
GAZİ ÜNİVERSİTESİ
Gazi
1926

ELİPTİK EĞRİLER İLE KİMLİK TABANLI SERTİFİKASIZ KRİPTOGRAFİ

Yasin GENÇ

DOKTORA TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAYIS 2023

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Yasin GENÇ

08/05/2023

ELİPTİK EĞRİLER İLE KİMLİK TABANLI SERTİFİKASIZ KRİPTOGRAFİ

(Doktora Tezi)

Yasin GENÇ

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2023

ÖZET

Kriptografi, bulut bilişimden akıllı ulaşım sistemlerine kadar birçok alanda yaygın olarak kullanılmaktadır. Bu tez çalışmasında, araçların interneti için koşullu gizliliği koruyan özgün bir kimlik doğrulama şeması önerilmiştir. Bu şema, eliptik eğrileri kullanan eşlemesiz, kimlik tabanlı ve sertifikasız bir kriptografi yöntemine dayanmaktadır. Ayrıca araç ağı sisteminde kullanıcıların gizliliğini sağlamak için anonim kimlikler kullanılmaktadır. Önerilen yöntemde matematiksel olarak maliyetli olan çift doğrusal eşleme kullanılmamıştır. Bu sayede eliptik eğri kriptografisi etkin olarak kullanılarak verimli ve hafifsiklet bir kimlik doğrulama şeması tasarlanmıştır. Eşleme tabanlı kriptografideki temel şifreleme ve imza şemaları hakkında bilgi sunulmuştur. Ayrıca önerilen şema kimlik tabanlı kriptografinin dezavantajı olan anahtar emanet problemini ortadan kaldıran sertifikasız kriptografi temelli bir özellik taşımaktadır. Ağ sistemine dahil olan kullanıcıların her mesaj imzalama sırasında merkezi birimlerle iletişim kurarak imzalama için gerekli bilgileri talep etmesi araç ağlarında hesaplama yüküne ve bağlantı sorunlarına sebep olabilmektedir. Önerilen şemada, kullanıcıların her mesaj imzalama sırasında merkezi birimlerle iletişim kurmasına gerek yoktur. Böylece merkezi birimlerdeki hesaplama yükü ve bağlantı sorunları en aza indirilmektedir. Bununla birlikte önerilen şema, mesaj doğrulama aşamasındaki gecikmeleri en aza indiren verimli bir toplu mesaj doğrulaması yöntemini de desteklemektedir. Önerilen kimlik doğrulama şemasının diğer mevcut şemalara göre performans değerlendirmesi için hesaplama maliyeti, iletişim maliyeti ve enerji tüketim maliyeti karşılaştırılmıştır. Önerilen kimlik doğrulama şemasının güvenlik değerlendirilmesi rasgele kâhin modeline göre yapılmıştır. Güvenli bir araç ağının tasarımı için gerekli olan temel güvenlik gereksinimleri kontrol edilmiş ve önerilen şemanın siber saldırılara karşı dayanıklı olduğu gösterilmiştir. Bu çalışmada ayrıca eliptik eğri tabanlı özgün bir şifreleme algoritması ve verimli bir eliptik eğri tabanlı imza şeması da önerilmiştir. Tasarım ve hesaplamalar C++ programlama dili ile gerçekleştirilmiştir.

Bilim Kodu	:	90501
Anahtar Kelimeler	:	Kriptografi, eliptik eğriler, kimlik tabanlı kriptografi, sonlu cisimler, sertifikasız kriptografi, araçların interneti
Sayfa Adedi	:	118
Danışman	:	Prof. Dr. Erkan AFACAN

IDENTITY BASED CERTIFICATELESS CRYPTOGRAPHY WITH ELLIPTIC CURVES

(Ph. D. Thesis)

Yasin GENÇ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

May 2023

ABSTRACT

Cryptography is widely used in many areas from cloud computing to intelligent transport systems. In this thesis, a novel conditional privacy-preserving authentication scheme is proposed for the Internet of Vehicles. This scheme is based on a pairing-free, identity-based and certificateless cryptography method using elliptic curves. In addition, anonymous identities are used to ensure the privacy of users in the vehicular network system. In the proposed method, mathematically costly bilinear pairing has not been used. In this way, an efficient and lightweight authentication scheme has been designed by using elliptic curve cryptography effectively. Information about the basic encryption and signature schemes in pairing-based cryptography is presented. It also includes a certificateless cryptography-based feature that eliminates the key escrow problem, which is the disadvantage of identity-based cryptography. The fact that the users involved in the network system communicate with the central units during each message signing and request the necessary information for signing may cause computational overhead and connectivity problems in vehicular networks. In the proposed scheme, users do not need to communicate with the central units during each message signing. This reduces the computational overhead and connectivity problems in central units to a minimum. In addition, the proposed scheme supports an efficient batch message authentication method that reduces the delays in the message authentication phase to a minimum. For the performance evaluation of the proposed authentication scheme against other existing schemes, the computation cost, communication cost and energy consumption cost have been compared. The security evaluation of the proposed authentication scheme is based on the random oracle model. The basic security requirements for the design of a secure vehicular network have been checked and the proposed scheme has been shown to be robust against cyber-attacks. A novel elliptic curve-based encryption algorithm and an efficient elliptic curve-based signature scheme is also proposed. The design and computations are implemented in C++ programming language.

Science Code : 90501

Key Words : Cryptography, elliptic curves, identity-based cryptography, finite fields, certificateless cryptography, internet of vehicles

Page Number : 118

Supervisor : Prof. Dr. Erkan AFACAN

TEŞEKKÜR

Bu zorlu ve uzun doktora çalışmam süresince beni her aşamada destekleyen, tecrübe ve bilgisini esirgemeyen, farklı fikirlere gösterdiği saygı sayesinde özgürce düşünmemi sağlayan değerli danışmanım Sayın Prof. Dr. Erkan AFACAN hocama en içten şükran ve saygılarımı sunarım.

Doktora çalışmalarım süresince desteklerini her zaman hissettiğim, tecrübe ve bilgilerini esirgemeyen tez izleme komitesi üyeleri Sayın Prof. Dr. Erdem YAZGAN ve Sayın Prof. Dr. Murat Hüsnü SAZLI hocalarıma teşekkürlerimi sunarım.

Bana vermiş oldukları destek için aileme ve çalışmalarım boyunca bana destek olan, sabırla bekleyen ve bu süreçte büyük fedakârlık gösteren sevgili eşim Merve'ye şükranlarımı sunarım.

Ayrıca doktora öğrenim boyunca TÜBİTAK BİDEB 2211/E doğrudan doktora burs programı kapsamında beni destekleyen TÜBİTAK'a teşekkür ederim.

Bu çalışmamı Kahramanmaraş depreminde zarar gören doğup ve büyüdüğüm güzel memleketlerim Afşin ve Elbistan ile depremde etkilenen ve depremde hayatını kaybeden herkese ithaf ediyorum.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xii

1. GİRİŞ 1

2. TEMEL KAVRAMLAR 9

2.1. Kriptoloji 9

2.2. Kriptoanaliz 9

2.3. Kriptografi 10

2.4. Kriptografik Sistemler 12

2.4.1. Gizli anahtarlı (simetrik) kriptografi 14

2.4.2. Açık anahtarlı (asimetrik) kriptografi 14

2.5. Diffie-Hellman Anahtar Değişim Protokolü 15

2.6. Açık Anahtar Altyapısı 16

2.7. Özet Fonksiyonu 17

2.8. Dijital İmza 19

2.9. Mesaj Doğrulama Kodu 20

3. ELİPTİK EĞRİ KRİPTOGRAFİSİ 23

3.1. Temel Cebirsel Kavramlar 23

3.2. Eliptik Eğriler 25

3.3. Sonlu Cisimler Üzerinde Tanımlı Eliptik Eğriler 28

Sayfa

3.4. Sonlu Asal Cisimler Üzerinde Matematiksel İşlemler 32
3.5. Eliptik Eğri Kriptografisi 37
3.6. Sonlu Cisimler Üzerinde Tavsiye Edilen Eliptik Eğriler 38
3.7. Eliptik Eğri Diffie-Hellman Anahtar Değişim Protokolü 39
3.8. Eliptik Eğri Tabanlı Kriptografi Algoritması 40
3.9. Eliptik Eğri Dijital İmza Algoritması 45
3.10. Eliptik Eğri Dijital İmza Algoritmasının Güvenliği 47
3.11. Eliptik Eğri Dijital İmza Algoritmasının Performansı ve Algoritma Önerisi .. 48
3.12. Önerilen ECDSA İmza Algoritmasının Güvenliği 51
3.13. Önerilen İmza Algoritmasının Uygulaması ve Karşılaştırılması 51
4. EŞLEME TABANLI KRİPTOGRAFİ 59
4.1. Çift Doğrusal Eşleme 60
4.2. Bölen Teorisi 61
4.3. Weil Eşlemesi 63
4.4. Tate Eşlemesi 64
4.5. Joux Üç Taraflı Anahtar Değişim Protokolü 65
4.6. Kimlik Tabanlı Kriptografi 66
4.7. Boneh-Franklin Kimlik Tabanlı Şifreleme 68
4.8. Boneh-Boyen Kimlik Tabanlı Şifreleme 72
4.9. BLS İmza Şeması 76
4.10. Sertifikasız Kriptografi 77
5. KİMLİK TABANLI SERTİFİKASIZ KRİPTOGRAFİ SİSTEMİNİN TASARIMI 81
5.1. Önerilen Modelin Ağ Mimarisi 82
5.2. Önerilen Kimlik Doğrulama Şemasının Algoritması 83

Sayfa

5.3. Güvenlik Değerlendirmesi 90

5.4. Performans Değerlendirmesi 97

6. SONUÇ VE ÖNERİLER 105

KAYNAKLAR 109

ÖZGEÇMİŞ 117

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. Kriptografi sisteminde kullanılan parametreler ve tanımları 13

Çizelge 3.1. $E / \mathbb{F}_{41} : y^2 = x^3 + 3x + 7$ eliptik eğrisinin noktaları 29

Çizelge 3.2. NIST eliptik eğrileri..... 39

Çizelge 3.3. Eliptik eğri tabanlı ElGamal kriptografi sisteminin parametreleri 41

Çizelge 3.4. Unicode tablosunda bulunan bazı karakterlerinin onaltılık sayı tabanındaki değerleri..... 42

Çizelge 3.5. Önerilen şifreleme algoritmasında kullanılan parametreler..... 43

Çizelge 3.6. NIST P-521 eliptik eğrisinin parametreleri 52

Çizelge 3.7. Sonlu asal cisimler üzerinde gerçekleştirilen işlemlerin süreleri (μs) 55

Çizelge 3.8. Önerilen imza algoritmasının mevcut imza algoritmaları ile karşılaştırılması..... 56

Çizelge 3.9. Önerilen imza algoritmasının klasik ECDSA ile karşılaştırılması 57

Çizelge 4.1. Boneh-Franklin kimlik tabanlı şifrelemede kullanılan parametreler..... 69

Çizelge 4.2. Önerilen imza algoritmasının klasik ECDSA ile karşılaştırılması 73

Çizelge 5.1. Önerilen şemanın algoritmasında kullanılan parametre ve tanımları 83

Çizelge 5.2. Matematiksel işlemlerin ortalama gerçekleştirim süresi..... 98

Çizelge 5.3. Hesaplama maliyetlerinin karşılaştırılması..... 99

Çizelge 5.4. Önerilen şemanın mevcut şemalara göre performans artışları..... 101

Çizelge 5.5. Önerilen şemanın mevcut şemalara kıyasla iletişim maliyetinin ve performans artışının karşılaştırılması..... 102

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Temel bir kriptografi sistemi 10

Şekil 2.2. Açık anahtar altyapısı 17

Şekil 2.3. SHA-1 ve SHA-512 özet fonksiyonları 18

Şekil 2.4. Temel dijital imza algoritması 20

Şekil 3.1. $y^2 = x^3 - x$ eliptik eğrisi 27

Şekil 3.2. $E / \mathbb{F}_{41} : y^2 = x^3 + 3x + 7$ eliptik eğrisinin noktalarının gösterimi 30

Şekil 3.3. $E / \mathbb{F}_{787}$ eliptik eğrisi üzerindeki noktaların gösterimi 30

Şekil 3.4. $E / \mathbb{F}_{20011}$ eliptik eğrisi üzerindeki noktaların gösterimi 31

Şekil 3.5. Eliptik eğri üzerinde toplama işlemi ($P + Q = R$) 32

Şekil 3.6. Eliptik eğri üzerinde nokta çiftleme ($P + P = 2 \cdot P = R$) 33

Şekil 3.7. $P = -P$ olması durumunda nokta çiftleme ($P + P = 2 \cdot P = \mathcal{O}$) 34

Şekil 3.8. Eliptik eğri üzerindeki noktalar arasında bağlantı 36

Şekil 3.9. Eliptik eğri Diffie-Hellman anahtar değişim protokolü 40

Şekil 4.1. Joux üç taraflı anahtar değişim protokolü 65

Şekil 4.2. Kimlik tabanlı kriptografinin genel şeması 67

Şekil 4.3. Boneh-Franklin kimlik tabanlı şifreleme şeması 69

Şekil 4.4. Boneh-Boyen kimlik tabanlı şifreleme şeması 76

Şekil 4.5. Sertifikasız açık anahtarlı kriptografinin genel şeması 78

Şekil 5.1. IoV ağ mimarisi 81

Şekil 5.2. Önerilen şemanın ve mevcut şemaların hesaplama maliyetinin karşılaştırılması 99

Şekil 5.3. Önerilen şema ve diğer şemaların enerji tüketim maliyetlerinin karşılaştırılması 103

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
µs	Mikro saniye
ms	Mili saniye
e	Eşleme

Kısaltmalar	**Açıklamalar**
AES	Advanced Encryption Standard (Gelişmiş Veri Standardı)
ABE	Attribute Based Encryption (Öznitelik Tabanlı Şifreleme)
AUS	Akıllı Ulaşım Sistemleri
BDHP	Bilinear Diffie-Hellman Problem (Çift Doğrusal Diffie-Hellman Problemi)
CLC	Certificateless Cryptography (Sertifikasız Kriptografi)
CL-AS	Certificateless Aggregate Signature (Sertifikasız Toplu İmza)
CL-PKC	Certificateless Public Key Cryptography (Sertifikasız Açık Anahtar Kriptografisi)
CPPA	Conditional Privacy-Preserving Authentication (Koşullu Gizliliği Koruyan Kimlik Doğrulama)
CU	Central Unit (Merkezi Birim)
DES	Data Encryption Standart (Veri Şifreleme Standardı)
DBDHP	Decisional Bilinear Diffie-Hellman Problem (Çift Doğrusal Diffie-Hellman Karar Verme Problemi)
DHP	Diffie-Hellman Problem (Diffie-Hellman Problemi)
DSRC	Dedicated Short-Range Communications (Özel Kısa Menzilli İletişim)
ECC	Elliptic Curve Cryptography (Eliptik Eğri Kriptografisi)

Kısaltmalar	Açıklamalar
ECDLP	Elliptic Curve Discrete Logarithm Problem (Eliptik Eğri Ayrık Logaritma Problemi)
ECDSA	Elliptic Curve Digital Signature Algorithm (Eliptik Eğri Dijital İmza Algoritması)
IoT	Internet of Things (Nesnelerin İnterneti)
IoV	Internet of Vehicles **(**Araçların İnterneti)
IBC	Identity Based Cryptography **(**Kimlik Tabanlı Kriptografi)
KGU	Key Generation Unit (Anahtar Üretim Merkezi)
MAC	Message Authentication Code (Mesaj Doğrulama Kodu)
MDM	Mesaj Doğrulama Maliyeti
MİM	Mesaj İmzalama Maliyeti
NIST	National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
OBU	On-Board Unit (Araç Üstü Birim)
PBC	Pairing Based Cryptography (Eşleme Tabanlı Kriptografi)
PKC	Public Key Cryptography (Açık Anahtar Kriptografisi)
PKI	Public Key Infrastructure (Açık Anahtar Altyapısı)
PPK	Partial Private Key (Kısmi Gizli Anahtar)
RSU	Road Side Unit **(**Yol Kenarı Birimi)
TMDM	Toplu Mesaj Doğrulama Maliyeti
VANET	Vehicular Ad-Hoc Network (Geçici Araç Ağı)
V2G	Vehicle-to-Grid (Araçtan Şebekeye İletişim)
V2I	Vehicle-to-Infrastructure (Araçtan Altyapıya İletişim)
V2N	Vehicle-to-Network (Araçtan Ağa İletişim)
V2P	Vehicle-to-Pedestrian (Araçtan Yayaya İletişim)
V2S	Vehicle-to-Sensor (Araçtan Sensöre İletişim)
V2V	Vehicle-to-Vehicle (Araçtan Araca İletişim)

1. GİRİŞ

Akıllı ulaşım sistemleri (AUS) ile birlikte araçların birbirleriyle veya çevredeki olası her nesne ile haberleşmesi önemli bir konu haline gelmiştir ve bu haberleşme araçların interneti (Internet of Vehicles; IoV) ile sağlanmaktadır [1]. Araç ağları şehirlerdeki trafik sıkışıklığı ve trafik kazaları gibi birçok nedenden dolayı ön plana çıkmaktadır. Bu ağlar hız uyarıları, viraj alma, yol durumu bilgisi, hava durumu, yaya geçidi ve kavşak uyarıları gibi trafikteki birçok hayati konuda kullanıcıları bilgilendirmektedir [2-5]. Geçici araç ağları (Vehicular Ad-hoc Network; VANET) araçtan araca (Vehicle-to-Vehicle; V2V) ve araçtan altyapıya (Vehicle-to-Infrastructure; V2I) iletişimi gerçekleştirmek için kullanılmaktadır [5-7]. Ayrıca, araçlar yol kenarı birimleriyle (Road Side Unit; RSU) de iletişim kurabilir, bu da araçtan RSU'ya (V2R) iletişim olarak adlandırılır. Araç ağları, IEEE 802.11p standardına göre özel kısa menzilli iletişim (Dedicated Shaort-Range Communications; DSRC) kullanarak V2V ve V2I iletişimini sağlar [9]. VANET'lerden farklı olarak IoV ile etrafındaki yaya (Vehicle-to-Pedestrian; V2P), ağ (Vehicle-to-Network; V2N), sensör (Vehicle-to-Sensor; V2S), bulut (Vehicle-to-Cloud; V2C) ve şebeke (Vehicle-to-Grid; V2G) ile haberleşen bir model ağ kurulabilmektedir [8-11]. Bu ağ, nesnelerin internetinin (IoT) araç ağlarında kullanımını da modellemektedir [12]. Böylece gelecekte ağa farklı kullanıcılar ve farklı nesneler eklenerek çok daha kullanışlı bir araç ağı oluşturulabilir. Ancak, IoV tabanlı akıllı şehir ortamında hassas mesajlar güvensiz bir kanal üzerinden iletildiği için araçlar arasındaki iletişim çeşitli güvenlik tehditlerine açıktır. Bu nedenle, akıllı şehir ortamında araç ağlarında çeşitli hizmetleri sağlamak için güvenli bir mesaj kimlik doğrulama protokolü kullanılması önemli bir araştırma konusudur [2-8, 13-16].

Araç ağlarında güvenlik ve gizlilik sorunlarını çözmek için koşullu gizliliği koruyan kimlik doğrulama (Conditional Privacy-Preserving Authentication; CPPA) şemaları yaygın olarak kullanılmaktadır. CPPA, araç ağlarında karşılıklı kimlik doğrulamayı sağlayan ve araç kimliklerinin anonim (koşullu gizlilik) olarak kullanıldığı bir kimlik doğrulama şemasıdır. Bu şemalar hem karşılıklı kimlik doğrulamayı hem de gizlilik korumasını aynı anda destekler [5]. Araç sistemlerini ve uygulamalarını verimli bir şekilde desteklemek için protokollerin dikkatli bir şekilde tasarlanması ve etkili bir şekilde uygulanması gerekmektedir [17]. Araçlarda, diğer araç veya nesneler tarafından üretilen verileri toplama ve işleme yeteneğine sahip güçlü işlemci olarak kabul edilen araç üstü birimler (On-Board

Unit; OBU) bulunur. Yol kenarı birimi (RSU), OBU ile veri alışverişinden ve verilerin daha sonraki işlemler için trafik kontrol merkezi tarafından yetkilendirilen bulut sunucusuna iletilmesinden sorumlu olan altyapıyı ifade eder [18].

Akıllı ulaşım sistemi büyük miktarda dinamik gerçek zamanlı kritik veri içerir, bu nedenle bu sistemlerin güvenliği oldukça önemlidir [19]. Günümüz araçlarının, araç iletişimini hedef alan siber saldırılara maruz kalması çok muhtemel bir risktir [20]. Araç ağlarının tasarımı güvenlik, gizlilik ve verimlilik açısından kritik öneme sahiptir. Çünkü güvenlik ve gizlilik eksikliği varsa, ağ sistemi siber saldırıya uğrayabilir. Bu siber saldırı kişisel verilerin çalınmasına, trafik kazalarına, yaralanmalara ve hatta ölümlere yol açabilir [21]. Bu nedenle bu ağların tasarımında güvenli iletişim için gizliliği koruyan kimlik doğrulama şemaları kullanılmaktadır.

Bu çalışmada araç ağlarının güvenliğini sağlamak için kriptografik yöntemlere dayanan koşullu gizliliği koruyan yeni ve özgün bir kimlik doğrulama (CPPA) şeması önerilmiştir. Kimlik doğrulama şemalarının tasarımında üç önemli kriter vardır: araç kimliklerinin anonimliği, siber saldırılara karşı güvenlik, düşük hesaplama ve iletişim maliyetleri. Bununla birlikte farklı ek özellikleri sağlayacak şekilde de tasarım gerçekleştirilebilir. Araçların kimliklerinin anonim olmasının nedeni araç takibini önlemektir. Eğer mesajı gönderen araç, saldırganlar tarafından tespit edilirse araç ağında ciddi bir güvenlik sorunu ortaya çıkacaktır. Bununla birlikte, herhangi bir araç tarafından gerçekleştirilen kötü niyetli bir davranış tespit edilirse, araç yalnızca merkezi otorite tarafından araç ağından çıkarılabilir, bu nedenle aracın gizliliği şartlıdır [3, 22].

CPPA şemalarının tasarımı ile ilgili literatürde açık anahtar altyapı tabanlı, eliptik eğri tabanlı, eşleme tabanlı, kimlik tabanlı ve serifikasız kriptografi tabanlı farklı kriptografik yöntemler kullanılmaktadır [2, 3, 5, 7, 8, 23-34, 46]. Ayrıca V2V, V2I, V2P gibi farklı iletişim modları için de tasarım gerçekleştirilebilir [7, 26, 35]. CPPA şemalarının tasarımında eliptik eğri kriptografisi (Elliptic Curve Cryptography; ECC) ve çift doğrusal eşlemeye dayalı kriptografi yaygın olarak kullanılmaktadır. Ancak bu yöntemlerin verimlilik ve gerçekleştirim süreleri farklılık göstermektedir. Çift doğrusal eşleme yönteminde matematiksel işlemlerin hesaplama maliyetleri oldukça yüksektir. Buna karşın araç ağındaki kullanıcılar tarafından gönderilen mesajların imzalanması ve doğrulanması aşamalarının oldukça hızlı ve verimli olması gerekmektedir. Muhtemel gecikmeler trafik

güvenliğini ve trafik akışını olumsuz etkilemektedir. Bu sorundan kaçınmak için gecikme süreleri en aza indirilmelidir. Bu nedenle, tasarlanan CPPA şemaları için ECC tabanlı eşlemesiz hafif bir şifreleme yöntemi kullanmak verimliliği artırır [34].

ECC aynı güvenlik seviyesi için diğer şemalara kıyasla daha küçük bir anahtar boyutu kullanmaktadır. Böylece hesaplama maliyeti, depolama alanı, bant genişliği ve güç tüketimi gibi parametrelerde avantaj sağlamaktadır [36, 37]. Ancak ECC tabanlı tasarlanan CPPA şemasının açık anahtar kriptografisine (Public Key Cryptography; PKC) dayanması durumunda, açık anahtar altyapısı (Public Key Infrastructure; PKI) ve sertifika ihtiyacı nedeniyle araç ağlarında kullanılması pratik olmayacaktır [42]. Bu sebeple tasarımda kimlik tabanlı kriptografinin (Identity Based Cryptography; IBC) [38] kullanılması açık anahtar altyapısı ve sertifika sorununu ortadan kaldıracaktır [39, 40]. Bu yapıda kullanıcıların açık anahtarı e-posta gibi kendilerine özgü bir bilgidir. Bu kriptografi yönteminde PKI ve sertifika yerine gizli anahtarın üretildiği gizli anahtar üreticisi (PKG) mevcuttur. Ancak tüm gizli anahtarların PKG tarafından üretilmesi anahtar emaneti sorununa neden olmaktadır [43, 45]. Eğer PKG siber saldırıya uğrarsa sisteme kayıtlı kullanıcıların gizli anahtarı kötü niyetli üçüncü tarafların eline geçebilir. Bu durumda araç sisteminde güvenlik sorunları ortaya çıkacaktır. Anahtar emanet sorununu çözmek için sertifikasız kriptografi yöntemi önerilmiştir [44]. Sertifikasız kriptografide IBC'ye benzer olarak kullanıcıların herkes tarafından bilinen bir kimliği ve anahtar üretim merkezi (KGC) vardır. Ancak bu merkez yarı güvenilir bir konumda olup gizli anahtarın sadece bir kısmını üretebilmektedir. Tam gizli anahtar KGC tarafından üretilmediğinden anahtar emanet sorunu olmayacaktır.

Raya ve Hubaux tarafından açık anahtar altyapısına dayalı anonim bir kimlik doğrulama şeması önerilmiştir [24]. Anonim bir kimlik oluşturmak için anahtar çiftleri ve sertifikalar araç üstü birime (OBU) yüklenir. Ancak, bu şema çok fazla depolama alanı gerektirmesi ve açık anahtar altyapısı kullanması sebebiyle verimsizdir.

He tarafından çift doğrusal eşleme ve noktadan noktaya özet fonksiyonu içermeyen kimlik tabanlı bir CPPA şeması önerilmiştir [5]. Bu şema söz konusu matematiksel işlemlerin olmaması sayesinde oldukça verimlidir. Ayrıca toplu mesaj doğrulamasını da desteklemektedir.

Ali, VANET'ler için çift doğrusal eşleme ve tek yönlü özet fonksiyonlarını kullanan verimli bir CPPA şeması önermiştir. Bu şema V2I iletişimi gerçekleştirir ve toplu mesaj doğrulamayı desteklemektedir [26].

Lu, yol kenarı birimi (RSU) tarafından araçlara geçici anonim sertifika dağıtımına dayanan yeni bir çözüm önermiştir [27]. Ancak bu şema verimsizdir çünkü RSU'ların yayın alanındaki her araç için geçici sertifikalar gerektirmektedir. PKI'da araçlar kimlik doğrulama için açık anahtar, özel anahtar ve merkezi otoriteden alınan sertifikaları kullanır.

İlk sertifikasız kriptografi (CLC) şeması Al-Riyami ve Paterson tarafından önerilmiştir [44]. Ayrıca CLC tabanlı şema olarak Kamil tarafından çift doğrusal eşleme gerektirmeyen sertifikasız toplu imza (Certificateless Aggregate Signature; CL-AS) şeması önerilmiştir [30].

Li, IoV için bir sertifikasız koşullu gizliliği koruyan kimlik doğrulama (CL-CPPA) şeması önermiştir. Kimlik doğrulama şemalarındaki güvenlik eksiklikleri incelenmiş ve bu güvenlik açıklarını gidermek için bir şema tasarlanmıştır [32].

Thumbur tarafından toplu mesaj doğrulamasını destekleyen RSU'nun yükünü azaltan bir CL-AS şeması önerilmiştir [43]. Bu şemada, anonim kimlik ve kısmi özel anahtar bir kez üretilir ve araçtaki OBU'da saklanır. Ancak mesaj imzalandıktan sonra, yayınlanan parametrelerdeki tüm değişikliklere rağmen anonim kimlik değişmez. Bu durumda bir saldırgan, kullanıcı tarafından aynı anonim kimlik kullanılarak gönderilen mesajlarla bağlantı kurabilir ve kullanıcıyı tespit edebilir.

Zhang VANET'ler için PA-CRT adı verilen Çinlilerin kalan teoremine dayalı bir CPPA önermiştir. Bu şemada, sürücüler araçlarındaki kurcalamaya dayanaklı cihazı (Tamper Proof Device; TPD) etkinleştirmek için parmak izi kullanırlar. Böylece, sürücüler gerçek kimlikleri veya şifreleri yerine parmak izlerini kullanarak TPD'yi etkinleştirirler [47].

Zhou tarafından VANET'ler için imzalama ve doğrulama süreleri ve bant genişliği açısından mevcut şemalardan daha iyi performans gösteren bir CL-CPPA önerilmiştir [48]. Bu şemada, anonim kimlik ve kısmi gizli anahtar, merkezi birim tarafından toplu olarak

üretilir ve araçlardaki OBU'lara önceden yüklenir. Yüklenen anonim kimlikler ve kısmi gizli anahtarlar tükenirse, araçlar merkezi birimle tekrar iletişime geçerek gizli anahtar talep eder. Ancak bu durum araçların depolama kapasitesini olumsuz etkilemektedir. Ayrıca, araç iyi niyetli olmayan kişiler tarafından ele geçirilir veya merkezi otorite tarafından cezalandırılırsa, OBU'ya önceden yüklenmiş anonim kimlikleri ve kısmi gizli anahtarları kullanmaya devam edebilir.

Chen tarafından VANET'ler için verimli, ECC tabanlı CL-AS şeması önerilmiştir [40]. Bu şemada, anonim kimlik OBU'ya önceden yüklenmiştir. Ancak, bu sistemde anonim kimlik her mesaj gönderildiğinde değişmez. Sadece, anonim kimlik için belirli bir geçerlilik süresi belirlenmiştir. Bu süre zarfında imzalanan mesajlar aynı anonim kimliği içerir. Mesajlar bu anonim kimlik ile ilişkilendirilebilir. Bu nedenle, önerilen şema bağlantısızlık sağlamaz.

Bu tez kapsamında, literatürde bulunan kimlik doğrulama şemalarının güvenlik zafiyetleri, performansları ve kriptografik yöntemleri detaylı olarak incelenmiştir. Bu çalışmada önemli hususlar dikkate alınarak sertifikasız koşullu gizliliği koruyan yeni ve özgün bir kimlik doğrulama (CL-CPPA) şeması tasarlanmıştır. Önerilen şemada anonim kimlik üretimi tek bir merkezde gerçekleştirilmez. Merkezi birim tarafından üretilen kısmi anonim kimlikten yola çıkılarak tam anonim kimlik araç tarafından üretilir. Böylece merkezi birimden sürekli anonim kimlik talep edilmez. Bu sayede merkezi birimin aşırı yüklenmesine engel olunur. Ayrıca, mesaj imzalama için tek bir anonim kimliğin kullanılması, aracın kimliğini siber saldırılara maruz bırakabileceğinden önerilen şemada her mesaj için farklı bir anonim kimlik üretilir. Bu sayede anonim kimlik ve mesajlar arasında bağlantı kurulamaz. Önerilen şema trafikteki verimliliği artırmak için toplu mesaj doğrulamayı da desteklemektedir. Böylece, birden fazla mesaj hızlı bir şekilde doğrulanır. Araçlar ve RSU'lar aynı anda birçok araçtan mesaj alacağından, doğrulama aşamalarının öncelikli olarak dikkate alınması gerekmektedir. Bu nedenle önerilen şemada bu husus göz önüne alınmıştır. Ayrıca her mesaj iletimi sırasında araçlarda anonim kimlik oluşturulur ve saldırganlar mesaj ile anonim kimlik arasında bağlantı kuramaz. Araç ağı sistemine kayıtlı kullanıcılar birbirleriyle haberleşirken anonim kimliklerini kullanırlar. Ancak bu durum araçların anonim kimliklerini kullanarak kimse tarafından tespit edilmeden sistemi istediği gibi manipüle etmesine olanak vermez. Merkezi birim aracı izleyebilir ve gerçek kimliğini ortaya çıkarabilir. Önerilen yöntemde, kullanıcının gizli anahtarının bir kısmı KGU adı verdiğimiz anahtar üretim birimi tarafından üretilir. Daha sonra, araçlar ve RSU'lar kendi

gizli anahtarlarını oluştururken bu kısmi gizli anahtarı (Partial Private Key; PPK) kullanırlar. Böylece gizli anahtar, merkezi bir birim tarafından üretilmediği için anahtar emanet sorunu ortadan kalkmaktadır.

Ayrıca literatürdeki şemaların çoğunda enerji tüketimi hesaplaması ve karşılaştırması eksiktir, bu husus dikkate alınarak enerji tüketim maliyeti hesaplanmış ve diğer mevcut şemalarla karşılaştırılmıştır. Ek olarak önerilen şemanın detaylı bir güvenlik değerlendirmesi gerçekleştirilerek bu tez çalışmasında sunulmuştur.

Kriptografi sistemlerinin tasarımında birçok tasarım parametresinin önemli olması sebebiyle bu tez çalışmasında özgün bir eliptik eğri tabanlı şifreleme sistemi ve verimli bir imza algoritması da önerilmiş ve tasarlanmıştır. Ayrıca konunun daha iyi açıklığa kavuşturulabilmesi için ECC ve eşleme tabanlı kriptografi detaylı olarak incelenmiştir. Uygulamalar ve hesaplamalar verimlilik ve etkinlik açısından C++ programlama dilinde gerçekleştirilmiştir.

Bu tez çalışmasının ana katkıları aşağıda sunulmuştur.

- Bu çalışma ile hafifsiklet, enerji tasarruflu, çift doğrusal eşleme kullanmayan, verimli ve güvenli bir özgün CL-CPPA şeması sunulmuştur. Bu şemada anonim kimlik tek bir merkezde üretilmemektedir. Bu sayede merkezi birimde herhangi bir sorun oluşması durumunda araç ağındaki kesinti olasılığı en aza indirilir.
- V2V, V2I, I2I ve I2V gibi farklı iletişim modlarında çalışabilir. Ayrıca önerilen şema V2P ve V2G gibi V2X iletişimi için genişletilebilir.
- Bu şema sertifikasız kriptografi özelliği sağlaması sayesinde kimlik tabanlı kriptografide bulunan anahtar emanet problemini çözer ve toplu mesaj doğrulamayı destekler.
- Anonim kimlikler ve araç gizli anahtarı veya RSU gizli anahtarı, araç veya RSU tarafından her mesaj gönderildiğinde tekrar oluşturulur. Böylece araç ve RSU tarafından yayınlanan genel parametreler sürekli değişmekte ve mesaj ile genel parametreler arasındaki bağlantı engellenmektedir.
- Önerilen şemada merkezi birim (Central Unit; CU) ve anahtar üretim birimi (Key Generation Unit; KGU) merkezi otoritelerdir ve sadece belirli aralıklarla araçlarla ve

RSU ile iletişim kurarlar. Araçların ve RSU'ların gizli kısmi anonim kimlikleri CU, KGU tarafından üretilir. Son olarak, araç ve RSU, her mesaj gönderiminden önce tam anonim kimliklerini oluşturur. Böylece merkezi birimin aşırı yüklenmesi engellenmiş olunur.

- Önerilen şema, imzalama ve doğrulama aşamalarında IoV ağının verimliliği göz önünde bulundurularak modellenmiştir. Araç ağlarında mesaj alma ve mesaj gönderme farklı yoğunluklara sahip olacaktır. Özellikle bir aracın veya RSU'nun çok sayıda mesaj alması ve doğrulaması gerekecektir, bu durum dikkate alınarak doğrulama aşamasının hesaplama maliyeti üzerinde çalışılmış ve önerilen şemanın performansı artırılmıştır.
- Araçların ve RSU'ların depolama ve işleme kapasitelerinin birbirinden farklı olacağı göz önünde bulundurularak iletişim maliyeti, depolama kapasitesi maliyeti ve enerji tüketim maliyeti optimize edilmiştir.
- Bu tez çalışmasında ayrıca eliptik eğri tabanlı özgün bir mesaj şifreleme sistemi tasarlanmıştır. Bu tasarımda ASCII tablosunun sınırlı sayıda karakter içermesi dikkate alınarak Unicode karakterlerini içerecek şekilde özgün bir yöntem sunulmuştur.
- Bununla birlikte, klasik eliptik eğri tabanlı dijital imza algoritması olan ECDSA'da bulunan modüler aritmetikte ters alma işlemi matematiksel olarak maliyetli bir işlemdir. Söz konusu matematiksel işlem imza algoritmasından kaldırılarak verimli ve özgün bir ECDSA önerilmiştir. Bu kapsamda gerçek bir uygulamada kullanılabilecek NIST (National Institute of Standards and Technology) tarafından kullanılması tavsiye edilen parametreler ile algoritma gerçekleştirilmiştir. Ayrıca güvenlik ve performans analizleri de bu tez çalışmasında sunulmuştur.

2. TEMEL KAVRAMLAR

Bu bölümde, kriptoloji ile ilgili temel kavramlar sunulmuştur. Özellikle ilerleyen bölümlerde konunun daha açık anlaşılabilmesi için kriptografi konusuna ağırlık verilmiştir. Bu kapsamda kriptografi, kriptoanaliz, kriptografi sistemleri, gizli anahtarlı kriptografi, açık anahtarlı kriptografi, açık anahtar altyapısı, dijital imza, özet fonksiyonu, kriptografideki matematiksel arka plan ve mesaj doğrulama kodu hakkında bilgiler sunulmuştur.

2.1. Kriptoloji

Kriptoloji, gizli ve güvenli bir haberleşme gerçekleştirebilmek için matematiksel yöntemler kullanılarak mesajların şifrelenmesi ve şifreli mesajların çözülmesi bilimidir. Kriptoloji kavramı kriptografi ve kriptoanaliz olmak üzere iki alt bölümde incelenir. Kriptografi, mesajın gizliliğini sağlamak amacıyla kriptograflar tarafından matematiksel yöntemler kullanılarak gerçekleştirilen tasarım ve analizleri ifade eder. Kriptoanaliz ise şifrelenmiş mesajın şifreleme kuralını bilmeden mesajın şifresini çözme teknikleriyle ilgilenen bilimdir [49, 50, 51]. Kriptoloji başlı başına bir bilim dalı gibi gözükse de aslında altyapısının büyük bir bölümünü matematik bilimi oluşturmaktadır. Sayılar teorisi, asal sayılar, gruplar, halkalar, cisimler, çarpanlara ayırma, matrisler ve modüler aritmetik gibi matematiğin önemli konularından birçoğu kriptoloji altyapısında kullanılmaktadır. Bu tez çalışması kriptografi üzerine olduğundan dolayı aşağıda kriptoanaliz hakkında kısa bir bilgi verildikten sonra kriptografi konusu detaylı olarak incelenmiştir.

2.2. Kriptoanaliz

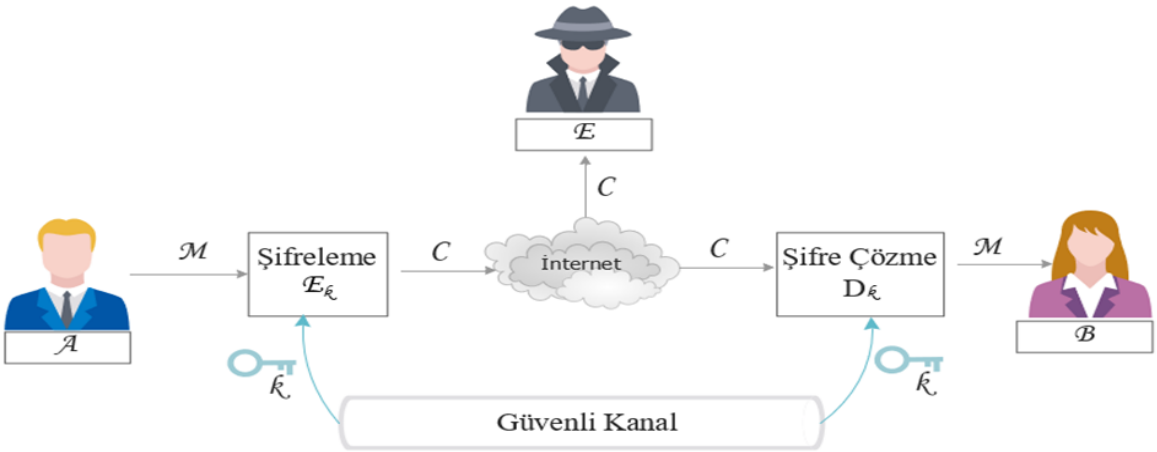
Kriptoanaliz, kriptografi sistemlerinin şifresini kırma bilimidir. İstihbarat veya çok gizli haberleşmelerde yaygın olarak kullanılmaktadır. Ayrıca modern kriptografi sistemlerinin güvenliğini test etmek için kritik bir öneme sahiptir. Kötü niyetli saldırganlar tarafından kriptografi sistemlerini kırmak üzere farklı kriptoanaliz yöntemleri uygulanır [51]. Eğer kriptografi sistemleri bu saldırılara karşı yeteri kadar dayanabiliyorsa güvenli bir kriptografi sistemi olarak tanımlanabilir. Kriptoanaliz; klasik kriptoanaliz, uygulama

saldırıları ve sosyal mühendislik saldırıları olmak üzere farklı yöntemlerle gerçekleştirilebilir [49].

