediği mesaj trafiğini tarafların haberi olmadan önce kendi yönetir vaziyete geçmesi ve kullanıcıların mesajlarını ele geçirme, mesajları değiştirme, bozma, onlara karşı taraf gibi davranma algısı yaratması açısından açık anahtarlı protokoller için tehdit unsuru oluşturmasıdır.

Temel Diffie-Hellman protokolünün sınırlılıklarından bir tanesi de aslında en sıkıntılı durumu olarak görülen kimlik doğrulamasının olmayışıdır. Bu da protokolü şifrelemede yer alan taraflar arasında kalan ortadaki adam saldırısına açık bir durum haline getirir (Katz et al., 1996).

Saldırıyı düzenlemek isteyen düşmana Erdem ismini verelim. Erdem'in prensibi Ayşe'ye Bora gibi, Bora'ya da Ayşe gibi gözükerek ortak anahtarı kendinde oluşturup iletilmek istenen mesajı ifşa etmektir.



Şekil 4.1. Ortadaki Adam Saldırısı

1. Ayşe rassal gizli anahtarı a ile $A \equiv g^a$ yı hesaplar ve Bora'ya gönderir.
2. Davetsiz misafir olan Erdem araya girerek kendi gizli anahtarı e ile oluşturduğu $E \equiv g^e$ yi Ayşe'nin mesajı yerine gönderir.
3. Bora rassal gizli anahtarı b ile $B \equiv g^b$ yı hesaplar ve Ayşe'ye gönderir.
4. Davetsiz misafir olan Erdem araya girerek kendi gizli anahtarı E yi Bora'nın mesajı yerine gönderir.
5. Ayşe, Bora zannettiği Erdem'den aldığı g^e ile $\mathbf{K}_A = (g^a)^e$ yi hesaplar.
6. Erdem, Ayşe ve Bora'nın açık ettiği g^a ve g^b değerleri ile olması gereken $\mathbf{K}_A = \mathbf{K}_B = g^{ab}$ yi hesaplar.
7. Bora, Ayşe zannettiği Erdem'den aldığı g^e ile $\mathbf{K}_B = (g^b)^e$ yi hesaplar.
8. Erdem kötü niyetli olarak iletişimin merkezine geçtiği için saldırısını aynı şekilde yineleyebilir.

Ayşe ve Bora, düşmanın varlığını hissetmezler ve önlem almazlar ise tekrar saldırıya uğrayabilirler. Bundan dolayı anonim Diffie-Hellman kullanımı risk teşkil ettiği için kullanıcıların kendini tanıtmalarını sağlayan statik anahtarlara, dijital imzalara veya bir otorite tarafından kabul edilmiş sertifikaları kullanmaları gerekmektedir. Güvenlik ihtiyacı dolayısıyla Sertifikalı Diffie-Hellman anahtar anlaşması türetilmiştir.

4.3. Sertifikalı Diffie-Hellman Anahtar Anlaşması

Diffie-Hellman Anahtar Anlaşmasının geçici kullanımı, pasif düşmanların bulunduğu durumlarda zımnen anahtar kimlik doğrulamasını gerçekleştirse de aktif saldırılara karşı tek başına herhangi bir yarar sağlamaz (Blake-Wilson and Menezes, 1998). Bu durumda ortadaki adam saldırısına açık hale gelip mesajların değiştirilebilir olması olasılığı artar. Sıkıntılı durumun düzeltilmesi için ancak güvenilir bir CA (Sertifika Otoritesi) tarafından saldırıdan korunmak amacıyla mesajların DSS veya RSA sertifikası onaylanmış genel anahtar yardımıyla aşılabilir (Koçoğlu, 2004; Dierks and Rescorla, 2006).

Ayşe		**Bora**
$a \in \mathbb{Z}_p^*$ $Cert(A) = Sertifika$		$b \in \mathbb{Z}_p^*$ $Cert(B) = Sertifika$
$A \equiv g^a \mod p$		$B \equiv g^b \mod p$
	$\xrightarrow{Cert(A), A}$	
	$\xleftarrow{Cert(B), B}$	
$\mathbf{K} \equiv B^a = g^{ba}$		$\mathbf{K} \equiv A^b = g^{ab}$

Tablo 4.2. Sertifikalı Diffie-Hellman Anahtar Anlaşma Protokolü

Protokolün ilerleyişi Temel Diffie-Hellman ile benzeşik olduğundan adımları tek tek vermek yerine açık anahtarlar A ve B türetilip güvenli kabul edilen bir CA ile sertifikasını imzalattıktan sonra iletişimin devam etmesi gerektiği görülür. Bu şekilde PKI ile de organik bir bağ kurulmuş olur.

Protokolün her çalışması esnasında farklı anahtar çiftlerinin kullanımı "ileriye dönük mükemmel gizliliği" sağladığı için herhangi bir şekilde bir anahtarın açığa çıkması geçmiş mesajların güvenliğini tehlikeye sokmaz. Ancak yine de bazı güvenlik gereksinimleri vardır. Bundan dolayı da Diffie-Hellman bazlı bir çok anahtar anlaşması türetilmiştir.

Temel Diffie–Hellman protokolünde üretilmiş olan $\mathbf{K} = g^{xy}$ ortak anahtarı, sadece keyfi olarak seçilen değerlere bağlı olduğu için ve sadece ortak anahtarı üretilene kadar ömrünü sürdürdüğünden dolayı kısa vadeli(kısa ömürlü) Diffie–Hellman anahtarı olarak isimlendirilmiştir. Ancak, iletişimin tarafları (a, b) gibi iki uzun vadede gizli birer anahtar belirleyip $A = g^a$ ve $B = g^b$ şeklinde uzun vadeli ortak anahtarlarını değiştirirlerse, ikisinin de $\mathbf{K} = g^{ab}$ ortak anahtarı hesap edilmiş olur. Bu durum da herhangi bir keyfi seçime bağlı olmadığından uzun vadeli (statik) Diffie-Hellman anahtarı olarak isimlendirilir. Bu bölümde işlenecek olan Diffie-Hellman'ı baz alan protokoller, kendilerinden istenen özellikleri yerine getirebilmek adına sertifikalı anlaşmaların maliyeti sebebiyle statik ve geçici anahtarı karışım haline getirdikleri tasarımlar geliştirmişlerdir (Boyd et al., 2003).

