



**T.C.  
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**YÜKSEK LİSANS TEZİ**

**BLOKZİNCİR TABANLI MERKEZİ OLMAYAN ÜNİVERSİTE ÖLÇEKLİ  
OYLAMA SİSTEMİ**

**Ahmet Can ÇİÇEK**

**DANIŞMAN  
Prof. Dr. Muhammed Ali AYDIN**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı**

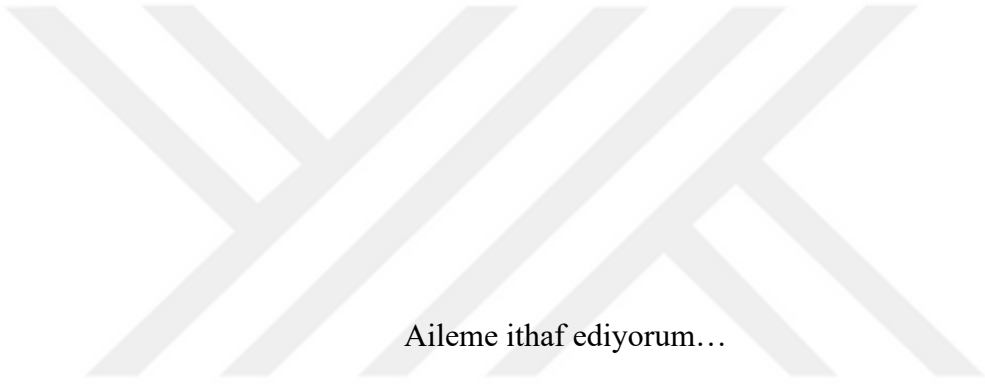
**Haziran, 2025**

## TEZ KABUL VE ONAYI

**Ahmet Can ÇİÇEK** tarafından, **Prof. Dr. Muhammed Ali AYDIN** danışmanlığında hazırlanan **"BLOKZİNCİR TABANLI MERKEZİ OLMAYAN ÜNİVERSİTE ÖLÇEKLİ OYLAMA SİSTEMİ"** başlıklı bu çalışma, jürimiz tarafından **30/06/2025** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

### Tez Jürisi

|                 | İmza                                  | Sonuç                           |
|-----------------|---------------------------------------|---------------------------------|
| <b>DANIŞMAN</b> | Prof. Dr. Muhammed Ali AYDIN          | <input type="checkbox"/>        |
|                 | İstanbul Üniversitesi-Cerrahpaşa      | Kabul                           |
|                 | Bilgisayar Mühendisliği Anabilim Dalı | <input type="checkbox"/><br>Ret |
| <b>ÜYE</b>      | Dr. Öğr. Üyesi Özgür Can TURNA        | <input type="checkbox"/>        |
|                 | İstanbul Üniversitesi-Cerrahpaşa      | Kabul                           |
|                 | Bilgisayar Mühendisliği Anabilim Dalı | <input type="checkbox"/><br>Ret |
| <b>ÜYE</b>      | Dr. Öğr. Üyesi Oğuzhan ÖZTAŞ          | <input type="checkbox"/>        |
|                 | İstanbul Gelişim Üniversitesi         | Kabul                           |
|                 | Bilgisayar Mühendisliği Anabilim Dalı | <input type="checkbox"/><br>Ret |



## **BÜTÇE DESTEKLERİ**

### **BLOKZİNCİR TABANLI MERKEZİ OLMAYAN ÜNİVERSİTE ÖLÇEKLİ OYLAMA SİSTEMİ**

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

## TEŞEKKÜR

Yüksek lisans eğitim sürecimde, bilgi ve tecrübeleriyle rehberlik eden, bu tez çalışmasının gerçekleştirilmesi sürecinde sunduğu değerli destekle yol gösteren danışmanım Prof. Dr. Muhammed Ali AYDIN'a teşekkür ederim.