Klasik kriptoanaliz: bir şifreli metinden düz metni elde etme veya şifreli metinden gizli anahtarı elde etme bilimi olarak tanımlanabilir. Klasik kriptoanaliz ikiye ayrılır: birincisi kriptografi sisteminin matematiksel iç yapısını dikkate alan analitik saldırılar, ikincisi kriptografi algoritmasını bir kara kutu gibi ele alan ve tüm olası anahtarları test eden kaba kuvvet saldırıları. Uygulama saldırıları ise daha çok fiziki sistemlerde kullanılabilir. Örneğin; yan kanal analizi ile gizli anahtar üzerinde çalışan bir işlemcinin elektriksel güç tüketimi ölçülerek kriptografi sisteminin gizli anahtarı elde edilebilir. Güç tüketimine ek olarak, elektromanyetik radyasyon veya algoritmaların çalışma zamanındaki davranışı, gizli anahtar hakkında bilgi verebilir [49]. Sosyal mühendislik saldırıları ise günümüzde oldukça yaygın kullanılan bir kriptoanaliz yöntemidir. Bu tür saldırılarda kullanıcıların dikkatsizliğinden, dalgınlığından ve bilgisizliğinden faydalanılarak kriptografi sisteminin gizli anahtarı elde edilebilir.

2.3. Kriptografi

Güvenli olmayan bir kanal üzerinden iletişimin güvenli olarak gerçekleştirilmesi için iki kullanıcı arasında mesajlar şifrelenerek gönderilir. Temel bir kriptografi sistemi Şekil 2.1'deki gibi modellenebilir. Şekilde görüldüğü üzere $\mathcal{M}$ ile gösterilen şifrelenmemiş düz metni (plaintext), C ile gösterilen şifreleme işlemi gerçekleştirilerek elde edilen şifreli metni (ciphertext) ve k ise anahtarı (key) ifade eder.



Şekil 2.1. Temel bir kriptografi sistemi

$\mathcal{A}$ tarafından şifrelenen mesajın $\mathcal{B}$ tarafında çözülerek düz metin haline getirilmesi için her iki kullanıcının da k anahtarına sahip olması gerekmektedir. Anahtarın, $\mathcal{A}$ ve $\mathcal{B}$ arasında dağıtılması için güvenli bir kanala ihtiyaç vardır. $\mathcal{A}$ tarafından gönderilen mesaj E_k şifreleme kuralı ile şifrelenerek güvenli olmayan bir kanal üzerinden $\mathcal{B}$'ye gönderilir. $\mathcal{B}$, şifreli metnin şifresini çözmesi için gerekli anahtara sahip ise D_k şifre çözme kuralı ile şifreli metni çözerek $\mathcal{M}$ düz metnini elde eder. Kötü niyetli $\mathcal{E}$ ise iki kişi arasındaki haberleşmeyi dinlemek ve bilgileri elde etmek için farklı yöntemler kullanan bir saldırgandır. $\mathcal{E}$'nin, güvenli olmayan kanal üzerinden elde etmiş olduğu şifreli metni çözmesi için k anahtarına sahip olması gerekir. Ancak k anahtarı güvenli bir kanal üzerinden iletildiğinden $\mathcal{E}$ tarafından elde edilmesi zordur. Düz metin $\mathcal{M}$ ve şifreli metin C arasındaki ilişki aşağıdaki eşitlikler ile ifade edilebilir [52].

$$C = E_k(\mathcal{M}) \tag{2.1}$$

$$\mathcal{M} = D_k(C) \tag{2.2}$$

Kriptografi sistemlerinde anahtar dağıtımı oldukça önemli bir konudur [50]. $\mathcal{A}$ ve $\mathcal{B}$ anahtarı birbirleriyle gizli olarak paylaşabilecekleri gibi matematiksel yöntemlerle uygun bir anahtar seçebilirler. Bu sebeple anahtar dağıtım algoritması, kriptografi sisteminin yapısını belirlemede oldukça önemli bir unsurdur. Kriptografi sistemlerinde dikkat çekici diğer bir husus aynı zamanda hem şifreleme hem de şifre çözme algoritmalarının herkes tarafından bilinmesidir. Bu algoritmaların gizli tutulmasının kriptografi sistemlerinin kırılmasını zorlaştıracağı açıktır ancak gizli algoritmalar aynı zamanda test edilmemiş algoritmalar anlamına da gelir. Bir kriptografi sisteminin güçlü olup olmadığını, yani kararlı bir saldırgan tarafından kırılıp kırılamayacağını öğrenmenin yollarından birisi onu herkese açık hale getirmek ve diğer kriptograflara analiz ettirmektir.

Şekil 2.1'de verilen örnek kriptografi sisteminde kötü niyetli $\mathcal{E}$, anahtarı ele geçirirse, algoritma herkes tarafından bilindiği için mesajın şifresini kolayca çözebilir. Burada dikkat edilecek husus, bir mesajı güvenli bir şekilde iletme probleminin, bir anahtarı gizlice iletme ve anahtarı güvenli bir şekilde saklama problemlerine indirgenmiş olmasıdır. Bu senaryoda gizlilik problemine dikkat çekilmiş olsa da güvenli bir kriptografi sisteminin tasarımında farklı unsurlar mevcuttur. Kötü niyetli $\mathcal{E}$'nin, mesaj üzerinde fark edilmeyen

değişiklikler yapmasını önlemek (mesaj bütünlüğü), bir mesajın gerçekten $\mathcal{A}$ tarafından gönderildiğine emin olmak (kimlik doğrulaması) ve $\mathcal{A}$ tarafından gönderilen mesajın yine kendisi tarafından gönderildiğinin inkâr edilememesi gibi güvenli bir kriptografi sisteminin taşıması gereken özellikler vardır.

Yukarıda özetlenen senaryolar dikkatli bir şekilde incelendiğinde $\mathcal{A}$ ve $\mathcal{B}$ arasında güvenli iletişim için kriptografi sistemlerinin tasarımında aşağıdaki temel unsurlar dikkate alınır. Bunlar aynı zamanda kriptografi sistemlerin temel amaçlarındandır.

- Gizlilik (Privacy / Confidentiality): Gönderilen şifrelenmiş mesajı sadece yetkisi olan kişilerin görebilmesini sağlar [53]. $\mathcal{A}$ tarafından gönderilen mesajın kötü niyetli üçüncü kişi olan $\mathcal{E}$ tarafından okunamaması gizlilik sayesinde sağlanır.
- Bütünlük (Integrity): Gönderilen mesajın yetkisiz kişiler tarafından değiştirilmesinin mümkün olmamasıdır [53, 54]. $\mathcal{A}$'nın gönderdiği mesaj üzerinde $\mathcal{E}$ tarafından herhangi bir değişiklik yapılmışsa $\mathcal{B}$'nin bu durumu tespit etmesi mesajın bütünlüğü kontrol edilerek gerçekleştirilir.
- Kimlik denetimi (Authentication/Identification): $\mathcal{B}$ tarafından alınan mesajın gerçekten doğru bir kaynaktan gelip gelmediğini belirlemek için kimlik denetimine ihtiyaç duyulur. $\mathcal{B}$, $\mathcal{A}$ tarafından gönderildiği iddia edilen şifrelenmiş mesajın gerçekten de $\mathcal{A}$ tarafından gönderildiğini doğrulayabilmelidir [53, 54].
- İnkâr edilemezlik (Non-repudation): Gönderilen şifreli metnin göndericisi tarafından inkâr edilememesidir. Oldukça önemli bir husus olup gizli mesaj sonrası ortaya çıkacak ciddi problemlerde mesajı gönderen taraf kendisi tarafından bu mesajın gönderildiğini inkâr edememelidir [53, 54]. $\mathcal{B}$, $\mathcal{A}$'dan geldiği iddia edilen bir mesaj aldığında, yalnızca $\mathcal{B}$, mesajın $\mathcal{A}$'dan geldiğine ikna olmakla kalmaz, aynı zamanda $\mathcal{B}$, tarafsız üçüncü bir kişiyi ikna edebilir; bu nedenle $\mathcal{A}$, mesajı $\mathcal{B}$'ye gönderdiğini inkâr edemez.

2.4. Kriptografik Sistemler

Kriptografik sistemler gizli anahtarlı (simetrik) ve açık anahtarlı (asimetrik) kriptografi sistemleri olmak üzere ikiye ayrılır [55]. Bu başlık altında öncelikle temel bir kriptografi sisteminin tasarım adımları tanımlanmıştır. Sonrasında gizli anahtarlı ve açık anahtarlı

kriptografi sistemleri hakkında bilgi verilmiştir. Çizelge 2.1'de temel bir kriptografi sisteminde kullanılan parametrelerin tanımları verilmiştir.

Çizelge 2.1. Kriptografi sisteminde kullanılan parametreler ve tanımları

Parametre	Tanım
$\mathcal{K}$	Gizli anahtarları içinde barındıran anahtar uzayı
$k_{\mathcal{A}}$, $k_{\mathcal{B}}$	$\mathcal{A}$ ve $\mathcal{B}$ taraflarının gizli anahtarları
$\mathcal{M}$	Şifrelenmemiş düz metin (plaintext)
C	Şifreli metin (ciphertext)
$\mathcal{E}_k$	Şifreleme kuralı
$\mathcal{D}_k$	Şifre çözme kuralı

Temel bir kriptografi sisteminin şifreleme ve şifre çözme adımları aşağıdaki gibidir.

1. $\mathcal{K}$, anahtar uzayı olarak adlandırılan sonlu anahtar kümesi olarak belirlenir. Şifrelemede kullanılacak anahtar $k_{\mathcal{A}}$, $\mathcal{K}$ kümesinin elemanlarından seçilir ($k_{\mathcal{A}} \in \mathcal{K}$).
2. Her $k_{\mathcal{A}} \in \mathcal{K}$ için düz metin $\mathcal{M}$'yi şifreli metin C'ye dönüştürecek birebir fonksiyon özelliğine sahip $\mathcal{E}_k$ şifreleme kuralı belirlenir. Şifreleme kuralı belirlenirken şifreli metnin tekrar düz metin haline dönüştürülmesi gerektiği düşünülerek $\mathcal{E}_k$ şifreleme kuralının birebir fonksiyon olması gerekmektedir.
3. Şifrelenmiş mesajı çözmek için kullanılacak anahtar $k_{\mathcal{B}}$, $\mathcal{K}$ kümesinin elemanlarından seçilir ($k_{\mathcal{B}} \in \mathcal{K}$).
4. Her $k_{\mathcal{B}} \in \mathcal{K}$ için şifreli metin C'yi düz metin $\mathcal{M}$'ye dönüştürecek birebir fonksiyon özelliğine sahip $\mathcal{D}_k$ şifre çözme kuralı belirlenir.
5. Bir kriptografi sisteminin algoritması $\{\mathcal{E}_k : k_{\mathcal{A}} \in \mathcal{K}\}$ şifreleme kuralı ve $\{\mathcal{D}_k : k_{\mathcal{B}} \in \mathcal{K}\}$ şifre çözme kuralından oluşur. Her $k_{\mathcal{A}} \in \mathcal{K}$ için $\mathcal{D}_k = \mathcal{E}_k^{-1}$ eşitliğini sağlayacak benzersiz $k_{\mathcal{B}} \in \mathcal{K}$ anahtarı vardır. Böylece her $m \in \mathcal{M}$ düz metni için $\mathcal{D}_k(\mathcal{E}_k(m)) = m$ eşitliği sağlanır.

Bu adıma kadar olan özellikler temel kriptografi sisteminin tasarımında kullanılabilir. Ancak gizli anahtarlı veya açık anahtarlı kriptografi sistemlerinde, anahtar üretimi ve dağıtımı farklılık göstermektedir. $\{k_{\mathcal{A}}, k_{\mathcal{B}}\}$ anahtar çiftindeki anahtarların birbirine eşit

olması durumunda gizli anahtarlı (simetrik) kriptografi sistemi gerçekleştirilir. Ancak anahtarlar arasında matematiksel yöntemlerle ilişki oluşturularak gizli ve açık anahtar belirlenmesi durumunda ise açık anahtarlı (asimetrik) kriptografi sistemi tasarlanır.

2.4.1. Gizli anahtarlı (simetrik) kriptografi

Gizli anahtarlı kriptografide hem şifreleme hem de şifre çözme için aynı anahtar kullanılır. Bu özelliğinden dolayı simetrik kriptografi sistemi olarak da adlandırılmaktadır. Gizli anahtarlı kriptografi sisteminde tasarım kolay olup şifreleme ve şifre çözme süreçleri hızlıdır. Ancak iletişim halindeki kullanıcılar arasında gizli anahtarın paylaşımı kullanıcı sayısı arttıkça zorlaşır ve güvenlik riskleri oluşabilir. Kötü niyetli üçüncü taraf şifreleme için kullandığımız anahtarı olan k'yı ele geçirirse, gönderdiğimiz herhangi bir mesajın şifresini çözebilir. Sezar şifreleme [56, 57], DES [56, 58] ve AES [56, 58] gizli anahtarlı kriptografiye örnek olarak verebileceğimiz kriptografi sistemleridir. Gizli anahtarlı kriptografide anahtar dağıtımının oldukça zordur [59]. Bu sebeple çok kullanıcılı sistemlerde anahtar dağıtımı problemi ortaya çıkmaktadır.

2.4.2. Açık anahtarlı (asimetrik) kriptografi

Gizli anahtarlı kriptografideki anahtar dağıtım problemine çözüm üretmek amacıyla ilk pratik öneri 1976 yılında Diffie-Hellman [60] tarafından ortaya atılmıştır. Bu çalışma sonrasında açık anahtarlı kriptografi üzerine çalışmalar hızlanmıştır. Açık anahtarlı kriptografide hem şifreleme hem de şifre çözme için farklı anahtarlar kullanılır. Bu anahtarlar; herkes tarafından bilinen açık anahtar (public key) ve sadece kullanıcısı tarafından bilinen gizli anahtar (private key) olarak adlandırılır. Çalışma mantığına göre açık olan anahtara herkes ulaşabilir ve bu anahtarı kullanarak matematiksel yollarla gizli olan anahtara ulaşmanın bir yolu olmamalıdır yani gizli anahtarı elde etmek matematiksel olarak oldukça zor olmalıdır. Açık anahtara sahip olan herkes mesajı şifreleyebilir ancak mesajın şifresinin gizli anahtar kullanılarak çözülmesi gerekmektedir.

$\mathcal{A}$, mesajı göndermeden önce alıcı $\mathcal{B}$'nin açık anahtarını kullanarak mesajı şifreler, mesajı alan $\mathcal{B}$ ise $\mathcal{A}$ tarafından gönderilen şifreli mesajı kendisine ait gizli anahtarını kullanarak çözer. RSA [58, 61], ElGamal [49], DSA [49] ve ECC [62] kriptografi sistemleri açık anahtarlı kriptografiye örnek olarak verilebilir. Bu kriptografi sistemlerinin güvenliği; tam

sayı çarpanlara ayırma problemi, ayrık logaritma problemi ve eliptik eğri ayrık logaritma problemi gibi farklı matematiksel problemlere dayanmaktadır. Açık anahtarlı kriptografi sistemi için temel adımlar aşağıdaki gibi tanımlanabilir.

1. Öncelikle kriptografi sisteminin kullanıcıları $\mathcal{A}$ ve $\mathcal{B}$, açık anahtarlı kriptografi sistemi üzerinde anlaşırlar. Üzerinde anlaştıkları kriptografi sistemi yukarıda belirtilen RSA, ECC gibi açık anahtarlı kriptografi sistemlerinden birisi olabilir.
2. $\mathcal{A}$, göndereceği mesajı şifrelemek için $\mathcal{B}$ tarafından gönderilen açık anahtarı kullanır.
3. Mesaj $\mathcal{A}$ tarafından şifrelendikten sonra gizli mesaj alıcısı olan $\mathcal{B}$'ye bir kanal üzerinden gönderilir.
4. $\mathcal{B}$ ise almış olduğu şifreli mesajın şifresini sadece kendisinin bildiği gizli anahtarı kullanarak çözer ve düz mesajı elde eder.

Yukarda verilmiş olan adımlar temel bir açık anahtarlı kriptografi sisteminin genel adımlarını içermektedir. Burada dikkat edilecek husus gizli anahtar ve açık anahtar arasındaki ilişkinin nasıl sağlanabildiği ve her bir kullanıcının sadece gizli anahtarını kullanarak şifreli mesajı nasıl çözebildiğidir. Bu soruların cevabı olan matematiksel altyapının nasıl sağlandığına yönelik bilgi anahtar değişim protokolleri ile kolayca açıklanabilmektedir.

2.5. Diffie-Hellman Anahtar Değişim Protokolü

Diffie-Hellman anahtar değişim protokolü, 1976 yılında Whitfield Diffie ve Martin Hellman tarafından anahtar dağıtım problemine karşı önerilen ilk pratik çözümdür [60]. Bu anahtar dağıtım protokolü bir kriptografi sistemi olmayıp güvenli bir şekilde haberleşmek isteyen iki kişinin, güvenli olmayan bir iletişim kanalı üzerinden ortak bir gizli anahtarı üretebildiği ve paylaşabildiği bir protokoldür. Bu anahtar değişim protokolünün altyapısı ayrık logaritma problemine dayanmaktadır.

2.5.1. Tanım

p asal tamsayı ve $q \in \mathbb{Z}_p^*$ ilkel kök olsun. $t \in \mathbb{Z}_p^*$ seçilir. $0 \le a \le p-2$ olmak üzere;

$$q^a \equiv t \pmod{p} \tag{2.3}$$

eşitliğini sağlayan a tamsayısının bulunmasına $\mathbb{Z}_p$'de ayrık logaritma problemi denir [63].

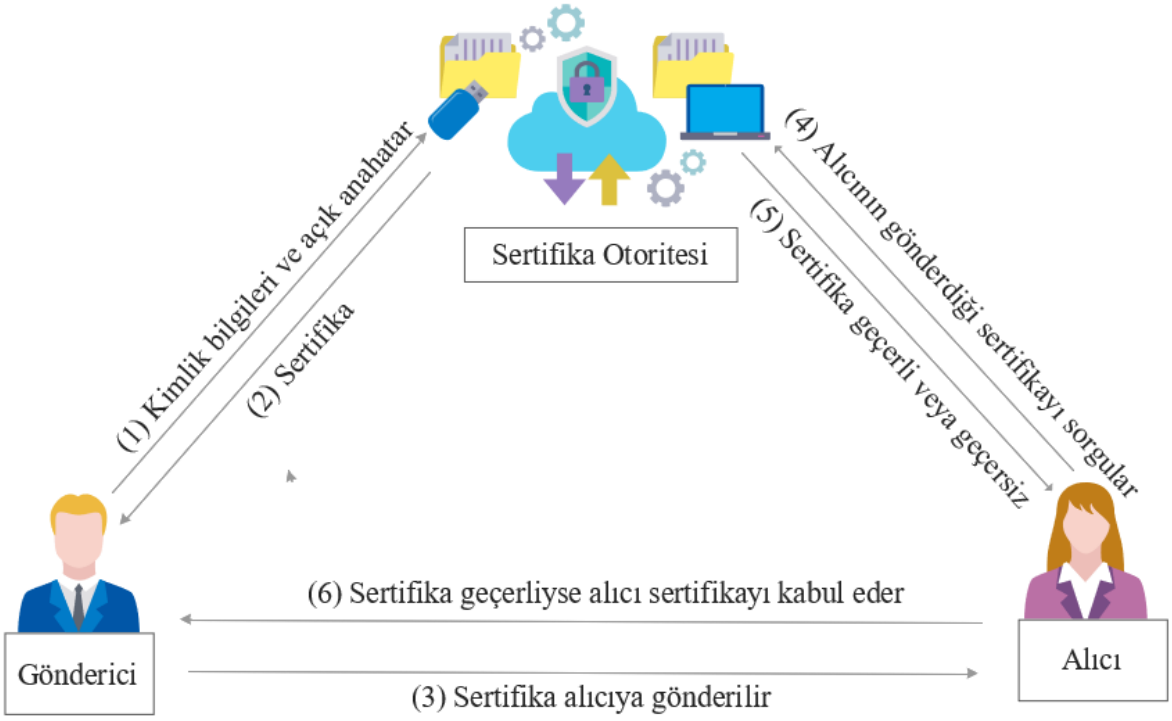
Açık anahtarlı kriptografi sistemlerinin temelini oluşturan Diffie-Hellman anahtar değişim protokolünün algoritması aşağıda verilmiştir [64].

1. Kriptografi sistemleri için yeterli büyükte p asal tamsayısı ve $q \in \mathbb{Z}_p^*$ ilkel kökü seçilir. p ve q herkes tarafından bilinen parametreler olup $\mathbb{Z}_p$'de ayrık logaritma problemini çözmek oldukça zordur.
2. $\mathcal{A}$ tarafından $0 \le a \le p-2$ şartını sağlayan rasgele gizli a değeri seçilir. Daha sonra $q^a \pmod p$ değeri hesaplanarak $\mathcal{B}$'ye gönderilir.
3. $\mathcal{B}$ tarafından $0 \le b \le p-2$ şartını sağlayan rasgele gizli b değeri seçilir. Daha sonra $q^b \pmod p$ değeri hesaplanarak $\mathcal{A}$'ya gönderilir.
4. $\mathcal{A}$, $\mathcal{B}$ tarafından gönderilen q^b değerini ve gizli anahtarı a'yı kullanarak ortak anahtarı $K = (q^b)^a$ hesaplaması ile elde eder.
5. Aynı şekilde $\mathcal{B}$, $\mathcal{A}$ tarafından gönderilen q^a değerini ve gizli anahtarı b'yi kullanarak ortak anahtarı $K = (q^b)^a$ hesaplaması ile elde eder.
6. Dikkat edilecek olursa $\mathcal{A}$ ve $\mathcal{B}$ tarafından elde edilen değer $K = (q^b)^a = (q^a)^b$ olduğundan ortak anahtar üzerinde anlaşma sağlanmış olur.

Anahtar değişim protokolünde herhangi bir mesajın şifreleme işlemi gerçekleştirilmez, sadece kullanıcılar arasında ortak anahtar üzerinde anlaşma sağlanmaktadır.

2.6. Açık Anahtar Altyapısı

Açık anahtar altyapısı (Public Key Infrastructure; PKI), açık anahtar dağıtımı ve sertifikalarını yönetmek amacıyla açık anahtarlı kriptografi sistemlerinde kullanılmaktadır. Güvenli bir otorite olan sertifika yetkilisi tarafından desteklenen bu altyapı açık anahtarların doğruluğunu kontrol etmek ve sertifikaları iptal etmek gibi temel işlemleri gerçekleştirir [50, 65]. Şekil 2.2'de genel bir açık anahtar altyapısı verilmiştir. Sertifika yetkilisi tek bir otoriteden oluşabileceği gibi birçok otoriteden oluşan karmaşık bir yapıda da olabilir.



Şekil 2.2. Açık anahtar altyapısı

Mesajı gönderen tarafın kimliğini doğrulamak için bir ortak anahtar sertifikası kullanılır. Bir kullanıcı, güvenilen sertifika otoritesinden bir sertifika almak amacıyla önce kimliğiyle birlikte açık anahtarını otoriteye gönderir. Sertifika otoritesi kullanıcının açık anahtarını ve kimliğini doğruladıktan sonra sertifikayı verir. Bu sertifikanın üretilmesi kullanıcının ortak anahtarı, kimliği ve otoritenin gizli anahtarı kullanılarak gerçekleştirilir. Kullanıcı daha sonra ilgili sertifikayı mesaj gönderdiği alıcıya sunar. Mesajı alan taraf, sertifikayı otoritenin ortak anahtarı aracılığıyla doğrular ve ardından kabul eder. Açık anahtarlı kriptografi sistemlerinde bir sertifikanın üretilmesi, iletilmesi, saklanması, doğrulanması ve iptal edilmesi hesaplama ve iletişim maliyetinin artmasına neden olur.

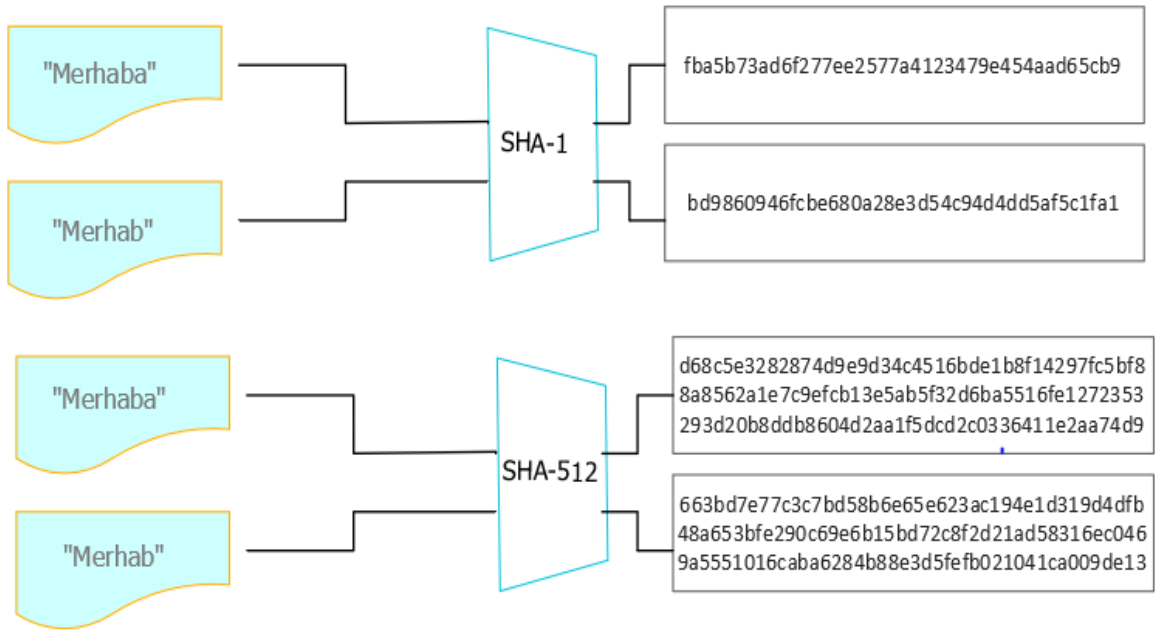
2.7. Özet Fonksiyonu

Özet (Hash) fonksiyonu rasgele uzunluktaki ikili dizileri sabit uzunluktaki ikili dizilere eşleyen hesaplama açısından oldukça verimli tek yönlü matematiksel işlemdir. Özet fonksiyonunda girdi olarak kullanılan mesajın terabayt veya bit büyüklüğünde olmasının herhangi bir önemi olmayıp çıktı olarak sabit uzunlukta bir değer üretir. İşlem sonucunda elde edilen çıktıdan tersine mühendislik yöntemleri kullanılarak girdinin elde edilmesi

mümkün değildir. Özet fonksiyonları yaygın olarak dijital imzalarda ve veri bütünlüğü doğrulamasında kullanılmaktadır. Giriş verilerinin değiştirilmediğini doğrulamak için, eldeki girdi kullanılarak özet değer yeniden hesaplanır ve daha önce hesaplanmış özet değerle eşitlik açısından karşılaştırılır. Eğer eşitlik sağlanmazsa veri bütünlüğünün bozulduğu ve veri üzerinde değişiklik yapıldığı tespit edilir. SHA (Secure Hash Algorithm) ve MD (Message Digest) ailesi yaygın kullanılan özet fonksiyonlardır [49, 66, 67].

Özet fonksiyonun kriptografi sistemlerinde güvenle kullanılabilmesi için hesaplanmasının kolay, tek yönlü ve çakışmaya dayanıklı olması gerekir. Çakışma (collision), farklı girdi değerleri için aynı çıkış özet değerinin üretilmesidir [56]. M_1 ve M_2 birbiri ile aynı olmayan iki farklı metin olsun. $h()$ ise özet fonksiyonu olup farklı iki metin için farklı özet değerler üretmesi beklenir. Eğer $h()$ fonksiyonu farklı metinler için $M_1 \neq M_2$ olmasına rağmen aynı özet değeri $h(M_1) = h(M_2)$ üretmesi durumunda bu durum çakışma olarak tanımlanır.

Günümüzde yaygın olarak kullanılan SHA özet fonksiyonunun SHA-1, SHA-224, SHA-256, SHA-384 ve SHA-512 gibi farklı türleri mevcuttur [68]. Bu özet fonksiyonlarının aralarındaki temel fark mesaj girdilerinin blok bit uzunluğu ve özet fonksiyon işlemi sonunda üretilen özet değer bit uzunluğudur.



Şekil 2.3. SHA-1 ve SHA-512 özet fonksiyonları

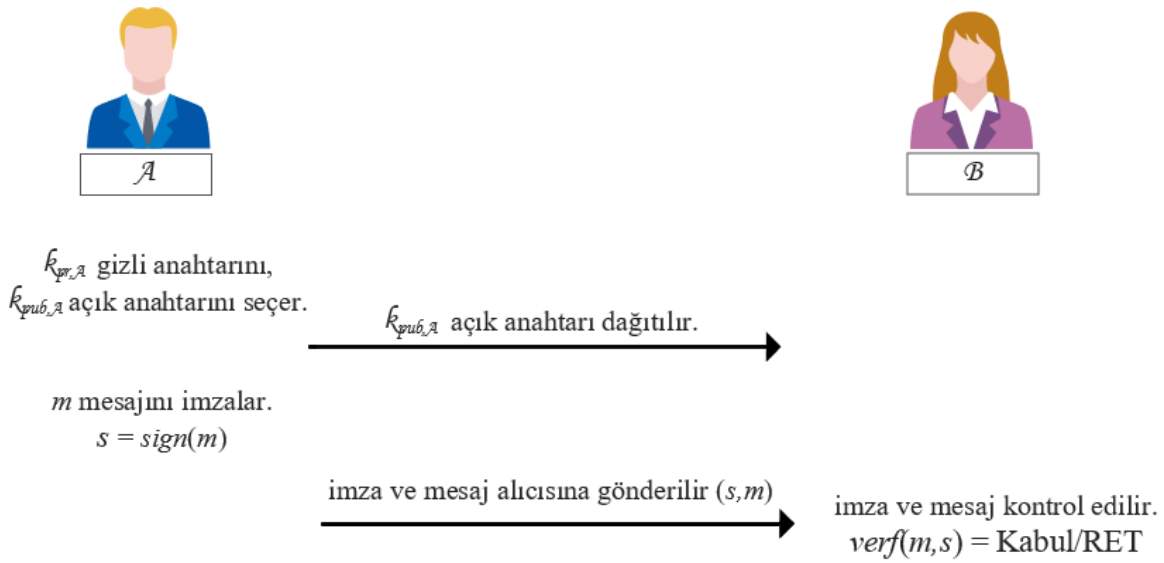
Şekil 2.3'te SHA-1 ve SHA-512 özet fonksiyonlarının "Merhaba" ve "Merhab" gibi iki kelimenin tek bir harf değişikliği ile sonucu üretilen özet değerlerinin değişimi gösterilmiştir.

2.8. Dijital İmza

Dijital imza, mesajı gönderen tarafın kimliğini doğrulamak için kullanılan bir elektronik imzadır. Genellikle, bir mesajın aslında mesajı oluşturduğunu iddia eden kişiden geldiğini garanti etmek için kullanılan bir yöntemdir. Aynı zamanda; gönderilen mesajın değiştirilmediğini yani içeriğinin aynı kaldığını doğrulamak içinde kullanılmaktadır. Bu açıklamalardan yola çıkarak; kriptografinin temel özelliklerini içerisinde barındıran dijital imzaların; bütünlük, kimlik doğrulama ve inkâr edilemezliği sağlamak için kullanıldığı görülmektedir. Modern kriptografide dijital imzalar; mesaj doğrulama, nesnelerin interneti, araçların haberleşmesi, kimlik doğrulama şemaları ve e-ticaret gibi farklı birçok uygulamada kullanım alanı bulmaktadır. Dijital imzalara örnek olarak RSA imza şeması [49, 76], ECDSA [69], DSA [49] ve ElGamal imza şemaları [49] örnek olarak verilebilir.

Dijital imzaların pratikte kullanılabilmesi için; imzalama işleminin hızlı gerçekleştirilmesi, imzanın ilgililer tarafından kolay bir şekilde doğrulanabilmesi ve imzanın sınırlı bir kullanım ömrüne sahip olması gerekir. Temel bir imza algoritması: kurulum (setup), anahtar üretimi (key generation), mesaj imzalama (message signing) ve imza doğrulama (signature verification) olmak dört aşamadan oluşur. Bu aşamalar aşağıda tanımlanmıştır [70]. Ayrıca Şekil 2.4'te temel imzalama algoritmasının adımları görsel olarak verilmiştir.

1. Kurulum (Setup): Bu aşamada dijital imza algoritması için gerekli temel parametreler belirlenir ve bunlar yayınlanır.
2. Anahtar üretimi (Key generation): Bu aşamada açık anahtar ve gizli anahtar çiftleri oluşturulur $\{k_{pr,A}, k_{pub,A}\}$.
3. Mesaj imzalama (Mesage signing): Bu aşamada imzalama fonksiyonu *sign*() ile mesaj imzalanır ve elde edilen imza ile mesaj alıcıya gönderilir.
4. İmza doğrulama (Signature verification): Bu aşamada doğrulama fonksiyonu *verfy*() ile imza doğrulanırsa mesaj kabul edilir, aksine imza doğrulanmazsa mesaj reddedilir.



Şekil 2.4. Temel dijital imza algoritması

2.9. Mesaj Doğrulama Kodu

Mesaj doğrulama kodunun (Message Authentication Code (MAC)) amacı, bir kullanıcıdan diğerine gönderilen mesajın saldırgan tarafından değiştirilip değiştirilmediğinin tespit edilmesini sağlamaktır. Mesaj doğrulama algoritmasının çıktısı bir sayısal değer olup buna MAC etiketi denir. Bu değerin üretilmesi için kullanıcının sahip olduğu bir gizli anahtar ve gönderilecek mesaj, algoritmanın girdisi olarak kullanılırken işlem sonucu elde edilen çıktı mesaj doğrulama kodudur. Mesajın alıcısı ise almış olduğu kodu kontrol etmek için kendisinde de bulunan gizli anahtarı ve mesajı kullanarak elde etmiş olduğu kodu gönderilen kod ile karşılaştırır. Eğer mesaj doğrulama kodları eşit ise mesajın bir saldırgan tarafından değiştirilmemiş olduğu belirlenir. Mesaj doğrulama kodunun güvenliği hiçbir saldırganın herhangi bir mesajı kullanarak geçerli bir mesaj doğrulama kodu üretememesine dayanmaktadır. Saldırgan mesaj üzerinde değişiklik yaptığında alıcı tarafından bu durum mesaj doğrulama kodu ile tespit edilebilir. Güvenlik işlevselliği açısından MAC'ler, aynı zamanda mesaj bütünlüğü ve mesaj kimlik doğrulaması da sağladıkları için dijital imzalarla bazı özelliklere sahiptirler. Bununla birlikte, dijital imzalardan farklı olarak, MAC'ler simetrik anahtar tabanlı olup aynı zamanda reddedilemezlik özelliğini sağlamazlar. MAC'lerin önemli bir özelliği, blok şifrelere veya özet fonksiyonlarına dayalı oldukları için dijital imzalardan çok daha hızlı olmalarıdır. MAC'ler ve dijital imzalar arasındaki en önemli fark ise, MAC'lerin hem kimlik

doğrulama etiketi oluşturmak hem de doğrulamak için simetrik bir k anahtarı kullanmalarıdır. Uygulamada MAC'ler, blok kriptografi veya özet fonksiyonu temelli olmak üzere iki farklı şekilde oluşturulur [49].

MAC algoritması anahtar üretimi ($Gen()$), mesaj doğrulama kodu üretimi ($Mac()$) ve doğrulama ($Vrfy()$) olmak üzere üç aşamadan oluşur. Anahtar üretim aşamasında girişin n uzunluğunda bir dizi (1^n) olduğu varsayılarak gizli k anahtarı üretilir. $Mac()$ aşamasında, gizli k anahtarını ve m mesajını bir t etiketine eşleyerek mesaj doğrulama kodu üretilir. Bu aşamadaki işlem $Mac_k(m)$ şeklinde gösterilir. Son olarak, doğrulama $Vrfy()$ aşaması bir k anahtarını, bir m mesajını ve bir t etiketini girdi olarak alır ve 1 (mesaj kodu geçerli) veya 0 (mesaj kodu geçersiz) çıktısı verir ($Vrfy_k(m,t)$). MAC algoritması aşağıda verilmiştir [56].

1. 1^n girişi üzerine $Gen(1^n)$ fonksiyonu gizli k anahtarını üretir, $k \leftarrow Gen(1^n)$.
2. $Mac_k(m)$ aşamasında ise gizli $k \in \{0,1\}^n$ anahtarını ve $m \in \{0,1\}^*$ mesajını bir $t \in \{0,1\}^*$ etiketine eşleyerek t etiketi üretilir.
3. Doğrulama aşamasında $Vrfy()$ ise bir $k \in \{0,1\}^n$ anahtarı, bir $m \in \{0,1\}^*$ mesajı ve bir $t \in \{0,1\}^*$ etiketi giriş olarak alınıp $Vrfy_k(m, Mac_k(m)) = b$ ile çıktı üretilir $b \in \{0,1\}$. Eğer $b = 1$ ise mesaj kodu geçerli aksi durumda mesaj kodu geçersizdir.

3. ELİPTİK EĞRİ KRİPTOGRAFİSİ

Bu bölümde, eliptik eğri kriptografisi ile ilgili temel kavramlar sunulmuştur. Eliptik eğrilerin ve eliptik eğri tabanlı kriptografinin daha anlaşılır olması için matematiksel kavramlar açıklanmıştır. Sonrasında eliptik eğriler detaylı olarak incelenmiştir. Ayrıca bu bölümde eliptik eğri tabanlı kriptografi algoritması ve dijital imza algoritması tasarlanmıştır.

3.1. Temel Cebirsel Kavramlar

Bu alt bölümde eliptik eğrilerin kriptografide kullanılmasını daha anlaşılır bir şekilde açıklamak için ikili işlem, grup, cisim, halka, sonlu cisim, asal cisimler gibi soyut cebir kavramlarını içeren temel matematiksel ifadeler sunulmuştur.

3.1.1. Tanım

A boş olmayan bir küme olsun. $A \times A$'dan A'ya bir $*$ fonksiyonuna A'da bir ikili işlem ve $(A,*)$ ikilisine de bir cebirsel yapı denir [71].

i) $\forall a,b \in A$ için A'da bir $a*b$ elemanı vardır. Bu özellik kapalılık olarak adlandırılır.
ii) Bu eleman tek türlü olarak belirlidir. Bu özellik ise iyi tanımlılık olarak adlandırılır.

3.1.2. Tanım

G boş olmayan bir küme ve $*$, G kümesi üzerinde tanımlı ikili işlem olsun. $(G,*)$ cebirsel yapısı aşağıdaki özellikleri sağlıyorsa grup olarak ifade edilir [52, 71].

i) $*$, G'de bir ikili işlemdir.
ii) $*$ işleminin G'de birleşme özelliği vardır. $\forall a,b,c \in G$ için, $a*(b*c)=(a*b)*c$ eşitliği sağlanır.
iii) $*$ işleminin, G'de birim elemanı vardır. $\forall a \in G$ için, $a*e=e*a=a$ eşitliğini sağlayacak $\exists e \in G$ tanımlıdır.

iv) $*$ işlemine göre, G'deki her elemanın tersi vardır. $a \in G$ için, $a * a^{-1} = a^{-1} * a = e$ eşitliğini sağlayacak $\exists a^{-1} \in G$ bulunabilir.

3.1.3. Tanım

$(G,*)$ bir grup ve $\forall a,b \in G$ için $a * b = b * a$ değişme özelliğini de sağlayan gruba, değişmeli grup veya Abel grup denir. Değişmeli gruplarda işlem $+$ ise toplamsal grup denir. Eğer grubun işlemi $\cdot$ ise çarpımsal grup olarak ifade edilir.

3.1.4. Tanım

G sonlu bir küme ise $(G,*)$ grubuna bir sonlu grup denir. Bu sonlu grubun eleman sayısına da grubun mertebesi denir [52].

3.1.5. Tanım

R boş olmayan bir küme ve bu küme üzerinde tanımlı iki ikili işlem $+$ ve $\cdot$ olsun. $(R,+,\cdot)$ iki işlemli cebirsel yapısı aşağıdaki özellikleri sağlıyorsa halka olarak ifade edilir [52, 71].

i) $(R,+)$ bir değişmeli gruptur.
ii) $\cdot$ işleminin R'de birleşme özelliği vardır.
iii) $\cdot$ işleminin $+$ işlemi üzerinde sağdan ve soldan dağılma özelliğine sahiptir. Böylece, $\forall a,b,c \in R$ için, $a \cdot (b+c) = a \cdot b + a \cdot c$ ve $(a+b) \cdot c = a \cdot c + b \cdot c$ eşitlikleri sağlanır.

R halkasının $+$ işlemine göre etkisiz elemanına halkanın sıfır elemanı denir ve 0_R ile gösterilir. Halkanın $\cdot$ işlemine göre etkisiz elemanı olmayabilir. Eğer varsa bu halkaya birimli halka denir ve birim eleman 1_R ile gösterilir. Bununla birlikte halka $\cdot$ işlemine göre değişme özelliğine sahipse değişmeli halka olarak ifade edilir [52].

3.1.6. Tanım

R birimli ve değişmeli bir halka olsun. $R^* = R - \{0_R\}$ ikinci işlem olan $\cdot$ işlemine göre bir grup ise R bir cisim olarak tanımlanır. Bir cismin sıfırdan farklı her elemanının çarpımsal

tersi vardır ve tektir [52]. Bir cisim halkanın özel bir hali olup birimli, değişmeli ve sıfırdan farklı her elemanının tersi olan bir halkadır [71].