4.4. MQV Anahtar Anlaşma Protokolü

Menezes, Qu ve Vanstone tarafından MQV adını verdikleri protokol, 1995 senesinde türetilmiş daha sonra 1998'de Law ve Solinas tarafından revize edilmiştir (Kaliski Jr, 2001; Aumasson, 2017). Çalışma prensibi şu şekildedir:

Ayşe		**Bora**
$a \in \mathbb{Z}_q$, $x \in_r \mathbb{Z}_q$		$b \in \mathbb{Z}_q$, $y \in_r \mathbb{Z}_q$
Statik anahtarlar: (A, a)		**Statik anahtarlar:** (B, b)
$A = g^a \mod q$	$\xrightarrow[dogrulama]{A}$	
	$\xleftarrow[dogrulama]{B}$	$B = g^b \mod q$
Geçici anahtarlar: (X, x)		**Geçici anahtarlar:** (Y, y)
$X = g^x \mod q$		
$\bar{X} = 2^\ell + (X \mod 2^\ell)$		
$S_A = (x + \bar{X}a) \mod q$	$\xrightarrow[gecici(ephemeral)]{X, S_A}$	
		$Y = g^y \mod q$
		$\bar{Y} = 2^\ell + (Y \mod 2^\ell)$
	$\xleftarrow[gecici(ephemeral)]{Y, S_B}$	$S_B = (y + \bar{Y}b) \mod q$
($\ell = (\lceil \log_2(q) \rceil + 1)/2$ ve $\ell = \lvert q \rvert / 2$ olmak üzere)		
$t_A = Y \cdot B^{\bar{Y}}$		$t_B = X \cdot A^{\bar{X}}$
$\mathbf{K} = {t_A}^{S_A} = (Y \cdot B^{\bar{Y}})^{x + \bar{X}a}$		$\mathbf{K} = {t_B}^{S_B} = (X \cdot A^{\bar{X}})^{y + \bar{Y}b}$
$\mathbf{K} = (g^y \cdot g^{b \cdot \bar{Y}})^{x + \bar{X}a}$ =	$g^{(x + a \cdot \bar{X}) \cdot (y + b \cdot \bar{Y})}$	= $(g^x \cdot g^{a \cdot \bar{X}})^{y + \bar{Y}b} = \mathbf{K}$
($h \leq 4$ ve $h = \frac{\lvert G \rvert}{q}$ da h kofaktör olmak üzere)		
$X = xP$	$\xrightarrow{X}$	
	$\xleftarrow{Y}$	$Y = yP$
$\mathbf{K} = h \cdot S_A \cdot S_B$		$\mathbf{K} = h \cdot S_B \cdot S_A$
$\mathbf{K} = h \cdot (x + \bar{X}a) \cdot (y + \bar{Y}b)$		$\mathbf{K} = h \cdot (y + \bar{Y}b) \cdot (x + \bar{X}a)$

Tablo 4.3. MQV Anahtar Anlaşma Protokolü

Diffie-Hellman kriptosisteminin ilk sunulan halinin kimlik doğrulamayı sağlayamamasından dolayı zaman içinde tarafların sertifikalarını teyit etme veya dijital imzalara başvurma görüşü oluşmuştur (Yang et al., 2005). Zaman içinde algoritmalar üzerinde oynamalar ve nitelik kazandırmalarla açık anahtarlı sistemler daha aktif kullanılır hale gelmiştir. Yine de iletişimin taraflarından birinin yerine iletişimi aksatmayı planlayan bir saldırganın geçtiği ve diğer tarafın bundan haberinin olmadığını düşünüldüğünde Diffie-Hellman anahtar anlaşmasının temel hali yetersiz kalabilmektedir. Bu noktada Geçici (Kısa Süreli) Diffie-Hellman algoritmasını eliptik eğrilerle bütünleştiren MQV protokolü önemli bir yer tutmaktadır.

Tablo 4.3 de açıklık getirilmesi gereken parametreler şunlardır:

- A, B kimlik doğrulamada kullanılan statik açık anahtarlardır
- a, b kimlik doğrulamada kullanılan statik gizli anahtarlardır.
- X, Y kısa süreli (geçici) açık anahtarlardır.
- x, y kısa süreli (geçici) gizli anahtarlardır.
- h, q boyutlu bir G devirli grubu için Lagrange teoremi gereğince; alt grubun mertebesinin grubun mertebesini bölmesiyle oluşan kofaktördür. Kriptografi uygulamalarında $h \leq 4$ alınır (Van Tilborg and Jajodia, 2014).
- **K** oturum (ortak/oluşturulan) anahtarıdır.

MQV kriptosistemi için G'nin bir sonlu cisim üzerinde tanımlı eliptik eğri grubunun asal dereceli bir alt grubu (toplama notasyonu kullanılır) ya da bir sonlu çarpımsal grubunun asal mertebeden bir alt grubu (neredeyse hep çarpma notasyonu kullanılır) olduğu düşünülür (Brown, 2014). Bu gruplar ayrık logaritma gerektiren kriptografik şemalar için en yaygın kullanılan gruplardır.

İletişimin tarafları statik anahtarlarını daha önceden belirlenmiş (pre-established), dijital imzalara başvurmuş ya da sertifika kullanmış olmakla kendi kimliklerini birbirine karşı ispatlamış olurlar. Statik ve geçici değerlerin bir arada kullanılıyor olması da bir anlamda "örtük" kimlik doğrulaması anlamındadır (Blake et al., 2005).

Bir yüzey söz konusu ise eğri üzerindeki noktaların koordinatları ikililerden oluşmaktadır. $R = (x, y)$ eliptik eğri üzerindeki bir nokta olmak üzere $\ell = ([\log_2(q)] + 1)/2$ de ℓ, q bitinin yarı eğri uzunluğunda iken $\bar{R} = 2^{\ell} + (X \mod 2^{\ell})$ [ya da Y] şeklinde özel işlem tanımlanır ve X ve Y değerlerinin ℓ bit uzunluklarıdır.

$\bar{X}$ veya $\bar{Y}$ nin hesaplanmasında $\ell = |q|/2$ uzunluğu baz alındığı için bir "yarım üs"(half exponentiation) sayıldığı not edilmelidir (Krawczyk, 2005).