Haziran 2025

Ahmet Can ÇİÇEK



# İÇİNDEKİLER

Sayfa No

|                                                         |       |
|---------------------------------------------------------|-------|
| TEZ KABUL VE ONAYI.....                                 | ii    |
| BEYAN .....                                             | iii   |
| BÜTÇE DESTEKLERİ .....                                  | v     |
| TEŞEKKÜR.....                                           | vi    |
| İÇİNDEKİLER.....                                        | vii   |
| ŞEKİL LİSTESİ .....                                     | xi    |
| TABLO LİSTESİ.....                                      | xv    |
| SİMGE VE KISALTMA LİSTESİ .....                         | xvii  |
| ÖZET .....                                              | xviii |
| ABSTRACT .....                                          | xx    |
| 1. GİRİŞ.....                                           | 1     |
| 1.1. KONU VE ÖNEMİ .....                                | 3     |
| 1.2. ÖZGÜN DEĞERİ.....                                  | 5     |
| 1.3. AMACI VE HEDEFLERİ .....                           | 6     |
| 1.4. KAPSAMI VE SINIRLAMALARI.....                      | 8     |
| 1.5. TEZİN YAPISI .....                                 | 8     |
| 2. KAVRAMSAL ÇERÇEVE .....                              | 10    |
| 2.1. MERKEZİ, MERKEZİ OLMAYAN ve DAĞITIK AĞLAR .....    | 10    |
| 2.1.1. Merkezi Ağlar .....                              | 11    |
| 2.1.2. Merkezi Olmayan Ağlar .....                      | 11    |
| 2.1.3. Dağıtık Ağlar.....                               | 12    |
| 2.2. KRİPTOGRAFİK ALGORİTMALAR .....                    | 12    |
| 2.2.1. Simetrik ve Asimetrik Anahtar Algoritmaları..... | 13    |
| 2.2.2. Özetleme Fonksiyonları .....                     | 14    |
| 2.3. SIFIR BİLGİ İSPATI.....                            | 15    |
| 2.4. BLOKZİNCİR .....                                   | 17    |
| 2.4.1. Blokzincir Türleri.....                          | 18    |
| 2.4.2. Blokzincir Özellikleri.....                      | 19    |

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| 2.4.3. Blok Yapısı.....                                              | 20        |
| 2.4.4. Merkle Tree .....                                             | 23        |
| 2.4.5. Konsensüs Algoritmaları.....                                  | 24        |
| 2.4.6. Blokzincir Evrimi .....                                       | 26        |
| 2.4.7. Ethereum .....                                                | 28        |
| 2.5. OYLAMA SİSTEMLERİ.....                                          | 32        |
| 2.5.1. Üniversitelerde Kullanılan Oylama Sistemleri .....            | 33        |
| 2.5.2. Kağıt Tabanlı ve Elektronik Oylama Sistemleri .....           | 34        |
| 2.5.3. Merkezi ve Merkezi Olmayan Oylama Sistemleri .....            | 37        |
| 2.5.4. Güvenli Elektronik Oylama Sistemleri için Gereksinimler ..... | 38        |
| <b>3. YÖNTEM .....</b>                                               | <b>40</b> |
| 3.1. SİSTEMİN AKTÖRLERİ .....                                        | 40        |
| 3.1.1. Seçmen (Voter) .....                                          | 40        |
| 3.1.2. Oylama Yetkilisi (Election Authority).....                    | 42        |
| 3.1.3. Teknik Personel (Operator).....                               | 43        |
| 3.2. SİSTEM GEREKSİNİMLERİ.....                                      | 43        |
| 3.3. KULLANILAN TEKNOLOJİ VE ARAÇLAR .....                           | 44        |
| 3.3.1. Ethereum .....                                                | 46        |
| 3.3.2. Solidity .....                                                | 46        |
| 3.3.3. Hardhat .....                                                 | 46        |
| 3.3.4. React.js .....                                                | 46        |
| 3.3.5. JavaScript .....                                              | 47        |
| 3.3.6. Material UI .....                                             | 47        |
| 3.3.7. Ethers.js .....                                               | 47        |
| 3.3.8. Express .....                                                 | 47        |
| 3.3.9. Java.....                                                     | 47        |
| 3.3.10. Circom.....                                                  | 47        |
| 3.3.11. CircomlibJS.....                                             | 48        |
| 3.3.12. SnarkJS.....                                                 | 48        |
| 3.3.13. Paillier Şifreleme Algoritması.....                          | 48        |
| 3.3.14. Relay Mimarisi (İşlem Ücreti Aracısı).....                   | 48        |
| 3.3.15. Diamod Pattern (EIP-2535).....                               | 49        |
| 3.3.16. Poseidon Hash Fonksiyonu .....                               | 49        |
| 3.3.17. OpenZeppelin .....                                           | 49        |