3.1.7. Tanım

Sonlu sayıda elemanı bulunan cisimler sonlu cisim olarak ifade edilir. Sonlu cisim $\mathbb{F}$ ile gösterilir. Bu sonlu cisim üzerinde $+$ ve $\cdot$ ikili işlemlerini barındırır ve aşağıdaki özellikleri sağlar.

i) $(\mathbb{F},+)$ bir değişmeli gruptur.

ii) $\cdot$ işleminin $+$ işlemi üzerinde sağdan ve soldan dağılma özelliğine sahiptir. Böylece, $\forall a,b,c \in R$ için, $a\cdot(b+c)=a\cdot b+a\cdot c$ ve $(a+b)\cdot c=a\cdot c+b\cdot c$ eşitlikleri sağlanır.

iii) $(\mathbb{F}-\{0\},\cdot)$ bir değişmeli gruptur.

3.1.8. Tanım

p asal bir sayı olsun. $\{0,1,2,3,....,p-1\}$ tamsayı kümesi modüler (mod p) toplama $+$ ve çarpma $\cdot$ işlemlerine göre cisim özelliğini sağlar. Bu cisme sonlu asal cisim denir ve $\mathbb{F}_p$ ile ifade edilir [62].

3.2. Eliptik Eğriler

K bir cisim olsun. K cismi üzerindeki E eliptik eğrisi, aşağıdaki eşitlik ile tanımlanır.

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (3.1)$$

Weierstrass denklemi olarak adlandırılan bu denklemde $a_1,a_2,a_3,a_4,a_6 \in K$ olarak verilmektedir. K cismi rasyonel sayılar kümesi $\mathbb{Q}$, gerçel sayılar kümesi $\mathbb{R}$, kompleks sayılar kümesi $\mathbb{C}$ veya sonlu asal cisim $\mathbb{F}_p$ olabilir. Denklem katsayılarının K cisminin elemanı olmasından dolayı E eliptik eğrisinin K cismi üzerinde tanımlandığı ifade edilir. E eliptik eğrisinin K cismi üzerinde tanımlandığı E/K şeklinde gösterilir. K cismi üzerinde tanımlanan Weierstrass denklemi, kabul edilebilir katsayı değişiklikleri uygulanarak önemli ölçüde basitleştirilebilir [62]. K cisminin karakteristik değerini $C(K)$

notasyonu ile ifade ederek farklı karakteristik $C(K)$ değerleri için eliptik eğri denklemlerini aşağıda inceleyelim.

- K cisminin karakteristik değerinin $C(K)=2$ olması durumunda a_1 değişkeninin almış olduğu değere göre eliptik eğri denklemi değişmektedir. Eğer $a_1 \neq 0$ ise aşağıdaki dönüşümle,

$$(x,y) \rightarrow \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 + a_3^2}{a_1^3} \right) \tag{3.2}$$

Eş. 3.3'teki eliptik eğri eşitliği elde edilir.

$$E: y^2 + xy = x^3 + ax^2 + b \quad . \tag{3.3}$$

Eğer $a_1 = 0$ ise aşağıdaki dönüşümle,

$$(x,y) \rightarrow (x + a_2, y) \tag{3.4}$$

Eş. 3.5'teki eliptik eğri eşitliği elde edilir.

$$E: y^2 + cy = x^3 + ax + b \tag{3.5}$$

- K cisminin karakteristik değerinin $C(K)=3$ olması durumunda, eğer $a_1^2 \neq -a_2$ ise aşağıdaki dönüşümle,

$$(x,y) \rightarrow \left(x + \frac{a_4 - a_1 a_3}{{a_1}^2 + a_2}, y + a_1 x + a_1 \frac{a_4 - a_1 a_3}{{a_1}^2 + a_2} + a_3 \right) \tag{3.6}$$

Eş. 3.7'deki eliptik eğri eşitliği elde edilir.

$$E: y^2 = x^3 + ax^2 + b \tag{3.7}$$

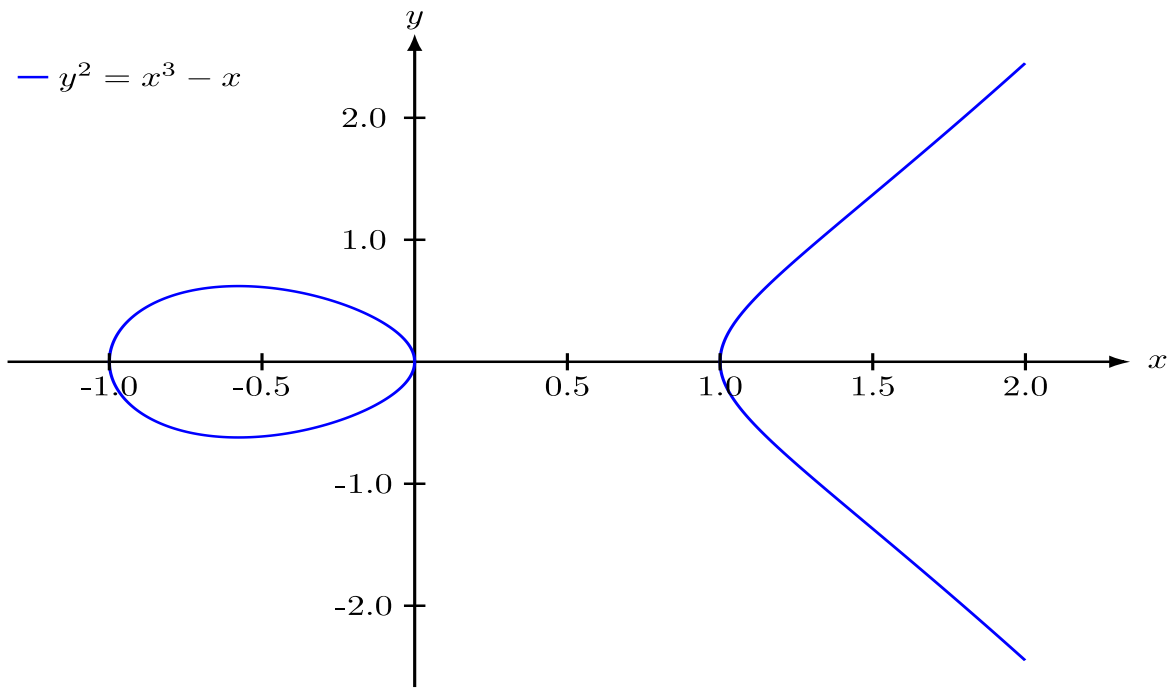
- K cisminin karakteristik değerinin $C(K) \neq 2,3$ olması durumunda ise aşağıdaki dönüşümle,

$$(x, y) \rightarrow \left(\frac{x - 3a_1^{\,2} - 12a_2}{36}, \frac{y - 3a_1 x}{216} - \frac{a_1^3 + 4a_1 a_2 - 12a_3}{24} \right) \tag{3.8}$$

Eş. 3.9'daki eliptik eğri eşitliği elde edilir.

$$E : y^2 = x^3 + ax + b \tag{3.9}$$

Eliptik eğri denkleminde yer alan değişkenler $x, y, a, b \in K$ cisminin elemanlarıdır. Seçilen a, b değerlerine göre E eliptik eğrisinin denklemi belirlenir. Daha sonra eşitliği sağlayan x, y değerlerine göre K cismi üzerinde yer alan E eliptik eğrisinin noktaları hesaplanır. Bu noktaların belirlenmesinde bazı kıstaslar olup bunlar eliptik eğrilerin kriptografide kullanılması için gerekli şartlardandır. K cismi gerçel sayılar kümesi ise E eliptik eğrisinin belirlenmesinde $x, y, a, b \in \mathbb{R}$ olarak seçilir. E eliptik eğrisi kendi eşitliğini sağlayan (x, y) noktaları ile birlikte $\mathcal{O}$ sonsuzluk noktasına sahiptir [55]. Şekil 3.1'de $a = -1$ ve $b = 0$ seçilerek elde edilen eliptik eğrisi gösterilmiştir.



Şekil 3.1. $y^2 = x^3 - x$ eliptik eğrisi

Kriptografi üzerine uygulamalarda E eliptik eğrisinin denkleminde a,b değişkenleri belirlenirken denklemin katlı köke sahip olmaması gerekli şart olup bunun için diskriminantının $\Delta \neq 0$ şartını sağlayıp sağlamadığı kontrol edilir. Eş. 3.9'daki Weierstrass denkleminde diskriminant $\Delta = -16(4a^3 + 27b^2)$ eşitliği ile belirlenir. Bundan dolayı eliptik eğri seçiminde a,b değişkenleri için $4a^3 + 27b^2 \neq 0$ eşitliği kontrol edilmelidir. Eğer bu şart sağlanmıyorsa a,b değişkenleri değiştirilerek yeniden seçilmelidir. Bu eşitliğin kontrol edilmesinin önemi eliptik eğrinin pürüzsüz olmasını (şifreleme ve şifre çözme fonksiyonlarının bire bir olması) sağlamak istememizden dolayıdır. Bundan dolayı gerçel sayılar üzerinde yapılan hesaplamalar hız olarak yavaştır ve yuvarlamalardan dolayı hatalı sonuçlar verebilmektedir. Bu sebeple kriptografide gerçel sayılar üzerinde tanımlı cisimlerden ziyade sonlu cisimler üzerinde işlemler gerçekleştirilir [55].

3.3. Sonlu Cisimler Üzerinde Tanımlı Eliptik Eğriler

Sonlu cisimler üzerinde tanımlı eliptik eğriler, sağlamış oldukları pratik çözümden dolayı kriptografi uygulamalarında yaygın olarak kullanılmaktadır. Sonlu cisimler üzerinde sonlu asal cisim $\mathbb{F}_p$, sonlu ikili cisim $\mathbb{F}_{2^m}$ ve sonlu cisim genişlemesi $\mathbb{F}_{p^m}$ tanımlanabilir [62].

Sonlu asal cisim $\mathbb{F}_p$ ile gösterilir ve burada p yeterince büyük bir asal sayıdır. Sonlu asal cisim üzerindeki E eliptik eğrisi $E(\mathbb{F}_p)$ veya $E/\mathbb{F}_p$ notasyonu ile de ifade edilebilir. $\mathbb{F}_p$, p mertebesinde ve $\{0,1,2,3.....p-1\}$ elemanlarından oluşan sonlu asal cisim olup p asal sayısı $\mathbb{F}_p$ cisminin modülüdür (mod p). Sonlu asal cisim üzerinde gerçekleştirilecek herhangi bir matematiksel işlemin sonucunda $\{0,1,2,3.....p-1\}$ kümesinin elemanlarından biri elde edilir. Benzer şekilde $\mathbb{F}_p$ sonlu asal cismi üzerinde tanımlı eliptik eğrilerde sınırlı sayıda (x, y) noktası mevcuttur. Sonlu asal cisim $\mathbb{F}_p$ üzerinde E eliptik eğrisi Eş. 3.10'daki gibi gösterilir.

$$E/\mathbb{F}_P : y^2 = x^3 + ax + b \pmod{p} \tag{3.10}$$

Eliptik eğri üzerinde noktaların bulunmasını bir örnekle açıklayalım. Sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlı E eliptik eğrisini seçelim. p asal sayısı, a ve b değerleri sırasıyla (41, 3,

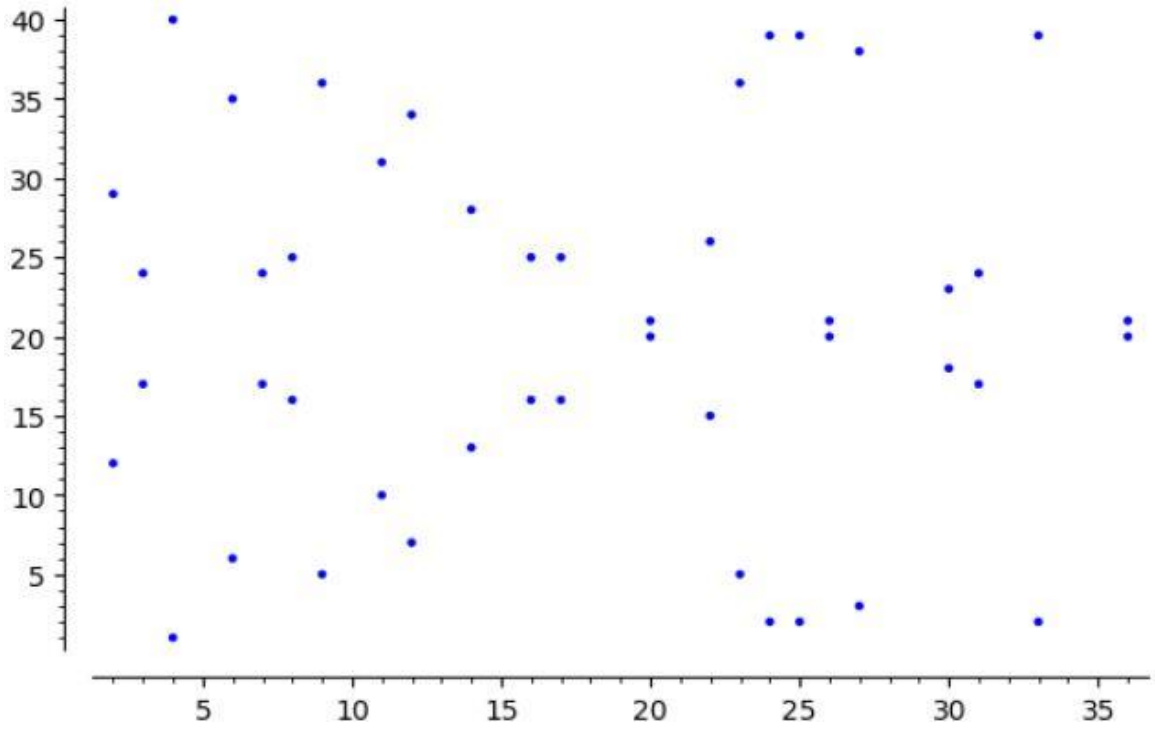
7) olarak seçilsin. Seçilen değerler $4(3)^3 + 27(7)^2 \neq 0$ eşitliğini sağladığından eliptik eğrinin kriptografide kullanılması uygundur.

$$E / \mathbb{F}_{41} : y^2 = x^3 + 3x + 7 \pmod{41} \tag{3.11}$$

Eş. 3.11'de x değerleri yerine konulup y değerleri elde edilir. $x = 2$ için; $y^2 = (2)^3 + 3.2 + 7 = 21 \pmod{41}$ eşitliği düzenlenerek $y^2 = 21 + 41.k$ elde edilir. Buradan $k = 3$ için $y = 12$ bulunur. Elde edilen nokta (2,12) noktasıdır. Benzer işlem diğer x değerleri için tekrarlanarak tüm (x, y) noktaları bulunur. Sonuç olarak söz konusu $\mathbb{F}_{41}$ sonlu asal cismi üzerinde tanımlı E eliptik eğrisi için toplam nokta sayısı 47 olup bunlardan bir tanesi $\mathcal{O}$ sonsuzluk noktasıdır. $\mathbb{F}_{41}$ sonlu asal cismi üzerinde tanımlı E eliptik eğrisinin noktaları Çizelge 3.1'de verilmiştir. Şekil 3.2'de ise eliptik eğri noktaları koordinat sistemi üzerinde gösterilmiştir.

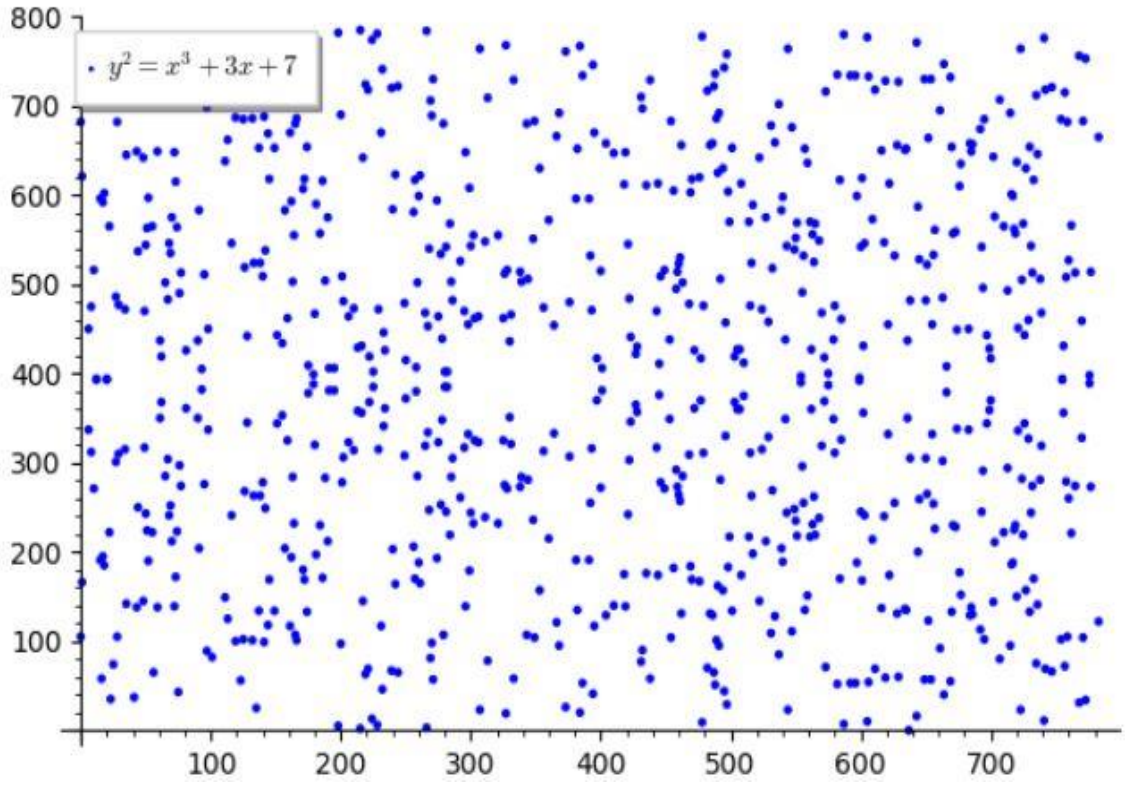
Çizelge 3.1. $E / \mathbb{F}_{41} : y^2 = x^3 + 3x + 7$ eliptik eğrisinin noktaları

(2,12)	(7,17)	(12,7)	(20,20)	(25,2)	(31,17)
(2,29)	(7,24)	(12,34)	(20,21)	(25,39)	(31,24)
(3,17)	(8,16)	(14,13)	(22,15)	(26,20)	(33,2)
(3,24)	(8,25)	(14,28)	(22,26)	(26,21)	(33,39)
(4,1)	(9,5)	(16,16)	(23,5)	(27,3)	(36,20)
(4,40)	(9,36)	(16,25)	(23,36)	(27,38)	(36,21)
(6,6)	(11,10)	(17,16)	(24,2)	(30,18)	$\mathcal{O}$
(6,35)	(11,31)	(17,25)	(24,39)	(30,23)	-



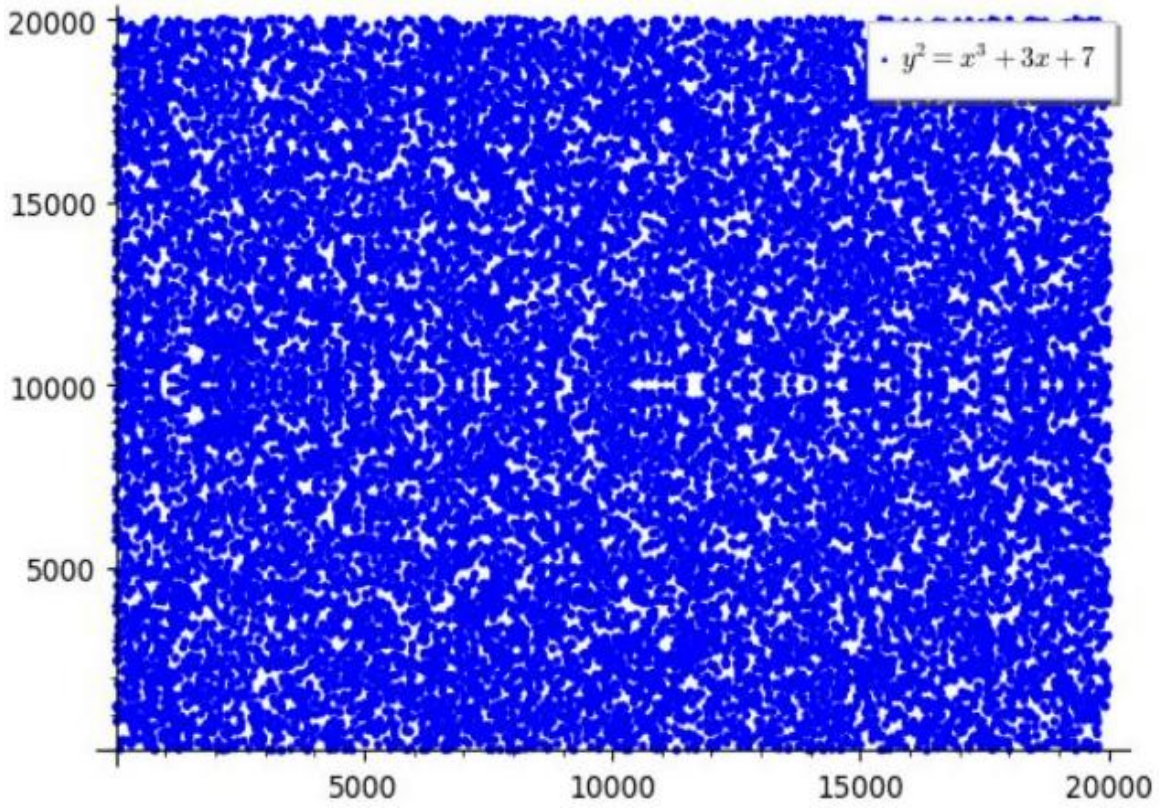
Şekil 3.2. $E / \mathbb{F}_{41} : y^2 = x^3 + 3x + 7$ eliptik eğrisinin noktalarının gösterimi

Örnek uygulamada kullanılan *p* asal sayısı 41 olup bu değer oldukça küçük bir asal sayıdır.



Şekil 3.3. $E / \mathbb{F}_{787}$ eliptik eğrisi üzerindeki noktaların gösterimi

Kriptografik sistemlerin tasarımında kullanılmak üzere seçilecek *p* asal sayısı yeterince büyük olmalıdır, bu durum sistemin güvenliği açısından oldukça önemlidir. *p* değerinin 787 gibi bir asal sayı seçilmesi durumunda $E / \mathbb{F}_{787}$ eliptik eğrisi üzerindeki noktalar Şekil 3.3'te gösterildiği şekilde elde edilir. $E / \mathbb{F}_{787}$ eliptik eğrisi üzerinde 782 nokta bulunmaktadır. Dikkat edilecek olursa bir önceki *p* asal sayısı 41 iken oldukça az sayıda nokta mevcut iken yeni seçilen *p* asal sayısı 787 için eliptik eğri üzerindeki nokta sayısı artmaktadır.



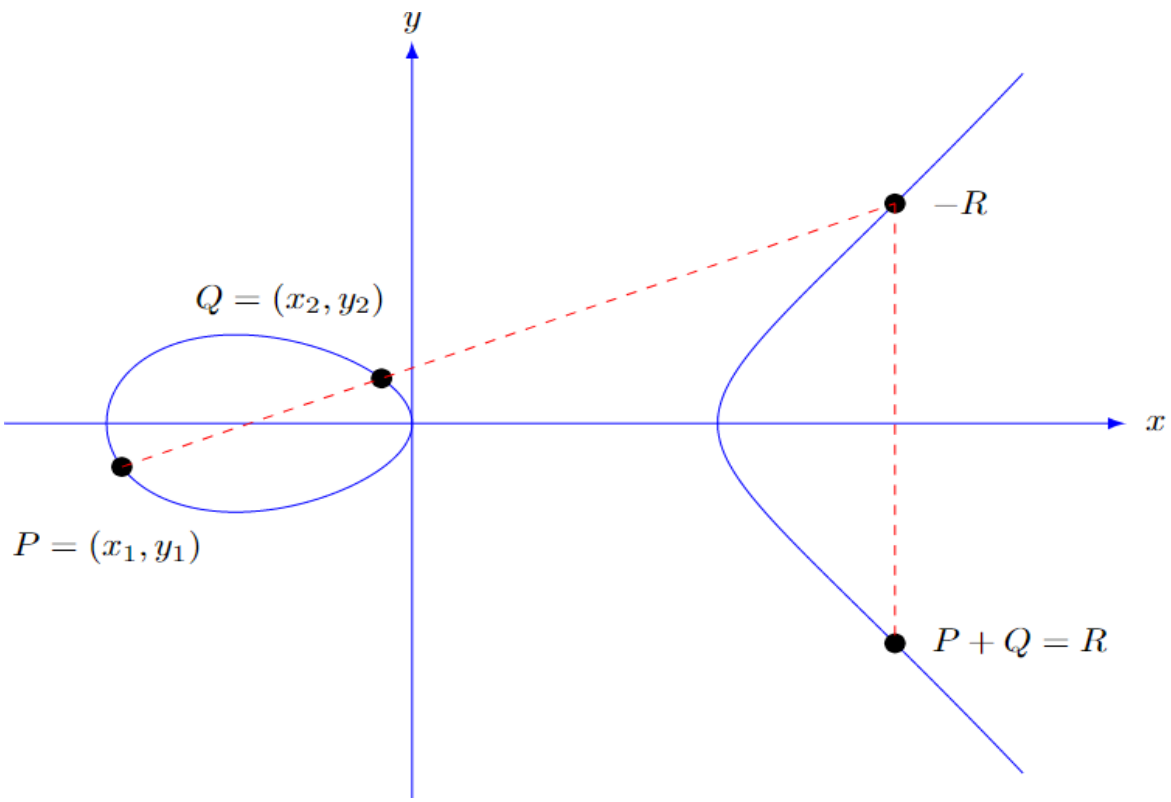
Şekil 3.4. $E / \mathbb{F}_{20011}$ eliptik eğrisi üzerindeki noktaların gösterimi

p asal sayısının daha büyük bir asal sayı olarak seçilmesi sonucunda eliptik eğri üzerindeki nokta sayısı artmakla birlikte bu değer halen bir kriptografik sisteminin tasarımında kullanılmak için yeterli değildir. Bir sonraki adım olarak *p* asal sayısının 20011 olarak seçildiğini varsayalım. Bu durumda $E / \mathbb{F}_{20011}$ eliptik eğrisi üzerindeki noktalar Şekil 3.4.'te gösterilmiştir. $E / \mathbb{F}_{20011}$ eliptik eğrisi üzerinde 19978 nokta bulunmaktadır. Dikkat edilecek olursa *p* değerinin önceki değerden çok daha büyük bir asal sayı seçilmesi ile eliptik eğri üzerinde tanımlanan noktalar artmakta ve noktalar arasında bağlantı kurulması oldukça zorlaşmaktadır. Ancak *p* değeri olarak seçilen 20011 asal sayısının da güvenli bir

kriptografik sisteminin tasarımında da oldukça yetersiz bir değer olduğunu burada belirtmekte fayda var.

3.4. Sonlu Asal Cisimler Üzerinde Matematiksel İşlemler

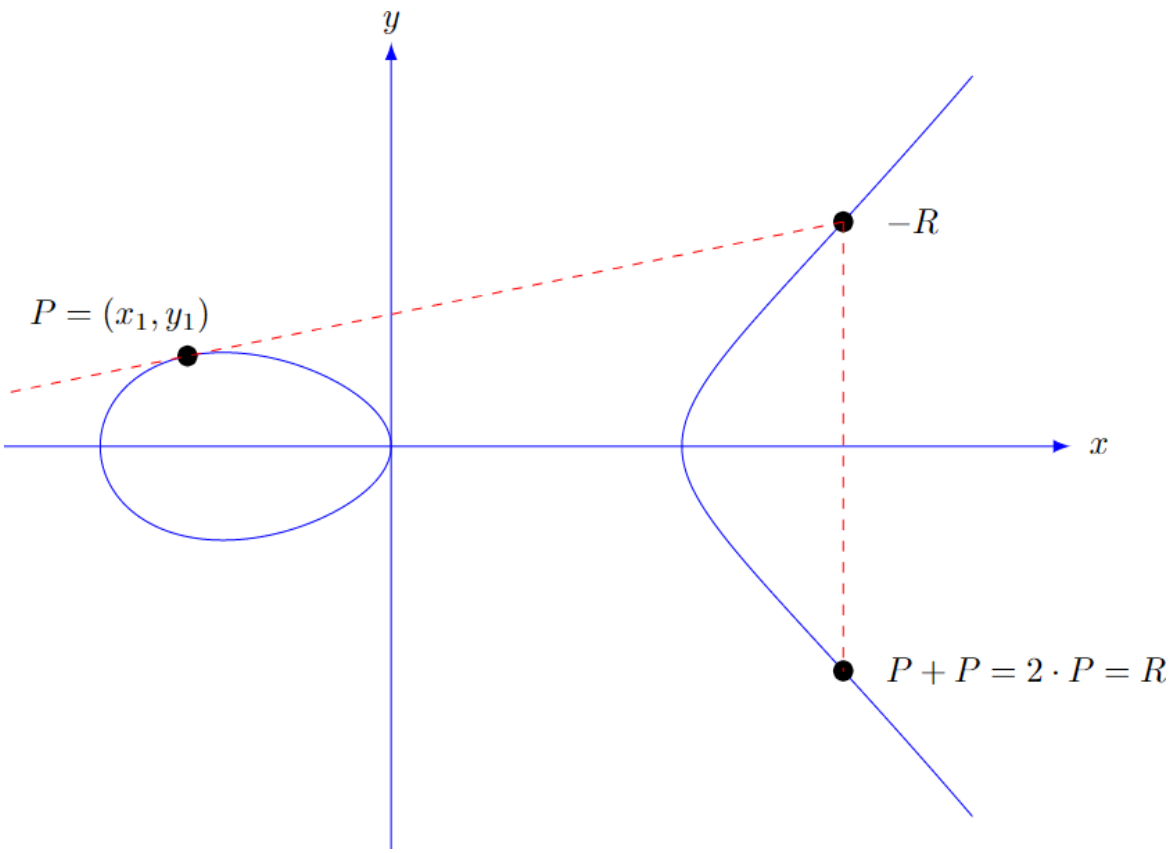
Bir sonlu asal cisim üzerinde tanımlı E eliptik eğrisinin noktaları arasındaki bağlantı temel olarak toplama işlemi ile kurulabilir. Sonlu asal cisim üzerinde tanımlı eliptik eğrinin iki noktası arasındaki toplama geometrik olarak açıklanabilir. Bu durum, eliptik eğri üzerindeki noktaların toplama işlemine göre değişmeli grup oluşturması olarak da açıklanmaktadır. Değişmeli grup diğer bir ifadeyle Abel grup eliptik eğri kriptografi sistemlerinin oluşturulmasında kullanılan bir cebirsel yapıdır. Bu grup aynı zamanda sıfır eleman ve ters eleman özelliklerine de sahiptir. Eliptik eğrinin noktaları arasında nokta toplama, nokta çiftleme ve bu işlemler temel alınarak çıkarma ve skaler çarpma işlemleri gerçekleştirilmektedir.



Şekil 3.5. Eliptik eğri üzerinde toplama işlemi ($P+Q=R$)

E eliptik eğrisi sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlı ve $P(x_1, y_1)$, $Q(x_2, y_2)$ şeklinde iki farklı noktaya sahip olsun. Ayrıca Q noktasının x eksenine göre simetriği olan $-Q$ için

$P \neq -Q$ şartı sağlanmalıdır. P ve Q noktalarını birleştirecek şekilde çizilen çizgi, eliptik eğriyi üçüncü bir nokta olan $-R$ noktasında keser ve bu $-R$ noktasının x eksenine göre simetriğinin alınmasıyla P ve Q noktalarının toplamı olan R noktası bulunur. Sonlu asal cisim $\mathbb{F}_p$ üzerinde geometrik olarak gerçekleştirilen $P+Q=R$ toplama işlemi Şekil 3.5'te gösterilmiştir.



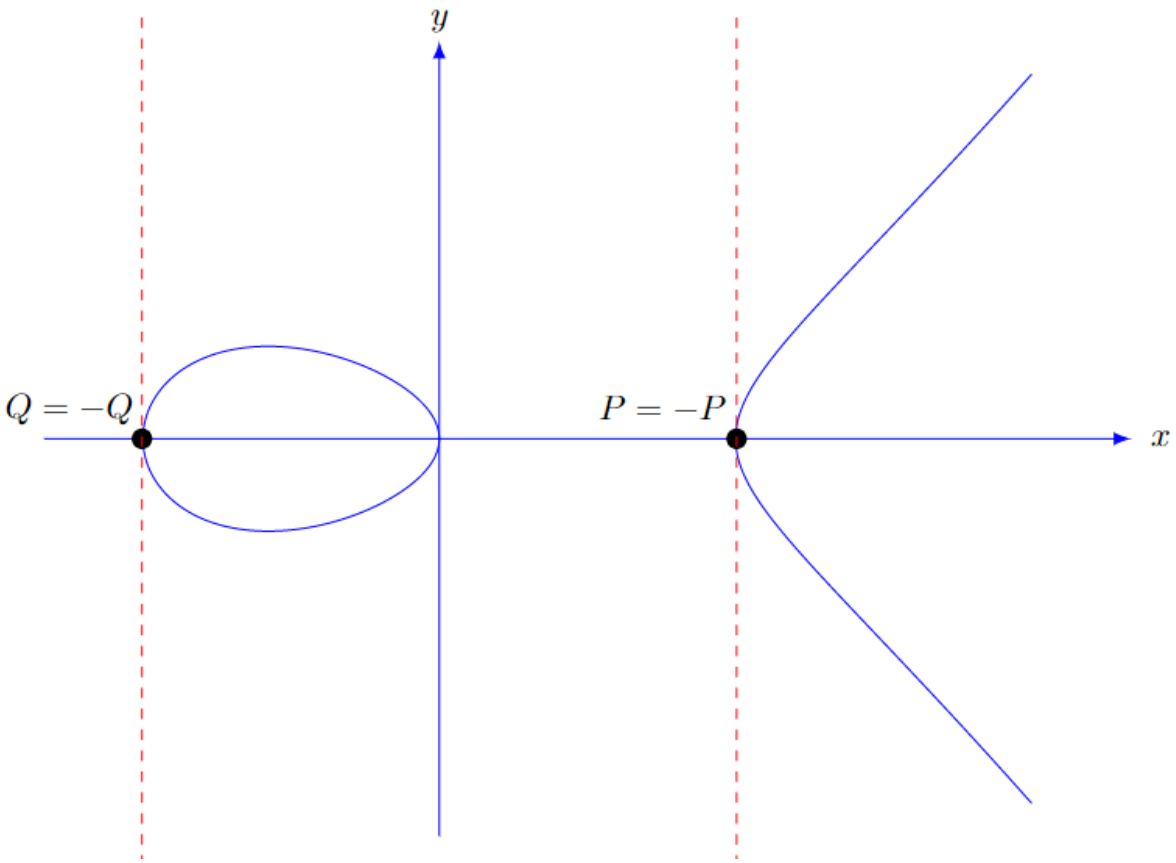
Şekil 3.6. Eliptik eğri üzerinde nokta çiftleme ($P+P=2\cdot P=R$)

E eliptik eğrisi üzerindeki P noktasının x eksenine göre simetriği olan $-P$ noktası ile toplanması durumunda ise çizilen çizgi, eliptik eğriyi üçüncü bir noktada kesmez ve toplama işlemi sonsuzluk noktasına gider. Bu işlemin sonucu $P+(-P)=\mathcal{O}$ sonsuzluk noktasını verir. Aynı zamanda $P+\mathcal{O}=P$ şeklinde ifade edilebilir.

P noktasının ikinci bir P noktası ile toplanması işlemi nokta çiftleme işlemi olarak adlandırılmaktadır. $P \neq -P$ olmak üzere P noktasından teğet olarak çizilen çizgi eliptik eğriyi $-R$ noktasında keser ve $-R$ noktasının x eksenine göre simetriğinin alınmasıyla P

noktasının çiftlemesi olan R noktası bulunur. Sonlu cisim $\mathbb{F}_p$ üzerinde geometrik olarak gerçekleştirilen $P+P=2\cdot P=R$ çiftleme işlemi Şekil 3.6'da gösterilmiştir.

Nokta çiftleme işleminde $P=-P$ olması yani $y=0$ olması durumunda P noktasına çizilen teğet çizgi eliptik eğriyi başka bir noktada kesmez. Bundan dolayı bu işlem sonucunda $P+P=2\cdot P=\mathcal{O}$ sonsuzluk noktası elde edilir. Benzer durumda $Q=-Q$ olması durumunda aynı şekilde $Q+Q=2\cdot Q=\mathcal{O}$ sonsuzluk noktası elde edilir.



Şekil 3. 7. $P=-P$ olması durumunda nokta çiftleme ($P+P=2\cdot P=\mathcal{O}$)

Sonlu asal cisim $\mathbb{F}_p$ üzerindeki E eliptik eğrisinin noktaları kullanılarak geometrik olarak gerçekleştirilen nokta toplama ve nokta çiftleme işlemleri cebirsel yöntemlerle de gerçekleştirilebilir.

Nokta toplama

Eliptik eğri üzerinde $P(x_1, y_1)$ ve $Q(x_2, y_2)$ iki farklı nokta olsun ve $P \neq -Q$ şartı sağlansın. İki noktanın toplamında $P(x_1, y_1) + Q(x_2, y_2) = R(x_3, y_3)$ işlemi ile elde edilen üçüncü bir nokta olan $R(x_3, y_3)$ noktası için x_3, y_3 Eş. 3.12 ve Eş. 3.13'teki gibi hesaplanır [62].

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2 \tag{3.12}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 (x_1 - x_3) - y_1 \tag{3.13}$$

Nokta çiftleme

Eliptik eğri üzerinde $P \neq -P$ olmak üzere seçilen $P(x_1, y_1)$ noktasının kendisi ile toplanarak $P(x_1, y_1) + P(x_1, y_1) = 2 \cdot P = R(x_3, y_3)$ işlemi sonucu elde edilen x_3, y_3 değerleri Eş. 3.14 ve Eş 3.15'teki gibi hesaplanır [62].

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1 \tag{3.14}$$

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)(x_1 - x_3) - y_1 \tag{3.15}$$

Nokta çıkarma

Eliptik eğri üzerindeki çıkarma işleme toplama işlemi ile aynı cebirsel yöntemle gerçekleştirilir. $P(x_1, y_1)$ ve $Q(x_2, y_2)$ eliptik eğri üzerinde iki farklı nokta olsun. $Q(x_2, y_2)$ noktasının negatifi y_2 noktasının *x* eksenine göre simetriği alınarak elde edilen

$Q(x_2, -y_2)$ noktasıdır. Nokta çıkarma işlemi Eş.3.16'daki gibi düzenlenerek toplama işlemi ile gerçekleştirilir.

$$P(x_1, y_1) - Q(x_2, y_2) = P(x_1, y_1) + Q(x_2, -y_2) = R(x_3, y_3) \tag{3.16}$$

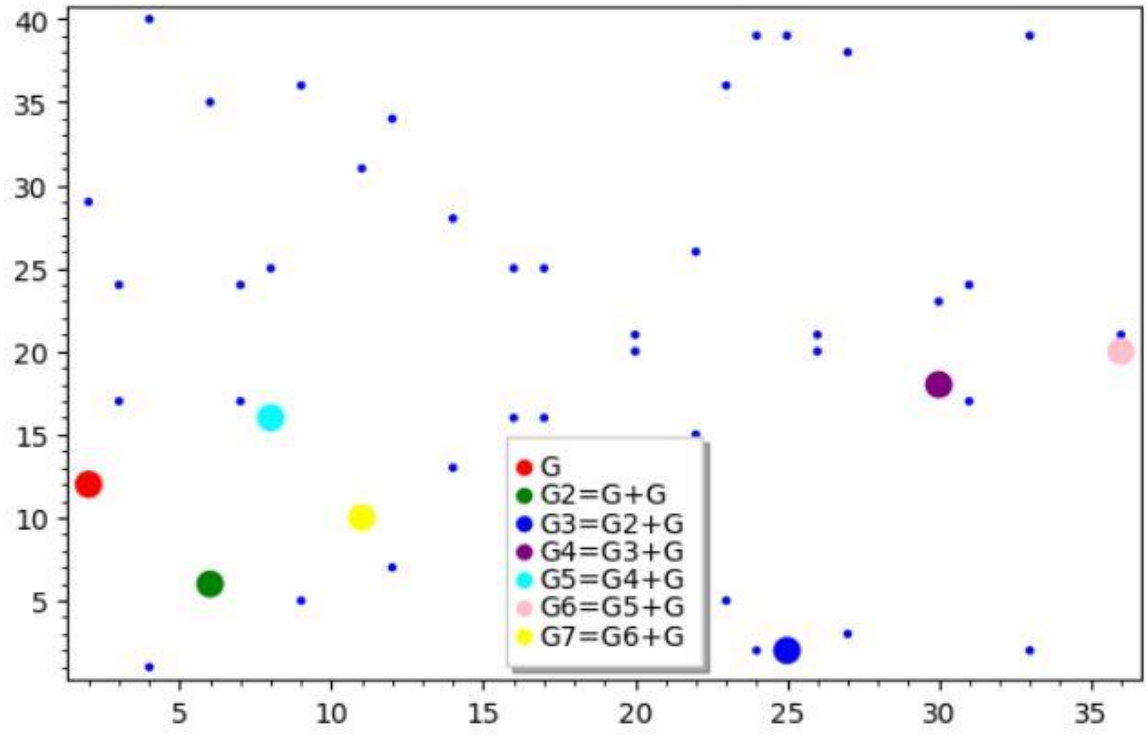
Skaler Çarpım

Eliptik eğri üzerinde bulunan $P(x_1, y_1)$ noktasının *k* ile çarpılması ile gerçekleştirilen $k \cdot P$ işlemi skaler çarpım olarak adlandırılır. Bu işlemi gerçekleştirmek için Eş. 3.17'deki gibi *P* noktası *k* defa kendisi ile toplanır [72].

$$k \cdot P = \underbrace{P + P + P .... + P}_{k} \tag{3.17}$$

Skaler çarpma işlemini gerçekleştirmek için nokta toplama ve nokta çiftleme işlemleri kullanılır. Örnek olarak $11 \cdot P$ skaler çarpma işlemi Eş. 3.18'deki gibi gerçekleştirilir.

$$11 \cdot P = 2(2(P + P) + P) + P \tag{3.18}$$



Şekil 3.8. Eliptik eğri üzerindeki noktalar arasında bağlantı

Eliptik eğri üzerindeki noktalar arasında bağlantı Şekil 3.8'de gösterilmiştir. *E* eliptik eğrisi üzerinde *G* başlangıç noktası seçilerek nokta çiftleme ve nokta toplama işlemleri ile G2, G3, G4, G5, G6, G7 noktaları hesaplanarak elde edilen değerlerin temsil ettiği noktalar şekilde gösterilmiştir.