S_A ve S_B örtük imzalarının hesaplanması ile geçici açık değerler değiştirilir. G grubunun tanımlı olduğu çalışma alanına göre çarpma notasyonu kullanımlarında t_A ve t_B hesaplanarak DH deki gibi üs almadan, toplama notasyonu kullanımlarında ise eliptik eğri üzerindeki P noktasının türettiği (X, Y) açık anahtarlarının h kofaktöründen yararlanılarak **K** ortak anahtarını oluşturduğu görülür.

K ortak anahtarını elde etmek isteyen saldırgan x i öğrenir ama a yı öğrenemezse **K** yı hesaplayamaz. Bundan dolayı bir kullanıcının iki ayrı (x ve a) gizli anahtarı olmalıdır. Ayrıca bu durumun ileriye yönelik gizliliğe de katkısı olur (Krawczyk, 2005). **K** anahtarı oluşturulduktan sonra geçici özel anahtarlar yok edilir (Brown, 2014). Protokolün her çalışması farklı oturum anahtarlarını vermek üzerine tasarlanmıştır.

MQV protokolü, bir ortak anahtarın oluşumunu ve kimlik doğrulamasını aynı anda yapmaktadır. Eliptik eğriler üzerinde türetildiği için de ECMQV (Elliptic Curve MQV) ismini alır. Aynı zamanda bu özelliklerinden olayı otoritelerden bazılarına göre DH protokollerinden en verimlisi olduğu düşünülmektedir. Hatta kendine en yaklaşan üç üs alma gerektirdiği halde daha az güvenlik verdiği düşünülen Unified (Birleştirilmiş) Model yaklaşımıdır. Ancak kusursuzdur denilemez öyle ki saldırılar sırasında UKS (Unknown Key Share/Bilinmeyen Anahtar Paylaşımı bkz.4.1.3), PFS (Perfect Forward Secrecy/İleriye Dönük Mükemmel Gizlilik bkz.4.1.2) ve KCI (Key Compromise Impersonation/Anahtar Mutabakatında Kimliğe Bürünmeye Karşı Dayanım bkz.4.1.4) özelliklerini sağlamada savunmasız kalabilmektedir (Krawczyk, 2005).

Tasarımına ve güvenliğine karşın MQV kullanımı, DH patentleri ile çelişmesi

ve kimliği doğrulanmış DH algoritmalarına oranla daha karmaşık olması sebebiyle pek yaygınlaşamamıştır (Aumasson, 2017).

Canetti-Krawczyk'nin MQV kriptanalizi sonucu temel güvenlik amaçlarının çoğunun geçersiz kılındığı ve çeşitli saldırılara karşı da dirençsiz olduğunu görülmüştür. Böylece güvenlik açıklarını kapatmaya yönelik daha üstün performans ve işlevsellik sağlayan HMQV sunulmuştur. MQV den ayrılan özelliği paylaşılan değerlerin özet değerleri alınarak geçici değerlerle iletimi olarak gösterilebilir. UKS ve diğer tehditlere karşı dayanımı artsa da PFS yi sağlamada zorluk çektiği belirtilmiştir (Çakmak, 2018).

4.5. MTI Anahtar Anlaşma Protokolü

Matsumoto, Takashima ve Imai tarafından Diffie-Hellman anahtar anlaşması protokolünün örtük kimlik doğrulamasını sağlayabilmek adına 1986 senesinde geliştirilmiş bir anahtar anlaşma protokolüdür. ElGamal tek geçişli anahtar anlaşmasına paralel bir uygulama olarak görülebilmektedir. A, B ve C şeklinde üç hali bulunur. MTI A(0), MTI B(0), MTI C(0) ve MTI C(1) bu ailenin üyeleri olup en çok tercih edilenleri MTI A(0) ve MTI C(0) durumlarıdır (Dutta and Barua, 2005; Wang et al., 2006).

Herhangi bir A protokolü için tanımlanan k değeri ile A(k), protokol sınıfını ifade etsin. $k = 0$ durumu A, B ve C sınıflarının her biri için temel protokol anlamına gelir (Boyd et al., 2003).

MTI protokolleri, anahtar onayına (protokolün kullanıcılardan birinin diğer kullanıcının şifreli mesajı çözebilecek özel anahtarlara sahip olduğundan emin olması) gerek kalmadan anahtar doğrulaması sağlar. Tarafların protokolün çalışması esnasında üslendikleri rolleri simetriktir (her bir taraf doğrudan benzer işlemleri yürütür) (Katz et al., 1996).

Sistemin güvenliği, Diffie-Hellman probleminin zor hesap edilmesine indirgenebilir iken uzun vadeli (statik) ve geçici (kısa ömürlü) girdileri tek bir den-

klemde birleştirerek kimlik doğrulamayı Diffie-Hellman algoritmasına dahil ederler (Boyd et al., 2003).

MTI protokolleri çift geçişli (two-pass) anahtar anlaşması olarak da adlandırılır çünkü şemanın her oturumunda gerçekleştirilen yalnızca iki ayrı bilgi aktarımı vardır. Buna karşılık, Başlık 4.6 de daha ayrıntılı açıklanacak olan STS anahtar anlaşması da üç geçişli bir protokoldür (Stinson and Paterson, 2018).

MTI protokollerinin çalışabilmesi için büyük bir p asal saysı seçilir ve $\bmod p$ ye göre çarpımsal grup olan $\mathbb{Z}_p^*$ 'i tanımlanır. Bu grupta ilkel (primitif) kök olarak $g \in \mathbb{Z}_p^*$ sayısı seçilir.

4.5.1. MTI A(0) Anahtar Anlaşması

Protokol şeması ve çalışma prensibi şu şekilde verilmiştir:

Ayşe		**Bora**
$a \in \mathbb{Z}_p^*$, $x \in_r \mathbb{Z}_p^*$		$b \in \mathbb{Z}_p^*$, $y \in_r \mathbb{Z}_p^*$
Statik anahtarlar: A, a		**Statik anahtarlar:** B, b
$A = g^a \mod p$	$\xrightarrow[dogrulama]{A}$	
	$\xleftarrow[dogrulama]{B}$	$B = g^b \mod p$
Geçici anahtarlar: X, x		**Geçici anahtarlar:** Y, y
$X = g^x \mod p$	$\xrightarrow{X}$	
	$\xleftarrow{Y}$	$Y = g^y \mod p$
$\mathbf{K} = B^x \cdot Y^a = (g^b)^x \cdot (g^y)^a$		$\mathbf{K} = A^y \cdot X^b = (g^a)^y \cdot (g^x)^b$
$\mathbf{K} = g^{bx+ay}$		$\mathbf{K} = g^{ay+bx}$

Tablo 4.4. MTI A(0) Anahtar Anlaşma Protokolü

1. Ayşe doğrulama için uzun süreli anahtarlar seçilerek $A = g^a \mod p$ açık anahtarını hesaplayıp gönderir. Bora da $B = g^b \mod p$ açık anahtarını hesaplayıp gönderir.