|                                                              |            |
|--------------------------------------------------------------|------------|
| 3.4. OYLAMA SİSTEMİNİN YAŞAM DÖNGÜSÜ .....                   | 49         |
| 3.4.1. Aşama 1: Sistemin Kurulması.....                      | 50         |
| 3.4.2. Aşama 2: Kayıt Sistemi.....                           | 58         |
| 3.4.3. Aşama 3: Seçim Yönetimi .....                         | 61         |
| 3.4.4. Aşama 4: Oy Kullanımı .....                           | 65         |
| 3.4.5. Aşama 5: Oyların Saklanması .....                     | 69         |
| 3.4.6. Aşama 6: Sonuçların İlanı .....                       | 69         |
| 3.5. OYLAMA SÜRECİ VE İŞ AKIŞI .....                         | 75         |
| 3.6. SİSTEM MİMARİSİ .....                                   | 77         |
| 3.7. KULLANICI ARAYÜZÜ .....                                 | 79         |
| 3.8. RELAYER SERVİSİ.....                                    | 80         |
| 3.9. KİMLİK DOĞRULAMASI .....                                | 80         |
| 3.10. KRİPTOGRAFİK TEKNİKLER.....                            | 83         |
| 3.11. KRİPTOGRAFİK TEKNİKLERİN KULLANIMI.....                | 91         |
| 3.11.1. Oy Kullanma İşlemi .....                             | 91         |
| 3.11.2. Oy Doğrulama Süreci.....                             | 101        |
| 3.11.3. Oyun Saklanması.....                                 | 103        |
| 3.11.4. Seçim Sonuçlarının Hesaplanması .....                | 105        |
| 3.12. AKILLI SÖZLEŞMELER.....                                | 107        |
| 3.12.1. Kimlik Doğrulama ve Yetkilendirme.....               | 108        |
| 3.12.2. Yetkili İşlemleri.....                               | 109        |
| 3.12.3. Oylama Süreci .....                                  | 111        |
| 3.12.4. Diamod Pattern (EIP-2535).....                       | 112        |
| 3.12.5. Snapshot Sistemi .....                               | 115        |
| 3.12.6. Oy Doğrulama İşlemleri.....                          | 116        |
| 3.13. GELİŞTİRME VE TEST ORTAMI .....                        | 117        |
| 3.14. TEST YÖNTEMİ .....                                     | 118        |
| <b>4. BULGULAR .....</b>                                     | <b>121</b> |
| 4.1. MALİYET ANALİZİ .....                                   | 121        |
| 4.1.1. Sistem Dağıtım ve Başlangıç Maliyetleri .....         | 123        |
| 4.1.2. Yetkili Tabanlı İşlem Maliyetleri .....               | 124        |
| 4.1.3. Seçmen Tabanlı İşlem Maliyetleri .....                | 125        |
| 4.2. PERFORMANS ANALİZİ.....                                 | 126        |
| 4.2.1. Oy Verme Sürecinin Bütünleşik Performans Ölçümü ..... | 126        |

|                                                                          |            |
|--------------------------------------------------------------------------|------------|
| 4.2.2. Kriptografik Hash Üretim Performansı.....                         | 129        |
| 4.2.3. Kanıt Üretme Performans Analizi.....                              | 131        |
| 4.2.4. Kanıt Doğrulama Performans Analizi.....                           | 134        |
| 4.2.5. Paillier Tabanlı Oy Verisi Şifreleme Performans Analizi.....      | 143        |
| 4.2.6. Paillier Şifreleme Algoritması Sistemdeki Zaman Karmaşıklığı..... | 147        |
| <b>5. TARTIŞMA.....</b>                                                  | <b>149</b> |
| <b>6. SONUÇ VE ÖNERİLER .....</b>                                        | <b>156</b> |
| <b>KAYNAKLAR.....</b>                                                    | <b>158</b> |
| <b>İNTİHAL RAPORU İLK SAYFASI .....</b>                                  | <b>164</b> |
| <b>ETİK KURUL İZİN YAZISI .....</b>                                      | <b>165</b> |
| <b>KURUM İZNİ YAZILARI.....</b>                                          | <b>166</b> |
| <b>ÖZGEÇMİŞ .....</b>                                                    | <b>167</b> |