3.5. Eliptik Eğri Kriptografisi

Eliptik eğrilerin kriptografide kullanılması fikri 1980'li yılların ortalarına rastlamaktadır. Neal Koblitz [73] ve Victor Miller [74] birbirlerinden bağımsız olarak eliptik eğrilerin kriptografide kullanılmasını önermişlerdir. Eliptik eğri kriptografisi (Elliptic Curve Cryptography; ECC) temel olarak açık anahtarlı kriptografiye dayanmaktadır. Basitliği ve yüksek performansı nedeniyle sonraki yıllarda birçok kriptografi uygulamasında eliptik eğrilerin kullanımı yaygınlaşmıştır. Eliptik eğriler, Diffie-Hellman anahtar değişim şeması [75] ve ElGamal [53, 77] imza şemalarını uygulamak için kullanılabilir. Eliptik eğri kriptografisi mevcut diğer açık anahtar kriptografi yöntemlerinin sağlamış olduğu güvenlik seviyesini daha kısa anahtar uzunlukları ile sağlayabilmektedir. Örneğin; ECC'deki 160 bit anahtar boyutu, 1024 bit anahtar boyutuna sahip RSA ile aynı seviyede güvenliği sağlar. Böylece daha düşük bant genişliği, daha az bellek ihtiyacı ve daha düşük güç tüketimi sağlanmış olunur. Bu özellikler nesnelerin interneti, akıllı kartlar gibi donanım olarak kısıtlı kaynağa sahip nesnelerin kullanıldığı uygulamalarda oldukça önemli parametrelerdir. Eliptik eğri kriptografisinin güvenliği eliptik eğri ayrık logaritma problemine (Elliptic Curve Discrete Logarithm Problem; ECDLP) dayanmaktadır [62].

3.5.1. Tanım

E eliptik eğrisi sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlı olsun. Bu eliptik eğri üzerinde P, Q şeklinde iki farklı nokta alalım. $k \in \mathbb{Z}_p^*$ olmak üzere;

$$Q = k \cdot P \tag{3.19}$$

eşitliğinde k değerini bulma işlemi eliptik eğri ayrık logaritma problemi (ECDLP) olarak adlandırılır [62].

Eliptik eğri ayrık logaritma problemi büyük *p* asal sayıları için hesaplanması oldukça zor bir cebirsel problemdir.

Grup mertebesi

E eliptik eğrisi üzerinde bulunan noktaların sayısı olan *n* değeri eliptik eğrinin mertebesini verir ve $\#E(F_p)$ ile gösterilir. *E* eliptik eğrisinin mertebesi kriptografi sistemlerin tasarımında oldukça önemli olup özellikle *n* değerinin asal olmasına ve $E/\mathbb{F}_P$'nin devirli grup olmasına dikkat edilmelidir. Grup mertebesinin alt sınırının ve üst sınırının belirlenmesinde Hasse teoremi kullanılmaktadır.

Hasse Teoremi

Sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlanan *E* eliptik eğrisinin mertebesi olan $\#E(\mathbb{F}_p)=n$ Eş. 3.20'deki gibi Hasse aralığı olarak adlandırılan aralıkta ifade edilebilir [72].

$$p+1-2\sqrt{p} \le n \le p+1+2\sqrt{p} \tag{3.20}$$

Sistem Parametreleri

Eliptik eğri temelli kriptografi sistemlerinin tasarımında öncelikle temel parametreler belirlenerek üzerinde anlaşılır. Sistem parametreleri olarak da adlandırılan bu temel parametreler kriptografi sisteminin tasarımında kullanılmaktadır. Eliptik eğri temelli bir kriptografi sisteminde sistem parametreleri, sonlu asal cisim $\mathbb{F}_p$, başlangıç noktası $G \in E/\mathbb{F}_p$ ve eliptik eğrinin mertebesi *n* olarak belirlenir. Parametreler, ECDLP bilinen tüm saldırılara karşı dirençli olacak şekilde seçilmelidir. Güvenlik veya uygulama nedenleriyle sistem parametrelerine başka parametreler de eklenebilir.

3.6. Sonlu Cisimler Üzerinde Tavsiye Edilen Eliptik Eğriler

ABD Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) sonlu cisimler üzerinde asal cisimler ve ikili cisimler için özel rasgele eliptik eğriler önermiştir. NIST tarafından

yayınlanan FIPS 186-2 standardında sonlu asal cisim üzerinde P-192, P-224, P-256, P-384, P-521 eliptik eğrileri tanımlanmıştır [78]. Tavsiye edilen eliptik eğriler farklı parametrelere göre seçilmiş olup sonlu asal cismin mertebesine göre Çizelge 3.2'de gösterilmektedir.

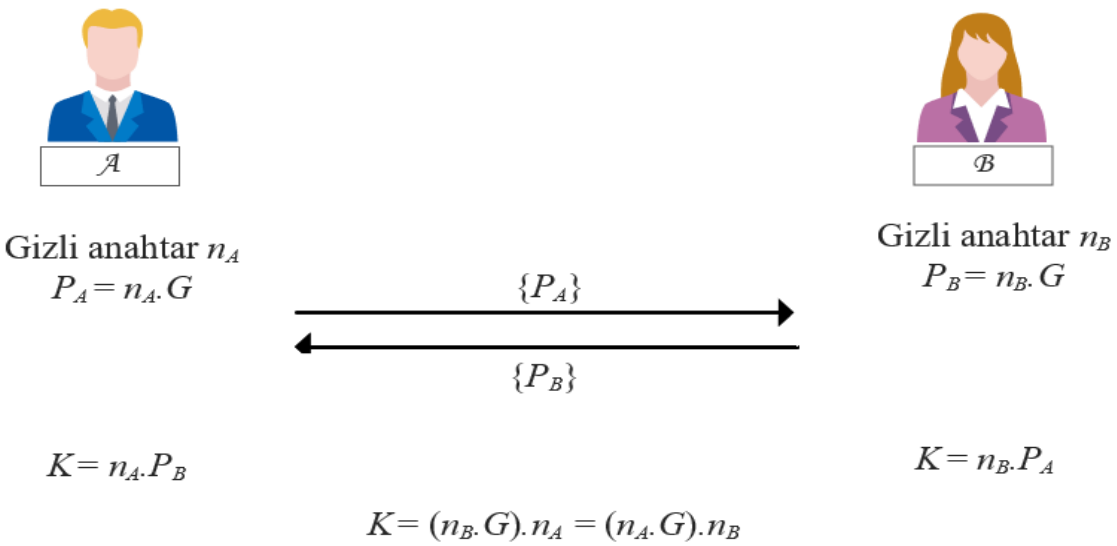
Çizelge 3.2. NIST eliptik eğrileri

Eliptik Eğri	Asal cismin mertebesi (p)
P-192	$2^{192}-2^{64}-1$
P-224	$2^{224}-2^{96}+1$
P-256	$2^{256}-2^{224}+2^{192}+2^{96}-1$
P-384	$2^{384}-2^{128}-2^{96}+2^{32}-1$
P-521	$2^{521}-1$

3.7. Eliptik Eğri Diffie-Hellman Anahtar Değişim Protokolü

Anahtar değişim protokolleri açık anahtarlı kriptografi sistemleri için oldukça önemli bir konudur. Anahtar dağıtım probleminin ilk çözümü Diffie-Hellman [60] tarafından önerilmiş ve sonrasında açık anahtarlı kriptografi sistemlerinin türlerinin ve kullanımının yaygınlaştığı görülmektedir. Açık anahtarlı kriptografiye dayalı eliptik eğri tabanlı kriptografide kullanılan anahtar değişimi, Diffie-Hellman anahtar değişimine dayanmaktadır. Açık anahtarlı kriptografi sistemlerinde, açık anahtar ve gizli anahtar olmak üzere iki anahtar vardır. A ve B taraflarının iletişim kurmak istediklerinde, açık ve gizli anahtarlara sahip olmaları gerekir.

Eliptik eğri temelli kriptografi sistemlerinde kullanılan Diffie-Hellman anahtar değişim protokolünü tanımlamak için sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlı E eliptik eğrisini ve bu eğride bulunan G başlangıç noktasını seçelim. A tarafı; gizli anahtar olarak n_A değerini seçer ve $P_A = n_A \cdot G$ skaler çarpımı ile P_A açık anahtarını hesaplar. Aynı şekilde, B tarafı, gizli anahtar olarak n_B değerini seçer ve $P_B = n_B \cdot G$ skaler çarpımı ile P_B açık anahtarını hesaplar. Bu hesaplamalar sonucunda; A ve B tarafları açık anahtarlarını birbirlerine gönderirler. Her iki taraf karşı taraftan gönderilen açık anahtarı kullanarak kendi özel anahtarı ile skaler çarpım işlemini gerçekleştirerek K ortak anahtarını hesaplar. Eliptik eğri Diffie-Hellman anahtar değişim protokolü Şekil 3.9'da gösterilmiştir.



Şekil 3.9. Eliptik eğri Diffie-Hellman anahtar değişim protokolü

Diffie-Hellman anahtar değişim protokolünün güvenliği ECDLP'ye dayanmaktadır. Şöyle ki P_A, P_B açık anahtarları ve G başlangıç noktası herkes tarafından bilinse bile n_A, n_B gizli anahtarlarını hesaplanması oldukça zor bir problemdir.

3.7.1. Tanım

E eliptik eğrisi sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlı olsun. E eliptik eğrisi üzerindeki bir nokta $G \in E/\mathbb{F}_p$ ve $a, b \in \mathbb{Z}_p^*$ olmak üzere; bilinen G, $a \cdot G$ ve değerlerinden yola çıkılarak $ab \cdot G$ değerinin hesaplanması oldukça zor olup bu problem Diffie-Hellman problemi olarak ifade edilmektedir [62, 70].

3.8. Eliptik Eğri Tabanlı Kriptografi Algoritması

Eliptik eğriler kullanılarak sonlu cisimler üzerinde şifreleme algoritmaları tasarlanmaktadır. Yaygın olarak eliptik eğrilerle şifreleme algoritması olarak Menezes-Vanstone [77] ve ElGamal şifreleme [62, 80] algoritmaları kullanılsa da aslında literatürde birçok farklı şifreleme algoritması mevcuttur. Geçmişten günümüze farklı uygulamalarda ortaya çıkan kriptografi istekleri şifreleme algoritmalarının da zaman içerisinde değişmesine sebebiyet vermektedir. ElGamal Eliptik Eğri Kriptografisi, ECDLP'ye dayalı ElGamal şifreleme şemalarının açık bir anahtar kriptografi analoğudur [62, 80]. Aşağıda eliptik eğri tabanlı temel ElGamal şifreleme şemasının anahtar değişim, şifreleme ve şifre

çözme aşamaları verilmiştir. Algoritmanın adımlarına geçmeden önce Çizelge 3.3'te algoritmada kullanılan parametreler tanımlanmıştır.

Çizelge 3.3. Eliptik eğri tabanlı ElGamal kriptografi sisteminin parametreleri

Parametre	Tanım
P_A, P_B	A ve B taraflarının açık anahtarı
n_A, n_B	A ve B taraflarının gizli anahtarı
G	Başlangıç noktası
K	Ortak anahtar
P_M	Şifrelenmemiş düz mesaj
C_M	Şifrelenmiş mesaj

Anahtar değişim aşaması

1. Anahtar değişim protokolü Diffie-Hellman anahtar değişim yöntemini temel almaktadır. Sonlu cisim üzerinde tanımlı E eliptik eğrisinde başlangıç noktası olarak G noktası seçilir. A tarafı; gizli anahtar olarak n_A değerini seçer ve $P_A = n_A \cdot G$ skaler çarpımı ile P_A açık anahtarını hesaplar. Aynı şekilde B tarafı, gizli anahtar olarak n_B değerini seçer ve $P_B = n_B \cdot G$ skaler çarpımı ile P_B açık anahtarını hesaplar. A ve B tarafları şifreli mesajın gönderimi sırasında açık anahtarlarını da birbirlerine gönderirler.
2. A ve B tarafları mesajın şifrelenmesi ve şifresinin çözülmesi sırasında $K = n_A \cdot P_B$ ve $K = n_B \cdot P_A$. işlemleri ile K ortak anahtarını hesaplarlar.

Şifreleme aşaması

1. Eliptik eğri üzerinde bulunan bir noktayı temsil eden P_M noktası düz mesajımıza karşılık gelen bir nokta olsun. A tarafı P_M mesajını şifreleyerek B tarafına göndermek istediğinde şifreli C_M mesajını $C_M = (P_A, P_M + n_A \cdot P_B)$ şeklinde hesaplar ve B tarafına gönderir.

Şifre çözme aşaması

1. C_M şifreli mesajını alan B tarafı gizli anahtarı n_B ve A'nın açık anahtarı P_A'yı kullanarak $P_M + n_A \cdot P_B - n_B \cdot P_A = P_M$ hesaplaması ile P_M düz mesajını elde eder. Burada dikkat edilirse $n_A \cdot P_B = n_B \cdot P_A = K$ ile her iki taraf birbirlerinin gizli anahtarını bilmeden ortak anahtarı hesaplayabilirler.

Eliptik eğri tabanlı ElGamal şifreleme algoritmasının adımları yukarıda sunulmuştur ancak bu algoritma ile farklı dillerdeki karakterleri şifrelemek mümkün değildir. Sadece bazı alfabe karakterleri bu yöntemlerle şifrelenebilmektedir. Onaltılık sayı sisteminde ASCII kodlarına karşılık gelen ifadeler eliptik eğri üzerindeki bir nokta ile temsil edilebilir. Ancak ASCII karakterleri içerisinde bulunmayan farklı dillere ait karakterlerin şifrelenmesi sırasında sorun yaşanacaktır. Kullanılan ASCII değerleri 0-255 arası ondalık değerlerdir ve onaltılık sayı sistemine göre şifrelenir. Bu tez kapsamında bu sorun dikkate alınarak ASCII'den çok daha fazla karakter içeren Unicode tablosunda bulunan karakterleri de içeren eliptik eğri tabanlı bir şifreleme sistemi önerilmiştir [37]. Tüm Unicode karakterleri, alfabe ve karakter kısıtlamaları olmadan onaltılık sayı tabanındaki değerler kullanılarak şifrelenir. Önerilen yöntemde, mesajın her karakteri onaltılık Unicode değerine dönüştürülür ve ardından bu değer bir karakter içeren bloklara ayrılır. Unicode tablosunda, karakterlerin onaltılık değerleri bir ile altı basamak arasında değişir. Farklı alfabelerde bulunan karakterler ve emojiler için Unicode tablosundaki bazı karakterlerin onaltılık sayı sistemindeki karşılıkları olan değerler Çizelge 3.4'te sunulmuştur. Unicode tablosunda toplam yüz yirmi binden fazla karakter ve emoji bulunmaktadır.

Çizelge 3.4. Unicode tablosunda bulunan bazı karakterlerinin onaltılık sayı tabanındaki değerleri

Karakter	Değer	Karakter	Değer	Karakter	Değer
H	48	ا	627	e (Kiril)	435
e	65	ب	628	c	441
😎	1F60E	你	4F60	✈	2708
🌥	26C5	好	597D	🫗	1F510

Önerilen yeni şifreleme algoritması, sadece ASCII tablosundaki değerleri kullanarak değil, Unicode tablosundaki tüm değerleri kullanılarak farklı alfabe ve karakterleri şifreleyebilmektedir [37]. Önerilen şifreleme algoritması; (1) kurulum aşaması, (2) anahtar değişim aşaması, (3) şifreleme ve (4) şifre çözme aşamaları olmak üzere dört aşamadan oluşmaktadır. Çizelge 3.5'te önerilen şifreleme algoritmasında kullanılan parametreler sunulmuştur.

Çizelge 3.5. Önerilen şifreleme algoritmasında kullanılan parametreler

Algoritma Aşaması	Parametreler	Tanım
Kurulum	n, a, b, p, G	Sistem parametreleri
Anahtar değişim	P_A, P_B, n_A, n_B, K	Anahtar parametreleri
Şifreleme	$C = \{c_1 c_2 c_3 ... c_z\}$	Şifreli mesaj
Şifre çözme	$M = \{m_1 m_2 m_3 m_4 .... m_z\}$	Düz mesaj

Kurulum aşaması

1. Aralarında gizli olarak mesajlaşmak isteyen A ve B öncelikle kriptografi sisteminin parametrelerini belirlerler. Taraflar bir p asal sayısı ve p'den küçük bir a ve b tamsayısı seçerek, sonlu asal cisim $\mathbb{F}_p$ üzerinde bir E eliptik eğrisini kullanmak üzere anlaşırlar.
2. $E/\mathbb{F}_p$ eliptik eğrisi üzerindeki noktaların sayısı yani mertebesi n ve bir başlangıç noktası G belirlenir.

Anahtar değişim aşaması

1. Anahtar değişim protokolü Diffie-Hellman anahtar değişim yöntemi ile aynıdır. K ortak anahtarı hesaplanır ve ortak anahtarın $K = (x, y)$ olduğunu varsayalım.

Şifreleme aşaması

1. *M* mesajımız $\{m_1 m_2 m_3 m_4 ....m_z\}$ olmak üzere *z* karakterden oluştuğunu kabul edelim. *M*'nin her karakterini Unicode tablosunda birden altı karaktere kadar değişen onaltılık değerine dönüştürelim.
2. $\{m_1\} = [p_1, p_2, ..., p_t]$ on altı tabanında yazıldıktan sonra her bir karakter onluk tabana çevrilerek sırasıyla aşağıdaki adımlar izlenir.
3. $K = (x, y)$ ortak anahtar olmak üzere mesaj karakteri aşağıdaki gibi şifrelenir.

$$\begin{aligned} C_1 &= p_1 + x + y \\ C_2 &= p_2 + p_1 \\ &... \\ C_t &= p_t + p_{t-1} \end{aligned} \tag{3.21}$$

4. $\{m_1\}$ karakterinin şifrelenmesi $\{c_1\} = [C_1, C_2, ..., C_t]$ olarak hesaplanır. Daha sonra diğer karakterler $\{m_2 m_3 m_4 ....m_l\}$ için aynı işlemler uygulanarak $\{c_2, c_3, c_4 ...c_z\}$ şifreli karakterleri hesaplanır.
5. Son olarak *A* tarafından şifreli mesaj $C = \{c_1 c_2 c_3 ...c_z\}$, *B* tarafına gönderilir.

Şifre çözme aşaması

1. *B* tarafından alınan $C = \{c_1 c_2 c_3 ...c_z\}$ şifreli mesajı aşağıdaki adımlarla her bir karakter $\{c_1\} = [C_1, C_2, ..., C_t]$ haline dönüştürülerek $\{m_1\} = [p_1, p_2, ..., p_t]$ düz mesajı haline getirilir.

$$\begin{aligned} p_1 &= C_1 - x - y \\ p_2 &= C_2 - p_1 \\ &... \\ p_t &= C_t - p_{t-1} \end{aligned} \tag{3.22}$$

2. $\{m_1\} = [p_1, p_2, ..., p_t]$ hesaplanır. Daha sonra diğer şifreli karakterler $\{c_2, c_3, c_4 ...c_z\}$ için aynı işlemler uygulanarak $\{m_2 m_3 m_4 ....m_z\}$ düz mesaj karakterleri hesaplanır.

3. Bu şekilde *B* tarafından $M = \{m_1 m_2 m_3 m_4 .... m_z\}$ düz mesajı elde edilmiş olur.

Önerilen şifreleme algoritması ile sonlu cisimler üzerinde eliptik bir eğri kullanılarak yeni bir mesaj şifreleme algoritması tasarlanmıştır. Örnek uygulama ve detaylı bilgi için yazar ve danışman tarafından yayınlanmış olan bildiriye bakılabilir [37]. Birçok farklı alfabe, ifade ve sembol, önerilen şifreleme algoritmasında kullanılabilir. Mesaj şifreleme ve şifre çözme, İngilizce, Türkçe, Çince, Arapça ve Rusça dahil olmak üzere farklı dillerde ve karakterlerde uygulanmaktadır. Böylece önerilen yöntem ile şifreleme uygulamaları için ASCII tablosunun sınırlı karakter sorununa çözüm getirilmiştir. Bu uygulama ile matematiksel olarak çarpma ve ters alma işlemlerine ihtiyaç duymadan sadece toplama ve çıkarma ile şifreleme ve şifre çözme yapılabilmektedir. Bu sayede daha hızlı şifreleme ve şifre çözme işlemleri gerçekleştirilir [37].

3.9. Eliptik Eğri Dijital İmza Algoritması

Kendi el yazımız ile oluşturulan imzaların sayısal karşılıkları olan algoritmalar ile tasarlanan dijital imzalar, bir mesajı gönderenin veya bir belgeyi imzalayanın kimliğini doğrulamak ve muhtemelen gönderilen mesajın veya belgenin orijinal içeriğinin değişmediğinden emin olmak için günümüzde yaygın olarak kullanılmaktadır. Eliptik eğri tabanlı kriptografi sistemlerinin yaygınlaşması ile birlikte dijital imza olarak kullanılmak üzere eliptik eğri tabanlı imza algoritmaları önerilmiştir. İlk olarak Scott Vanstone tarafından 1992 yılında ECDSA (Elliptic Curve Digital Signature Algorithm) olarak adlandırılan eliptik eğriler üzerinde dijital imza algoritmasına (DSA) dayanan bir imza şeması önerilmiştir [69]. Önerilen imza şemasının güvenliği ECLDP'ye dayanmaktadır. ECDSA imza algoritması üç aşamadan oluşur: (1) algoritma oluştururken sistem parametrelerinin belirlendiği ve gizli anahtarın seçildiği anahtar üretim aşaması, (2) imzalama aşaması ve (3) imza doğrulama aşaması [69]. Bu üç aşama aşağıda incelenmiştir.

<u>Anahtar üretim aşaması</u>

1. Sonlu asal cisim $\mathbb{F}_p$ üzerinde *E* eliptik eğrisi tanımlı olsun. Burada *p,* kriptografik uygulamalar için yeterince büyük bir asal sayı ve *G* başlangıç noktası olarak seçilsin.
2. İmzacı *S,* $d \in \mathbb{Z}_p^*$ olmak üzere rasgele gizli anahtar olarak d tamsayısını seçsin.

3. Daha sonra imzacı *S*, *G* başlangıç noktası ve *d* gizli anahtarını kullanarak $Q(x, y) = d \cdot G$ skaler çarpma işlemi ile Q açık anahtarını üretir.

İmzalama aşaması

1. İmzacı *S*, mesajını imzalamak için öncelikle rasgele *k* tamsayısını seçer ($k \in \mathbb{Z}_p^*$).
2. Daha sonra *G* başlangıç noktasını kullanarak $k \cdot G = (x_1, y_1)$ skaler çarpma işlemi ile x_1, y_1 noktalarını hesaplar.
3. $r = x_1 \pmod p$ işlemi ile *r* hesaplanır. Eğer $r = 0$ ise birinci adıma gidilerek tekrar *k* değeri seçilerek sonraki adıma geçilir. Eğer $r \neq 0$ ise dördüncü adımdan devam edilir.
4. *k* tamsayısının *p* modülüne göre tersi $k^{-1} \pmod p$ hesaplanır.
5. *M* mesajının SHA-1 özet fonksiyonu ile *Hash*(*M*) = *h* özet değeri hesaplanır.
6. Hesaplanan değerler kullanılarak $s = k^{-1}(h + dr) \pmod p$ işlemi ile *s* hesaplanır. Eğer $s = 0$ ise birinci adıma gidilerek tekrar *k* değeri seçilir. Eğer $s \neq 0$ ise imzalama işlemi tamamlanmış olur.
7. İmzacı *S*, *M* mesajının imzası (r, s) olarak oluşturulur ve mesajla birlikte alıcı *R*'ye $\{E, p, Q, G, M, (r, s)\}$ parametrelerini gönderir ($r, s \in \mathbb{Z}_p^*$).

İmza doğrulama aşaması

1. Mesaj *M* ile $\{E, p, Q, G, (r, s)\}$ parametrelerini alan alıcı *R*, mesajın içeriğinin ve kaynağının doğruluğunu ispatlamak amacıyla imza doğrulama aşamasını başlatır.
2. İlk olarak $w = s^{-1} \pmod p$ hesaplanır.
3. *M* mesajının SHA-1 özet fonksiyonu ile *Hash*(*M*) = *h* özet değeri hesaplanır.
4. $u_1 = hw \pmod p$ ve $u_2 = rw \pmod p$ hesaplamaları ile u_1, u_2 değerleri hesaplanır.
5. $(x_1, y_1) = u_1 \cdot G + u_2 \cdot Q$ işlemi ile x_1, y_1 noktaları ve $v = x_1 \pmod p$ hesaplanır.
6. Son olarak mesajı alan *R*, hesaplamalar sonucunda $v = r$ eşitliğini kontrol eder, eğer $v = r$ ise imza geçerli, $v \neq r$ ise imza geçersiz kabul edilir.

Eliptik eğri tabanlı imza şeması olan ECDSA'nın tasarım aşamaları yukarıda tanımlanmıştır. İmza doğrulama aşaması aşağıdaki işlem adımları ile ispatlanabilir.

$$(x_1, y_1) = u_1 \cdot G + u_2 \cdot Q \tag{3.23}$$

Eş. 3.23'te verilen eşitlikte $(x_1, y_1), u_1, u_2$ değerlerinin yerine sırasıyla eşdeğerleri olan $k \cdot G$, *hw*, *rw* değerlerini yazalım.

$$k \cdot G = hw \cdot G + rw \cdot Q \tag{3.24}$$

Elde edilen Eş. 3.24'teki eşitlikte w değeri yerine $w = s^{-1}$ ve Q değeri yerine $Q = d \cdot G$ yazalım.

$$k \cdot G = hs^{-1} \cdot G + rs^{-1} d \cdot G \tag{3.25}$$

Son olarak Eş. 3.25 düzenlenirse Eş. 3.26 ve Eş. 3.27 elde edilir.

$$k = s^{-1}(h + dr) \tag{3.26}$$

$$s = k^{-1}(h + dr) \tag{3.27}$$

Böylece, Eş. 3.27'de imzalama eşitliği elde edilerek imza doğrulama aşaması ispatlanmıştır.

3.10. Eliptik Eğri Dijital İmza Algoritmasının Güvenliği

ECDSA'nın güvenliği, eliptik eğri ayrık logaritma problemini (ECDLP) çözmenin zorluğuna dayanmaktadır. *C*'nin, imzamızı taklit etmek isteyen siyah şapkalı bir bilgisayar korsanı olduğunu varsayalım. *C*, imzamıza ve sistem parametrelerine $\{E, p, Q, G, M, (r, s)\}$ sahip olsun. Eğer kötü niyetli bu bilgisayar korsanı, anahtar üretme ve imzalama aşamasında kullandığımız gizli *d* ve *k* anahtarlarımızı ele geçirirse imzamızı her yerde kullanabilir. Bu nedenle eliptik eğri parametrelerinin belirlenmesinde uluslararası standartlara uygun olarak dikkatli seçmeliyiz. Dikkat edilecek olursa bilgisayar korsanı *C*, herkes tarafından bilinen (r, s) imzasını kullanarak ve *Hash*(*M*) = *h* özet değerini hesaplayarak Eş. 3.26 eşitliğine göre *k* ve *d* arasındaki eşitliği bulur. Bu durumu önlemek

amacıyla birden fazla mesaj imzalanırken *k* değeri her mesaj için bağımsız olarak seçilmelidir. *k* değeri farklı seçilmezse bilgisayar korsanı tarafından daha önceki imzalardan elde edilen eşitlikten *d* değeri hesaplanabilir.

İki farklı mesaj için aynı *k* gizli anahtarını kullandığımızı varsayalım. Aynı *k* değeri kullanıldığı için *r* değeri aynı olacaktır. Elde edilen imzalar (r, s_1) ve (r, s_2) olacaktır. İki farklı mesaj için elde edilen özet değerleri h_1 ve h_2 birbirinden farklıdır. Her iki imza denkleminin kullanılması ile *d* değeri aşağıdaki adımlar izlenerek elde edilir.

1. $s_1 = k^{-1}(h_1 + dr) \pmod{p}$ (Birinci imza).
2. $s_2 = k^{-1}(h_2 + dr) \pmod{p}$ (İkinci imza).
3. Her iki imza denklemi düzenlenir: $ks_1 - ks_2 = (h_1 + dr) - (h_2 + dr)$.
4. Elde edilen eşitlik basitleştirilir: $k(s_1 - s_2) = (h_1 - h_2)$.
5. *k* değeri $k = (h_1 - h_2) / (s_1 - s_2)$ şeklinde elde edilir.
6. *k* değerinden yola çıkarak *d* değeri bulunur.

Yukarıdaki işlem adımlarından görüldüğü gibi eğer *k* her imzalama aşamasında farklı seçilmezse *d* gizli anahtarı elde edilerek imzamız kötü niyetli bilgisayar korsanı *C* tarafından taklit edilebilir. Ancak *k* değeri her imzalama aşamasında rasgele bir tamsayı seçilerek bu olasılık ortadan kaldırılabilir. Bu durumda ECDLP sayesinde *k* ve *d*'yi bulmak çok zordur. Zira dikkat edilecek olursa işleminde *d* gizli anahtarını bulmak ECDLP problemine dayanmaktadır.

3.11. Eliptik Eğri Dijital İmza Algoritmasının Performansı ve Algoritma Önerisi

ECDSA'nın imzalama ve imza doğrulama aşamalarında kullanılan modüler aritmetikte ters alma işlemi matematiksel olarak oldukça maliyetli bir işlemdir. Buna bağlı olarak imzalama ve imza doğrulama sürelerinde gecikmeler oluşmaktadır. Algoritmada bulunan imzalama ve imza doğrulama aşamalarının hızı sayesinde ECDSA giderek daha çok alanda kullanılmaktadır. Günümüzdeki nesnelerin interneti, araçların iletişimi gibi gecikme süresinin düşük olması istenen uygulamalarda yaygın olarak dijital imzalar kullanılmaktadır. ECDSA için performans kriterleri dikkate alındığında modüler

aritmetikte ters alma işlemi, eliptik eğrilerde nokta toplama ve skaler çarpma işlemleri oldukça maliyetli matematiksel işlemler olarak dikkat çekmektedir. Ancak düşük gecikme süreleri elde etmek amacıyla algoritma üzerinde gerçekleştirilen işlemler güvenlik seviyesinde zafiyet oluşturabilmektedir. Algoritmanın güvenliğini artırmak için gerçekleştirilen işlemler performans artırmak için gereken işlemlere göre daha kolay ve uygulanabilirdir.

Literatür incelendiğinde eliptik eğri tabanlı birçok imza algoritması önerildiği görülmektedir [81-85]. Önerilen algoritmalar farklı güvenlik ve performans parametreleri dikkate alınarak gerçekleştirilmektedir. Bu çalışmalar dikkate alınarak eliptik eğri tabanlı yeni bir imza algoritması önerilmiştir. Önerilen algoritmada, anahtar üretim aşamasındaki gerçekleştirilen değişiklik ile imzalama ve imza doğrulama aşamalarındaki modüler aritmetikte ters alma işlemi kaldırılmıştır. Bu işlemleri gerçekleştirmek için anahtar üretim aşamasında; açık anahtar Q, rasgele seçilen gizli d tamsayısının modüler aritmetikte tersi alınarak sadece bir kez hesaplanır. Önerilen ECDSA imza algoritması üç aşamadan oluşur: (1) anahtar üretim aşaması, (2) imzalama aşaması, (3) imza doğrulama aşaması [67]. Önerilen imza algoritmasının tasarım aşamaları aşağıda verilmiştir.

Anahtar üretim aşaması

1. Sonlu asal cisim $\mathbb{F}_p$ üzerinde E eliptik eğrisi tanımlı olsun. Burada p, kriptografik uygulamalar için yeterince büyük bir asal sayı ve G başlangıç noktası olarak seçilsin.
2. İmzacı S, $d \in \mathbb{Z}_p^*$ olmak üzere rasgele gizli anahtar olarak d tamsayısını seçsin.
3. Daha sonra imzacı S, G başlangıç noktasını ve d gizli anahtarını kullanarak $Q = (d^{-1} \bmod p) \cdot G$ hesaplaması ile Q açık anahtarını üretir. Bu adım klasik ECDSA algoritmasında olduğundan farklıdır.

İmzalama aşaması

1. İmzacı S, mesajını imzalamak için öncelikle rasgele k tamsayısını seçer $(k \in \mathbb{Z}_p^*)$.
2. G başlangıç noktasını kullanarak $k \cdot G = (x_1, y_1)$ işlemi ile x_1, y_1 noktaları hesaplanır.
3. $r = x_1 \pmod p$ işlemi ile r hesaplanır. Eğer $r = 0$ ise birinci adıma gidilerek tekrar k değeri seçilerek sonraki adıma geçilir. Eğer $r \neq 0$ ise dördüncü adımdan devam edilir.

4. *M* mesajının SHA-512 özet fonksiyonu ile *Hash*(*M*) = *h* özet değeri hesaplanır. Klasik ECDSA algoritmasında SHA-1 özet fonksiyonu kullanılmaktadır.
5. Hesaplanan değerler kullanılarak $s = d(k - h) \pmod{p}$ işlemi ile *s* hesaplanır. Eğer $s = 0$ ise birinci adıma gidilerek tekrar *k* değeri seçilir. Eğer $s \neq 0$ ise imzalama işlemi tamamlanır.
6. *M* mesajının imzası (r, s) olarak oluşturulur ve mesajla birlikte alıcı *R*'ye $\{E, p, Q, G, M, (r, s)\}$ parametreleri gönderilir $(r, s \in \mathbb{Z}_p^*)$.

İmza doğrulama aşaması

1. Mesaj *M* ile $\{E, p, Q, G, (r, s)\}$ parametrelerini alan alıcı *R*, mesajın içeriğinin ve kaynağının doğruluğunu ispatlamak amacıyla imza doğrulama aşamasını başlatır.
2. *M* mesajının SHA-512 özet fonksiyonu ile *Hash*(*M*) = *h* özet değeri hesaplanır.
3. $u_1 = h \pmod{p}$ ve $u_2 = s \pmod{p}$ hesaplamaları ile u_1, u_2 değerleri hesaplanır.
4. $(x_1, y_1) = u_1 \cdot G + u_2 \cdot Q$ işlemi ile x_1, y_1 noktaları ve $v = x_1 \pmod{p}$ hesaplanır.
5. Son olarak mesajı alan *R*, hesaplamalar sonucunda $v = r$ eşitliğini kontrol eder, eğer $v = r$ ise imza geçerli, $v \neq r$ ise imza geçersiz kabul edilir.

Önerilen ECDSA imza algoritmasının tasarım aşamaları yukarıda tanımlanmıştır. İmza algoritmasının doğruluğu, imza doğrulama aşamasındaki işlem adımları ile aşağıdaki gibi ispatlanabilir.

$$(x_1, y_1) = u_1 \cdot G + u_2 \cdot Q \tag{3.28}$$

Eş. 3.28'de verilen eşitlikte $(x_1, y_1), u_1, u_2$ değerlerinin yerine sırasıyla eşdeğerleri olan $k \cdot G,\ s,\ h$ değerlerini yazalım.

$$k \cdot G = h \cdot G + s \cdot Q \tag{3.29}$$

Elde edilen Eş. 3.29'da açık anahtar Q değeri yerine $Q = (d^{-1} \bmod p) \cdot G$ eşdeğerini yazalım ve eşitliği aşağıdaki gibi düzenleyelim.

$$s \cdot (d^{-1} \cdot G) = (k - h) \cdot G \tag{3.30}$$

Son olarak Eş. 3.30 sadeleştirilir ve düzenlenirse aşağıdaki eşitlik elde edilir.

$$s = d(k - h) \tag{3.31}$$

Böylece, Eş. 3.31'de imzalama eşitliği elde edilerek imza doğrulama aşaması ispatlanmıştır.

3.12. Önerilen ECDSA İmza Algoritmasının Güvenliği

Önerilen imza algoritmasının güvenliğini ispat etmek amacıyla iki farklı imza oluşturulur ve aşağıdaki işlem adımları gerçekleştirilir.

1. $s_1 = d(k_1 - h_1) \pmod{p}$ (Birinci imza).
2. $s_2 = d(k_2 - h_2) \pmod{p}$ (İkinci imza).
3. Her iki imza eşitliği düzenlenirse: $s_1 - s_2 = d(k_1 - k_2 - h_1 + h_2)$.
4. Gizli k anahtarının her imzalama aşamasında farklı seçilmesinden dolayı (k_1, k_2), k değeri her imzalamada değişecektir.
5. $s_1 - s_2 = d(k_1 - k_2 - h_1 + h_2)$ eşitliğinde k_1, k_2, d anahtarları gizli olduğundan ve k gizli anahtarının her imzalama aşamasında değişmesinden dolayı gizli anahtarlar arasında bağlantı kurulması oldukça zordur.

Bu durumda önerilen ECDSA imza algoritmamızın güvenliği ECDLP nedeniyle k ve d'yi bulmak çok zor olmasına dayanır. Zira dikkat edilecek olursa $Q = (d^{-1} \bmod p) \cdot G$ işleminde d gizli anahtarını bulmak ECDLP problemine dayandığından oldukça zordur.

3.13. Önerilen İmza Algoritmasının Uygulaması ve Karşılaştırılması

Önerilen imza algoritmasının tasarım aşamaları ve güvenliği ispatlandıktan sonra bilgisayar programı ile uygulaması gerçekleştirilmiştir. Çizelge 3.6'da önerilen imza algoritmasında kullanılan NIST tarafından tavsiye edilen NIST P-521 eliptik eğrisinin parametreleri verilmiştir [78]. Çizelgede verilen parametrelerden p ile sonlu asal cismin

mertebesi, *a* ve *b* ile eliptik eğri denkleminin parametreleri, *n* ile eliptik eğrinin mertebesi, *x* ve *y* ile ise *G* başlangıç noktasının parametreleri ifade edilmektedir. Önerilen imza algoritmasının gerçekleştirim adımları aşağıda verilmiştir.

Çizelge 3.6. NIST P-521 eliptik eğrisinin parametreleri

Parametre	Değer
p	6864797660130609714981900799081393217269435300143305409394463459185543183397656052122559640661454554977296311391480858037121987999716643812574028291115057151
a	-3
b	0x 051 953eb961 8e1c9a1f 929a21a0 b68540ee a2da725b 99b315f3 b8b48991 8ef109e1 56193951 ec7e937b 1652c0bd 3bb1bf07 3573df88 3d2c34f1 ef451fd4 6b503f00
n	6864797660130609714981900799081393217269435300143305409394463459185543183397655394245057746333217197532963996371363321113864768612440380340372808892707005449
x	0x c6 858e06b7 0404e9cd 9e3ecb66 2395b442 9c648139 053fb521 f828af60 6b4d3dba a14b5e77 efe75928 fe1dc127 a2ffa8de 3348b3c1 856a429b f97e7e31 c2e5bd66
y	0x 118 39296a78 9a3bc004 5c8a5fb4 2c7d1bd9 98f54449 579b4468 17afbd17 273e662c 97ee7299 5ef42640 c550b901 3fad0761 353c7086 a272c240 88be9476 9fd16650

Anahtar üretim aşaması

1. Sonlu asal cisim $\mathbb{F}_p$ üzerinde tanımlanacak *E* eliptik eğrisi için NIST tarafından önerilen P-521 eliptik eğrisini seçelim. NIST P-521 eliptik eğrisinin parametreleri Çizelge 3.6'da gösterilmiştir. Çizelgede; *p* ile sonlu asal cismin derecesi, *a* ve *b* ile eliptik eğri denkleminin parametreleri, *n* ile eliptik eğrinin derecesi, *x* ve *y* ile ise *G* başlangıç noktası (x, y) temsil edilmektedir.
2. İmzacı *S*, rasgele gizli anahtar olarak d tamsayısını seçsin $(d \in \mathbb{Z}_n^*)$.

$d =$ 2738551815154143238629604620434095212002380032088698751378522915984114965235391041520729090359610602655212404092228365548874698617979374804880397822124412005

3. İmzacı *S*, $G(x, y)$ başlangıç noktasını ve d gizli anahtarını kullanarak $Q = (d^{-1} \bmod n) \cdot G$ hesaplaması ile Q açık anahtarını hesaplar.

$d^{-1}(\text{mod } n)$ = 19416518938517003184466105236933678645940172608213839353184504863432821682355600505951501188497597366456719729120856073614393420468983281050833046757163165 40

Q = (27981906457861405044428152874157991981621885789649844222435065192846266903521847620629424204575047888731190996761440220266328857117954454330234977617078055 6,
11911343402287581599337907953061492403306052818563913148136560127213331562407053788396916910485276253677641801576541534067955852174620239699012481889308552 84)

<u>İmzalama aşaması</u>

1. İmzacı *S*, mesajını imzalamak için öncelikle rasgele *k* tamsayısını seçer $(k \in \mathbb{Z}_n^*)$.

k = 67181709246772475249811115371613144816498172986114956322541418021897598055918041615362919725470012266643281620254318693095678844155379200064407988399145774 8

2. $k \cdot G(x, y) = (x_1, y_1)$ işlemi ile x_1, y_1 noktaları hesaplanır.

x_1 = 37796482561053347957005484992577554400457026734420243468653164675431394303871435484025861984778133963225915102067696902901638511845445729655469709140965525 19

y_1 = 68516115587219135179432285125090408560819974121383906635040087799395996113150245769241097746291661997159969543824710300153990352697020425136598795633211397 47

3. *"Gazi Gelecektir"* mesajının SHA-512 özet fonksiyonu ile *Hash* (*"Gazi Gelecektir"*) = *h* özet değeri hesaplanır ve tamsayıya çevrilir.