2. Ayşe $1 \leq x \leq p-2$ olacak şekilde rassal bir $x \in \mathbb{Z}$ i belirler. $g^x \mod p$

değerini hesaplayıp gönderir. Bora da $1 \leq y \leq p-2$ olacak şekilde rassal bir $y \in \mathbb{Z}$ i belirler. $g^y \mod p$ değerini hesaplayıp gönderir.

3. Ayşe ve Bora, Diffie-Hellman uyarınca **K** ortak anahtarını hesaplar.

Şimdi de MTI ailesinin B sınıfı incelenecektir:

4.5.2. MTI B(0) Anahtar Anlaşması

Protokol şeması ve çalışma prensibi şu şekilde verilmiştir:

Ayşe		**Bora**
$x \in_r \mathbb{Z}_p^*$		$y \in_r \mathbb{Z}_p^*$
$A \equiv Y^a = (g^y)^a \mod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B \equiv X^b = (g^x)^b \mod p$
$\mathbf{K} \equiv B^{x^{-1}} \cdot g^a = ((g^x)^b)^{x^{-1}} \cdot g^a$		$\mathbf{K} \equiv A^{y^{-1}} \cdot g^b = ((g^y)^a)^{y^{-1}} \cdot g^b$
$\mathbf{K} = g^{b+a}$		$\mathbf{K} = g^{a+b}$

Tablo 4.5. MTI B(0) Anahtar Anlaşma Protokolü

1. Uzun süreli ve geçici anahtarlar seçilerek hesaplamalar yapılır ve taraflarca birbirlerine gönderilir.

2. Son alınan açık anahtarla statik gizli anahtarlarla üstellenerek gönderilir.

3. Taraflar statik gizli anahtarlarının çarpmaya göre terslerinin üslerini g^a ve g^b çarpımı yardımıyla hesaplar ve **K** ortak anahtarını elde eder.

Son olarak de MTI ailesinin C sınıfı incelenecektir:

4.5.3. MTI C(0) Anahtar Anlaşması

Protokol şeması ve çalışma prensibi şu şekilde verilmiştir:

Ayşe		**Bora**
$x \in_r \mathbb{Z}_p^*$		$y \in_r \mathbb{Z}_p^*$
$A \equiv Y^a = (g^y)^a \mod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B \equiv X^b = (g^x)^b \mod p$
$\mathbf{K} \equiv (B^{x^{-1}})^a = ((g^x)^b)^{x^{-1} \cdot a}$		$\mathbf{K} \equiv (A^{y^{-1}})^b = ((g^y)^a)^{y^{-1} \cdot b}$
$\mathbf{K} = g^{ba}$		$\mathbf{K} = g^{ab}$

Tablo 4.6. MTI C(0) Anahtar Anlaşma Protokolü

1. Uzun süreli ve geçici anahtarlar seçilerek hesaplamalar yapılır ve taraflarca birbirlerine gönderilir.
2. Son alınan açık anahtarla statik gizli anahtarlarla üstellenerek gönderilir.
3. Taraflar statik gizli anahtarlarının çarpmaya göre tersleri ve geçici gizli anahtarlarının çarpımını üstelleyerek **K** ortak anahtarını hesaplar.

4.5.4. MTI C(1) Anahtar Anlaşması

Protokol şeması ve çalışma prensibi şu şekilde verilmiştir:

Ayşe		**Bora**
$x \in_r \mathbb{Z}_p^*$		$y \in_r \mathbb{Z}_p^*$
$A \equiv Y^{ax} = (g^y)^{ax} \mod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	$B \equiv X^{by} = (g^x)^{by} \mod p$
$\mathbf{K} \equiv B^a = ((g^x)^{by})^a$		$\mathbf{K} \equiv A^b = ((g^y)^{ax})^b$
$\mathbf{K} = g^{xyab}$		$\mathbf{K} = g^{abxy}$

Tablo 4.7. MTI C(1) Anahtar Anlaşma Protokolü

1. Uzun süreli ve geçici anahtarlar seçilerek hesaplamalar yapılır ve taraflarca birbirlerine gönderilir.

2. Son alınan açık anahtarla statik ve geçici gizli anahtarların çarpımı üstellenerek birbirlerine gönderilir.

3. Taraflar statik gizli anahtarlarla üstelleme yaparak **K** ortak anahtarını hesaplar.

A(0) ve B(0) protokollerinde hesaplama gayreti aynı ilerlemektedir: bir üs ve anahtarı elde etmek için bir çoklu üsden yararlanılır. Protokol C(0) karmaşıklığı diğer sınıflara göre daha azdır , anahtarı elde etmek için yalnızca sıradan bir üs alma işlemi gereklidir (Boyd et al., 2003).

MTI A(0) ve MTI C(0) pasif saldırılar karşısında güvenli gibi gözükse de çeşitli aktif saldırılara karşı savunmasız kalabilmektedir. MTI A(0) protokolünün bazen bilinmeyen anahtar paylaşımını yerine getirmediği gibi MTI C(0) protokolünün de küçük alt grup saldırısı karşısında örtük anahtar kimlik doğrulaması sağlamadığı bilinmektedir (Law et al., 2003).

Tarafların g^a ve g^b değerlerini hesaplaması sırasında ayrık logaritma probleminin zorluğundan dolayı düşman tarafından hesaplanamadığı varsayılır. Statik anahtarın kullanımı kimlik doğrulamasını bu şekilde yerine getirirken bu özelliğe bazen örtük anahtar kimlik doğrulaması denir.

MTI A(0) ve MTI C(0) sınıfları incelendiğinde ikisinde de düşman **K** yi hesaplamak yerine alıcı tarafın yanlış bilgilere sahip olmasına neden olabilir. Bu türden ortadaki adam saldırıları gibi tehditler örnek vermek gerekirse bir dijital imza yoluyla önlenebilir (Katz et al., 1996).