## ŞEKİL LİSTESİ

### Sayfa No

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Şekil 2. 1. Merkezi ağ tasarımı.....                                                          | 11 |
| Şekil 2. 2. Merkezi olmayan ağ tasarımı. ....                                                 | 12 |
| Şekil 2. 3. Dağıtık ağ tasarımı. ....                                                         | 12 |
| Şekil 2. 4. Görüntü, ikinci ön görünüş ve çakışma direnci. ....                               | 15 |
| Şekil 2. 5. Sıfır bilgi ispatı için kanıtlayıcı ve doğrulayıcı gösterimi.....                 | 15 |
| Şekil 2. 6. Bitcoin’de kullanılan blokzinciri modeli. ....                                    | 18 |
| Şekil 2. 7. Merkle Tree ve Mekle Root modeli. ....                                            | 22 |
| Şekil 2. 8. Merkle Tree veri yapısının örnek modeli.....                                      | 24 |
| Şekil 2. 9. Ethererum için kullanılan blok modeli.....                                        | 29 |
| Şekil 2. 10. Ethereum blok yapısında header bölümü. ....                                      | 30 |
| Şekil 3. 1. Seçmen aktörü için kullanım durumu (use case) diyagramı. ....                     | 41 |
| Şekil 3. 2. Oylama yetkilisi için kullanım senaryosu (use case) diyagramı. ....               | 43 |
| Şekil 3. 3. Oylama sisteminin aşamalarını gösteren model.....                                 | 50 |
| Şekil 3. 4. Oylama sisteminin kurulumunu gösteren diyagram. ....                              | 51 |
| Şekil 3. 5. Java ile geliştirilen MetaMask uyumlu Ethereum cüzdan üretici program.....        | 52 |
| Şekil 3. 6. Seçmenlerin ve seçim otoritesi için cüzdan üretilmesi ve sisteme aktarılması..... | 52 |
| Şekil 3. 7. MetaMask uyumlu Ethereum cüzdan üretilmesi. ....                                  | 53 |
| Şekil 3. 8. Başlangıçta seçim otoritelerin belirlenmesini sağlayan JSON dosyası. ....         | 53 |
| Şekil 3. 9. Başlangıçta fakültelerin belirlenmesini sağlayan JSON dosyası. ....               | 54 |
| Şekil 3. 10. Başlangıçta bölümlerin belirlenmesini sağlayan JSON dosyası. ....                | 54 |
| Şekil 3. 11. Başlangıçta seçmenlerin belirlenmesini sağlayan JSON dosyası. ....               | 54 |

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| Şekil 3. 12. Hardhat ile blokzinciri ağının oluşturulması.....                  | 55 |
| Şekil 3. 13. Circom ile geliştirilen devrenin derlenmesi. ....                  | 55 |
| Şekil 3. 14. Akıllı sözleşmelerin derlenmesi ve dağıtılması. ....               | 56 |
| Şekil 3. 15. Seçim yetkilisi paneli ana sayfa arayüzü. ....                     | 57 |
| Şekil 3. 16. Seçmen paneli ana sayfa arayüzü.....                               | 58 |
| Şekil 3. 17. Seçim yetkilisi panelinde seçmen yönetimi arayüzü. ....            | 59 |
| Şekil 3. 18. Seçim yetkilisi panelinde seçmen ekleme işlemi. ....               | 60 |
| Şekil 3. 19. Seçim yetkilisi paneli fakülte ve bölüm yönetimi arayüzü. ....     | 60 |
| Şekil 3. 20. Seçim yetkilisi paneli yetkili yönetimi arayüzü. ....              | 61 |
| Şekil 3. 21. Seçim yetkilisi paneli seçim yönetimi arayüzü.....                 | 62 |
| Şekil 3. 22. Seçim yetkilisi paneli seçim ekleme arayüzü. ....                  | 63 |
| Şekil 3. 23. Seçim yetkili paneli seçim detayları arayüzü.....                  | 64 |
| Şekil 3. 24. Seçmen ve seçim yetkilisi giriş arayüzü. ....                      | 65 |
| Şekil 3. 25. Seçmen paneli ana sayfasında aktif seçimlerin görüntülenmesi. .... | 66 |
| Şekil 3. 26. Seçmen paneli üzerinden seçim listesinin görüntülenmesi.....       | 66 |
| Şekil 3. 27. Seçmen paneli üzerinden oy kullanma arayüzü. ....                  | 67 |
| Şekil 3. 28. Seçmen paneli başarılı oy kullanma arayüzü. ....                   | 68 |
| Şekil 3. 29. Seçmen paneli seçim sonuçları arayüzü.....                         | 69 |
| Şekil 3. 30. Seçim yetkilisi paneli seçim detayları arayüzü. ....               | 70 |
| Şekil 3. 31. Seçmen paneli seçim sonucu arayüzü. ....                           | 71 |
| Şekil 3. 32. Seçim yetkilisi panel seçim sonuçları arayüzü.....                 | 72 |
| Şekil 3. 33. Seçmen paneli seçim doğrulama arayüzü. ....                        | 73 |
| Şekil 3. 34. Seçim yetkilisi paneli seçim kontrolü arayüzü.....                 | 74 |
| Şekil 3. 35. Oylama işleminin akış diyagramı. ....                              | 76 |
| Şekil 3. 36. Oylama sisteminin sistem mimarisi. ....                            | 78 |
| Şekil 3. 37. Seçim otoritesi süreçlerine ait sistem mimarisi diyagramı. ....    | 78 |
| Şekil 3. 38. Oy kullanma sürecine ait sistem mimarisi diyagramı. ....           | 79 |