SHA-512 = 5f96015075b3013039614cea89f3cdc65ab8387d60a720cc5cf1e18ec92771389a76c3edd250a5fdf055ef096aeb0923e8e8d03763c642689420860325b4ba05

$h =$ 133915885858867386326001395563861002659711888107837963217693922204222203477275939277177841743055290986660893250522468858167742253646957765678250923790112 42

Hesaplanan değerler kullanılarak $s = d(k-h) \pmod{n}$ işlemi ile s hesaplanır. Eğer $s = 0$ ise birinci adıma gidilerek tekrar k değeri seçilir. Eğer $s \neq 0$ ise imzalama işlemi tamamlanır. $r = x_1 \pmod{n}$ olmak üzere mesajın imzası (s, r) parametreleri olarak alıcı R'ye gönderilir.

$s =$ 150693329679089927227309522713651557189492463630293889892538352595158883311836834363716372718965194310887184477286952626966151902043231944951282005488404895

$r = x_1 =$ 377964825610533479570054849925775544004570267344202434686531646754313943038714354840258619847781339632259151020676969029016385118454457296554697091409655251 9

Mesaj doğrulama aşaması

1. Mesaj olarak gönderilen *"Gazi Gelecektir"* metni ile $\{E, p, Q, G, (s, r)\}$ parametrelerini alan alıcı R, mesajın içeriğinin ve kaynağının doğruluğunu ispatlamak amacıyla imza doğrulama aşamasını başlatır.
2. *"Gazi Gelecektir"* mesajının SHA-512 özet fonksiyonu ile Hash(*"Gazi Gelecektir"*) = h özet değerini hesaplanır.

$h =$ 133915885858867386326001395563861002659711888107837963217693922204222203477275939277177841743055290986660893250522468858167742253646957765678250923790112 42

3. $u_1 = h \pmod{n}$ ve $u_2 = s \pmod{n}$ hesaplamaları ile u_1, u_2 değerleri hesaplanır.

$u_1 =$ 500624281129393824625504726382865759752232397780767976688451627613268211318977575314654120426010842258350677620702638986794947706320009577981569394047232 5

$u_2 =$ 511494794927024022091770186637392967283727396101915950862243335852392801775726383867468986979456176147304014138389850752432361405720796810122064065121407802 2

4. $(x_1, y_1) = u_1 \cdot G + u_2 \cdot Q$ işlemi ile x_1, y_1 noktaları ve $v = x_1 \pmod n$ hesaplanır.

$x_1 =$ 377964825610533479570054849925775544004570267344202434686531646754313943038714354840258619847781339632259151020676969029016385118454457296554696709140965525 19

$y_1 =$ 685161155872191351794322851250904085608199741213839066350400877993959961131502457692410977462916619971599695438247103001539903526970204251365987956332113974 7

$v =$ 377964825610533479570054849925775544004570267344202434686531646754313943038714354840258619847781339632259151020676969029016385118454457296554696709140965525 19

5. Son olarak *R*, hesaplamalar sonucunda $v = r$ eşitliğinin sağlanıp sağlanamadığını kontrol eder, $v = r$ eşitliği sağlandığından imzanın geçerli olduğu ispatlanır.

Önerilen imza algoritması ile ECDSA'nın daha verimli ve güvenli bir versiyonu tasarlanmış ve uygulanmıştır. ECDSA'da hem imzalama hem de doğrulama aşamasında yer alan modüler aritmetikte ters alma işlemi, matematiksel olarak maliyetli ve zaman alan bir işlemdir. NIST tarafından önerilen sonlu asal cisimler $\mathbb{F}_{P-192}$, $\mathbb{F}_{P-224}$, $\mathbb{F}_{P-256}$, $\mathbb{F}_{P-384}$ ve $\mathbb{F}_{P-521}$ üzerinde gerçekleştirilen modüler aritmetikte toplama, çıkarma, çarpma ve ters alma işlemleri ile eliptik eğri üzerindeki nokta toplama ve nokta çiftleme işlemlerin uygulama süreleri (mikro saniye- μs) Çizelge 3.7'de verilmiştir [67].

Çizelge 3.7. Sonlu asal cisimler üzerinde gerçekleştirilen işlemlerin süreleri (μs)

Matematiksel işlem	$\mathbb{F}_{P-192}$	$\mathbb{F}_{P-224}$	$\mathbb{F}_{P-256}$	$\mathbb{F}_{P-384}$	$\mathbb{F}_{P-521}$
Toplama	0,221	0,264	0,241	0,244	0,237
Çıkarma	0,217	0,228	0,223	0,228	0,225
Çarpma	0,338	0,408	0,320	0,823	1,020
Ters alma	3,790	4,4	4,020	5,62	7,340
Nokta toplama	50,4	52,9	51	54,4	57,9
Nokta çiftleme	64,1	65,8	65,6	69	74,8

Eliptik eğrilerdeki nokta toplama ve nokta çiftleme işlemleri haricinde modüler aritmetikteki ters alma işleminin diğer işlemlerden çok daha fazla gerçekleştirim süresi gerektirdiğini görülmektedir. Donanım tabanlı uygulamalarda, modüler aritmetikteki ters alma işlemini gerçekleştirmek için yazılım tabanlı uygulamalara göre daha fazla zaman gerektiğinden önerilen imza algoritmasının önemi artmaktadır. Önerilen imza algoritmasında, yalnızca anahtar üretim aşamasında ters alma işlemi gerçekleştirilir. Fakat bu işlem sadece bir kez gerçekleştirildiği için imzalama ve imza doğrulama süresini etkilemeyecektir.

Literatürdeki diğer eliptik eğri tabanlı imzalar ve klasik ECDSA ile karşılaştırıldığında, önerilen imza algoritması daha düşük süre ve daha az matematiksel işlem sayesinde daha iyi sonuçlar vermektedir. Önerilen imza algoritmasının, klasik ECDSA ve mevcut diğer eliptik eğri tabanlı imzalarla karşılaştırılması Çizelge 3.8'de verilmiştir.

Çizelge 3.8. Önerilen imza algoritmasının mevcut imza algoritmaları ile karşılaştırılması

Algoritma	Aşama	$M_{Toplama}$	$M_{Çıkarma}$	$M_{Çarpma}$	$M_{Tersalma}$	E_{Skaler}	$E_{Toplama}$
Klasik ECDSA [69]	İmzalama	1	-	2	1	1	-
	Doğrulama	-	-	2	1	2	1
[81]	İmzalama	-	-	4	2	1	-
	Doğrulama	-	-	1	1	1	-
[82]	İmzalama	-	1	2	1	-	-
	Doğrulama	-	-	2	1	2	1
[83]	İmzalama	2+1(⊕)	-	3	-	2	-
	Doğrulama	1(⊕)	-	-	1	2	1
[84]	İmzalama	2	-	3	1	2	-
	Doğrulama	1	-	1	1	2	1
[85]	İmzalama	2	-	3	1	2	-
	Doğrulama	-	-	4	1	2	1
Önerilen Algoritma	İmzalama	-	1	1	-	1	-
	Doğrulama	-	-	-	-	2	1

Çizelgedeki kısaltmalardan $M_{Toplama}$: modüler aritmetikte toplama, $M_{Çıkarma}$: modüler aritmetikte çıkarma, $M_{Çarpma}$: modüler aritmetikte çarpma, $M_{Tersalma}$: modüler aritmetikte ters alma, E_{Skaler}: eliptik eğri üzerinde skaler çarpma, $E_{Toplama}$: eliptik eğri üzerinde nokta toplama, $\oplus$: XOR işlemini ifade etmektedir.

Klasik ECDSA'nın ve önerilen imza algoritmasının imzalama ve imza doğrulama aşamalarının gerçekleştirim süreleri (milisaniye: ms) Çizelge 3.9'da gösterilmiştir. Önerilen imza algoritmasında, imzalama ve imza doğrulama aşamaları klasik ECDSA'ya kıyasla sırasıyla %25 ve %18 daha düşük sürede gerçekleştirilmektedir.

Çizelge 3.9. Önerilen imza algoritmasının klasik ECDSA ile karşılaştırılması

İmza algoritması	İmzalama süresi (ms)	İmza doğrulama süresi (ms)
Klasik ECDSA	48	83
Önerilen Algoritma	36	63

Önerilen imza algoritması göstermiş olduğu performans sayesinde hız ve güvenlik gerektiren birçok yazılım ve donanım uygulamasında kullanılabilmektedir. Özellikle günümüzde önemi giderek popüler hale gelen nesnelerin interneti gibi alanlarda var olan düşük güç, sınırlı bellek ve düşük bilgi işlem kapasitesi gibi kısıtlı kaynaklı uygulamalar için uygundur.

4. EŞLEME TABANLI KRİPTOGRAFİ

Eşleme tabanlı kriptografi, gelişmiş güvenlik özelliklerine sahip, geleneksel açık anahtarlı kriptografiye kıyasla daha verimli olan matematiksel eşleme kavramına dayanan bir açık anahtarlı kriptografi türüdür. Eşlemeler, iki nokta grubunu üçüncü bir nokta grubuna eşleyen çift doğrusal fonksiyonlardır. Eşlemelerin kullanımı ek maliyet gerektirir ve ilgili hesaplamaları verimli bir şekilde gerçekleştirmek için özel donanım veya yazılım ihtiyacı ortaya çıkarabilir. Eşleme tabanlı kriptografide bu eşlemeler şifreleme, şifre çözme, dijital imzalama ve anahtar anlaşması gibi çeşitli kriptografik işlemleri gerçekleştirmek için kullanılır. Eşleme tabanlı kriptografinin temel avantajlarından biri, geleneksel açık anahtarlı kriptografiye kıyasla gelişmiş güvenlik seviyesidir. Geleneksel açık anahtar kriptografisinde güvenlik, büyük tamsayıların çarpanlarına ayrılması veya ayrık logaritmaların hesaplanması gibi belirli matematiksel problemlerin çözülmesinin zorluğuna dayanır. Buna karşın eşleme tabanlı kriptografinin güvenliği, çift doğrusal Diffie-Hellman karar verme problemi (Decisional Bilinear Diffie-Hellman Problem; DBDHP) veya çift doğrusal Diffie-Hellman problemi (Bilinear Diffie-Hellman Problem; BDHP) gibi eşleme hesaplamalarıyla ilgili belirli matematiksel problemlerin zorluğuna dayanır [70, 75, 86, 87, 93].

Eşleme tabanlı kriptografi bağlantılı araçlar, nesnelerin interneti, mobil ödemeler, kablosuz sensör ağları ve bulut bilişim gibi çeşitli uygulamalarda da kullanılmaktadır. Bununla birlikte, eşleme tabanlı kriptografinin kullanımı hesaplama açısından oldukça maliyetli olabilir ve bu da belirli kaynak kısıtlı ortamlarda uygulanabilirliğini sınırlayabilir. Eşleme tabanlı kriptografi fikrinin temeli matematiksel olarak eliptik eğri gruplarına dayanır [92].

Eşleme tabanlı kriptografinin en iyi bilinen uygulamalarından biri, bir kullanıcının açık anahtarının e-posta adresi veya telefon numarası gibi kendisine özel bilgilerden oluşturulmasına dayanan kimlik tabanlı kriptografidir [90]. Bu kriptografi türünde açık anahtarların önceden değiştirilmesine gerek olmadığı için basitleştirilmiş bir anahtar dağıtım sürecine olanak tanınır. Bir diğer uygulama alanı ise öznitelik tabanlı şifreleme (Attribute-Based Encryption; ABE)'dir [91]. ABE, iş unvanı veya yaş gibi belirli özniteliklere dayalı olarak şifrelenmiş verilere erişim sağlayan bir kriptografi türüdür. Eşleme tabanlı kriptografi, daha verimli ve güvenli ABE sistemleri oluşturmak için

kullanılabilir. Bu uygulamalarla birlikte eşleme tabanlı kriptografi, kullanıcıların kimliklerini açıklamadan kendileri hakkında belirli nitelikleri kanıtlamalarına olanak tanıyan anonim kimlik bilgi sistemleri oluşturmak için de kullanılabilir. Tüm kriptografik sistemlerde olduğu gibi, eşleme tabanlı kriptografinin pratikte güvenliğini sağlamak için uygun tasarım ve uygulama oldukça önemlidir.

Bu bölümde eşleme tabanlı kriptografinin temelini oluşturan cebirsel kavramlar ve uygulamalar hakkında bilgi verilmiştir. Eşleme tabanlı kriptografi kullanan şifreleme ve imza algoritmaları tanıtılmıştır. Ayrıca kimlik tabanlı kriptografi ve sertifikasız kriptografi hakkında da bilgi verilmiştir.

4.1. Çift Doğrusal Eşleme

Bu başlıkta çift doğrusal eşlemenin (bilinear pairing) nasıl tanımlandığı ve hangi özelliklere sahip olduğu hakkında bilgi verilmiştir. Çift doğrusal eşlemeyi tanımlamak için G_1, p mertebeli toplamsal değişmeli bir grup olsun ve G_T ise, p mertebeli çarpımsal değişmeli bir grup olsun. Çift doğrusal eşleme olarak adlandırılan e aşağıdaki gibi ifade edilebilir [87].

$$e : G_1 \times G_1 \rightarrow G_T \tag{4.1}$$

Kriptografide kullanıma uygun bir eşleme; dejenere olmama, kolay hesaplama ve çift doğrusallık özelliklerini sağlamalıdır. Bu özellikleri aşağıda inceleyelim.

Çift doğrusallık: e eşlemesinin çift doğrusal olması için $Q_1, P_1, P_2 \in G_1$ olmak üzere $e(P_1 + P_2, Q_1) = e(P_1, Q_1)\ e(P_2, Q_1)$ ve $e(P_1, P_2 + Q_1) = e(P_1, P_2)\ e(P_1, Q_1)$ şeklinde tanımlanan iki özelliği sağlaması gerekir. Bu iki özelliğin sonucu olarak $a, b \in \mathbb{Z}_p$ olmak üzere $e(a \cdot P_1, b \cdot P_2) = e(b \cdot P_1, a \cdot P_2) = e(P_1, P_2)^{ab}$ eşitliklerini sağlayan özellik eşleme tabanlı kriptografide çift doğrusallık özelliğine dayanır.

Dejenere olmama: $P, Q \in G_1$ ve $P, Q \neq \mathcal{O}$ olmak üzere $e(P, Q) \neq 1$ özelliğini sağlar.

Kolay hesaplanabilme: Çift doğrusal eşleme verimli bir şekilde hesaplanabilir olmalıdır. $P_1, P_2 \in G_1$ olmak üzere, $e(P_1, P_2)$ eşlemesini hesaplamak için çift doğrusal eşlemedeki hesaplama işlemlerini verimli bir şekilde gerçekleştiren bir algoritma olmalıdır.

4.1.1. Tanım

G_1 ve G_T, p mertebeli sırasıyla toplamsal değişmeli $(G_1, +)$ ve çarpımsal değişmeli $(G_T, \cdot)$ birer grup, $e: G_1 \times G_1 \to G_T$ çift doğrusal eşleme ve $P \in G_1$ olsun. P, $a \cdot P$, $b \cdot P$ ve $c \cdot P$ verildiği zaman, $e(P, P)^{abc}$ hesaplaması oldukça zor olup bu problem çift doğrusal Diffie-Hellman problemi (BDHP) olarak ifade edilir [86, 89].

4.1.2. Tanım

G_1 ve G_T, p mertebeli sırasıyla toplamsal değişmeli $(G_1, +)$ ve çarpımsal değişmeli $(G_T, \cdot)$ birer grup, $e: G_1 \times G_1 \to G_T$ çift doğrusal eşleme ve $P \in G_1$ olsun. P, $a \cdot P$, $b \cdot P$, $c \cdot P$ ve x verildiği zaman $x = e(P, P)^{abc}$ olup olmadığının belirlenmesi oldukça zor problem olup bu problem çift doğrusal Diffie-Hellman karar verme problemi (DBDHP) olarak ifade edilir [75].

4.2. Bölen Teorisi

Bu alt bölümde, kriptografik eşleme hesaplamalarının gerçekleştirilmesi için kullanılan bölenler hakkında bilgi verilmiştir. Rasyonel bir f fonksiyonunun sıfırlarını ve kutuplarını bölen olarak adlandırılır ve $div(f)$ olarak ifade edilebilir. Böylece bir bölen, f'in bir sıfır veya kutba sahip olduğu noktaların toplamı olarak, sıfırların çokluklarına göre pozitif ağırlıklar ve kutupların çokluklarına göre ise negatif ağırlıklar alması kuralıyla, sıfırların ve kutupların çoklukları ile ağırlıklandırması olarak ifade edilir [75].

4.2.1. Tanım

Bir S kümesinin biçimsel toplamı, S'nin elemanlarının serisidir $\{s_0, s_1, s_2, ...\}$. Biçimsel toplam genellikle bir yer tutucu kullanılarak yazılır [75].

Sonlu K cismi üzerinde tanımlanan E eliptik eğrisinin sahip olduğu noktaların biçimsel toplamına bölen denir. $P_i \in E$ ve $a_i \in \mathbb{Z}$ olmak üzere bir D böleni aşağıdaki gibi tanımlanır [75, 89, 94].

$$D = \sum_{P \in E} a_i [P] \tag{4.2}$$

$P = \{P_1, P_2, P_3, ...P_i\}$ olmak üzere bir eliptik eğri üzerindeki noktalar kümesi ise, P'nin elemanlarının biçimsel toplamı $D = a_1 \cdot P_1 + a_2 \cdot P_2 + ... + a_i \cdot P_i$ şeklinde ifade edilir [75].

E eliptik eğrisinin bölenlerinden oluşan bir grup $div(E)$ olarak gösterilir ve bu grup E üzerindeki noktalardan oluşan bir değişmeli gruptur.

4.2.2. Tanım

D böleninin mertebesi;

$$\deg(D) = \deg\left(\sum_{P \in E} a_i [P] \right) = \sum_{P \in E} a_i \in \mathbb{Z} \tag{4.3}$$

Eş. 4.3 ile tanımlanır. Eğer $\deg(D) = 0$ ise bölen sıfır mertebelidir [89]. Derecesi 0 olan tüm bölenlerin kümesi bir alt grup oluşturur ve $div^0(E)$ ile gösterilir [94].

4.2.3. Tanım

D böleninin toplamı;

$$sum\left(\sum_{P \in E} a_i [P] \right) = \sum_{P \in E} a_i P \tag{4.4}$$

Eş. 4.4 ile tanımlanır [89].

Bir bölenin amacı, eliptik eğride tek bir nokta olarak değerlendirilmek değil, ağırlıkları eliptik eğri noktaları kümesindeki öğelerle ilişkilendirmektir. Bazı durumlarda bölenler ilgili noktalarda rasyonel fonksiyonların mertebelerini temsil eder. Bir f fonksiyonunun mertebesi $ord_P(f)$ ile gösterilmek üzere f fonksiyonunun böleni Eş. 4.5 ile tanımlanır [94].

$$div(f) = \sum_{P \in E} ord_p(f)[P] \tag{4.5}$$

4.3. Weil Eşlemesi

Weil eşlemesinin farklı tanımları mevcut olup bu kapsamda teorik veya kriptografide kullanımına yönelik tanımlarını yapmak mümkündür. Weil eşlemesinin hesaplanması, Victor Miller'ın algoritması [95] kullanılarak verimli bir şekilde yapılabilir. Weil eşlemesi ve özellikleri aşağıda tanımlanmıştır [94].

K cismi üzerinde E eliptik eğrisi tanımlansın ve n tamsayısı seçilsin. Seçilen $n \in \mathbb{Z}^+$ pozitif tamsayısı K cisminin karakteristiği ile bölünemeyecek bir değer olsun. Bu durumda Weil eşlemesi Eş. 4.6 ile tanımlanır [89].

$$e_n : E[n] \times E[n] \rightarrow \mu_n \in \overline{K} \tag{4.6}$$

Eş. 4.6'da verilen Weil eşlemesinde μ_n, $\overline{K}$ cismindeki n'inci kökler grubunu ifade eder. P ve Q, $E[n]$ üzerinde olduğu varsayılarak $f_P, f_Q \in \overline{K}$ olmak üzere bölenleri $div(f_P) = n \cdot P$ ve $div(f_Q) = n \cdot Q$ olarak hesaplanır. Böylece, P ve Q noktalarının Weil eşlemesi Eş. 4.7'deki şekilde tanımlanır [94].

$$e_n(P,Q) = \frac{f_P(Q)}{f_Q(P)} \tag{4.7}$$

Weil eşlemesi aşağıdaki özelliklere sahiptir [94].

- Weil eşlemesi e_n, çift doğrusallık özelliğini sağlar. Şöyle ki; $Q,P,P_1,P_2 \in E[n]$ olmak üzere $e_n(P_1+P_2,Q)=e_n(P_1,Q)\ e_n(P_2,Q)$ ve $e_n(P_1,P_2+Q)=e_n(P_1,P_2)\ e_n(P_1,Q)$ şeklinde tanımlanan iki özelliğe sahiptir.
- Weil eşlemesi e_n, dejenere olmama özelliğini sağlar. Şöyle ki; $e_n(P,Q)=1$ olması durumunda $\forall P \in E[n]$ ise $Q=\mathcal{O}$ veya $\forall Q \in E[n]$ ise $P=\mathcal{O}$ olur.
- $\forall P \in E[n]$ için $e_n(P,P)=1$ ve $\forall Q \in E[n]$ için $e_n(Q,Q)=1$ olur.
- Weil eşlemesi e_n, $\forall P,Q \in E[n]$ için $e_n(P,Q)=e_n(Q,P)^{-1}$ özelliğini sağlar [103].

4.4. Tate Eşlemesi

Tate eşlemesi tanımında farklı olarak bölenler ve bölenler üzerinde rasyonel fonksiyonların değerlendirilmesini kullanır. $P \in E(F_{q^k})[n]$ olmak üzere, k Tate gömme derecesi olarak adlandırılan uygun bir cisim genişleme boyutudur. n, q^k-1 ile bölünebilen $n \in \mathbb{Z}^+$ tamsayısı olsun $(n \mid q^k-1)$. $Q \in E(F_{q^k})/nE(F_{q^k})$ noktası bir eşkümeyi (coset) temsil etmektedir. Tate eşlemesi Eş.4.8'deki şekilde tanımlanır [94].

$$\tau_n : E(\mathbb{F}_{q^k})[n] \times E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}_{q^k}^{\ *}/(\mathbb{F}_{q^k}^{\ *})^n \tag{4.8}$$

D_P, mertebesi sıfır olan bir bölen olup $\mathbb{F}_{q^k}$ cismi üzerinde tanımlansın. $div(f_{D_P})=n \cdot D_P$ ve $D_Q \sim (Q)-(\mathcal{O})$ olmak üzere P ve Q noktalarının Tate eşlemesi Eş. 4.9'daki şekilde verilir.

$$\tau_n(P,Q) = f_{D_P}(D_Q) \tag{4.9}$$

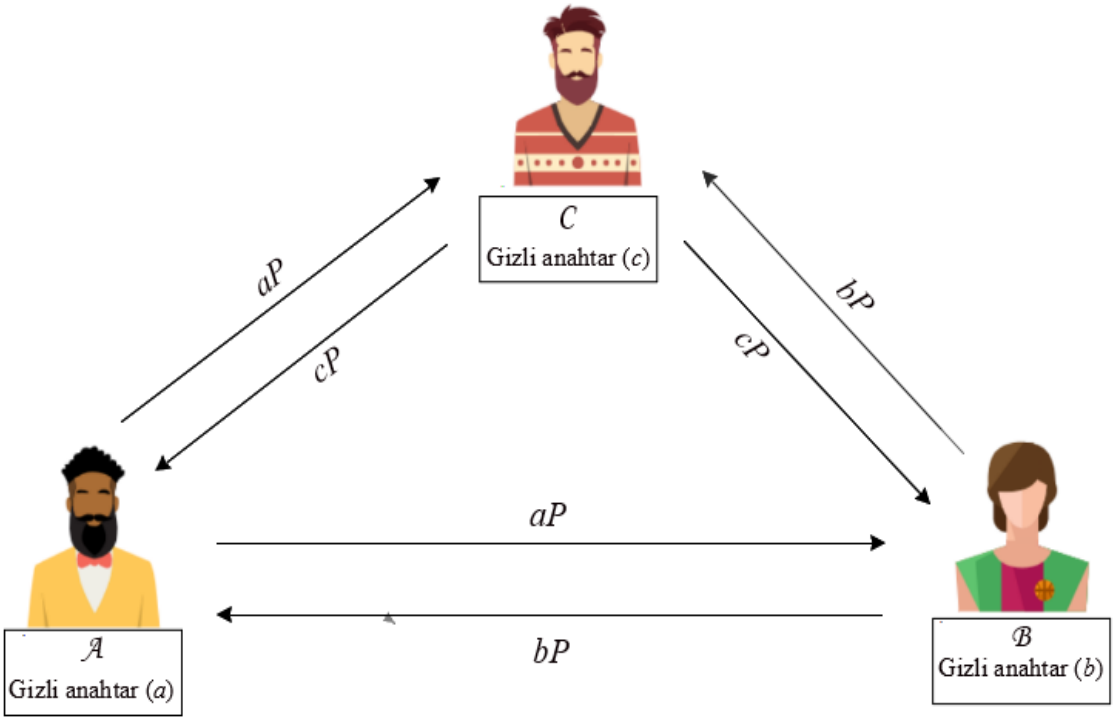
Tate eşlemesi aşağıdaki özelliklere sahiptir.

- Tate eşlemesi çift doğrusallık özelliğini sağlar. $P_1,P_2 \in E[n]$ ve $Q \in E(F_{q^k})/nE(F_{q^k})$ olmak üzere $\tau_n(P_1+P_2,Q)=\tau_n(P_1,Q)\ \tau_n(P_2,Q)$ ve $\tau_n(P_1,P_2+Q)=\tau_n(P_1,P_2)\ \tau_n(P_1,Q)$ eşitlikleri şeklinde tanımlanan iki özelliğe sahiptir.

- Tate eşlemesi dejenere olmama özelliğini sağlar. $\forall P \in E[n]$ ve $\exists Q \in E(F_{q^k}) / nE(F_{q^k})$ olmak üzere $\tau_n(P,Q) \neq 1$ olur [94].

4.5. Joux Üç Taraflı Anahtar Değişim Protokolü

Joux anahtar değişimi protokolü, kriptografide çift doğrusal eşleme şemasının kullanıldığı ilk uygulama olarak bilinir. Bu anahtar değişim protokolünde, üç tarafın ortak anahtarlarının güvenli olmayan bir kanal aracılığıyla tek turlu iletişimle güvenli bir şekilde oluşturulması sağlanır. Bu özelliğinden dolayı bu anahtar değişim protokolü, üç taraflı tek turlu anahtar değişim protokolü olarak da adlandırılır [96]. Joux'un anahtar değişim protokolü Diffie-Hellman anahtar değişim protokolünü temel olarak almakla birlikte bu üç taraf ancak çift doğrusal eşleme yardımıyla tek turda ortak bir anahtar oluşturabilir [86]. Şekil 4.1'de Joux üç taraflı anahtar değişim protokolü gösterilmiştir.



Şekil 4.1. Joux üç taraflı anahtar değişim protokolü

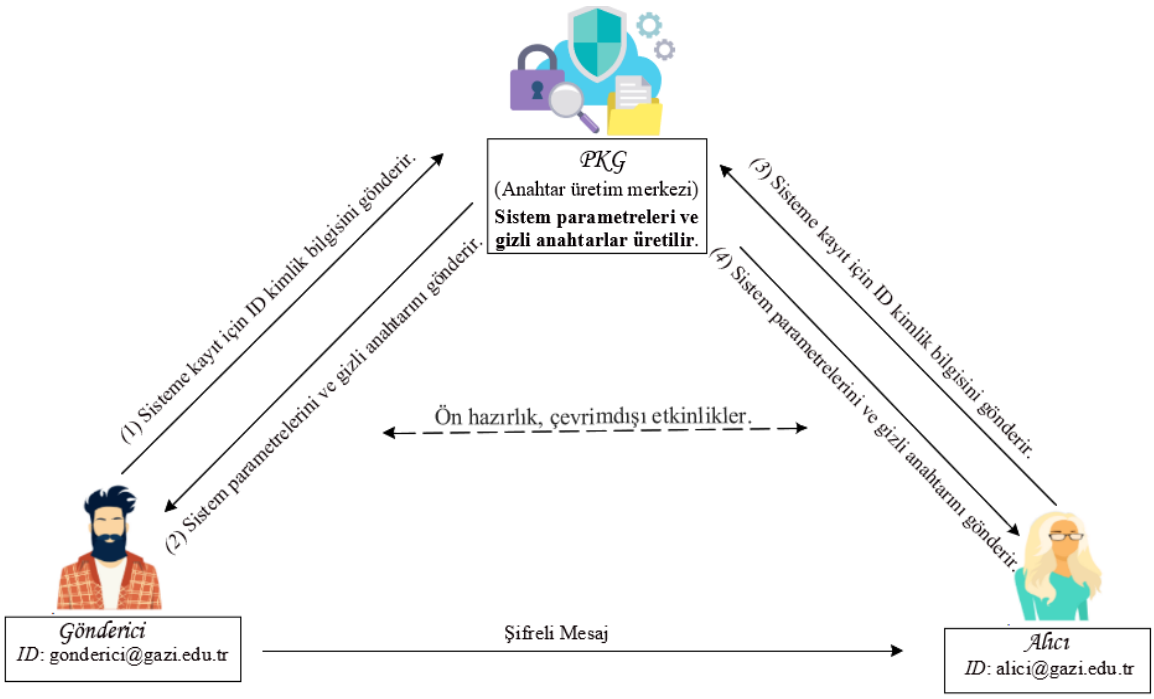
$\mathcal{A}$, $\mathcal{B}$ ve C taraflarının çift doğrusal eşleme kullanarak Joux anahtar değişim protokolünü nasıl gerçekleştirdiği aşağıdaki adımlarla açıklanmıştır.

1. Joux anahtar değişim protokolünün tasarımı için G_1 ve G_2, p mertebesinde sırasıyla toplamsal değişmeli $(G_1,+)$ ve çarpımsal değişmeli $(G_2,\cdot)$ birer grup, $e: G_1 \times G_1 \to G_2$ çift doğrusal eşleme ve $P \in G_1$ olsun.
2. $\mathcal{A}$ tarafından rasgele gizli $a \in \mathbb{Z}_p^*$ tamsayısı seçilir ve $a \cdot P$ skaler çarpımı ile açık anahtar hesaplanarak $\mathcal{B}$ ve $\mathcal{C}$ taraflarına gönderilir.
3. $\mathcal{B}$ tarafından rasgele gizli $b \in \mathbb{Z}_p^*$ tamsayısı seçilir ve $b \cdot P$ skaler çarpımı ile açık anahtar hesaplanarak $\mathcal{A}$ ve $\mathcal{C}$ taraflarına gönderilir.
4. $\mathcal{C}$ tarafından rasgele gizli $c \in \mathbb{Z}_p^*$ tamsayısı seçilir ve $c \cdot P$ skaler çarpımı ile açık anahtar hesaplanarak $\mathcal{A}$ ve $\mathcal{B}$ taraflarına gönderilir.
5. $\mathcal{A}$ tarafı, $\mathcal{B}$ ve $\mathcal{C}$ taraflarından gönderilen $b \cdot P$ ve $c \cdot P$ açık anahtarlarını kullanıp çift doğrusal eşlemenin özelliğinden faydalanarak $K = e(b \cdot P, c \cdot P)^a = e(P,P)^{abc}$ hesaplaması ile *K* ortak anahtarını elde eder.
6. $\mathcal{B}$ tarafı, $\mathcal{A}$ ve $\mathcal{C}$ taraflarından gönderilen $a \cdot P$ ve $c \cdot P$ açık anahtarlarını kullanıp çift doğrusal eşlemenin özelliğinden faydalanarak $K = e(a \cdot P, c \cdot P)^b = e(P,P)^{abc}$ hesaplaması ile *K* ortak anahtarını elde eder.
7. $\mathcal{C}$ tarafı, $\mathcal{A}$ ve $\mathcal{B}$ taraflarından gönderilen $a \cdot P$ ve $b \cdot P$ açık anahtarlarını kullanıp çift doğrusal eşlemenin özelliğinden faydalanarak $K = e(a \cdot P, b \cdot P)^c = e(P,P)^{abc}$ hesaplaması ile *K* ortak anahtarını elde eder.

4.6. Kimlik Tabanlı Kriptografi

Kimlik tabanlı kriptografi (Identity Based Cryptography; IBC), gizli mesaj göndermek isteyen kullanıcının; alıcının e-posta adresi, IP adresi, kullanıcı adı gibi kendine özel kimlik bilgilerini açık anahtar olarak kullanarak bir mesajı şifrelemesini sağlayan bir açık anahtarlı kriptografi sistemidir. Alıcının önceden var olan bir açık anahtara sahip olması gereken geleneksel açık anahtar kriptografi sistemlerinin aksine IBC, önceden bir açık anahtar değişimine gerek kalmadan herkesin mesaj alıcısı olmasına izin verir. Bu kriptografi sistemi gizli anahtar üreticisi (PKG) olarak bilinen ve her kullanıcının kimliğine karşılık gelen özel anahtarları üretmekten sorumlu olan güvenilir bir otorite kavramına dayanır. PKG, bir kullanıcının kimliğine ve yalnızca PKG tarafından bilinen bir ana gizli anahtara dayalı olarak o kullanıcı için gizli anahtar üretir. Kullanıcının özel anahtarı daha

sonra kullanıcıya güvenli bir şekilde teslim edilir. Bu sayede geleneksel açık anahtar temelli kriptografide açık anahtarların dağıtımını ve doğrulanmasını yönetmek için kullanılan bir açık anahtar altyapısı (PKI) ihtiyacı ortadan kalkar. Dikkat edilirse kimlik tabanlı kriptografi sisteminde açık anahtar ayrı olarak üretilmez ve dağıtılmaz, bunun yerine kullanıcının kimliğinden üretilir. PKG, gizli anahtarları kullanıcılara güvenli bir şekilde dağıtmaktan sorumludur ve gerekirse anahtarları iptal edebilir. Bununla birlikte, özel anahtarları yönetmek için güvenilir bir otoriteye ihtiyaç duyulmakta olup bu görevi yerine getiren PKG'nin saldırıya uğraması durumunda sistemin tümünün olumsuz etkilenmesi söz konusu olabilir. Bu nedenle, kriptografi sisteminin güvenliğini sağlamak için kimlik tabanlı kriptografinin tasarımını dikkatli bir şekilde gerçekleştirmek oldukça önemlidir. Şekil 4.2'de kimlik tabanlı kriptografinin genel şeması verilmiştir.



Şekil 4.2. Kimlik tabanlı kriptografinin genel şeması

PKG, sistem parametrelerini hem göndericiye hem de alıcıya gönderir. Hem gönderici hem de alıcı, gizli anahtarlarını almak için açık anahtarlarını (e-posta adresleri, gonderici@gazi.edu.tr ve alici@gazi.edu.tr) PKG'ye gönderir. PKG gizli anahtarları oluşturur ve bunları hem göndericiye hem de alıcıya gönderir. Bir taraf alıcıya bir mesaj göndermek için kendi gizli anahtarını kullanarak bir mesajı imzalar ve bunu keyfi bir kanal üzerinden gönderir. Gizli mesajı alan alıcı, genel parametreleri ve gizli anahtarını kullanarak mesajın şifresini çözer. Bu senaryo dijital imza için de geçerlidir. Bu durumda

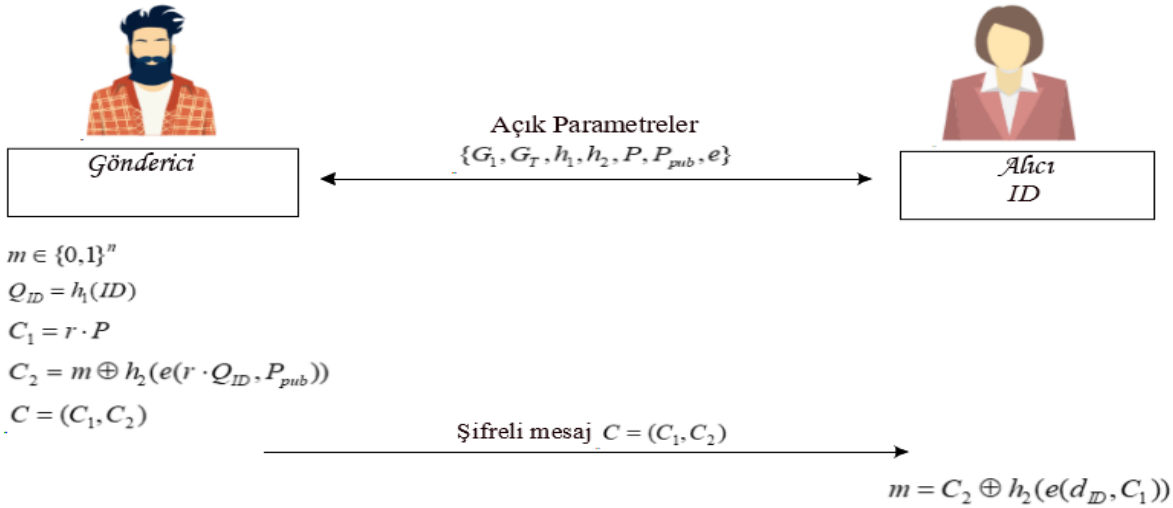
da alıcı açık parametreleri kullanarak dijital imzayı doğrulayabilir [70]. Kimlik tabanlı kriptografinin kullanımıyla, açık anahtarların üyeler arasında dağıtılmasına gerek kalmadan herkes bir imzayı doğrulayabilir (veya mesajı şifreleyebilir). Bununla birlikte, kimlik tabanlı kriptografide anahtar emanet problemi önemli bir sorun teşkil etmektedir. Anahtar emanet sorunu kullanıcıların gizli anahtarlarının PKG tarafından kullanılabilme riskini ifade eder.

Kimlik tabanlı kriptografi ilk olarak 1984 yılında Adi Shamir tarafından önerilmiştir [38] ancak belli bir süre pratik bir uygulamada kullanılmamıştır. 2001'de Boneh ve Franklin, kimlik tabanlı şifrelemenin pratik uygulamasında kullanılabilecek eliptik eğriler üzerinde Weil eşlemesine dayalı bir şema önermiş [90], daha sonra kimlik tabanlı kriptografinin kullanımı yaygınlaşmıştır. IBC, güvenli e-posta iletişimi, bulut bilişimde erişim kontrolü, geçici araç ağları, güvenli mobil mesajlaşma ve dijital imza şemaları gibi farklı uygulama alanlarında yaygın olarak kullanılmaktadır. IBC'nin en büyük avantajlarından birisi, özellikle büyük ölçekli sistemler için anahtar yönetimindeki basitlik ve esnekliktir. Örnek olarak e-posta kullanarak şifrelemede, e-posta mesajlarını yalnızca hedeflenen alıcının okuyabileceği şekilde şifrelemek için alıcının e-posta adresi açık anahtar olarak kullanılabilir. Anlık mesajlaşma, dosya paylaşımı veya video konferans gibi iki taraf arasındaki her türlü iletişimi şifrelemek için alıcının kimliği açık anahtar olarak kullanılabilir. IBC aynı zamanda kullanıcılara güvenli erişim kontrolü sağlayabilir. Kullanıcının kimliğine veya kullanıcının biyometrik verilerine dayalı olarak gizli verilerin şifrelenmesine ve daha sonra kullanıcının gizli verilere erişim sağlamasına imkân tanır. Ayrıca geleneksel açık anahtarlı kriptografi ile karşılaştırıldığında IBC, açık anahtar altyapı ihtiyacını ve sertifika yönetimi gibi altyapı bileşenlerini ortadan kaldırdığı için daha uygun maliyetli bir kriptografik sistem tasarımına imkân tanımaktadır.

4.7. Boneh-Franklin Kimlik Tabanlı Şifreleme

Boneh ve Franklin, 2001 yılında Weil eşlemesine dayalı ilk pratik kimlik tabanlı şifreleme şemasını açıklayan bir makale yayınladılar [90]. Bu çalışma kimlik tabanlı kriptografinin yaygınlaşması için bir dönüm noktası olmuştur. Önerilen şemanın güvenliği çift doğrusal Diffie-Hellman varsayımı ile rasgele kâhin modeline (Random Oracle Model; ROM) dayanır. Boneh-Franklin kimlik şemasında, alıcının kimlik bilgisinin şifrelenmesi ve şifre çözmede kullanılması, kimlik bilgisi olan *ID* bilgisinin bir eliptik eğri noktasına

gömülmesi ile sağlanır. Boneh-Franklin şeması temel şema ve tam şema olmak üzere ikiye ayrılır [75].



Şekil 4.3. Boneh-Franklin kimlik tabanlı şifreleme temel şeması

Temel bir Boneh-Franklin kimlik tabanlı şifreleme şeması dört aşamadan oluşur: (1) sistem parametrelerinin kurulum aşaması, (2) anahtar üretim aşaması, (3) şifreleme aşaması, (4) şifre çözme aşaması. Şekil 4.3'te Boneh-Franklin kimlik tabanlı temel şifreleme şeması gösterilmiştir. Bu şemada kullanılan parametreler Çizelge 4.1'de verilmiştir.