Tüm bunlara ek olarak MTI C(1)'in, MTI B(0) ve MTI C(0)'a göre uzun vadeli değerleri daha önceden hesaplayabilmesi gibi bir avantajı vardır.

4.6. İstasyondan İstasyona Anahtar Anlaşma Protokolü (STS Protokolü)

STS (Station to Station), Diffie-Hellman anahtar anlaşmasının modifiye edilmiş bir hali olarak kolay, verimi yüksek, kimliği doğrulanmış haliyle kullanıma sunulmuştur. İlk defa 1987 Uluslararası Anahtarlama Sempozyumu'nda tartışmaya açılan protokol şu anki formuna 1989 senesinde ulaşsa da zaman içinde gelişmiştir. Diffie–Hellman protokolüne kimlik doğrulama kazandırabilmek için değiştirilen mesajlarda dijital imzalamalardan destek alırlar (Diffie et al., 1992; Boyd et al., 2003; Stinson and Paterson, 2018).

İstasyondan İstasyona anahtar anlaşması diye adlandırabileceğimiz protokol güvenliğini kimlik doğrulama ve veri bütünlüğü için dijital imza şemalarının kullanımına dayandırmaktadır. Bundan dolayı tarafların üzerinde anlaşmaya varılan bir dijital imza şemasının kullanımı için birbirlerinin genel doğrulama anahtarlarının asıl kopyalarına sahip olması önemlidir (Dent and Mitchell, 2005).

Üç geçişli hareket eden bir protokol yapısına sahip olan STS, ileri yönlü mükemmel gizliliği, çift taraflı açık anahtar onayı ile kimlik doğrulamalı anahtar anlaşması tanımına uymaktadır.

Protokolün çalışma esasına geçmeden evvel **K** ortak anahtarı hakkında birkaç özelliğini tanımlamak yararlı olacaktır. Burada Ayşe'ye (veya Bora'ya) üç düzey güvence verilmiştir (Stinson and Paterson, 2018):

4.6.1. Örtük Anahtar Kimlik Doğrulaması

Ayşe'nin Bora'dan başka hiç kimse tarafından **K** değerini hesaplayamayacağından emin olması, anahtar anlaşma protokolünün Ayşe 'ye örtük anahtar kimlik doğrulaması sağladığı belirtilir.

4.6.2. Örtük Anahtar Onayı

Ayşe Bora'nın **K** değerini hesaplayabileceğinden emin olursa (Bora'nın protokolü kendi parametrelerine göre ilerlettiği düşünülürse) anahtar anlaşma protokolünün örtük anahtar onayını sağladığını belirtilir.

4.6.3. Açık Anahtar Onayı

Ayşe'nin Bora'dan başka hiç kimse tarafından **K** değerini hesaplayamayacağından emin olması ve bunu gerçekleştirirken protokolü kendi parametrelerine göre ilerletmesi durumunda anahtar anlaşma protokolünün açık anahtar onayını sağladığını belirtilir.

STS güvenli bir karşılıklı anlaşma protokolü olduğundan Ayşe ve Bora gibi taraflardan Ayşe, gerçekten Bora ile iletişim kurduğundan emin olmalıdır. Bora'nın dürüst bir katılımcı olduğunu varsayılırsa, (4.6.3 gereğince) Ayşe Bora'nın **K** değerini hesaplayabileceğinden ve Bora'dan başka hiç kimsenin **K** yı hesaplayamayacağından emin olabilir (Stinson and Paterson, 2018). Bu noktada protokolün çalışmasından bahsedilebilir (Diffie et al., 1992):

Ayşe		**Bora**
$0 \leq x \leq p-1$ ve $x \in_r \mathbb{Z}_p^*$		$0 \leq y \leq p-1$ ve $y \in_r \mathbb{Z}_p^*$
$X = g^x \mod p$	$\xrightarrow{X}$	$Y = g^y \mod p$
		$\mathbf{K} = g^{yx}$
$\mathbf{K} = g^{xy}$	$\xleftarrow{Y, E_{\mathbf{K}}(Sig_B\{g^y, g^x\})}$	$E_{\mathbf{K}}(Sig_B\{g^y, g^x\})$
$E_{\mathbf{K}}(Sig_A\{g^x, g^y\})$	$\xrightarrow{E_{\mathbf{K}}(Sig_A\{g^x, g^y\})}$	

Tablo 4.8. STS Anahtar Anlaşma Protokolü

Diffie-Hellman asimetrik şifrelemesinin anahtar kurulumundan ve sonrasında kimlik doğrulama için imzalarının değişiminden meydana gelmektedir. Anahtar kurulumu için parametrelerde devirli grubun tanımlanması ve ilkel kök g olduğu

taraflarca bilinmektedir. Diffie-Hellman işlemleri altında yatan grubun çarpımsal olduğunu ve üs alma olarak çalıştığını ancak toplamsal gruplara (sonlu bir alan üzerindeki eliptik eğri grubu) da eşit derecede iyi uygulandığı belirtilir.

1. Protokol, taraflardan Ayşe ile başlar, keyfi bir x sayısı belirler ve g^x'i hesaplayarak Bora'ya gönderir.

2. Bora rasgele bir y sayısı belirler ve $\mathbf{K} = g^{yx}$ anahtarını hesaplamak için Ayşe'nin üstel değerini kullanır. Bora, g^y ve uygun bir simetrik şifreleme algoritması kullanılarak **K** nin şifrelenmiş ve üstelleri imzalanmış $E_K(Sig_B\{g^y, g^x\})$ yanıtını iletir. Sig_B, Bora'nın imzasını ve $E_{\mathbf{K}}$, **K** anahtarının şifrelenmesini temsil eder.

3. Ayşe $\mathbf{K} = g^{xy}$ 'yı hesaplar, **K** 'yı kullanarak şifreyi çözer ve Bora'nın ortak anahtarını kullanarak onun imzasını doğrular. Ayşe de Bora'ya üstelleri şifreli imzası $E_K(Sig_A\{g^x, g^y\})$ yı iletir. Sig_A, Ayşe'nin imzasını belirtir.

4. Bora benzer şekilde Ayşe'nin şifreli imzasını; **K** ve Ayşe'nin açık anahtarını yardımıyla doğrular.

Burada varsayılan durum Bora'nın Ayşe'nin ortak anahtarını bildiğidir. İmzalama ile tek başına Bora'nın **K** değerini bilmesi için yeterli olmaz. Bu nedenle, Bora'nın **K** değerini gerçekten bildiğine dair güvence vermek üzere **K**'yı da şifrelemesi gerekir (Boyd et al., 2003). Ayşe, imzasını K ile şifreleyerek Bora'ya x değerini oluşturan taraf olduğunu kanıtlamış olur. Bu, Bora için anahtar değiş tokuşunu gerçekleştirdiği tarafın Ayşe olduğu yönünde güvence sağlar. Ayşe de Bora'dan benzer şekilde güvenceler alır.

Üstel anahtar değişiminin güvenliği, ayrık logaritma probleminin çözümünün zor oluşuna dayanır. Buna ek olarak şifreleme mekanizması bir MAC özet fonksiyonu tarafından da gerçekleştirilebilir (Boyd et al., 2003).

STS protokolünün gerçekleştirmesi istenen iki özelliği daha vardır (Diffie et al., 1992):

İlki, asimetrik anahtar tekniklerinin, geleneksel kriptografi yöntemleri kullanılarak mümkün olduğundan daha kolay ve daha emniyetli olmasıdır. Taraflar gizli anahtarlarını oluşturursalar bile, bu anahtarlarını protokolü başlatma esnasında dahi (güvenilir olduğu varsayılan herhangi bir taraf da dahil olmak üzere) hiç kimseye ifşa etmemesi gerekmektedir.

İkincisi, iletişim halindeki tarafların oturum çalıştırması başına merkezi kabul edilen bir otoriteyle iletişime geçmesine gerek olmamasıdır. Ortak anahtarları dağıtmak üzere sertifikalar kullanılıyor ise bir taraf kendi sertifikası ve güvenilir otoritenin açık anahtarına sahip olduğunda, merkezi bir otoriteye başvurmadan da diğer tarafla anahtar değiş tokuşunda bulunup kimliklerini doğrulayabilir. Protokol gereksiz öğeler kullanmadan basit ve güvenli olmasıyla zarif ve zor bir dengeyi birlikte sağlıyor gibi görünmektedir.

Paylaşılan anahtar, kısa ömürlü Diffie-Hellman anahtarı olduğundan, ileri yönlü gizlilik STS protokolü tarafından sağlanır. Ayrıca, uzun vadeli (statik) bir anahtar kaybolursa, bu, düşmanın farklı bir varlığın imzasını taklit etmesini sağlayamayacağından imzalar, anahtarın gizliliğini ihlal eden kimliğe bürünmeye karşı direnç sağlar (Boyd et al., 2003).

Bilinmeyen anahtar paylaşımı saldırılarını önlemek için simetrik şifrelemenin gerekli olduğu açıklanmıştır. Düşman, son mesajdaki Ayşe'nin imzasını kaldırabilir ve halka açık bir sertifika alıp onu kendininkiyle değiştirebilir. Bu yüzden bilinmeyen anahtar paylaşımı saldırısı olasılığı vardır. Bilinmeyen anahtar paylaşımı saldırıları, otoritenin kimliğinin değiştirilen imzalara dahil edilmesiyle önlenebilir (Boyd et al., 2003).

5. SONUÇ VE ÖNERİLER

Haberleşmenin güvenliğinin gün geçtikçe arttığı dünyamızda insanların ve kurumların gizlilikleri de o denli önem kazanmıştır. Bir çok bilginin yanlış kişilerce ve yanlış yerlerde ifşası devlet krizlerine bile neden olmuştur. Bu sebepten bilginin şifreli bir şekilde muhafaza edilmesi için insanlık tarihi boyunca süregelen tarihsel şifrelemeler, simetrik ve asimetrik şifrelemeler oluşturulmuştur. Anahtar anlaşmaları da asimetrik şifrelemenin yapı taşını oluşturmakta ve günümüz şifreleme tekniklerine öncülük etmektedir.

Tezde şifreleme terimleri ve tekniklerinden bahsedildikten sonra anahtar anlaşma protokollerinden literatürde en yaygın olarak kullanılanlarını vermeye gayret edilmiştir. Nitekim Diffie-Hellman protokolünün altyapısının kullanılarak üretilen çok fazla algoritma olduğundan dolayı hepsini incelemek zaman ve kaynak açısından sınırlılıklar oluşturmuştur.

Uzun gizlilik süreleri olan kriptografik uygulamalar için daha ekonomik bir eyleme ihtiyaç vardır. Büyük hacimli şifrelenmiş verilerin gelecekte bir kuantum bilgisayar kullanarak bu şifreyi çözmenin mümkün olma olasılığı vardır. Bugün kullanılan temel anlaşma protokollerinin de aynı nedenle uygun uzun bir süre boyunca güvenli olması gerekir.

Buna karşılık, kimlik doğrulama amaçlarına hizmet eden imzalar oldukça kısa bir çalışma ömrüne sahiptirler ve prensipte yalnızca doğrulandıklarında güvenli olmaları gerekir. Gelecekte kuantum bilgisayarların bir imza sistemini kırması mümkün olacaksa, bugünün imza sertifikaları muhtemelen çoktan güncelliğini yitiriyordur.

Er ya da geç, kuantum bilgisayar dirençli sistemlerin kullanımı çoğu kriptografik uygulama için standart haline gelecektir. Ancak, bahsedilen zorluklar (uygulanabilirlik, uyumluluk gibi) nedeniyle bu tür bir kullanım yakın gelecekte gerçekçi

olmayabilir. Bu sebeple şimdilik mümkün olduğunca hibrit (simetrik+asimetrik) çözümlerin kullanılması önerilmektedir.

Uygulamaların yeni ve daha çok geliştirilebilecek olması, akla gelebilecek tüm gelişmelere yanıt vermek, gelecekteki önerileri ve standartları uygulamak ve gelecekte daha zayıf olabilecek algoritmaları değiştirmek için mümkün olduğunca esnek tasarlanmalarının olması sağlanmalıdır.

Ülkemizde bu alanda yapılan çalışmaların bilgisayar bilimlerinde karşımıza çıkması, matematiksel altyapılarından yoksunluğu dolayısıyla ve Türkçe kaynakların çok az oluşu nispetinde literatüre kazandırdığımız tezin daha sonra yapılacak çalışmalar için yol gösterici olması sevindirici olacaktır.

(Bu tez LaTeXaltyapısı kullanılarak oluşturulmuştur.)