Çizelge 4.1. Boneh-Franklin kimlik tabanlı temel şifrelemede kullanılan parametreler

Parametre	Tanım
p	Sonlu asal $\mathbb{F}_p$ cisminin mertebesi
G_1	Toplamsal devirli grup
G_T	Çarpımsal devirli grup
e	Eşleme $e: G_1 \times G_1 \rightarrow G_T$
h_1, h_2	Güvenli kriptografik özet fonksiyonlar
P	Başlangıç noktası
s	Gizli anahtar $(s \in \mathbb{Z}_p^*)$
m	Mesaj $(m \in \{0,1\}^n)$
d_{ID}	Alıcının gizli anahtarı
Q_{ID}	Alıcının açık anahtarı
$C = (C_1, C_2)$	Şifreli Mesaj

Boneh-Franklin kimlik tabanlı temel şifreleme şemasının aşamaları aşağıda sunulmuştur.

Sistem parametrelerinin kurulum aşaması

1. Sistem parametrelerinin belirlenmesi ile Boneh-Franklin kimlik tabanlı şifreleme şemasının tasarımına başlanır. G_1 ve G_T, p mertebesinde sırasıyla toplamsal değişmeli $(G_1,+)$ ve çarpımsal değişmeli $(G_T,\cdot)$ birer grup, $e:G_1\times G_1\to G_T$ çift doğrusal eşleme olsun ve sonlu asal cisim üzerindeki $E/\mathbb{F}_p$ eliptik eğrisinde $P\in G_1$ başlangıç noktası seçilsin.
2. Rasgele gizli $s\in\mathbb{Z}_p^*$ tamsayısı seçilir ve P_{pub} açık anahtarı $P_{pub}=s\cdot P$ skaler çarpma işlemi ile hesaplanır.
3. $h_1:\{0,1\}^*\to G_1$ ve $h_2:G_T\to\{0,1\}^n$ olmak üzere h_1 ve h_2 kriptografik özet fonksiyonları seçilir.
4. Sistem parametrelerinden s gizli anahtarı dışındaki $\{G_1,G_T,h_1,h_2,P,P_{pub},e\}$ parametreleri açık parametrelerdir.

Anahtar üretim aşaması

1. Kullanıcılar kimlik bilgilerini kullanarak gizli mesajlaşma sistemine dahil olmak için *ID* kimlik bilgilerini güvenilir anahtar üretim merkezine gönderir.
2. *ID* bilgisi kullanılarak $Q_{ID}=h_1(ID)$ açık anahtarı ve kullanıcının gizli anahtarı $d_{ID}=s\cdot Q_{ID}$ skaler çarpma ile hesaplanır ve sisteme dahil olan kullanıcıya gönderilir.

Şifreleme aşaması

1. $m\in\{0,1\}^n$ mesajını şifreleyerek alıcısına göndermek için alıcının *ID* kimlik bilgisi kullanılarak $Q_{ID}=h_1(ID)$ işlemi ile alıcının açık anahtarı hesaplanır.
2. Rasgele $r\in\mathbb{Z}_p^*$ tamsayısı seçilir.
3. Şifreli mesaj *C*, C_1 ve C_2 olmak üzere iki parça halinde $C=(C_1,C_2)$ olarak ifade edilir.
4. $C_1=r\cdot P$ ve $C_2=m\oplus h_2(e(r\cdot Q_{ID},P_{pub}))$ işlemleri ile hesaplanır. Şifreli mesaj $C=(C_1,C_2)=(r\cdot P,m\oplus h_2(e(r\cdot Q_{ID},P_{pub})))$ alıcısına gönderilir.

Şifre çözme aşaması

1. Şifreli $C = (C_1, C_2)$ mesajını alan alıcı, mesajın şifresini, d_{ID} gizli anahtarını kullanarak $m = C_2 \oplus h_2(e(d_{ID}, C_1))$ hesaplaması ile çözer.

Şifre çözme aşamasındaki $m = C_2 \oplus h_2(e(d_{ID}, C_1))$ hesaplamasının doğrulaması aşağıdaki adımlarla gerçekleştirilir.

1. $C_2 = m \oplus h_2(e(r \cdot Q_{ID}, P_{pub}))$ eşitliğinin her iki tarafta $h_2(e(r \cdot Q_{ID}, P_{pub}))$ ile $\oplus$ (XOR) işlemi gerçekleştirilerek $m = C_2 \oplus h_2(e(r \cdot Q_{ID}, P_{pub}))$ ifadesi elde edilir.
2. $h_2(e(r \cdot Q_{ID}, P_{pub})) = h_2(e(r \cdot Q_{ID}, s \cdot P)) = h_2(e(Q_{ID}, P)^{rs})$ eşitlikleri çift doğrusal eşlemenin özellikleri kullanılarak elde edilir.
3. $h_2(e(Q_{ID}, P)^{rs}) = h_2(e(s \cdot Q_{ID}, r \cdot P))$ eşitliğinde $d_{ID} = s \cdot Q_{ID}$ ve $C_1 = r \cdot P$ değerleri yerine yazılırsa $h_2(e(s \cdot Q_{ID}, r \cdot P)) = h_2(e(d_{ID}, C_1))$ eşitliği elde edilerek hesaplamanın doğruluğu ispat edilir.

Yukarıda algoritma aşamaları verilen temel şema, seçilmiş şifreli metin saldırılarına karşı savunmasızdır. Bu güvenlik açığı Fujisaki-Okamoto dönüşümü ile ortadan kaldırılabilir. Böylece ek güvenlik parametreleri ile daha güvenli bir Boneh-Franklin şifreleme tam şeması elde edilir [75].

Boneh-Franklin kimlik tabanlı şifreleme tam şemasının aşamaları aşağıda sunulmuştur [75].

Sistem parametrelerinin kurulum aşaması

1. Boneh-Franklin kimlik tabanlı şifreleme temel şemasında bulunan $h_1 : \{0,1\}^* \to G_1$, $h_2 : G_T \to \{0,1\}^n$ kriptografik özet fonksiyonlarına ek olarak $h_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_p^*$ ve $h_4 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_p^*$ olmak üzere h_3 ve h_4 kriptografik özet fonksiyonları eklenir.
2. Sistem parametrelerinden s gizli anahtarı dışındaki $\{G_1, G_T, h_1, h_2, h_3, h_4, P, P_{pub}, e\}$ parametreleri açık parametrelerdir.

Anahtar üretim aşaması

1. Kullanıcılar kimlik bilgilerini kullanarak gizli mesajlaşma sistemine dahil olmak için *ID* kimlik bilgilerini güvenilir anahtar üretim merkezine gönderir.
2. *ID* bilgisi kullanılarak $Q_{ID} = h_1(ID)$ açık anahtarı ve kullanıcının gizli anahtarı $d_{ID} = s \cdot Q_{ID}$ skaler çarpma ile hesaplanır ve sisteme dahil olan kullanıcıya gönderilir.

Şifreleme aşaması

1. Rasgele $\lambda \in \{0,1\}^n$ değeri seçilir.
2. $m \in \{0,1\}^n$ mesajını şifreleyerek alıcısına göndermek için alıcının *ID* kimlik bilgisi kullanılarak $Q_{ID} = h_1(ID)$ işlemi ile alıcının açık anahtarı hesaplanır.
3. $r = h_3(\lambda, m)$ işlemi ile *r* hesaplanır.
4. Şifreli mesaj *C*, C_1, C_2 ve C_3 olmak üzere üç parça halinde $C = (C_1, C_2, C_3)$ olarak ifade edilir.
5. $C_1 = r \cdot P$, $C_2 = \lambda \oplus h_2(e(r \cdot Q_{ID}, P_{pub}))$ ve $C_3 = m \oplus h_4(\lambda)$ işlemleri ile hesaplanır. Şifreli mesaj $C = (C_1, C_2, C_3)$ alıcısına gönderilir.

Şifre çözme aşaması

1. Şifreli $C = (C_1, C_2, C_3)$ mesajını alan alıcı, d_{ID} gizli anahtarını kullanarak aşağıdaki adımlarla mesajın şifresini çözer ve mesajın doğruluğunu kontrol eder.
2. $\lambda = C_2 \oplus h_2(e(d_{ID}, C_1))$ hesaplanır.
3. $m = C_3 \oplus h_4(\lambda)$ hesaplaması ile mesaj elde edilir.
4. $r = h_3(\lambda, m)$ ve daha sonra $r \cdot P$ hesaplanır. Eğer $C_1 \neq r \cdot P$ ise şifreli mesaj geçersiz olarak reddedilir.

4.8. Boneh-Boyen Kimlik Tabanlı Şifreleme

D. Boneh ve X. Boyen tarafından 2004 yılında verimli bir kimlik tabanlı şifreleme şeması önerilmiştir. Boneh-Boyen kimlik tabanlı şifreleme şeması toplamsal ve çarpımsal olmak

üzere iki temel şema üzerine dayanır [97]. Bu şemada, bir eşleme sabit değeri hesaplanarak şifreleme ve şifre çözmede gönderici ve alıcı tarafından kullanılır. Boneh-Boyen kimlik tabanlı şifreleme şeması değişimli körleme ailesine ait olup Boneh-Franklin kimlik tabanlı şifreleme şemasıyla karşılaştırıldığında, kimliği eliptik eğri üzerindeki bir nokta yerine bir tamsayıya eşler. Bununla birlikte bu şema çift parçalı gizli anahtar kullanır. Bir düz metin mesajını şifrelemek için kullanılan sabit eşleme değeri, gönderen tarafından bir Boneh-Boyen kimlik tabanlı şifreleme şemasının açık parametreleri kullanılarak hesaplanır ve elde edilen şifreli mesajın alıcısı, sabit eşleme oranını hesaplayarak mesajın şifresini çözer. Bu bölümde Boneh-Boyen toplamsal şemasının algoritması verilmiştir. Algoritmada kullanılan parametreler Çizelge 4.2'de tanıtılmıştır. Bununla birlikte algoritma aşamasından sonra Şekil 4.4'te Boneh-Boyen kimlik tabanlı şifreleme şeması gösterilmiştir [75].

Çizelge 4.2. Boneh-Boyen kimlik tabanlı şifrelemede kullanılan parametreler

Parametre	Tanım
p	Sonlu asal $\mathbb{F}_p$ cisminin mertebesi
G_1	Toplamsal devirli grup
G_T	Çarpımsal devirli grup
e	Eşleme $e: G_1 \times G_1 \rightarrow G_T$
h_1, h_2	Güvenli kriptografik özet fonksiyonlar
P	Başlangıç noktası
α, β, γ	Gizli tamsayılar ($\alpha, \beta, \gamma \in \mathbb{Z}_p^*$)
s	Gizli anahtar ($s \in \mathbb{Z}_p^*$)
m	Mesaj $(m \in \{0,1\}^n)$
$d_{ID} = \{d_0, d_1\}$	Alıcının gizli anahtarı
Q_{ID}	Alıcının açık anahtarı
ϑ	$e(P,P)^{\alpha\beta}$ sabit değeri
$C = (c, C_0, C_1)$	Şifreli mesaj
$\{G_1, G_T, h_1, h_2, P_1, P_2, P_3, e, \vartheta\}$	Açık parametreler

Genel Boneh-Boyen kimlik tabanlı şifreleme şeması dört aşamadan oluşur: (1) sistem parametrelerinin kurulum aşaması, (2) anahtar üretim aşaması, (3) şifreleme aşaması, (4)

şifre çözme aşaması. Boneh-Boyen kimlik tabanlı şifreleme şemasının bir alt grubu olan toplamsal şemanın algoritma aşamaları aşağıda sunulmuştur [75].

Sistem parametrelerinin kurulum aşaması

1. Sistem parametrelerinin belirlenmesi ile Boneh-Boyen kimlik tabanlı şifreleme toplamsal şemasının tasarımına başlanır. G_1 ve G_T, p mertebesinde sırasıyla toplamsal değişmeli $(G_1,+)$ ve çarpımsal değişmeli $(G_T,\cdot)$ birer grup, $e:G_1\times G_1\to G_T$ çift doğrusal eşleme olsun ve sonlu asal cisim üzerindeki $E/\mathbb{F}_p$ eliptik eğrisinde $P\in G_1$ başlangıç noktası seçilsin.
2. Rasgele gizli $\alpha,\beta,\gamma\in\mathbb{Z}_p^*$ tamsayıları seçilir ve α,β,γ gizli anahtarları kullanılarak $P_1=\alpha\cdot P$, $P_2=\beta\cdot P$ ve $P_3=\gamma\cdot P$ skaler çarpma işlemleri ile P_1,P_2,P_3 açık anahtarları hesaplanır.
3. $h_1:\{0,1\}^*\to\mathbb{Z}_p$ ve $h_2:G_T\to\{0,1\}^n$ olmak üzere h_1 ve h_2 kriptografik özet fonksiyonları seçilir.
4. Boneh-Boyen şemasında $\vartheta=e(\alpha\cdot P,\beta\cdot P)=e(P,P)^{\alpha\beta}$ sabit değeri hesaplanır. Bu değer anahtar üretim merkezi tarafından gerçekleştirilerek açık parametre olarak dağıtılır. ϑ açık parametre olarak dağıtıldığından P parametresinin paylaşılması gerekmez.
5. α,β,γ gizli anahtarları dışındaki $\{G_1,G_T,h_1,h_2,P_1,P_2,P_3,e,\vartheta\}$ parametreleri açık parametrelerdir.

Anahtar üretim aşaması

1. Kullanıcılar kimlik bilgilerini kullanarak gizli mesajlaşma sistemine dahil olmak için *ID* kimlik bilgilerini güvenilir anahtar üretim merkezine gönderir.
2. *ID* kimlik bilgisi kullanılarak $Q_{ID}=h_1(ID)$ hesaplaması ile $Q_{ID}\in\mathbb{Z}_p^*$ belirlenir. Daha sonra $r\in\mathbb{Z}_p^*$ tamsayısı seçilir ve kullanıcının gizli anahtarı $d_{ID}=\{d_0,d_1\}$ şeklinde iki parça halinde hesaplanır. Burada $d_0=\{Q_{ID}r\cdot P_1+\alpha\cdot P_2+\gamma\cdot P_3\}$ ve $d_1=r\cdot P$ işlemleri ile hesaplanır.

Şifreleme aşaması

1. $m \in \{0,1\}^n$ mesajını şifreleyerek alıcısına göndermek için alıcının *ID* kimlik bilgisi kullanılarak $Q_{ID} = h_1(ID)$ işlemi ile $Q_{ID} \in \mathbb{Z}_p^*$ hesaplanır.
2. Rasgele $s \in \mathbb{Z}_p^*$ seçilir ve kullanıcılara gönderilen açık parametrelerden ϑ kullanılarak $k = \vartheta^s$ hesaplanır.
3. Şifreli mesaj *C;* c, C_0 ve C_1 olmak üzere üç parça halinde $C = (c, C_0, C_1)$ olarak ifade edilir.
4. $c = m \oplus h_2(k)$, $C_0 = s \cdot P$ ve $C_1 = Q_{ID} s \cdot P_1 + s \cdot P_3$ işlemleri ile hesaplanır. Şifreli mesaj $C = (c, C_0, C_1)$ alıcısına gönderilir.

Şifre çözme aşaması

1. Şifreli $C = (c, C_0, C_1)$ mesajını alan alıcı, $d_{ID} = \{d_0, d_1\}$ gizli anahtarını kullanarak aşağıdaki adımlarla mesajın şifresini çözer.
2. Çift doğrusal eşlemenin özelliği kullanılarak $k = \dfrac{e(C_0, d_0)}{e(C_1, d_1)}$ hesaplanır.
3. Son olarak $m = c \oplus h_2(k)$ hesaplaması ile mesaj elde edilir.

Şifre çözme aşamasındaki $k = \dfrac{e(C_0, d_0)}{e(C_1, d_1)}$ hesaplamasının doğrulaması aşağıdaki adımlarla gerçekleştirilir.

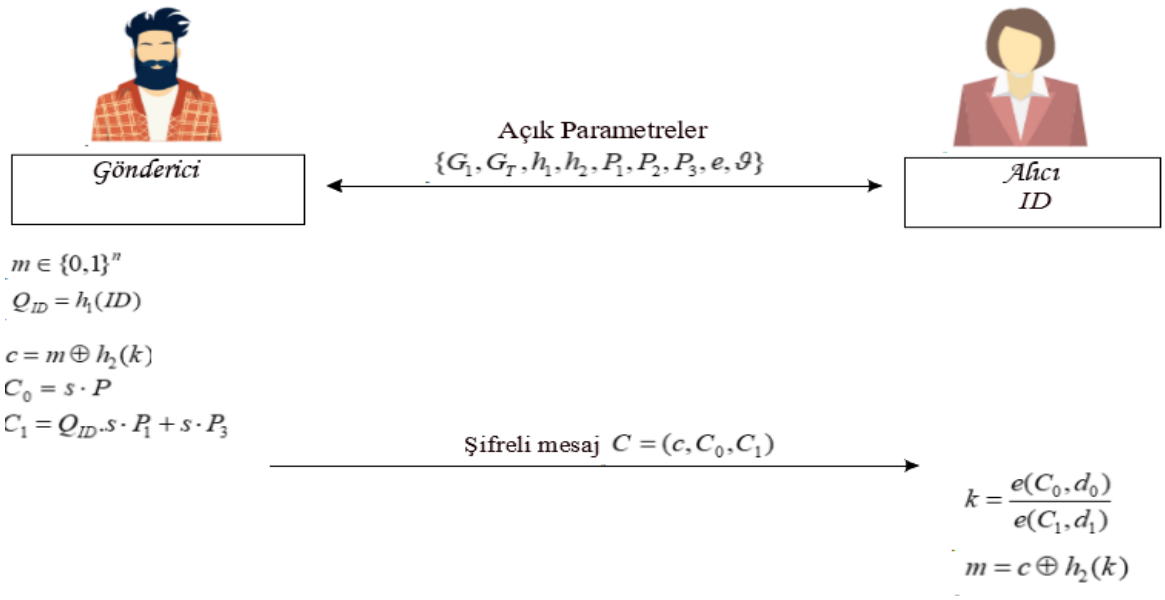
1. $e(C_0, d_0) = e(s \cdot P, Q_{ID} r \cdot P_1 + \alpha \cdot P_2 + r \cdot P_3)$ eşitliği çift doğrusal eşlemenin özellikleri kullanılarak Eş. 4.10'daki gibi düzenlenir.

$$\begin{aligned} e(C_0, d_0) &= e(s \cdot P, Q_{ID} r \cdot P_1)\ e(s \cdot P, \alpha \cdot P_2)\ e(s \cdot P, r \cdot P_3) \\ &= e(s \cdot P, Q_{ID} r\alpha \cdot P)\ e(s \cdot P, \alpha\beta \cdot P)\ e(s \cdot P, r\gamma \cdot P) \\ &= e(P, P)^{sQ_{ID} r\alpha}\ e(P, P)^{s\alpha\beta}\ e(P, P)^{sr\gamma} \end{aligned} \tag{4.10}$$

2. $e(C_1, d_1) = e(Q_{ID} s \cdot P_1 + s \cdot P_3, r \cdot P)$ eşitliği çift doğrusal eşlemenin özellikleri kullanılarak aşağıdaki gibi düzenlenir.

$$
\begin{aligned}
e(C_1, d_1) &= e(Q_{ID} s \cdot P_1 + s \cdot P_3, r \cdot P) \\
&= e(Q_{ID} s\alpha \cdot P + s\gamma \cdot P, r \cdot P) \\
&= e(Q_{ID} s\alpha \cdot P, r \cdot P)\ e(s\gamma \cdot P, r \cdot P) \\
&= e(P,P)^{Q_{ID} s\alpha r}\ e(P,P)^{s\gamma r}
\end{aligned}
\tag{4.11}
$$

3. $k = \dfrac{e(C_0, d_0)}{e(C_1, d_1)} = \dfrac{e(P,P)^{sQ_{ID} r\alpha}\, e(P,P)^{s\alpha\beta}\, e(P,P)^{sr\gamma}}{e(P,P)^{Q_{ID} s\alpha r}\, e(P,P)^{sr\gamma}} = e(P,P)^{s\alpha\beta} = \vartheta^s$ eşitliği elde edilir.



Şekil 4.4. Boneh-Boyen kimlik tabanlı şifreleme toplamsal şeması

4.9. BLS İmza Şeması

Eşlemeye dayanan bir dijital imza şeması olan Boneh-Lynn-Shacham (BLS) imza şeması eşleme tabanlı kriptografinin ilk imza uygulamalarındandır [98]. Bu imza şeması oldukça basit bir eşleme mantığına dayanmaktadır. BLS imza şeması; (1) anahtar üretimi, (2) imzalama ve (3) imza doğrulama olmak üzere üç aşamadan oluşmakta olup aşağıda bu aşamalar verilmiştir.

BLS imza şemasının tasarımında öncelikle sistem parametreleri belirlenir. G_1 ve G_2, p mertebesinde toplamsal değişmeli, G_T ise p mertebesinde çarpımsal değişmeli bir grup, $e: G_1 \times G_2 \to G_T$ çift doğrusal eşleme, $H: \{0,1\}^* \to G_1$ özet fonksiyonu ve $P \in G_2$ olsun.

Anahtar üretim aşaması

1. İmzacı *S*, rasgele $x \in \mathbb{Z}_p^*$ gizli anahtarını seçer ve $P_{pub} = x \cdot P$ işlemi ile açık anahtarı hesaplar ($P_{pub} \in G_2$).

İmzalama aşaması

1. İmzacı *S* tarafından gönderilecek *m* mesajının özet değeri $H(m) = h$ işlemi ile belirlenir. Daha sonra *x* gizli anahtarı kullanılarak aşağıdaki işlemle *m* mesajının imzası hesaplanır.

$$\sigma = x \cdot H(m) = x \cdot h \in G_1 \tag{4.12}$$

2. İmzacı *S*, mesaj ve imza ile birlikte diğer açık parametreleri $\{\sigma, m, H, P, P_{pub}\}$ mesajın alıcısına gönderir.

İmza doğrulama aşaması

1. Alınan mesaj ve imzanın doğrulanması için öncelikle mesajın özet değeri $H(m) = h$ işlemi ile hesaplanır. Daha sonra Eş. 4.13'teki eşitlik kontrol edilir. Eğer eşitlik sağlanırsa imza doğrulanır, aksi durumda imza reddedilir.

$$e(\sigma, P) \stackrel{?}{=} e(h, P_{pub}) \tag{4.13}$$

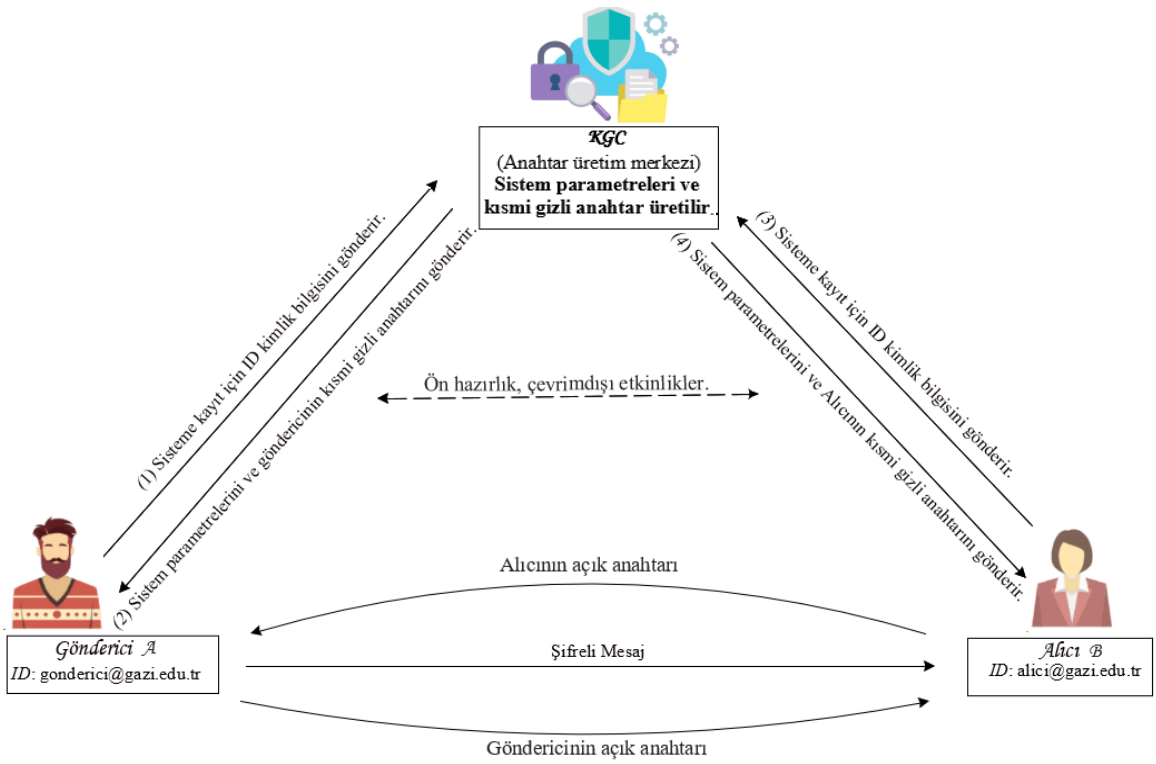
İmza doğrulama aşamasında kurulan Eş. 4.13 ifadesinin doğruluğu, eşlemenin sağlamış olduğu özelliğe dayanılarak aşağıdaki gibi ispatlanır.

$$e(\sigma, P) = e(x \cdot h, P) = e(h, P)^x = e(h, x \cdot P) = e(h, P_{pub}) \tag{4.14}$$

4.10. Sertifikasız Kriptografi

Sertifikasız açık anahtarlı kriptografi (Certificateless Public Key Cryptography; CL-PKC), geleneksel açık anahtarlı kriptografide bulunan açık anahtar altyapısı ve sertifika

yönetiminin ağır yükünü ve kimlik tabanlı kriptografide bulunan anahtar emanet sorununu ortadan kaldırır. 2003 yılında Al-Riyami ve Paterson, kimlik tabanlı kriptografide mevcut olan anahtar emanet problemine çözüm olarak sertifikasız açık anahtarlı kriptografiyi önermişlerdir [44]. Sertifikasız kriptografide, sisteme dahil olmak isteyen kullanıcılar kimlik bilgilerini anahtar üretim merkezine (Key Generation Center; KGC) gönderir ve bu merkez kısmi bir gizli anahtar üreterek kullanıcıya geri gönderir. Tam gizli anahtar, karşılık gelen kısmi gizli anahtar ve kullanıcı tarafından seçilen rasgele bir gizli değer kullanılarak oluşturulur. Bu sayede, KGC; kullanıcının tam gizli anahtarını saklayamaz ve elde edemez. Oldukça önemli olan bu detay sayesinde kimlik tabanlı kriptografide bulunan anahtar emanet problemi çözülmüş olur. Şekil 4.5'te sertifikasız açık anahtarlı kriptografinin genel şeması verilmiştir.



Şekil 4.5. Sertifikasız açık anahtarlı kriptografinin genel şeması

KGC, açık sistem parametrelerini, kriptografi sistemine dahil olan alıcıya ve göndericiye teslim eder. Kullanıcılar tarafından KGC'ye gönderilen kimlik bilgileri kullanılarak kısmi gizli anahtarlar üretilir ve kullanıcılara gönderilir. Daha sonra hem alıcı hem de gönderici, kendi açık ve gizli anahtar çiftlerini oluşturur ve açık anahtarlarını birbirleriyle paylaşır. $\mathcal{B}$'ye bir mesaj göndermek için $\mathcal{A}$, gizli anahtarını kullanarak bir mesajı imzalar ve ortak

kanal aracılığıyla $\mathcal{B}$'ye gönderir. $\mathcal{A}$'dan imzalı mesajı aldıktan sonra $\mathcal{B}$, $\mathcal{A}$'nın zaten biliyor olduğu açık anahtarı aracılığıyla bunu doğrular. Benzer şekilde $\mathcal{B}$, bir mesajı imzalayıp $\mathcal{A}$'ya gönderebilir. $\mathcal{A}$ kullanıcısı benzer şekilde imzayı doğrulayabilir. Tüm süreçte KGC hem $\mathcal{A}$ hem de $\mathcal{B}$'nin gizli anahtarlarını bilmez. Bu sayede, anahtar emanet sorunu çözülmüş olur.

Al-Riyami ve Paterson [44] tarafından tanımlanan sertifikasız kriptografi şeması (1) kurulum aşaması, (2) kısmi anahtar üretim aşaması, (3) gizli değerin üretim aşaması, (4) gizli anahtarın üretim aşaması, (5) açık anahtarın üretim aşaması, (6) şifreleme aşaması, (7) şifre çözme aşaması olmak üzere yedi aşamadan oluşmaktadır [45]. Aşağıda bu aşamalar sırasıyla incelenmiştir.

Kurulum aşaması

Sertifikasız kriptografinin kurulum aşamasında şifreleme sisteminde kullanılacak sistem parametreleri belirlenir. KGC; sistem parametrelerini, açık anahtarı ve kendi gizli anahtarını oluşturmak için bir güvenlik parametresi olan k'yı girdi olarak alarak alır ve kurulum aşamasını başlatır. Bu aşamada üretilen sistem parametreleri ve açık anahtar herkese yayınlanır ancak gizli anahtar kimseyle paylaşılmaz.

Kısmi anahtar üretim aşaması

KGC; bir kullanıcının kimliğini, kendi gizli anahtarını ve sistem parametrelerini girdi olarak alan ve kısmi gizli anahtarı üreten bir algoritmayı çalıştırır. Bu aşamada sisteme kayıt olan her kullanıcı için kısmi gizli anahtar üretilerek güvenli bir kanal aracılığıyla kullanıcıya gönderilir.

Gizli değerin üretim aşaması

Sistem parametrelerinin belirlenmesinden sonra şifreleme sisteminde kullanılmak üzere gizli bir değer üretmek amacıyla kullanıcı tarafından bir algoritma çalıştırılır.

Gizli anahtarın üretim aşaması

Bu algoritma; sistem parametrelerinin, kullanıcının kısmi gizli anahtarı ve bir kullanıcının gizli değerinin girilmesi üzerine kullanıcının tam gizli anahtarını oluşturmak için kullanıcı tarafından çalıştırılır.

Açık anahtarın üretim aşaması

Bu algoritma; sistem parametreleri ve kullanıcının gizli değerinin girilmesi üzerine kullanıcının açık anahtarını oluşturmak için kullanıcı tarafından çalıştırılır.

Şifreleme aşaması

Bu aşamada sistem parametrelerinin, kimlik bilgilerinin ve açık anahtar bilgilerinin kullanılması ile bir mesaj şifrelenerek karşı tarafa gönderilir.

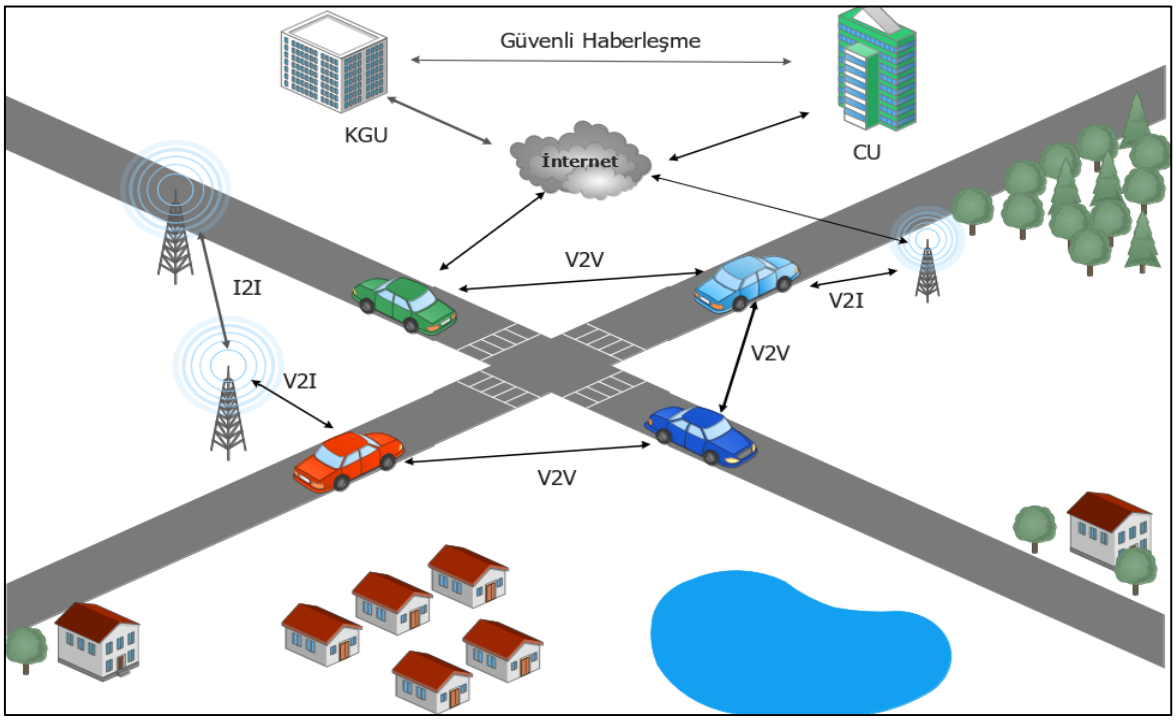
Şifre çözme aşaması

Bu aşamada sistem parametreleri ve kullanıcının gizli anahtarı kullanılarak mesajın şifresi çözülür.

5. KİMLİK TABANLI SERTİFİKASIZ KRİPTOGRAFİ SİSTEMİNİN TASARIMI

Bu bölümde eliptik eğri kriptografisi kullanılarak araçların interneti (IoV) için koşullu gizliliği koruyan kimlik tabanlı verimli ve eşlemesiz yeni bir sertifikasız kimlik doğrulama şeması tasarlanmıştır. Önerilen şemada kullanılan temel kavramlar ile birlikte gerçekleştirilen tasarım algoritması da bu bölümde sunulmuştur. Önerilen şemanın verimliliğini değerlendirmek için mevcut diğer şemalar ile performans karşılaştırması yapılmıştır. Ayrıca bu bölümde bir kriptografi sistemi için oldukça önemli olan güvenlik analizleri gerçekleştirilmiştir.

Önerilen şema, koşullu gizliliği koruyan bir kimlik doğrulama şeması olup kimlik tabanlı sertifikasız kriptografi sistemine dayanmaktadır. Ayrıca önerilen şemanın verimliliğini artırmak için eşleme kullanmadan eliptik eğriler üzerinde tasarım gerçekleştirilmiştir. İlk olarak önerilen modelin temelini oluşturan araçların interneti (IoV) ağ mimarisi ve bileşenleri hakkında bilgi verilmiştir. Sonrasında eşleme olmadan eliptik eğriler kullanılarak kimlik tabanlı sertifikasız bir kriptografi sisteminin tasarım aşamaları sunulmuştur.



Şekil 5.1. IoV ağ mimarisi

5.1. Önerilen Modelin Ağ Mimarisi

Önerilen modelde IoV ağ mimarisi; merkezi birim (CU), anahtar üretim birimi (KGU), yol kenarı birimi (RSU) ve araç olmak üzere dört bileşenden oluşmaktadır. Bu birimler arasında araçtan araca (V2V) ve araç ile RSU (V2R) arasındaki iletişim özel kısa mesafeli iletişim (DSRC) teknolojisi aracılığıyla gerçekleştirilmektedir. KGU ile CU arasındaki iletişim ise güvenli bir kanal üzerinden sağlanmaktadır. Önerilen modeldeki IoV ağ mimarisindeki bileşenler aşağıda açıklanmıştır. Şekil 5.1'de IoV ağ mimarisi verilmiştir.

Merkezi birim (CU)

CU, güvenilir bir otorite olarak araçların IoV ağına kaydedilmesinden ve kısmi anonim kimlik oluşturulmasından da sorumlu bir otoritedir. Araçların ve RSU'ların gerçek kimlikleri yalnızca CU tarafından bilinir ve sistemin güvenliğini tehlikeye düşürecek kötü niyetli bir olay tespit edildiğinde kullanıcıların gerçek kimliği ortaya çıkarılır. Araçlar ve RSU'lar, CU ile periyodik olarak iletişime geçerek kimlik bilgilerini güncelleyebilirler. Kullanıcıların kötü niyetli davranışı daha önce tespit edilmişse, kimlik güncellemeleri CU tarafından gerçekleştirilmez [34].

Anahtar üretim birimi (KGU)

Önerilen modelimizde KGU aynı zamanda güvenilir bir otoritedir. CU ile birlikte çalışarak araçların ve RSU'ların kısmi gizli anahtarlarını (PPK) üretir. Ayrıca araçların ve RSU'ların kısmi anonim kimliklerini oluşturmasında CU'ya yardımcı olur. Bu anonim kimlik kısmidir ve KGU bu kısmi kimlikten yola çıkarak gerçek kimliğe ulaşamaz. Gerçek kimliğe sadece CU tarafından ulaşılır[34].

Yol Kenarı Birimi (RSU)

RSU'lar yol kenarında bulunarak araçlarla V2R iletişimini gerçekleştirirler. Bu birimler trafikte belirli bölgelere yerleştirilir ve araçlara bölge ve trafik hakkında bilgi verir. RSU'lar araçlardan topladıkları birden fazla mesajı aynı anda diğer araçlara gönderebilir [34].

Araç

Araç içerisinde yerleşik üniteler (OBU) ve kurcalamaya karşı dayanaklı cihaz (TPD) bulunmakta olup bu bileşenler haberleşme altyapısı için oldukça önemlidir. Önerilen modelimizde araçlar diğer araçlarla ve yol birimleri ile haberleşebilir. Araçlar, trafik ile ilgili ürettikleri mesajları diğer araçlara ve RSU'ya imzalayarak gönderir. Eğer araç ağ sistemimize kayıtlı ise gönderilen mesajın doğruluğu alıcı araç veya RSU tarafından kontrol edilir [34].

5.2. Önerilen Kimlik Doğrulama Şemasının Algoritması

Önerilen kimlik doğrulama şemasının algoritması: (1) kurulum aşaması, (2) kayıt aşaması, (3) kısmi anonim kimlik oluşturma ve kısmi gizli anahtar oluşturma aşaması, (4) imza aşaması ve (5) doğrulama aşaması olmak üzere beş aşamadan oluşmaktadır. Ayrıca doğrulama aşamasından sonra çoklu mesajlar için büyük bir kolaylık sağlayan mesajların toplu doğrulanması aşaması sunulmuştur. Önerilen algoritmanın aşamalarına geçmeden önce Çizelge 5.1'de kriptografi sisteminde kullanılan parametreler ve tanımları verilmiştir. Sonrasında algoritmanın aşamaları sırasıyla açıklanmıştır.

Çizelge 5.1. Önerilen şemanın algoritmasında kullanılan parametreler ve tanımları

Parametre	Tanım
p ve q	Yeterince büyük iki asal sayı
$E / \mathbb{F}_p$	Sonlu asal cisim üzerinde eliptik eğri
G_1 ve G_T	Toplamsal ve çarpımsal devirli grup
V_i ve R_i	i'nci araç ve RSU
$VKA_{id,i}$	i'nci aracın kısmi anonim kimliği
$RKA_{id,i}$	i'nci RSU'nun kısmi anonim kimliği
$V_{gizli,i}$ ve $R_{gizli,i}$	i'nci araç ve RSU'nun gizli anahtarı
c ve C_{pub}	CU'nun gizli anahtarı ve açık anahtarı
k ve K_{pub}	KGU'nun gizli anahtarı ve açık anahtarı
T	Zaman bilgisi

Kurulum aşaması

Sistem parametrelerinin seçimi, etkili ve güvenli bir sertifikasız koşullu gizliliği sağlayan kimlik doğrulama şemasının (CL-CPPA) tasarımında en önemli adımıdır. Bu aşama aşağıdaki gibi gerçekleştirilir.

1. Kriptografi sisteminin parametrelerini belirlemek için CU ve KGU sonlu cisim üzerindeki $E/\mathbb{F}_p$ eliptik eğrisi üzerinde anlaşarak Eş. 3.9'da belirtilen eliptik eğri eşitliği için gerekli olan *a* ve *b* değerlerini seçer.
2. h_1, h_2, h_3 ve h_4 olmak üzere dört adet özet fonksiyon seçilir. $h_1 : G_1 \times \{0,1\}^* \to \mathbb{Z}_q^*$ $h_2 : G_1 \to \mathbb{Z}_q^*$, $h_3 : G_1 \to \mathbb{Z}_q^*$ ve $h_4 : \{0,1\}^* \times \{0,1\}^* \times G_1 \times G_1 \times \{0,1\}^* \to \mathbb{Z}_q^*$ özelliklerini sağlar.
3. CU gizli anahtarını seçer ($c \in \mathbb{Z}_q^*$) ve P başlangıç noktası olmak üzere açık anahtarını $C_{pub} = c \cdot P$ işlemi ile hesaplar.
4. KGU gizli anahtarını seçer ($k \in \mathbb{Z}_q^*$) ve açık anahtarını $K_{pub} = k \cdot P$ işlemi ile hesaplar.
5. CU ve KGU, sistem ağında $\{E, P, p, q, C_{pub}, K_{pub}, h_3, h_4\}$ açık parametrelerini yayınlar.
6. h_1 ve h_2 özet fonksiyonları yalnızca CU ve KGU tarafından kullanılır, bu nedenle yayınlanmazlar. CU ve KGU, kısmi anonim bir kimlik oluşturmak için h_1 ve h_2 özet fonksiyonlarını kullanır.

Kayıt aşaması

Önerilen kimlik doğrulama şemasının bu aşamasında araçlar ve RSU'lar, plaka numaraları ve cihaz numaraları gibi özel kimlikleri aracılığıyla CU'ya kaydolur. Bu numaralar belirli bir sıra ile yazılır, örneğin satın alındıkları yere göre kodlanabilirler. Böylece araç ve RSU bilgileri veri tabanında daha hızlı bulunabilir. Araçların plaka numaraları ve RSU'ların cihaz numaraları özeldir. Bu aşama aşağıdaki gibi gerçekleştirilir.

1. CU, sisteme kayıt olan V_i aracının gerçek kimliği $V_{id,i}$ ile c gizli anahtarını kullanarak $\zeta = V_{id,i} \oplus h_1(C_{pub} \| c)$ işlemi ile sabit ζ değerini hesaplar.

2. CU hesaplamış olduğu sabit ζ değerini V_i aracının kurcalamaya karşı korumalı cihazı olan TPD'ye yükler.
3. RSU R_i için de yukarıdaki işlemler gerçekleştirilir. CU, R_i'nin kimliği olan cihaz numarası $D_{id,i}$ ile c gizli anahtarını kullanarak $\Re = D_{id,i} \oplus h_1(C_{pub} \| c)$ işlemi ile sabit $\Re$ değerini hesaplar.
4. CU hesaplamış olduğu sabit $\Re$ değerini RSU R_i'nin kurcalamaya karşı korumalı cihazı olan TPD'ye yükler.

Kısmi anonim kimlik oluşturma ve kısmi gizli anahtar oluşturma aşaması

Önerilen kimlik doğrulama şemasında sisteme kayıtlı araçların ve RSU'ların gerçek kimliklerinin korunması gizlilik ve güvenlik açısından oldukça önemlidir. Bu aşamada, KGU tarafından kısmi anonim kimlik ve kısmi gizli anahtar üretimi gerçekleştirilir. Aracın kısmi anonim kimliği $VKA_{id,i}$ ve RSU'nun kısmi anonim kimliği $RKA_{id,i}$, merkezi otorite tarafından izlenebilirliğini sağlamak için yalnızca CU tarafından bilinir. Bu aşama, CU'da yapılan bir güncelleme ile düzenli aralıklarla tekrarlanır. Bu güncelleme ile kısmi anonim kimlik ve kısmi gizli anahtar araçlara yeniden yüklenir. Bu sayede her mesaj gönderiminde KGU ve CU'dan bu bilgi talep edilmediği için ağ sistemi aşırı yüklenmemiş olur. Bu aşama, araçlar ve RSU'lar için aşağıdaki adımlarla gerçekleştirilir.

Araç için algoritma adımları

1. V_i aracı, sabit ζ değerini güvenli bir kanal aracılığıyla KGU'ya gönderir.
2. KGU, CU tarafından gönderilen araç veri tabanından sabit ζ değerini kontrol eder. Kayıtlı araç veri tabanı CU tarafından KGU ile paylaşılmakta olup bu veri tabanında cezalı ve sisteme kayıtlı olmayan araçlar yer almaz. ζ, veri tabanında bulunursa, KGU hesaplamalarına devam eder, aksi takdirde aracın başvurusunu reddeder.
3. V_i aracının kısmi anonim kimliği KGU tarafından gizli anahtarı k kullanılarak $VKA_{id,i} = \zeta \oplus h_2(k \cdot C_{pub})$ işlemi ile hesaplanır.
4. KGU, $\lambda \in \mathbb{Z}_q^*$ gizli anahtarını seçer.

5. KGU, V_i aracının kısmi gizli anahtarını $V_{ppk,i} = k + \lambda h_1(VKA_{id,i} \| k \| K_{pub})$ işlemi ile hesaplar.
6. Son olarak KGU, güvenli bir kanal aracılığıyla $\{V_{ppk,i}, VKA_{id,i}\}$ parametrelerini gönderir.

RSU için algoritma adımları

1. RSU R_i, sabit $\Re$ değerini güvenli bir kanal aracılığıyla KGU'ya gönderir.
2. KGU, CU tarafından gönderilen cihaz veri tabanından sabit $\Re$ değerini kontrol eder. Kayıtlı RSU'ların tutulduğu veri tabanı CU tarafından KGU ile paylaşılmakta olup bu veri tabanında cezalı ve sisteme kayıtlı olmayan RSU'lar yer almaz. $\Re$ veri tabanında bulunursa, KGU hesaplamalarına devam eder, aksi takdirde RSU'nun başvurusunu reddeder.
3. R_i'nin kısmi anonim kimliği KGU tarafından gizli anahtarı k kullanılarak $RKA_{id,i} = \Re \oplus h_2(k \cdot C_{pub})$ işlemi ile hesaplanır.
4. KGU, $\varphi \in \mathbb{Z}_q^*$ gizli anahtarını seçer.
5. KGU, R_i'nin kısmi gizli anahtarını $R_{ppk,i} = k + \varphi h_1(RKA_{id,i} \| k \| K_{pub})$ işlemi ile hesaplar.
6. Son olarak KGU, güvenli bir kanal aracılığıyla $\{R_{ppk,i}, RKA_{id,i}\}$ parametrelerini gönderir.

İmzalama aşaması

İmzalama aşamasında araçlar ve RSU'lar, trafik güvenliğini sağlamak için diğer kullanıcılara gönderecekleri mesajları göndermeden önce imzalar. Böylece mesajın doğruluğu ve bütünlüğü, alıcı araç ve RSU tarafından doğrulanır. Araç ve RSU, KGU tarafından hesaplanan kısmi gizli anahtarlarını ve kısmi anonim kimliklerini aldıktan sonra diğer araçlara ve RSU'lara mesaj gönderebilir. Bu nedenle, önerilen şemada araçların ve RSU'ların her bir mesajlaşma sırasında KGU ve CU ile iletişim kurması gerekmez. Bu aşama, araçlar ve RSU'lar için aşağıdaki adımlarla gerçekleştirilir.

Araç için algoritma adımları

1. V_i aracı, rasgele $\omega_i \in \mathbb{Z}_q^*$ gizli anahtarını seçer.
2. V_i tarafından gizli anahtar ω_i ve kısmi gizli anahtar $V_{ppk,i}$ kullanılarak aracın gizli anahtarı $V_{gizli,i} = V_{ppk,i} + \omega_i$ hesaplanır.
3. V_i tarafından $V_{p,i} = V_{gizli,i} \cdot P$ hesaplanır. $V_{p,i}$ değerini kullanarak V_i aracı kendi açık anahtarını $V_{açık,i} = V_{p,i} - K_{pub}$ işlemi ile hesaplar.
4. V_i, gizli anahtarı ω_i ve $VKA_{id,i}$ kısmi anonim kimliğini kullanarak tam anonim kimliğini $VA_{id,i} = VKA_{id,i} h_3(\omega_i . C_{pub})$ işlemi ile üretir.
5. V_i tarafından $\phi_{v,i} = \omega_i \cdot P$ hesaplanır.
6. V_i; $T_{v,i}$ zaman bilgisi, $VA_{id,i}$ tam anonim kimliği, $V_{açık,i}$ açık anahtarı, göndermek istediği $m_{v,i}$ mesajını ve $\phi_{v,i}$ değerini kullanarak $S_{v,i} = h_4(m_{v,i} \| VA_{id,i} \| V_{açık,i} \| \phi_{v,i} \| T_{v,i})$ değerini hesaplar.
7. $m_{v,i}$ mesajının imzası Eş. 5.1'deki gibi V_i tarafından oluşturulur. Son olarak $\{\sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametreleri $m_{v,i}$ mesajı ile birlikte araçlara ve RSU'lara gönderilir.

$$\sigma_{v,i} = V_{gizli,i} + \omega_i S_{v,i} \pmod{q} \tag{5.1}$$

RSU için algoritma adımları

1. R_i, rasgele $\beta_i \in \mathbb{Z}_q^*$ gizli anahtarını seçer.
2. R_i tarafından gizli anahtar β_i ve kısmi gizli anahtar $R_{ppk,i}$ kullanılarak kendi gizli anahtarı $R_{gizli,i} = R_{ppk,i} + \beta_i$ hesaplanır.
3. R_i tarafından $R_{p,i} = R_{gizli,i} \cdot P$ hesaplanır. $R_{p,i}$ değerini kullanarak kendi açık anahtarını $R_{açık,i} = R_{p,i} - K_{pub}$ işlemi ile hesaplar.
4. R_i, gizli anahtarı β_i ve $RKA_{id,i}$ kısmi anonim kimliğini kullanarak tam anonim kimliğini $RA_{id,i} = RKA_{id,i} h_3(\beta_i \cdot C_{pub})$ işlemi ile üretir.

5. R_i tarafından $\phi_{R,i} = \beta_i \cdot P$ hesaplanır.
6. R_i; $T_{R,i}$ zaman bilgisi, $RA_{id,i}$ tam anonim kimliği, $R_{açık,i}$ açık anahtarı, $m_{R,i}$ mesajını ve $\phi_{R,i}$ değerini kullanarak $S_{R,i} = h_4(m_{R,i} \| RA_{id,i} \| R_{açık,i} \| \phi_{R,i} \| T_{R,i})$ değerini hesaplar.
7. $m_{R,i}$ mesajının imzası Eş. 5.2'deki gibi R_i tarafından oluşturulur. Son olarak $\{\sigma_{R,i}, RA_{id,i}, R_{açık,i}, \phi_{R,i}, T_{R,i}\}$ parametreleri $m_{R,i}$ mesajı ile birlikte araçlara ve RSU'lara gönderilir.

$$\sigma_{R,i} = R_{gizli,i} + \beta_i S_{R,i} \pmod{q} \tag{5.2}$$

Doğrulama aşaması

Bu aşamada, mesajı alan taraf mesajın doğruluğunu ve bütünlüğünü kontrol ederek mesajı kabul eder veya reddeder. Önerilen kimlik doğrulama şeması V2V, V2I, I2I ve I2V iletişim modlarını destekler. Araçtan RSU'ya (V2I) gönderilen mesaj için doğrulama aşaması aşağıdaki gibi gerçekleştirilir. Diğer iletişim modları içinde benzer işlemler gerçekleştirilebilir.

1. T zamanında alınan mesajın zaman damgası $T_{v,i}$, RSU tarafından kontrol edilir. $T - T_{v,i} > \Delta T$ ise mesaj reddedilir. Aksi takdirde, bir sonraki adıma geçilir. Böylece süresi dolmuş mesajlar değerlendirilmeden iptal edilir.
2. RSU R_i tarafından gönderilen parametreler kullanılarak $S_{v,i}$ değeri $S_{v,i} = h_4(m_{v,i} \| VA_{id,i} \| V_{açık,i} \| \phi_{v,i} \| T_{v,i})$ işlemi ile hesaplanır.
3. $m_{v,i}$ mesajını alan R_i, $\{\sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametrelerini kullanarak Eş. 5.3'ün sağlanıp sağlanamadığını kontrol eder.

$$\sigma_{v,i} \cdot P \stackrel{?}{=} K_{pub} + V_{açık,i} + S_{v,i} \cdot \phi_{v,i} \tag{5.3}$$

Eş. 5.3'ün doğruluğunu ispatlayalım. Eş. 5.1'de verilen imzalama eşitliğinin her iki tarafı *P* noktası ile çarpılarak Eş. 5.4 elde edilir.

$$\sigma_{v,i} \cdot P \stackrel{?}{=} (V_{gizli,i} + \omega_i S_{v,i}) \cdot P \tag{5.4}$$

Eş. 5.4, $V_{gizli,i} = V_{ppk,i} + \omega_i$ ve $V_{ppk,i} = k + h_1(VKA_{id,i} \| k \| K_{pub})$ eşitlikleri dikkate alınarak düzenlenirse Eş. 5.5 elde edilir.

$$\sigma_{v,i} \cdot P \stackrel{?}{=} (k + h_1(VKA_{id,i} \| k \| K_{pub}) + \omega_i) \cdot P + \omega_i S_{v,i} \cdot P \tag{5.5}$$

Eş. 5.5, $\phi_{v,i} = \omega_i \cdot P$, $V_{açık,i} = V_{p,i} - K_{pub}$ ve $V_{p,i} = V_{gizli,i} \cdot P$ eşitlikleri kullanılarak düzenlenirse Eş. 5.6 elde edilir. Böylece imza doğrulama aşamasında verilen Eş. 5.3 eşitliğinin doğruluğu ispatlanmıştır.

$$\sigma_{v,i} \cdot P = K_{pub} + V_{açık,i} + S_{v,i} \cdot \phi_{v,i} \tag{5.6}$$

Toplu doğrulama

Önerilen kimlik doğrulama şeması, trafikte verimliliği artırmak için toplu doğrulamayı destekler. RSU tarafından alınan birden çok mesaj toplu olarak aynı anda doğrulanır. RSU, birden çok $i = \{1, 2, 3..., x\}$ tane mesaj aldığında mesajların ve imzaların geçerliliğini aşağıdaki adımlarla kontrol eder [34].

1. R_i alınan mesajların $T_{v,i}$ değerlerine bakarak mesajların güncelliğini kontrol eder. Alınan mesajların zaman damgası $T - T_{v,i} > \Delta T$ eşitliği ile kontrol edilir ve eşitliği sağlamayan mesajlar reddedilir. Eğer eşitlik sağlanırsa, bir sonraki adıma geçilir. Böylece süresi dolmuş mesajlar değerlendirilmeden iptal edilir.
2. R_i tarafından $i = \{1, 2, 3..., x\}$ olmak üzere $S_{v,i} = h_4(m_{v,i} \| VA_{id,i} \| V_{açık,i} \| \phi_{v,i} \| T_{v,i})$ işlemi ile $S_{v,1}, S_{v,2}, S_{v,3}....S_{v,x}$ değerleri hesaplanır.
3. Daha sonra $i = \{1, 2, 3..., x\}$ olmak üzere $m_{v,i}$ mesajlarını alan R_i, $\{\sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametrelerini kullanarak Eş. 5.7'nin sağlanıp sağlanmadığını kontrol eder.

$$(\sum_{i=1}^{x}\sigma_{v,i})\cdot P \stackrel{?}{=} \sum_{i=1}^{x}K_{pub}+\sum_{i=1}^{x}V_{açık,i}+\sum_{i=1}^{x}S_{v,i}\cdot\phi_{v,i} \quad (5.7)$$

Toplu mesaj doğrulaması için Eş. 5.7'nin doğruluğunu ispatlayalım. Eş. 5.1'de verilen $\sigma_{v,i}=V_{gizli,i}+\omega_i S_{v,i}$ imzalama eşitliğini toplu doğrulama için Eş. 5.8'deki gibi düzenleyelim.

$$\sum_{i=1}^{x}\sigma_{v,i}=\sum_{i=1}^{x}V_{gizli,i}+\sum_{i=1}^{x}\omega_i S_{v,i} \quad (5.8)$$

Eş. 5.8'in her iki tarafı *P* noktası ile çarpılarak Eş. 5.9'daki gibi düzenlensin.

$$(\sum_{i=1}^{x}\sigma_{v,i})\cdot P \stackrel{?}{=} \sum_{i=1}^{x}V_{gizli,i}\cdot P+\sum_{i=1}^{x}\omega_i S_{v,i}\cdot P \quad (5.9)$$

Eş. 5.9 düzenlenirse Eş. 5.10 elde edilir. Burada $\phi_{v,i}=\omega_i\cdot P$, $V_{açık,i}=V_{p,i}-K_{pub}$, $V_{p,i}=V_{gizli,i}\cdot P$ ve $V_{ppk,i}=k+\hat{\lambda}h_1(VKA_{id,i}\,||\,k\,||\,K_{pub})$ eşitlikleri kullanılarak düzenleme gerçekleştirilir. Böylece Eş. 5.10 ile önerilen şemanın toplu doğrulamayı sağladığı ispat edilir.

$$(\sum_{i=1}^{x}\sigma_{v,i})\cdot P=\sum_{i=1}^{x}K_{pub}+\sum_{i=1}^{x}V_{açık,i}+\sum_{i=1}^{x}S_{v,i}\cdot\phi_{v,i} \quad (5.10)$$

Benzer şekilde V2V, I2V ve I2I iletişim modları içinde imzalama ve imza doğrulama aşamaları gerçekleştirilir.

5.3. Güvenlik Değerlendirmesi

Siber saldırılar, IoV ağının güvenliğini olumsuz etkiler ve hatta trafikte yaralanmalara ve ölümlere neden olabilir. Güvenli bir sistemin sağlaması gereken temel güvenlik gereksinimleri ve önerilen kimlik doğrulama şemasının bu saldırılar karşısındaki güvenliği aşağıda değerlendirilmiştir.

Göndericinin kimliğinin doğrulanması ve mesaj bütünlüğü

Trafik ile ilgili mesajı alan araç veya RSU, bu mesajı gönderen aracın ve RSU'nun IoV ağına kayıtlı olduğunu doğrulamalı ve alınan mesajın kendisine ulaştığı sürece kadar iyi niyetli olmayan saldırganlar tarafından değiştirilmediğinden emin olmalıdır. Önerilen kimlik doğrulama şemasında araç veya RSU, $\{\sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametrelerini kullanarak Eş. 5.3'teki imza doğrulama eşitliğinin sağlanıp sağlanmadığını kontrol eder. Mesajın alıcısı gönderilen parametreleri kullanarak $S_{v,i} = h_4(m_{v,i} \| VA_{id,i} \| V_{açık,i} \| \phi_{v,i} \| T_{v,i})$ değerini hesaplar. Dikkat edilecek olursa bu hesaplamada mesaj bilgisi, anonim kimlik, açık anahtar ve zaman bilgisi kullanılmakta olup bu değerlerin herhangi birinde değişme olması durumunda h_4 özet fonksiyonu farklı bir değer üreterek imza doğrulama eşitliğini sağlamayacaktır. Bu sayede, mesaj gönderenin kimlik doğrulaması ve alınan mesajın bütünlüğü kontrol edilir. Böylece önerilen kimlik doğrulama şemasının göndericinin kimlik doğrulamasını ve mesaj bütünlüğünü sağlamış olduğu ispatlanır.

Kimliğin anonim olması ve izlenebilirlik

Araçlar ve RSU'lar, güvenlik nedeniyle gerçek kimliklerinin gizlenmesi gerektiğinden anonim kimlik kullanırlar [104]. Ancak araçların veya RSU'ların anonim kimliklere sahip olması, yasa dışı faaliyetlere karıştıklarında veya saldırganlar tarafından ele geçirildiklerinde gerçek kimliklerinin tespit edilemeyeceği anlamına gelmez [34]. Bu durumda sisteme kayıtlı kullanıcıların gerçek kimlikleri merkezi otorite olan CU tarafından ortaya çıkarılabilir. Bu sayede araçların izlenebilirliği sağlanmış olur. Önerilen kimlik doğrulama şemasında araç veya RSU, anonim kimliğini her imza oluşturma sırasında tekrar oluşturur. CU ve KGU araçların ve RSU'ların kısmi anonim kimliklerini üreterek kullanıcılara teslim eder. Araç için açıklayacak olursak; bir araç üretmiş olduğu bir mesaj için $\omega_i \in \mathbb{Z}_q^*$ gizli anahtarını seçer. Bu gizli anahtarı kullanarak $VA_{id,i} = VKA_{id,i} h_3(\omega_i \cdot C_{pub})$ hesaplaması ile anonim kimliğini oluşturur. Benzer şekilde, RSU da kendi anonim kimliğini üretir. Bu anonim kimliklerden gerçek kimliğin ortaya çıkarılması sadece CU tarafından gerçekleştirilebilir. CU tarafından $\zeta = V_{id,i} \oplus h_1(C_{pub} \| c)$ ve $VKA_{id,i} = \zeta \oplus h_2(k \cdot C_{pub})$ eşitlikleri ve c gizli anahtarı kullanılarak gerçek kimlik Eş. 5.11'deki gibi elde edilir.

$$V_{id,i} = VKA_{id,i} \oplus h_2(c \cdot K_{pub}) \oplus h_1(C_{pub} \| c) \quad (5.11)$$

Eğer Eş. 5.11'de elde edilen gerçek kimliğin sisteme zarar verebilecek bir faaliyet içerisinde olduğu tespit edilmişse araç veri tabanından çıkarılır ve sisteme kayıtlı diğer araçlara mesaj gönderemez. Böylece, önerilen kimlik doğrulama şeması hem izlenebilirliği hem de kimlik gizliliğini korumayı (anonimlik) sağlar.

Bağlantısızlık

Bir saldırgan, aynı araç veya RSU tarafından üretilen iki imzayı ve mesajı ilişkilendirememelidir. Eğer saldırgan en az iki mesaj ve kimlik bilgisi arasında bağlantı kurarsa sistemin güvenliği tehlikeye düşebilir. Araç ve RSU, mesajı anonim bir kimlikle gönderse bile, her mesajın iletimi sırasında anonim kimlik değiştirilmelidir. Aksi halde mesaj ve anonim kimlik ilişkilendirilebilir. Önerilen kimlik doğrulama şemasında her mesaj gönderimi sırasında anonim kimlik değiştiği için saldırganlar tarafından mesajlar ve anonim kimlik arasında bağlantı kurulamaz [34].

Konum gizliliği

Araçların güvenliğini sağlamak için konum gizliliği oldukça önemlidir. Saldırganlar, araçların konumunu bulmak için çeşitli yöntemler kullanabilir. Araçların gerçek kimlikleri yerine anonim kimlikler kullanılması ve bu kimliklerin her mesaj gönderildiğinde değişmesi saldırganlara karşı konum gizliliğinin sağlanması açısından önemlidir. Önerilen kimlik doğrulama şemasında anonim kimlik kullanılması ve her mesaj gönderimi sırasında anonim kimliğin değişmesi nedeniyle konum gizliliği korunabilir.

İnkâr edilemezlik

Kriptografinin genel ilkelerinden biri olan inkâr edilemezlik, göndericinin göndermiş olduğu mesajdan sorumlu olmasını ifade eder. Önerilen ağ yapısına dahil olan araçlar ve RSU'lar gönderdikleri mesajlardan sorumludur. Anonim kimlikleri ile mesaj gönderip imza atsalar dahi inkâr edememeleri gerekir. Önerilen kimlik doğrulama şemasında merkezi

otorite kullanıcıların gerçek kimliklerini tespit edebildiği için inkâr edememe özelliği sağlanır.

Kimliğe bürünme saldırısı

Bir saldırgan, IoV ağına dahil olan aracı ve RSU'yu taklit edebilir ve ağ kullanıcısı gibi diğer araçlara ve RSU'lara mesajlar gönderebilir. Sistemin güvenliği açısından önemli bir sorun olan kimliğe bürünme saldırısına karşı önerilen kimlik doğrulama şeması güvenlidir. Açık $\{\sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametrelerinden yola çıkılarak saldırgan tarafından sahte mesaj ve Eş. 5.12'yi sağlayacak sahte açık parametreler üretilebilmesi gerekir.

$$\sigma_{v,i} \cdot P = K_{pub} + V_{açık,i} + S_{v,i} \cdot \phi_{v,i} \tag{5.12}$$

Ancak önerilen kimlik doğrulama şeması ECDLP'ye dayalı olduğu için bunu gerçekleştirmek oldukça zordur.

Sahte bilgi saldırısı

Sahte bilgi saldırısı ile saldırgan, trafik durumu hakkında sahte bilgiler üreterek IoV ağındaki araçlara veya RSU'lara gönderir. Bu sayede kendisine uygun bir trafik rotası oluşturabilir. Önerilen kimlik doğrulama şemasında, alıcı mesajı aldığında kimlik bilgisi ve mesaj bütünlüğü kontrol edilir. Önerilen kimlik doğrulama şemasının güvenliği ECDLP'ye dayalı olduğundan kayıtsız bir kullanıcının veya saldırganın, sahte bilgi saldırısı gerçekleştirmesi oldukça zordur.

Ortadaki adam saldırısı

Ortadaki adam saldırısında saldırgan iki araç ve/veya iki RSU arasında iletişim kurarak onları birbirleriyle iletişim kurduklarına inandırır. Bu sayede gelen gizli bilgileri toplar veya trafik mesajlarının içeriğini manipüle eder ve değiştirilmiş mesajları diğer araçlara veya RSU'lara iletir. Önerilen kimlik doğrulama şemasında karşılıklı iletişim kimlik doğrulamaya dayalı olduğundan, CU'ya kayıtlı olmayan bir saldırgan bu saldırıyı gerçekleştiremez. Çünkü kayıt olduktan sonra CU tarafından araca gönderilen

$\zeta = V_{id,i} \oplus h_1(C_{pub} \| c)$ değeri ile KGU ile iletişime geçerek kısmi anonim kimliğini ve kısmi gizli anahtarını alır. Bu sayede önerilen kimlik doğrulama şeması ortadaki adam saldırısına dayanaklıdır.

Rasgele kâhin modeli (ROM)

ROM analizinde, saldırgan tarafından kimlik doğrulama şemasının güvenliğinin polinom bir zaman içinde ihmal edilebilir olup olmadığı kontrol edilir. Önerilen şemanın güvenlik analizini yapmak ve bu şemaya karşı rakibin yeterliliğini ölçmek için bir oyun kurulur. Bu oyun meydan okuyucu *C* ve rakip *A* arasında oynanır. *A,* bu oyunu kazanırsa, kimlik doğrulama şeması güvenlik gereksinimlerini karşılayamaz [34].

Bu alt bölümde, önerilen kimlik doğrulama şemasının uyarlanabilir seçilmiş mesaj altında rasgele kahin modelinde güvenlik analizi gerçekleştirilmiştir. Bunun için, Oyun-I ve Oyun-II oyunları *C* tarafından rakiplere karşı başlatılır. Oyun-I için *C* meydan okuyucusu ile *A* rakibi arasında, aşağıdaki sorguları yapmasına izin verilen bir model belirleyelim. Bu adım araçlar için gerçekleştirilir. RSU için de benzer adımlar gerçekleştirilebilir.

Teorem: Eğer ECDLP varsayımı geçerliyse, önerilen kimlik doğrulama şeması ROM'daki *A* rakibi tarafından taklit edilemezdir.

İspat: *A,* önerilen kimlik doğrulama şemasında geçerli bir imza oluşturma yeteneğine sahip bir rakip olsun. Meydan okuyucu *C* ise ECDLP'yi çözebilme yeteneğine sahip olsun. Daha sonra *C*'ye $Q = x \cdot P$ gibi bir ECDLP örneği verildiğini varsayalım. *C, A* ile etkileşime girerek verilen $Q = x \cdot P$ eşitliğindeki x'i bulmayı amaçlar. Bu amaçla *C,* rakip *A*'ya meydan okumak için anonim kimlik olarak $VA^{*}_{id,i}$'yı seçer. Son olarak *C, A* tarafından aşağıdaki sorguları gerçekleştirir.

- Kurulum: *C,* rasgele k gizli anahtarını seçer ve $K_{pub} = k \cdot P$ hesaplamasını gerçekleştirir. Daha sonra $\{E, P, p, q, K_{pub}, C_{pub}, h_3, h_4\}$ parametrelerini *A*'ya gönderir. *C* aynı zamanda sorguları ve cevapları kaydetmek için başlangıçta boş olan $h_{liste-3}, h_{liste-4}$, $V_{liste}, PPK_{liste}, S_{liste}, Pub_{liste}$ listelerini tutar.

- h_3 sorgusu: Bu aşamada rakip *A*, $\{\omega_i, C_{pub}, H_i\}$ sorgusunu gerçekleştirir. Sonuç olarak, bu sorgu ile üretilen cevabın $h_{liste-3}$ içerisinde olup olmadığı kontrol edilir. Eğer listede ise *C*, $H_i = h_3(\omega_i \cdot C_{pub})$ sonucunu *A*'ya gönderir. Aksi halde *C*, rasgele bir H_i değeri üreterek *A*'ya gönderir ve $h_{liste-3}$'e parametreler kaydedilerek liste güncellenir.
- h_4 sorgusu: Rakip *A*, $\{m_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ sorgusunu gerçekleştirir. Sonuç olarak, bu sorgu ile üretilen cevabın $h_{liste-4}$ içerisinde olup olmadığı *C* tarafından kontrol edilir. Eğer listede ise *C*, $S_{v,i} = h_4(m_{v,i} \| VA_{id,i} \| V_{açık,i} \| \phi_{v,i} \| T_{v,i})$ sonucunu *A*'ya gönderir. Aksi halde *C*, rasgele bir $S_{v,i}$ değerini üreterek *A*'ya gönderir ve parametreler $h_{liste-4}$ listesine kaydedilerek liste güncellenir.
- Araç sorgusu: Rakip *A*, açık anahtar için aracın anonim kimliği olan $VA_{id,i}$ üzerinde $\{VA_{id,i}, V_{gizli,i}, V_{açık,i}, V_{p,i}, V_{ppk,i}, \omega_i\}$ sorgusunu gerçekleştirir. Daha sonra *C*, V_{liste} listesini kontrol eder, eğer V_{liste} listesi $\{VA_{id,i}, V_{gizli,i}, V_{açık,i}, V_{p,i}, V_{ppk,i}, \omega_i\}$ parametrelerini içeriyorsa, *C*, $V_{açık,i}$ açık anahtarını *A*'ya gönderir. Aksi durumda, *C* rasgele ω_i, $V_{ppk,i} \in \mathbb{Z}_q^*$ parametrelerini seçer ve $V_{gizli,i} = V_{ppk,i} + \omega_i$, $V_{p,i} = V_{gizli,i} \cdot P$ ve $V_{açık,i} = V_{p,i} - K_{pub}$ değerlerini hesaplar. Son olarak *C*, rakip *A*'ya $\{V_{açık,i}\}$ parametresini gönderir ve $h_{liste-3}, h_{liste-4}, V_{liste}$ listelerini günceller.
- Kısmi gizli anahtar (PPK) sorgusu: Rakip *A*, aracın anonim kimliği olan $VA_{id,i}$ üzerinde PPK için $\{VA_{id,i}, VKA_{id,i}, V_{açık,i}, k, V_{ppk,i}, K_{pub}\}$ sorgusunu gerçekleştirir. Daha sonra *C*, PPK_{liste} listesini kontrol eder, eğer PPK_{liste} listesi $\{VA_{id,i}, VKA_{id,i}, V_{açık,i}, k, V_{ppk,i}, K_{pub}, \lambda\}$ parametrelerini içeriyorsa, *C* tarafından $\{V_{ppk,i}\}$ parametresi rakip *A*'ya gönderilir. Aksi takdirde, *C* rasgele $VKA_{id,i}, k, \lambda \in \mathbb{Z}_q^*$ parametrelerini seçer ve daha sonra $V_{ppk,i} = k + \lambda h_1(VKA_{id,i} \| k \| K_{pub})$ değerini hesaplar. Son olarak, *C* tarafından *A*'ya $\{V_{ppk,i}\}$ parametresini gönderir ve $h_{liste-3}$, $h_{liste-4}$, V_{liste}, PPK_{liste} listelerini günceller.
- Gizli anahtar sorgusu: Rakip *A*, $V_{gizli,i}$ gizli anahtarı için aracın anonim kimliği olan $VA_{id,i}$ üzerinde $\{VA_{id,i}, VKA_{id,i}, V_{gizli,i}, \omega_i, V_{ppk,i}\}$ sorgusunu gerçekleştirir. Daha sonra *C*, S_{liste} listesini kontrol eder, eğer S_{liste} listesi $\{VA_{id,i}, VKA_{id,i}, V_{gizli,i}, \omega_i, V_{ppk,i}\}$ parametrelerini içeriyorsa, *C* tarafından $\{V_{gizli,i}\}$ gizli anahtarı *A*'ya gönderilir. Aksi

takdirde, *C* rasgele ω_i, $V_{ppk,i} \in \mathbb{Z}_q^*$ parametrelerini seçer ve $V_{gizli,i} = V_{ppk,i} + \omega_i$ değerini hesaplar. Son olarak, C tarafından $\{V_{gizli,i}\}$ parametresi gönderilir ve $h_{liste-3}$, $h_{liste-4}$, V_{liste}, PPK_{liste}, S_{liste} listelerini günceller.

- Açık anahtar değiştirme sorgusu: Rakip *A*, V_i aracının $V_{açık,i}^*$ açık anahtarını değiştirmek için $\{VA_{id,i}, V_{açık,i}^*\}$ sorgusunu gerçekleştirir. Daha sonra *C* tarafından Pub_{liste} listesi kontrol edilir ve eğer Pub_{liste} listesi $\{VA_{id,i}, V_{p,i}, V_{gizli,i}, V_{açık,i}, \omega_i, V_{ppk,i}\}$ parametrelerini içeriyorsa, $V_{açık,i}$ değerini $V_{açık,i}^*$ ile ve ω_i değeri ise $\perp$ ile değiştirir. Aksi takdirde, *C* tarafından $V_{açık,i} = V_{açık,i}^*$ ve $\omega_i = \perp$ olarak ayarlanır. Ardından *C* tarafından Pub_{liste} listesine $\{VA_{id,i}, V_{p,i}, V_{gizli,i}, V_{açık,i}^*, \perp, V_{ppk,i}\}$ parametreleri eklenir.
- İmza sorgusu: Rakip *A* tarafından $\{m_{v,i}, VA_{id,i}\}$ sorgusu gerçekleştirilir. Daha sonra *C* tarafından şu adımlar takip edilir: İlk olarak *C*, aracın $V_{ppk,i}$ kısmi anahtarını oluşturmak için kısmi gizli anahtar (PPK) sorgusu gerçekleştirir. *C*, rasgele $\omega_i \in \mathbb{Z}_q^*$ gizli anahtarını seçer ve $V_{gizli,i} = V_{ppk,i} + \omega_i$ değerini hesaplar. Daha sonra *C*, $V_{açık,i}$ parametresini oluşturmak için araç sorgusunu gerçekleştirir. *C* tarafından $\{S_{v,i}\}$ parametresini oluşturmak için h_4 sorgusunu gerçekleştirilir ve $\phi_{v,i} = \omega_i \cdot P$ değeri hesaplanır. Daha sonra *C*, $\sigma_{v,i} = V_{gizli,i} + \omega_i S_{v,i}$ değerini hesaplar ve $h_{liste-4}$, V_{liste}, PPK_{liste} listelerini günceller. Son olarak *C* tarafından $\{m_{v,i}, \sigma_{v,i}, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametreleri *A*'ya gönderilir. Rakip *A*, mesaj ve imza dahil olmak üzere bu parametreleri üretirse, *C* tarafından imzanın geçerliliği $\sigma_{v,i} \cdot P \stackrel{?}{=} K_{pub} + V_{açık,i} + S_{v,i} \cdot \phi_{v,i}$ eşitliği ile kontrol edilir. Eğer eşitlik sağlanırsa *C* oyunu bitirir.
- Sahtecilik: Fakat rakip *A*, sahte bir geçerli imza üretebilir. Sahtecilik varsayımına [102] dayanarak sahte bir geçerli imza ve mesajı içeren $\{m_{v,i}, \sigma_{v,i}^*, VA_{id,i}, V_{açık,i}, \phi_{v,i}, T_{v,i}\}$ parametrelerinin üretildiğini varsayalım. Böylece, $\sigma_{v,i}^* \cdot P = K_{pub} + V_{açık,i} + S_{v,i}^* \cdot \phi_{v,i}$ eşitliği sağlanır. Elde edilen eşitlik ve $\sigma_{v,i} \cdot P = K_{pub} + V_{açık,i} + S_{v,i} \cdot \phi_{v,i}$ eşitliği düzenlenirse Eş. 5.13 elde edilir.

$$
\begin{aligned}
(\sigma_{v,i} - \sigma^{*}_{v,i}) \cdot P &= S_{v,i} \cdot \phi_{v,i} - S^{*}_{v,i} \cdot \phi_{v,i} \\
&= (S_{v,i} - S^{*}_{v,i}) \cdot \phi_{v,i} \\
&= (S_{v,i} - S^{*}_{v,i}) \cdot x \cdot P
\end{aligned}
\tag{5.13}
$$

C tarafından ECDLP'ye dayalı oldukça zor bir problem olan $Q = x \cdot P$ eşitliğinde $x = (\sigma_{v,i} - \sigma^{*}_{v,i})(S_{v,i} - S^{*}_{v,i})^{-1}$ şeklinde elde edilir. Bu nedenle, önerilen kimlik doğrulama şeması ROM'da sahteciliğe karşı güvenlidir. Benzer işlemler Oyun-II ve RSU için de gerçekleştirilir.

Sonuç olarak önerilen kimlik doğrulama şemasının güvenlik gereksinimlerini sağladığı ve saldırılara karşı dirençli olduğu gösterilmiştir.

5.4. Performans Değerlendirmesi

Bu bölümde, önerilen kimlik doğrulama şemasının performansı mevcut şemalar [26], [32], [40], [43], [48], [99] ve [100] ile karşılaştırılmıştır. Performans değerlendirmesi için; hesaplama maliyeti, iletişim maliyeti ve enerji tüketim maliyeti kriterleri kullanılmaktadır.

Hesaplama maliyetinin değerlendirmesi

Bu alt bölümde önerilen kimlik doğrulama şeması ile literatürde bulunan şemalar [26], [32], [40], [43], [48], [99] ve [100] hesaplama maliyetleri açısından karşılaştırılmıştır. İlk olarak mesaj imzalama, tek mesaj doğrulama ve toplu mesaj doğrulama aşamalarında kullanılan kriptografik işlemlerin gerçekleştirim süreleri (ms: milisaniye) hesaplanarak Çizelge 5.2'de sunulmuştur. Matematiksel işlemlerin gerçekleştirim sürelerini hesaplamak için Intel i7-7500U CPU ve 8 GB RAM'e sahip Ubuntu 16.04 işletim sistemi üzerinde çalışan bir bilgisayar kullanılmıştır. C++ programlama dilinde PBC (Pairing-Based Cryptography) kütüphanesi [101] kullanılarak kriptografik işlemler 1000 defa tekrarlanmıştır ve yürütme sürelerinin ortalaması alınmıştır. XOR ve birleştirme (||) gibi diğer işlemler süre açısından ihmal edilebilir düzeyde olduğundan dikkate alınmamıştır [34].

Çizelge 5.2. Matematiksel işlemlerin ortalama gerçekleştirim süresi

İşlem	Süre (ms)	Tanım
T_{bp}	4,6216	Çift doğrusal eşleme işlemi
T_{smbp}	1,2506	Çift doğrusal eşlemede skaler çarpma
T_{pabp}	0,0034	Çift doğrusal eşlemede nokta toplama
T_{smecc}	0,3508	Eliptik eğri üzerinde skaler çarpma
T_{paecc}	0,0021	Eliptik eğri üzerinde nokta toplama
T_{ssmecc}	0,0168	Eliptik eğri üzerinde küçük sayı ile skaler çarpma
T_h	0,0002	Özet fonksiyon

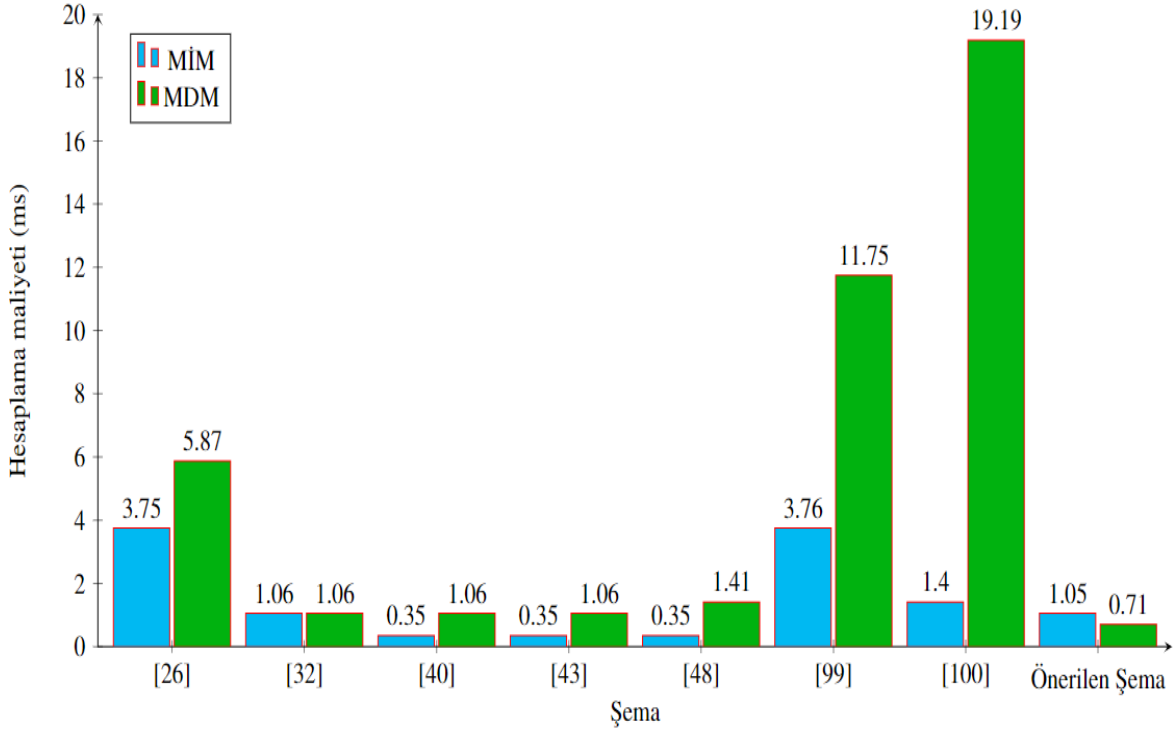
Önerilen kimlik doğrulama şemasının hesaplama maliyeti aşağıda belirlenmiştir. Mesaj imzalama maliyeti, mesaj doğrulama maliyeti ve toplu mesaj doğrulama maliyeti sırasıyla MİM, MDM ve TMDM kısaltmaları ile ifade edilmiştir.