KAYNAKLAR

Al Sultan, K., Saeb, M., and Badawi, U. E.-R. (2003). A new two-pass key agreement protocol. In *2003 46th Midwest Symposium on Circuits and Systems*, volume 1, s. 509–511. IEEE.

Altındiş, H. (1999). *Sayılar Teorisi ve Uygulamaları*. Kayseri: Erciyes Üniversitesi Yayın Komisyonu.

Altuncu, E. (2019). Blokzincir kullanılarak kimlik doğrulama seremonisini ortadan kaldıran bir güvenli mesajlaşma uygulamasının geliştirilmesi. *Yüksek Lisans Tezi, TOBB ETÜ Fen Bilimleri Enstitüsü, Ankara*.

Aumasson, J.-P. (2017). *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press.

Balcı, M. (1985). *Matematik Analiz 1*. Ankara Üniversitesi Fen Fakültesi Yayınları.

Bellare, M., Canetti, R., and Krawczyk, H. (1996). Message authentication using hash functions: The hmac construction. RSA Laboratories' CryptoBytes, 2(1):12–15.

Blake, I., Seroussi, G., Smart, N., and Society, L. M. (2005). *Advances in Elliptic Curve Cryptography.* Advences in Cryptography. Cambridge University Press.

Blake-Wilson, S., Johnson, D., and Menezes, A. (1997). Key agreement protocols and their security analysis. In *IMA international conference on cryptography and coding*, s. 30–45. Springer.

Blake-Wilson, S. and Menezes, A. (1998). Authenticated diffe-hellman key agreement protocols. In *International Workshop on Selected Areas in Cryptography*, s. 339–361. Springer.

Blake-Wilson, S. and Menezes, A. (1999). Unknown key-share attacks on the

station-to-station (sts) protocol. In *International Workshop on Public Key Cryptography*, s. 154–170. Springer.

Boyd, C., Mathuria, A., and Stebila, D. (2003). *Protocols for authentication and key establishment*, volume 1. Springer.

Bozkurt, Ö. Ö. (2005). Eliptik eğri şifreleme kullanarak güvenli soket katmanı protokolünün gerçeklenmesi ve performansının değerlendirilmesi. *Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul*.

Brown, D. R. (2014). Some theoretical conditions for menezes–qu–vanstone key agreement to provide implicit key authentication. *Work*.

Çakmak, A. (2018). Web güvenliğinde ssl/tls kriptografik protokolü: Açıklıklar, saldırılar ve güvenlik önlemleri. *Yüksek Lisans Tezi*.

Çallıalp, F. (2009). *Sayıların Teorisi*. Birsen Yayınevi.

Çimen, C. (2007). *Şifrelerin matematiği: kriptografi*. ODTÜ bilim ve toplum kitapları dizisi. ODTÜ.

Çubukçu, F. (2018). *Bilgi Güvenliği Yönetim Sistemi ISO27001: 2013 Uygulama Kılavuzu*. Pusula Yayıncılık.

Dent, A. and Mitchell, C. (2005). *User's Guide to Cryptography and Standards*. Artech House computer security series. Artech House.

Dierks, T. and Rescorla, E. (2006). Rfc 4346: The transport layer security (tls) protocol, version 1.1 (2006).

Diffie, W. and Hellman, M. (1976). *New directions in cryptography. IEEE transactions on Information Theory*, 22(6):644–654.

Diffie, W., Van Oorschot, P. C., and Wiener, M. J. (1992). Authentication and authenticated key exchanges. *Designs, Codes and cryptography*, 2(2):107–125.

Doğanaksoy, A., Sulak, F., Varıcı, K., Kocair, Ç., ve Atalay, F. (2008). Özet fonksiyon algoritması geliştirme proje önerisi.

Doğantimur, F. (2009). Iso 27001 standardı çerçevesinde kurumsal bilgi güvenliği. *Maliye Bakanlığı Strateji Geliştirme Başkanlığı, Mesleki Yeterlilik Tezi, Ankara*.

Durak, F. (2007). Gruplar ve cebirler üzerinde çaprazlanmış modüller. *Yüksek Lisans Tezi*.

Dutta, R. and Barua, R. (2005). Overview of key agreement protocols. IACR Cryptol. *ePrint Arch.*, 2005:289.

Eastlake, D. and Jones, P. (2001). Us secure hash algorithm 1 (sha1).

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472.

Erol, H. (2006). Kurumsal ağlarda açık anahtar altyapısı tabanlı elektronik imza uygulaması. *Yayınlanmamış Yüksek Lisans Tezi, Ankara: Gazi Üniversitesi*.

FIPS, P. (2013). 186-4. *Digital Signature Standard (DSS)*.

Fraleigh, J. B. (2003). *A first course in abstract algebra*. Pearson Education India.

Hammack, R. (2016). *Book of Proof*. Richard Hammack.

Hunt, R. (2001). Technological infrastructure for pki and digital certification. *Computer communications*, 24(14):1460–1471.

İmamoğlu, K. (2011). Görüntü sıkıştırma ve internet üzerinden güvenli aktarım sistemi. *Yüksek Lisans Tezi, Kocaeli Universitesi, Fen Bilimleri Enstitusu*.

Jirwan, N., Singh, A., and Vijay, D. S. (2013). Review and analysis of cryptography techniques. *International Journal of Scientific & Engineering Research*, 4(3):1–6.

Kaçmaz, E., Tuğrul, B., Germen, E., ve San, I. (2016). *Ağ Yönetimi ve Bilgi Güvenliği*. T.C. Anadolu Üniversitesi Yayını.

Kahate, A. (2013). *Cryptography and network security*. Tata McGraw-Hill Education.

Kaliski Jr, B. S. (2001). An unknown key-share attack on the mqv key agreement protocol. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):275–288.

Kaplan, Y. (2000). *Veri Haberleşmesi Temelleri*. Papatya Yayıncılık.

Karahan, A. (2014). Kripto sistemlerde türk alfabesi kullanımı: Şifreleme ve kriptanaliz.

Karataş, İ. (2019). Eliptik eğri şifrelemenin kısıtlamalı ağlardaki iletim katmanındaki dtls protokolünde doğrulama ve anahtar karşılaştırma aşamasında kullanılması. *Yüksek Lisans Tezi, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü*.

Katz, J., Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook Of Applied Cryptography*. CRC press.