İlk olarak MİM; eliptik eğri üzerinde üç skaler çarpma, bir nokta çıkarma (yaklaşık olarak T_{paecc}'ye eşit) ve iki özet fonksiyonu gerektirir. Böylece MİM gerçekleştirim süresi $3T_{smecc} + T_{paecc} + 2T_h \approx 1{,}0549$ ms olarak hesaplanır. Sonrasında MDM: iki skaler çarpma, iki nokta toplama ve bir özet fonksiyonu gerektirir, bu nedenle MDM gerçekleştirim süresi $2T_{smecc} + 2T_{paecc} + T_h \approx 0{,}7060$ ms olarak hesaplanır. Son olarak, TMDM hesaplanması $(x + 1)$ adet skaler çarpma, $(3x - 1)$ adet nokta toplama ve x adet özet fonksiyon gerektirir. Böylece TMDM gerçekleştirim süresi $(x + 1)T_{smecc} + (3x - 1)T_{paecc} + xT_h \approx 0{,}3487 + 0{,}3573x$ ms olarak hesaplanır. Benzer şekilde, karşılaştırma yapılan [26], [32], [40], [43], [48], [99] ve [100] şemalarının MİM, MDM ve TMDM gerçekleştirim süreleri hesaplanmıştır.

Sonuç olarak, önerilen kimlik doğrulama şemasının ve diğer mevcut [26], [32], [40], [43], [48], [99] ve [100] şemalarının MİM, MDM ve TMDM için hesaplama maliyetleri Çizelge 5.3'te sunulmuştur. Önerilen şemanın ve mevcut diğer şemaların grafiksel karşılaştırılması ise Şekil 5.2'de sunulmuştur.

Çizelge 5.3. Hesaplama maliyetlerinin karşılaştırılması

Şema	MİM (ms)	MDM (ms)	TMDM (ms)
[26]	$3T_{smbp} + T_h \approx 3,752$	$T_{bp} + T_{smbp} + T_h \approx 5,8724$	$T_{bp} + (x)T_{smbp} + (x)T_h \approx 4,6216 + 1,2508x$
[32]	$3T_{smecc} + 2T_{paecc} + T_h \approx 1,0568$	$3T_{smecc} + 3T_{paecc} + T_h \approx 1,0589$	$(3x+2)T_{smecc} + (3x)T_{paecc} + (x)T_h \approx 0,7016 + 1,0589x$
[40]	$T_{smecc} + T_h \approx 0,351$	$3T_{smecc} + 2T_{paecc} + 2T_h \approx 1,057$	$(2x+1)T_{smecc} + (2x)T_{paecc} + (2x)T_h \approx 0,3508 + 0,7062x$
[43]	$T_{smecc} + 2T_h \approx 0,3512$	$3T_{smecc} + 2T_{paecc} + 2T_h \approx 1,057$	$(2x+1)T_{smecc} + (3x-1)T_{paecc} + (2x)T_h \approx 0,349 + 0,708x$
[48]	$T_{smecc} + 2T_h \approx 0,3512$	$4T_{smecc} + 3T_{paecc} + 3T_h \approx 1,4101$	$(2x+2)T_{smecc} + (3x)T_{paecc} + (3x)T_h \approx 0,7016 + 0,7085x$
[99]	$3T_{smbp} + T_{pabp} + T_h \approx 3,755$	$2T_{bp} + 2T_{smbp} + 3T_{pabp} + 2T_h \approx 11,75$	$2T_{bp} + (2x)T_{smbp} + (3x)T_{pabp} + (2x)T_h \approx 9,2432 + 2,511x$
[100]	$4T_{smecc} + 6T_h \approx 1,4044$	$4T_{bp} + 2T_{smecc} + 4T_h \approx 19,1888$	$4T_{bp} + (2x)T_{smecc} + (3x+1)T_h \approx 18,4866 + 0,7022x$
Önerilen şema	$3T_{smecc} + T_{paecc} + 2T_h \approx 1,0549$	$2T_{smecc} + 2T_{paecc} + T_h \approx 0,706$	$(x+1)T_{smecc} + (3x-1)T_{paecc} + (x)T_h \approx 0,3487 + 0,3573x$



Şekil 5.2. Önerilen şemanın ve mevcut şemaların hesaplama maliyetinin karşılaştırılması

Önerilen şemanın performansını değerlendirmek için diğer mevcut şemalara kıyasla hesaplama maliyetinde sağlamış olduğu yüzdesel iyileştirmeyi hesaplayalım. İlk olarak MİM'de önerilen şemanın [99] numaralı şemaya göre hesaplama maliyetindeki iyileşme yüzdesi (3,755−1,0549) / 3,755 ≈ % 71,91 olarak hesaplanmıştır. Bu işlem diğer mevcut şemalar için benzer şekilde gerçekleştirilmiştir.

Böylece MİM'de önerilen şema; [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalara kıyasla sırasıyla %71,88, %0,18, %0, %0, %0, %71,91 ve %24,89 iyileştirme sağlar. Önerilen şema; [26], [32], [99] ve [100] numaralı şemalara göre hesaplama maliyetinde iyileştirme sağlarken ancak [40], [43] ve [48] numaralı şemalara kıyasla hesaplama maliyetinde iyileştirme sağlayamaz. Bunun nedeni [40], [43] ve [48] numaralı şemaların anonim kimliklerinin merkezi birim tarafından hesaplanması ve OBU'ya önceden yüklenmesidir. OBU'ya birçok anonim kimlik yüklenmiş olsa bile, her mesaj iletimi sırasında farklı anonim kimlikler gönderilmeyebilir. Aynı anonim kimlikle iki farklı mesaj gönderilirse, ağdaki haberleşmeyi izleyen saldırgan mesajlar arasında bağlantı kurabilir.

MDM için de benzer hesaplamalar gerçekleştirerek önerilen şemanın performansını inceleyelim. Önerilen şemanın [99] numaralı şemaya kıyasla hesaplama maliyetindeki iyileşme yüzdesi (11,75−0,706) / 11,75 ≈ %94 olarak hesaplanmıştır. Bu işlem diğer mevcut şemalar için benzer şekilde gerçekleştirilir. Böylece [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalar ile karşılaştırıldığında sırasıyla %87,98, %33,33, %33,21, %33,21, %49,93, %94 ve %96,32 iyileştirme sağlanır. Bu aşamada önerilen kimlik doğruma şeması, diğer mevcut şemalara kıyasla doğrulama aşamasında oldukça iyi sonuçlar göstermiştir.

TMDM için de benzer hesaplamalar gerçekleştirerek önerilen şemanın performansını inceleyelim. Bu aşama için toplu mesaj sayısı x = 1000 olsun. x değerini Çizelge 5.3'teki TMDM sütununda bulunan hesaplamalarda yerine yazalım. Önerilen şemanın, [99] numaralı şemaya göre hesaplama maliyetindeki iyileşme (2520,2430−357,6487) / 2520,2430 = % 85,81 olarak hesaplanmıştır. Bu işlem diğer şemalar için de benzer şekilde gerçekleştirilir. Böylece önerilen şemanın TMDM'de [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalara göre hesaplama maliyetinde %71,51, %66,25, %49,38, %49,51, %49,57, %85,81 ve %50,37 iyileştirme sağlanmıştır. Önerilen kimlik doğrulama şeması,

diğer mevcut şemalara kıyasla toplu doğrulama aşamasında oldukça iyi sonuçlar göstermiştir. Sonuç olarak, önerilen şemanın mevcut şemalara kıyasla hesaplama maliyetlerindeki iyileşme yüzdeleri Çizelge 5.4'te gösterilmektedir.

Çizelge 5.4. Önerilen şemanın mevcut şemalara göre performans artışları

Şema	MİM (%)	MDM (%)	TMDM (%)
[26]	71,88	87,98	71,51
[32]	0,18	33,33	66,25
[40]	0	33,21	49,38
[43]	0	33,21	49,51
[48]	0	49,93	49,57
[99]	71,91	94	85,81
[100]	24,89	96,32	50,37

İletişim maliyetinin değerlendirilmesi

Önerilen şemayı diğer mevcut şemalar ile iletişim maliyeti açısından kıyaslamak için karşı tarafa iletilen mesajın ve diğer açık parametrelerin boyutlarını belirleyerek hesaplama gerçekleştirilmiştir. $\mathbb{Z}_q^*$ kümesimin elemanları ve özet fonksiyonlar 20 bayt, T zaman damgası 4 bayt, toplamsal devirli grup olan G_1, 40 bayt ve çarpımsal devirli grup G_T, 128 bayt olsun. Önerilen şemada araç tarafından mesaj ile birlikte gönderilen açık parametrelerden $\{V_{açık,i}, \phi_{v,i}\} \in G_1$, $\{\sigma_{v,i}, VA_{id,i}\} \in \mathbb{Z}_q^*$ ve zaman damgası T içerir. Böylece önerilen iletişim maliyeti 2 x 40 + 2 x 20 + 4 =124 bayt olarak hesaplanmıştır. Benzer şekilde [26], [32], [40], [43], [48], [99] ve [100] numaralı şemaların iletişim maliyetleri 408, 408, 164, 184, 208, 388 ve 256 bayt olarak hesaplanmıştır.

Önerilen şemanın, [99] numaralı şemaya göre iletişim maliyetindeki yüzdesel iyileşme (388−124) / 388 = %68,04 olarak hesaplanmıştır. Bu işlem diğer mevcut şemalar için benzer şekilde gerçekleştirilir. Böylece önerilen şema; [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalarla karşılaştırıldığında iletişim maliyetlerinde sırasıyla %69,61,

%69,61, %24,39, %32,61, %40,38, %68,04 ve %51,56 performans artışı sağlamaktadır. Sonuç olarak, önerilen şema iletişim maliyeti bakımından mevcut şemalara göre daha iyi sonuç vermektedir. Önerilen şema ve diğer mevcut şemaların tek mesaj için iletişim maliyetleri ve önerilen şemanın diğer şemalara kıyasla performans artışı yüzde olarak Çizelge 5.5'te verilmiştir.

Çizelge 5.5. Önerilen şemanın mevcut şemalara kıyasla iletişim maliyetinin ve performans artışının karşılaştırılması

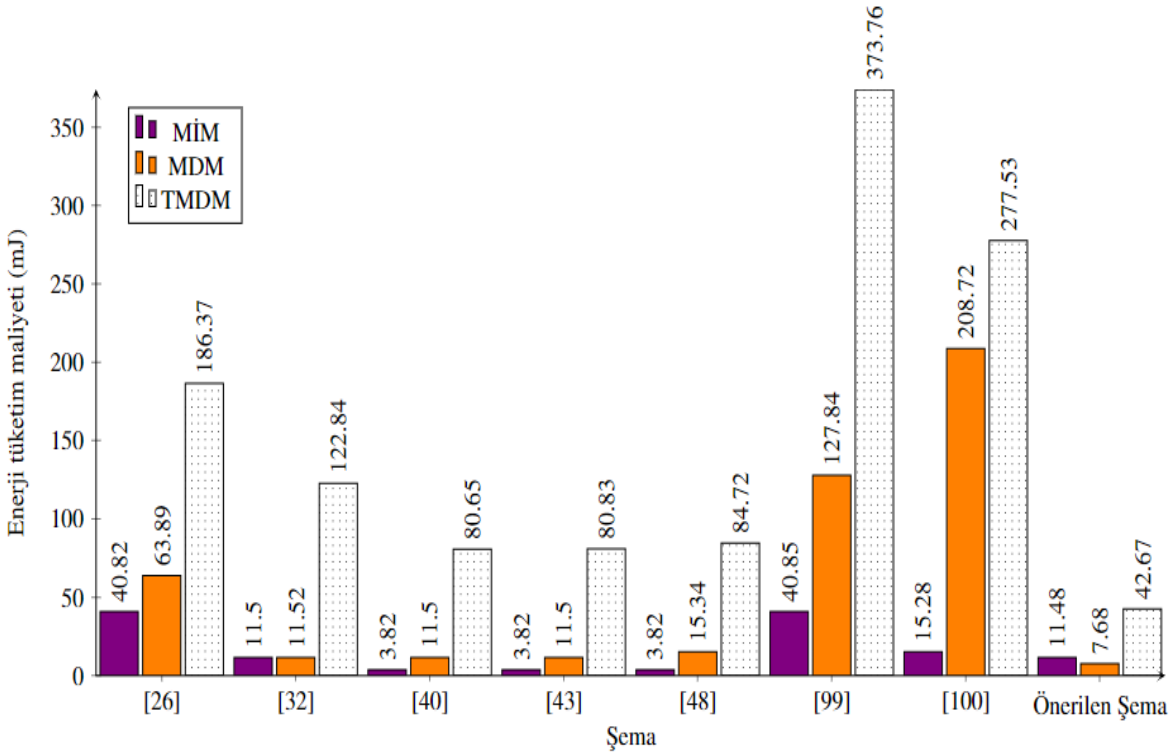
Şema	İletişim maliyeti (bayt)	Performans artışı (%)
[26]	$3\{G_T\}+\{\mathbb{Z}_q^*\}+T=408$	69,61
[32]	$3\{G_T\}+\{\mathbb{Z}_q^*\}+T=408$	69,61
[40]	$3\{G_1\}+2\{\mathbb{Z}_q^*\}+T=164$	24,39
[43]	$4\{G_1\}+\{\mathbb{Z}_q^*\}+T=184$	32,61
[48]	$4\{G_1\}+2\{\mathbb{Z}_q^*\}+2T=208$	40,38
[99]	$3\{G_T\}+T=388$	68,04
[100]	$2\{G_T\}=256$	51,56
Önerilen şema	$2\{G_1\}+2\{\mathbb{Z}_q^*\}+T=124$	-

Enerji tüketim maliyeti

Önerilen şemanın enerji tüketim maliyeti [43] numaralı çalışmada belirtilen yöntemle hesaplanmıştır. Kimlik doğrulama şemalarının enerji tüketimi $E = P.t$ eşitliği ile hesaplanır. Bu eşitlikte E; tüketilen enerjiyi, P; CPU'nun maksimum gücünü (10,88 Watt) ve t ise işlemin gerçekleştiği süreyi ifade etmektedir. Bu eşitlikten yola çıkarak önerilen şemanın ve diğer mevcut şemalar için mesajın imzalanması, mesajın doğrulanması ve toplu mesaj doğrulama süreçlerinin enerji tüketim maliyeti hesaplanmıştır.

MİM'de önerilen şemanın enerji tüketim maliyeti $E = P.t$ ile 10,88 x 1,0549 = 11,4773 mJ olarak hesaplanmıştır. Burada t değeri Çizelge 5.3'teki önerilen şemanın mesaj imzalama süresidir. Daha sonra MDM'deki enerji tüketim maliyeti 10,88 x 0,706 = 7,6813

mJ olarak hesaplanmıştır. Son olarak, x =10 olmak üzere TMDM'de enerji tüketim maliyeti 10,88 x 3,9217 = 42,6681 mJ olarak hesaplanmıştır. Benzer şekilde MİM, MDM ve TMDM'de [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalar için enerji tüketim maliyetleri hesaplanmıştır. Önerilen şemanın ve diğer mevcut şemaların MİM, MDM ve TMDM için enerji tüketim maliyetleri Şekil 5.3'te karşılaştırılmıştır.



Şekil 5.3. Önerilen şema ve diğer şemaların enerji tüketim maliyetlerinin karşılaştırılması

Sonuç olarak, önerilen şemanın mevcut şemalara kıyasla enerji tüketim maliyetlerindeki iyileşme yüzdeleri hesaplama maliyetlerindeki performans artışıyla aynıdır. Enerji tüketim maliyetinin hesaplandığı $E=P.t$ eşitliğinde P sabit olduğundan t değerine göre enerji tüketim maliyeti belirlenecektir. Bu sonuçtan yola çıkarak, MİM'de önerilen şemanın; [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalara göre sırasıyla enerji maliyetlerinde %71,88, % 0,18, %0, %0, %0, %71,91 ve %24,89 azalma sağlanır. Benzer hesaplama MDM için gerçekleştirilirse önerilen şema [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalar ile karşılaştırıldığında sırasıyla enerji maliyetlerinde %87,98, %33,33, %33,21, %33,21, %49,93, %94 ve %96,32 azalma sağlar. Son olarak toplu mesaj doğrulama için x=1000 olduğunu varsayalım. Bu durumda önerilen şema TMDM'de [26], [32], [40], [43], [48], [99] ve [100] numaralı şemalara göre enerji tüketim maliyetinde %71,51, %66,25, %49,38, %49,51, %49,57, %85,81 ve %50,37 azalma sağlar.

Enerji tüketim maliyeti açısından sonuçlar incelendiğinde önerilen şemanın mevcut diğer şemalara kıyasla oldukça iyi sonuçlar verdiği gözlemlenmiştir.

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, araçların interneti için eliptik eğri kriptografisini kullanan çift doğrusal eşlemesiz, kimlik tabanlı ve sertifikasız kriptografi yöntemine dayanan ve koşullu gizliliği sağlayan özgün bir kimlik doğrulama (CL-CPPA) şeması önerilmiştir. Eşleme tabanlı kriptografide kullanılan matematiksel hesaplamalar oldukça maliyetli ve süre gerektiren işlemlerdir. Bu çalışmada temel olarak eşleme tabanlı matematiksel işlemler kullanmayan eliptik eğri tabanlı kimlik doğrulama şeması üzerine çalışılmıştır. Eliptik eğri kriptografisi, küçük anahtar boyutu ve düşük güç tüketimi gibi özellikleri sayesinde birçok alanda yaygın olarak kullanılmaktadır. Ancak araç ağlarında kullanılacak bir kriptografi sisteminin verimliliğini artırmak için açık anahtar altyapısı ve sertifika kullanmayan kimlik tabanlı kriptografi sistemi de dikkate alınmıştır. Kimlik tabanlı kriptografi ile kullanıcılar inkâr edemeyecekleri bilgileri açık anahtar olarak kullanabilirler. Bu kriptografi sistemleri için oldukça önemli bir kriterdir. Kimlik tabanlı şifreleme ilk olarak Shamir tarafından önerilmiş ancak uzun bir süre herhangi bir pratik uygulamada kullanılmamıştır.

2001 yılında Boneh-Franklin çift doğrusal eşleme kullanan kimlik tabanlı bir şifreleme algoritması önermişlerdir. Fakat çift doğrusal eşlemede matematiksel işlemlerin maliyeti kriptografi sistemlerinin verimliliği açısından önemli bir problem olarak görülmektedir. Bu nedenle, eliptik eğriler kullanan eşlemesiz kimlik tabanlı kriptografi önerileri yaygınlaşmaktadır. Ancak kimlik tabanlı kriptografinin önemli bir sorunu olan anahtar emanet sorunu göz ardı edilmemelidir. Kimlik tabanlı kriptografide kullanılan PKG, kullanıcıların gizli anahtarlarını üretir ve saklar. Ancak siber saldırı veya fiziksel engeller PKG'deki gizli anahtar bilgilerinin ele geçirilmesi veya kullanılamaz hale gelmesi durumunda kriptografi sisteminin güvenliğini tehlikeye atar. Bu tezde, bu husus göz önünde bulundurularak kimlik tabanlı sertifikasız kriptografi yöntemini temel alan yeni ve özgün bir tasarım önerilmiştir. Bu yöntem araç ağlarında toplu mesaj doğrulamasını sağlayan ve mesaj doğrulama aşamasındaki gecikmeleri en aza indiren verimli bir yöntemle gerçekleştirilmiştir.

Önerilen kimlik doğrulama şemasında araç ağlarında V2V, V2I, I2V ve I2I gibi farklı modlarda iletişim sağlanmaktadır. Araçların internetindeki bu iletişim modlarına ek olarak araçtan ağa (V2N), araçtan şebekeye (V2G), araçtan yayaya (V2P) gibi araçtan her şeye

(V2X) iletişim modları da eklenebilir. Bu şemada, araçlar veya RSU'lar mesaj imzalama sırasında anonim kimliklerini oluştururlar. Böylece, anonim kimlik her mesajda değiştiği için konum ve kimlik gizliliği sağlanır. Aynı zamanda bu özellik, merkezi birimle sürekli iletişime ihtiyaç duymadığı için merkezi birimin aşırı yüklenmesini engeller ve araç ağında meydana gelebilecek gecikme sürelerini azaltır.

Önerilen şemanın hesaplama maliyeti, iletişim maliyeti ve enerji tüketim maliyetleri diğer mevcut şemalara göre karşılaştırıldığında oldukça yüksek performans göstermiştir. Söz konusu şema, mesaj doğrulama aşamasında diğer mevcut şemalara göre %33,21 ila %96,32 arasında iyileştirme sağlar. Toplu mesaj doğrulama aşamasında, önerilen şema diğer mevcut şemalara göre hesaplama maliyetinde %49,38 ila %85,81 arasında daha yüksek performans sağlar. Enerji tüketim maliyetlerinin hesaplanmasında ise mesaj imzalama, mesaj doğrulama ve toplu mesaj doğrulama süreleri kullanıldığından hesaplanacak değerler yüzdesel olarak hesaplama maliyetleri ile aynıdır. Böylece önerilen şema diğer mevcut şemalara göre enerji tüketim maliyeti açısından da oldukça iyi sonuçlar vermiştir. Önerilen şema iletişim maliyeti açısından da diğer mevcut şemalara kıyasla daha düşük iletişim maliyeti sağlamaktadır. İletişim maliyetindeki performans artışı diğer şemalara göre %24,39 ila %69,61 arasındadır.

Önerilen şemanın ROM'da kapsamlı bir güvenlik değerlendirmesi yapılmıştır. Ayrıca, önerilen şema güvenli bir araç ağı sisteminde bulunması gerekli güvenlik gereksinimleri ve saldırılara karşı göstermiş olduğu direnç açısından detaylı olarak incelenmiştir. Bu kapsamda; göndericinin kimliğinin doğrulanması ve mesaj bütünlüğü, kimliğin anonim olması ve izlenebilirlik, bağlantısızlık, konum gizliliği, inkâr edilemezlik, sahte bilgi saldırısı ve ortadaki adam saldırısı incelenmiş olup önerilen şemanın bu gereksinimleri sağladığı ve saldırılara karşı dirençli olduğu gösterilmiştir.

Böylece önerilen şemanın güvenlik analizleri sonucunda gizlilik ve güvenlik gereksinimlerini karşıladığı kanıtlanmıştır. Ayrıca, hesaplama, enerji tüketim ve iletişim maliyetleri açısından mevcut diğer şemalarla yapılan karşılaştırmalardan görülebileceği gibi daha uygun maliyetli olduğu kanıtlanmıştır.

Bu tez çalışmasında eliptik eğri tabanlı bir verimli imza algoritması ve eliptik eğri tabanlı özgün şifreleme algoritması da sunuluştur. Bu sayede kriptografi sistemlerinin tasarımında

birçok parametre ve algoritma incelenmiştir: Eliptik eğri sayısal imza algoritması (ECDSA) güvenliği ve verimliliği sayesinde birçok uygulamada kullanılmaktadır. Ancak algoritmadaki modüler aritmetikteki ters çevirme işlemi algoritmanın hızını yavaşlatmaktadır. Önerilen yöntemde, imzalama ve imza doğrulama aşamalarındaki modüler aritmetikteki ters çevirme işlemi kaldırılmıştır. Gerçekleştirilen tasarımla ECDSA'nın güvenliği azaltılmadan hızı artırılmıştır. Klasik ECDSA'nın ve önerilen yöntemin imza oluşturma ve doğrulama aşamalarındaki gerçekleştirim süreleri karşılaştırıldığında önerilen yöntemde sırasıyla %25 ve %18 performans artışı sağlanmaktadır. Diğer bir çalışmada ise sonlu cisimler üzerinde eliptik eğri kullanılarak yeni bir mesaj şifreleme algoritması tasarlanmıştır. Bu tasarımda birçok farklı alfabe ve sembol kullanılarak şifreleme ve şifre çözme gerçekleştirilmektedir. Önerilen yöntemde Unicode tablosu sayesinde daha fazla karakter kullanabilmektedir. Böylece, şifreleme uygulamaları için ASCII tablosunun sınırlı karakter sorununa bir çözüm önerilmiştir.

Bundan sonra yapılacak olan çalışmalarda; blokzincir tabanlı araç ağlarında koşullu gizliliği koruyan kimlik doğrulama şeması ve homomorfik tabanlı kriptografide farklı iletişim modları için araç iletişimi incelenebilir.

KAYNAKLAR

1. Zhang, M., Zhou, J., Zhang, G., Zou, M., and Chen, M. (2021). EC-BAAS: Elliptic curve-based batch anonymous authentication scheme for Internet of Vehicles. *Journal of Systems Architecture*, 117, 102161.

2. Yu, S., Lee, J., Park, K., Das, A. K., and Park, Y. (2020). IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. *IEEE Access*, 8, 167875-167886.

3. Li, J., Choo, K. K. R., Zhang, W., Kumari, S., Rodrigues, J. J., Khan, M. K., and Hogrefe, D. (2018). EPA-CPPA: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *Vehicular Communications*, 13, 104-113.

4. Xiong, H., Chen, J., Mei, Q., and Zhao, Y. (2020). Conditional privacy-preserving authentication protocol with dynamic membership updating for VANETs. *IEEE Transactions on Dependable and Secure Computing*, 19(3), 2089-2104.

5. He, D., Zeadally, S., Xu, B., and Huang, X. (2015). An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, 10(12), 2681-2691.

6. Alshudukhi, J. S., Al-Mekhlafi, Z. G., and Mohammed, B. A. (2021). A lightweight authentication with privacy-preserving scheme for vehicular ad hoc networks based on elliptic curve cryptography. *IEEE Access*, 9, 15633-15642.

7. Ali, I., Chen, Y., Ullah, N., Kumar, R., and He, W. (2021). An efficient and provably secure ECC-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs. *IEEE Transactions on Vehicular Technology*, 70(2), 1278-1291.

8. Vasudev, H., Das, D., and Vasilakos, A. V. (2020). Secure message propagation protocols for IoVs communication components. *Computers & Electrical Engineering*, 82, 106555.

9. Lai, C., Lu, R., Zheng, D., and Shen, X. (2020). Security and privacy challenges in 5G-enabled vehicular networks. *IEEE Network*, 34(2), 37-45.

10. Sikarwar, H., and Das, D. (2023, January). *SMMAP: Secure MAC-based Mutual Authentication Protocol for IoV*. In 24th International Conference on Distributed Computing and Networking, 330-335.

11. Sikarwar, H., and Das, D. (2021). Towards Lightweight Authentication and Batch Verification Scheme in IoV. *IEEE Transactions on Dependable and Secure Computing*, 19(5), 3244-3256.

12. Yu, S., Lee, J., Park, K., Das, A. K., and Park, Y. (2020). IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. *IEEE Access*, 8, 167875-167886.

13. Azees, M., Vijayakumar, P., and Deboarh, L. J. (2017). EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(9), 2467-2476.

14. Gayathri, N. B., Thumbur, G., Reddy, P. V., and Rahman, M. Z. U. (2018). Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks. *IEEE Access*, 6, 31808-31819.

15. Sucasas, V., Mantas, G., Saghezchi, F. B., Radwan, A., and Rodriguez, J. (2016). An autonomous privacy-preserving authentication scheme for intelligent transportation systems. *Computers and Security*, 60, 193-205.

16. Bagga, P., Sutrala, A. K., Das, A. K., and Vijayakumar, P. (2021). Blockchain-based batch authentication protocol for Internet of Vehicles. *Journal of Systems Architecture*, 113, 101877.

17. Jeong, H. H., Shen, Y. C., Jeong, J. P., and Oh, T. T. (2021). A comprehensive survey on vehicular networking for safe and efficient driving in smart transportation: A focus on systems, protocols, and applications. *Vehicular Communications*, 31, 100349.

18. Xiong, H., Hou, Y., Huang, X., and Zhao, Y. (2020). Secure message classification services through identity-based signcryption with equality test towards the internet of vehicles. *Vehicular Communications*, 26, 100264.

19. Sharma, S., and Kaushik, B. (2019). A survey on internet of vehicles: Applications, security issues and solutions. *Vehicular Communications*, 20, 100182.

20. El-Rewini, Z., Sadatsharan, K., Selvaraj, D. F., Plathottam, S. J., and Ranganathan, P. (2020). Cybersecurity challenges in vehicular communications. *Vehicular Communications*, 23, 100214.

21. Tanwar, S., Vora, J., Tyagi, S., Kumar, N., and Obaidat, M. S. (2018). A systematic review on security issues in vehicular ad hoc network. *Security and Privacy*, 1(5), 39.

22. Xiong, W., Wang, R., Wang, Y., Zhou, F., and Luo, X. (2021). CPPA-D: efficient conditional privacy-preserving authentication scheme with double-insurance in VANETs. *IEEE Transactions on Vehicular Technology*, 70(4), 3456-3468.

23. Bayat, M., Barmshoory, M., Rahimi, M., and Aref, M. R. (2015). A secure authentication scheme for VANETs with batch verification. *Wireless Networks*, 21, 1733-1743.

24. Raya, M., and Hubaux, J. P. (2007). Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1), 39-68.

25. Wang, S., and Yao, N. (2017). LIAP: A local identity-based anonymous message authentication protocol in VANETs. *Computer Communications*, 112, 154-164.

26. Ali, I., and Li, F. (2020). An efficient conditional privacy-preserving authentication scheme for Vehicle-To-Infrastructure communication in VANETs. *Vehicular Communications*, 22, 100228.

27. Lu, R., Lin, X., Zhu, H., Ho, P. H., and Shen, X. (2008, April). *ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications.* In IEEE INFOCOM 2008-The 27th Conference on Computer Communications, 1229-1237.

28. Zhang, J., Zhang, Q., Lu, X., and Gan, Y. (2021). A novel privacy-preserving authentication protocol using bilinear pairings for the VANET environment. *Wireless Communications and Mobile Computing*, 1-13.

29. Manvi, S. S., and Tangade, S. (2017). A survey on authentication schemes in VANETs for secured communication. *Vehicular Communications*, 9, 19-30.

30. Kamil, I. A., and Ogundoyin, S. O. (2019). An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks. *Journal of Information Security and Applications*, 44, 184-200.

31. Zhang, C., Lu, R., Lin, X., Ho, P. H., and Shen, X. (2008, April). *An efficient identity-based batch verification scheme for vehicular sensor networks*. In IEEE INFOCOM 2008-The 27th Conference on Computer Communications, 246-250.

32. Li, J., Ji, Y., Choo, K. K. R., and Hogrefe, D. (2019). CL-CPPA: Certificate-less conditional privacy-preserving authentication protocol for the Internet of Vehicles. *IEEE Internet of Things Journal*, 6(6), 10332-10343.

33. Wang, Y., Liu, Y., and Tian, Y. (2022). ISC-CPPA: Improverd-Security Certificateless Conditional Privacy-Preserving Authentication Scheme With Revocation. *IEEE Transactions on Vehicular Technology*, 71(11), 12304-12314.

34. Genc, Y., Aytas, N., Akkoc, A., Afacan, E., and Yazgan, E. (2023). ELCPAS: A new efficient lightweight certificateless conditional privacy preserving authentication scheme for IoV. *Vehicular Communications*, 39, 100549.

35. Mansour, A., Malik, K. M., Alkaff, A., and Kanaan, H. (2020). ALMS: Asymmetric lightweight centralized group key management protocol for VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1663-1678.

36. Miller, V. S. (1986). *Use of Elliptic Curves in Cryptography*. Proceedings of Crypto 85, 417-426.

37. Genç, Y., and Afacan, E. (2021, July). *Implementation of New Message Encryption Using Elliptic Curve Cryptography over Finite Fields*. In 2021 International Congress of Advanced Technology and Engineering (ICOTEN), 1-6.

38. Shamir, A. (1985). *Identity-based cryptosystems and signature schemes*. In Advances in Cryptology: Proceedings of CRYPTO 84, 47-53.

39. Ali, I., Hassan, A., and Li, F. (2019). Authentication and privacy schemes for vehicular ad hoc networks (VANETs): A survey. *Vehicular Communications*, 16, 45-61.

40. Chen, Y., and Chen, J. (2021). Cpp-clas: efficient and conditional privacy-preserving certificateless aggregate signature scheme for vanets. *IEEE Internet of Things Journal*, 9(12), 10354-10365.

41. Kate, A., and Goldberg, I. (2010). *Distributed private-key generators for identity-based cryptography.* In Security and Cryptography for Networks: 7th International Conference, SCN 2010, 436-453.

42. Genç, Y., and Afacan, E. (2021, June). *Identity-based encryption in the internet of Things*. In 2021 29th Signal Processing and Communications Applications Conference (SIU), 1-4.

43. Thumbur, G., Rao, G. S., Reddy, P. V., Gayathri, N. B., Reddy, D. K., and Padmavathamma, M. (2020). Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks. *IEEE Internet of Things Journal*, 8(3), 1908-1920.

44. Al-Riyami, S. S., and Paterson, K. G. (2003, November). Certificateless public key cryptography. In *Asiacrypt*, 2894, 452-473.

45. Xiong, H., Qin, Z., and Vasilakos, A. V. (2016). *Introduction to certificateless cryptography*. Boca Raton: CRC Press, 3-135.

46. Zhao, Y., Hou, Y., Wang, L., Kumari, S., Khan, M. K., and Xiong, H. (2020). An efficient certificateless aggregate signature scheme for the Internet of Vehicles. *Transactions on Emerging Telecommunications Technologies*, 31(5), 3708.

47. Zhang, J., Cui, J., Zhong, H., Chen, Z., and Liu, L. (2019). PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks. *IEEE Transactions on Dependable and Secure Computing*, 18(2), 722-735.

48. Zhou, X., Luo, M., Vijayakumar, P., Peng, C., and He, D. (2022). Efficient certificateless conditional privacy-preserving authentication for VANETs. *IEEE Transactions on Vehicular Technology*, 71(7), 7863-7875.

49. Paar, C., and Pelzl, J. (2009). *Understanding cryptography: a textbook for students and practitioners*. New York: Springer Science and Business Media, 1-357.

50. Ferguson, N., Schneier, B., and Kohno, T. (2011). *Cryptography engineering: design principles and practical applications*. Indianapolis: John Wiley and Sons, 1-300.

51. Stinson, D. R., and Paterson, M. (2018). *Cryptography: Theory and Practice*. Boca Raton: CRC Press, 1-481.

52. Afacan, E. (2016). *Kriptografiye Giriş*. Ankara: Epos Yayınları, 15-38.

53. Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (2018). *Handbook of applied cryptography*. Boca Raton: CRC Press, 1-432.

54. McAndrew, A. (2016). *Introduction to Cryptography with open-source software*. Boca Raton: CRC Press, 1-327.

55. Menezes, A. J. (1993). *Elliptic curve public key cryptosystems* . New York: Springer Science and Business Media, 1-116.

56. Katz, J., and Lindell, Y. (2020). *Introduction to modern cryptography*. Boca Raton: CRC Press, 1-495.

57. Zheng, Z. (2022). *Modern Cryptography Volume 1: A Classical Introduction to Informational and Mathematical Principle* . Singapore: Springer Nature, 153-250.

58. Martin, K. M. (2012). *Everyday Cryptography*: Fundamental Principles and Applications . New York: Oxford University Press, 1-488.

59. Mollin, R. A. (2000). *An introduction to cryptography*. Boca Raton: CRC Press, 1-156.

60. Diffie W. and Hellman, M.E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22, 644–654.

61. Mollin, R. A. (2002*). RSA and public-key cryptography.* Boca Raton: CRC Press, 1-179.

62. Hankerson, D., Menezes, A. J., and Vanstone, S. (2006). *Guide to elliptic curve cryptography.* New York: Springer Science and Business Media, 1-293.

63. Washington, L. C. (2008). *Elliptic curves: number theory and cryptography.* Boca Raton: CRC press, 1-489.

64. Stamp, M., and Low, R. M. (2007). *Applied cryptanalysis: breaking ciphers in the real world.* Indianapolis: John Wiley and Sons, 265-355.

65. Perlman, R. (1999). An overview of PKI trust models. *IEEE network*, 13(6), 38-43.

66. Nakajima, J., and Matsui, M. (2002). *Performance analysis and parallel implementation of dedicated hash functions.* In Advances in Cryptology-EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques Amsterdam, Netherlands, 165-180.

67. Genç, Y., and Afacan, E. (2021, April). *Design and implementation of an efficient elliptic curve digital signature algorithm (ECDSA)*. In 2021 IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS), 1-6.

68. Sobti, R., and Geetha, G. (2012). Cryptographic hash functions: a review. *International Journal of Computer Science Issues (IJCSI)*, 9(2), 461.

69. Johnson, D., Menezes, A., and Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security*, 1, 36-63.

70. Ali, I., Chen, Y., Faisal, M., Pohantūn, O. M., and Li, M. (2022). *Efficient and Provably Secure Schemes for Vehicular Ad-Hoc Networks*. Singapore: Springer, 1-32.

71. Çallıalp, F. (2011). *Örneklerle soyut cebir*. İstanbul: Birsen Yayınevi, 24-281.

72. Silverman, J. H. (2006). *An introduction to the theory of elliptic curves*. Brown University and NTRU Cryptosystem, 1-104.

73. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, 48(177), 203-209.

74. Miller, V. S. (1986). *Use of elliptic curves in cryptography* . Advances in Cryptology-CRYPTO, 85, 417-426.

75. Martin, L. (2008). *Introduction to identity-based encryption*. Norwood: Artech house, 1-176.

76. 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.

77. Menezes, A. J., and Vanstone, S. A. (1993). Elliptic curve cryptosystems and their implementation. *Journal of Cryptology*, 6, 209-224.

78. Chen, L., Moody, D., Randall, K., Regenscheid, A., and Robinson, A. (2023). *Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters.* NIST, 1-48.

79. Singh, L. D., and Singh, K. M. (2015). Implementation of text encryption using elliptic curve cryptography. *Procedia Computer Science*, 54, 73-82.

80. Boruah, D., and Saikia, M. (2014, February). *Implementation of ElGamal Elliptic Curve Cryptography over prime field using C*. In International Conference on Information Communication and Embedded Systems (ICICES2014), 1-7.

81. Lamba, S., and Sharma, M. (2013, December). *An efficient elliptic curve digital signature algorithm (ecdsa)*. In 2013 International Conference on Machine Intelligence and Research Advancement, 179-183

82. Junru, H. (2011, August). *The improved elliptic curve digital signature algorithm*. In Proceedings of 2011 international conference on electronic and mechanical engineering and information technology, 257-259.

83. Zhang, Q., Li, Z., and Song, C. (2011, August). *The Improvement of digital signature algorithm based on elliptic curve cryptography.* In 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 1689-1691.

84. Mehibel, N., and Hamadouche, M. H. (2020). A new enhancement of elliptic curve digital signature algorithm. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(3), 743-757.

85. Prabu, M., and Shanmugalakshmi, R. (2009, December). *A comparative analysis of signature schemes in a new approach of variant on ECDSA*. In 2009 International Conference on Information and Multimedia Technology, 491-494.

86. Menezes, A. (2009). An introduction to pairing-based cryptography. *Recent Trends in Cryptography*, 477, 47-65.

87. El Mrabet, N., and Joye, M. (Eds.). (2017). *Guide to pairing-based cryptography*. Boca Raton: CRC Press, 1-27.

88. Dutta, R., Barua, R., and Sarkar, P. (2004). Pairing-based cryptographic protocols: A survey. *Cryptology ePrint Archive*, 3-42.

89. Patil, H. K., and Szygenda, S. A. (2012). *Security for wireless sensor networks using identity-based cryptography*. Boca Raton: CRC Press, 33-90.

90. Boneh, D., and Franklin, M. (2001, August). *Identity-based encryption from the Weil pairing*. In Advances in Cryptology-CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, 213-229.

91. Odelu, V., Das, A. K., Rao, Y. S., Kumari, S., Khan, M. K., and Choo, K. K. R. (2017). Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment. *Computer Standards and Interfaces*, 54, 3-9.

92. Miret, J. M., Sadornil, D., and Tena, J. G. (2018). Pairing-based cryptography on elliptic curves. *Mathematics in Computer Science*, 12(3), 309-318.

93. Lynn, B. (2007). *On the implementation of pairing-based cryptosystems* . Doctoral dissertation, Stanford University.

94. Barbosa, M. (2005). *Identity based cryptography from bilinear pairings*. Universidade do Minho. Centro de Ciências e Tecnologias da Computação (CCTC), 1-46.

95. Miller, V. S. (2004). The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4), 235-261.

96. Joux, A. (2000, July). *A one round protocol for tripartite Diffie–Hellman.* In Algorithmic Number Theory: 4th International Symposium, ANTS-IV Leiden, Netherlands, 385-393.

97. Boneh, D., and Boyen, X. (2004, May). *Efficient selective-ID secure identity-based encryption without random oracles.* In Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 223-238.

98. Boneh, D., Lynn, B., and Shacham, H. (2001, December). *Short signatures from the Weil pairing*. In Advances in Cryptology-ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, 514-532.

99. Wang, H., Wang, L., Zhang, K., Li, J., and Luo, Y. (2022). A conditional privacy-preserving certificateless aggregate signature scheme in the standard model for VANETs. *IEEE Access*, 10, 15605-15618.

100. Liang, Y., and Liu, Y. (2022). Analysis and Improvement of an Efficient Certificateless Aggregate Signature With Conditional Privacy Preservation in VANETs. *IEEE Systems Journal*, 664-672.

101. İnternet: The pairing-based cryptography library. Web: https://crypto.stanford.edu/pbc/. Son Erişim Tarihi: 17.02.2023.

102. Pointcheval, D., & Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13, 361-396.

103. Meffert, D. (2009). *Bilinear pairings in cryptography*. Radboud Universiteit Nijmegen, Computing Science Department, Netherlands, 22-82.

104. Genç, Y., Habek, M., Aytaş, N., Akkoç, A., Afacan, E., and Yazgan, E. (2022, May). *Elliptic curve cryptography for security in connected vehicles*. In 2022 30th Signal Processing and Communications Applications Conference (SIU), 1-4.



Gazili olmak ayrıcalıktır