Keleş, C. (2012). Kaotik haritalar kullanarak görüntü şifreleme. *Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Trabzon*.

Kırımlı, M. (2007). Açık anahtar kriptografisi ile sayısal İmza tasarımı ve uygulaması. *Yüksek Lisans Tezi, Ankara: Gazi Üniversitesi/Fen Bilimleri Enstitüsü*.

Koçoğlu, İ. (2004). Telekomünikasyon yönetim ağının güvenliği. *Yüksek Lisans Tez,. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü*.

Kodaz, H. ve Botsalı, F. M. (2010). Simetrik ve asimetrik şifreleme algoritmalarının karşılaştırılması.

Krawczyk, H. (2005). Hmqv: A high-performance secure diffie-hellman protocol. In *Annual International Cryptology Conference*, s. 546–566. Springer.

Külen, F. (2013). Kriptolojide bazı Şifreleme yöntemlerinde cebirsel yaklaşımlar. *Yüksek Lisans Tezi, Gaziosmanpaşa Üniversitesi. Tokat*.

Law, L., Menezes, A., Qu, M., Solinas, J., and Vanstone, S. (2003). An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, 28(2):119–134.

List, P. T. C. L. (2007). 1.1 the lexicon of cryptography.

Mavroeidis, V., Vishi, K., Zych, M. D., and Jøsang, A. (2018). The impact of quantum computing on present cryptography. *arXiv preprint arXiv:1804.00200*.

Okuyucu, H. H. (2020). Hash fonksiyonlarının adli bilişimde uygulamaları ve c++ İle Şifreleme algoritması tasarımı. *Yüksek Lisans Tezi, Karabük Üniversitesi, Lisansüstü Eğitim Enstitüsü*.

Rivest, R. and Dusse, S. (1992). The md5 message-digest algorithm.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

Sağıroğlu, Ş. ve Alkan, M. (2005). *Her Yönüyle Elektronik İmza e-İMZA*. Grafiker Yayınları.

Sağlam, Ö. (2002). Halka protokolü üzerinde grup anahtarı belirlenmesi üzerine bir araştırma. *Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü,İzmir*.

Sağır, M. (2014). Açık anahtar altyapısı ve elektronik İmzanın mobil tabanlı uygulaması. *Yüksek Lisans Tezi, Kocaeli Üniversitesi*.

Schneier, B. (1996). *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons.

Şenay, H. (2007). Sayılar teorisi dersleri. *Dizgi Ofset Matbaacılık, Konya*.

Singh, S. (2000). *The code book: the science of secrecy from ancient Egypt to quantum cryptography*. Anchor.

Stallings, W. (2006). *Cryptography And Network Security, 4/E*. Pearson Education.

Stinson, D. R. (1995). Classical cryptography. *Cryptography, Theory and Practice, 2nd Edition, Chapman & Hall/CRC, Kenneth H. Rosen (Ed.)*, s. 1–20.

Stinson, D. R. and Paterson, M. (2018). Cryptography: Theory and Practice. CRC press.

Taş, O. ve Akın, E. (2002). Elektronik posta güvenliği İçin yeni bir anahtar dağıtım protokolü. In *Proceedings of the Fourth GAP Engineering Congress*, volume 6, s. 08.

Tuncal, T. (2008). Bilgisayar güvenliği üzerine bir araştırma ve şifreleme-deşifreleme üzerine uygulama. *Yüksek Lisans Tezi, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü*.

Turp, S. (2019). Ayrık logaritma problemi. *Yüksek Lisans Tezi. Bursa Uludağ Üniversitesi*.

Ural, N. ve Örenç, O. (2019). *Şifreleme ve Şifre Çözme Yöntemleri*. Pusula Yayıncılık.

Van Tilborg, H. C. and Jajodia, S. (2014). *Encyclopedia of cryptography and security*. Springer Science & Business Media.

Vesterås, B. (2006). Analysis of key agreement protocols. Master's thesis, Department of Computer Science and Media Technology Gjøvik University College.

Wang, S., Cao, Z., and Choo, K.-K. R. (2006). Provably secure identity-based authenticated key agreement protocols without random oracles. *CRYPTOGRAPHY ePRINT ARCHIVE*, s. 1–16.

Wayner, P. (1997). British document outlines early encryption discovery. *New York*

Times, 24.

Yang, C.-C., Yang, Y.-W., ve Chang, T.-Y. (2005). Cryptanalysis of an authentication key exchange protocol. *Journal of Applied Sciences*, 5(2):281–283.

Yeşilbaş, E. (2016). Cebirsel kriptoloji yöntemleri ve bazı uygulamaları. *Yüksek Lisans Tezi, Recep Tayyip Erdoğan Üniversitesi/Fen Bilimleri Enstitüsü*.

Yıldırım, K. (2006). Veri şifrelemesinde simetrik ve asimetrik anahtarlama algoritmalarının uygulanması (hybrid şifreleme). *Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü*.

Yıldırım, M., Yavaş, A. A., Özdemir, O., Mete, A., Demirbağ, E., ve diğerleri (2005). Nitelikli sertifikasyon altyapisi ve yetkilendirme.

Özkaynak, F. (2016). Kriptolojik rasgele sayı Üreteçleri. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 8:37 – 45.

Zuccherato, R. (2003). Root certificate management system and method. US Patent App. 09/906,504.

ÖZGEÇMİŞ

Ömer Kocabıyık. 1993 yılında Aydın'da doğdu. İlköğretimini 2008 yılında burada tamamladıktan sonra 2012 yılında Afyonkarahisar Dinar Anadolu Öğretmen Lisesi'nden mezun oldu. Lisans öğrenimini 2016 yılında Afyon Kocatepe Üniversitesi'nde İlköğretim Matematik Öğretmenliği bölümünde tamamladı. 2017 yılında Ağrı'da Milli Eğitim Bakanlığına bağlı bir ortaokulda matematik öğretmenliğine başlamıştır. 2019-2020 eğitim öğretim yılında başladığı Ağrı İbrahim Çeçen Üniversitesi Fen Bilimleri Enstitüsü'nde Matematik anabilim dalında yüksek lisansını tamamlamıştır. Halen Milli Eğitim Müdürlüğü İSG Biriminde öğretmen olarak görev yapmakta ve seramik sanatı ile ilgilenmektedir. Evli ve bir çocuk babasıdır.