|                                                                                                  |     |
|--------------------------------------------------------------------------------------------------|-----|
| Şekil 3. 39. Arayüzün yapısını gösteren şema. ....                                               | 80  |
| Şekil 3. 40. Kimlik doğrulama sürecine ait sıralama diyagramı (sequence diagram).....            | 82  |
| Şekil 3. 41. Şifrelenmemiş 10 elemanlı vektör gösterimi. ....                                    | 85  |
| Şekil 3. 42. Paillier şifreleme algoritmasıyla şifrelenecek 10 elemanlı vektör gösterimi. ....   | 86  |
| Şekil 3. 43. Paillier şifreleme algoritmasıyla şifrelenmiş 10 elemanlı vektör gösterimi. ....    | 86  |
| Şekil 3. 44. Circom devresinde kullanılan aşamalar. ....                                         | 88  |
| Şekil 3. 45. Circom devresinin açık ve gizlik parametreleri. ....                                | 88  |
| Şekil 3. 46. Circom devresindeki commitment kontrolü. ....                                       | 89  |
| Şekil 3. 47. Circom devresindeki oy ve aday kontrolü. ....                                       | 90  |
| Şekil 3. 48. Circom devresindeki zaman doğrulaması. ....                                         | 90  |
| Şekil 3. 49. Circom devresindeki nullifier üretimi ve doğrulaması. ....                          | 90  |
| Şekil 3. 50. Circom devresinin tüm kontrollerinin sonucu.....                                    | 91  |
| Şekil 3. 51. Circom devresinin ana bileşen tanımlanması.....                                     | 91  |
| Şekil 3. 52. Oylama işlemini gösteren sıralama diyagramı (sequence diagram). ....                | 92  |
| Şekil 3. 53. Commitment oluşturulmasına yönelik örnek JavaScript fonksiyonu. ....                | 93  |
| Şekil 3. 54. Commitment üreten JavaScript fonksiyonu örneği.....                                 | 94  |
| Şekil 3. 55. Nullifier oluşturulmasına yönelik örnek JavaScript fonksiyonu.....                  | 94  |
| Şekil 3. 56. Nullifier oluşturan JavaScript kodu örneği. ....                                    | 94  |
| Şekil 3. 57. Paillier algoritmasıyla şifreli oy üreten JavaScript fonksiyonu. ....               | 95  |
| Şekil 3. 58. Şifreleme işlemini gerçekleştiren JavaScript örneği. ....                           | 95  |
| Şekil 3. 59. Oy vektörünü üreten JavaScript fonksiyonu. ....                                     | 96  |
| Şekil 3. 60. Circom devresinin derlenmesi sonrası oluşturulan dosyalar.....                      | 96  |
| Şekil 3. 61. Kanıt oluşturma süreci JavaScript bağımlılıkları. ....                              | 97  |
| Şekil 3. 62. Devrenin giriş değerlerinin hazır hale getirilmesi.....                             | 97  |
| Şekil 3. 63. Sıfır bilgi ispatı kanıtı oluşturulmasına yönelik JavaScript fonksiyon örneği. .... | 98  |
| Şekil 3. 64. JavaScript ile sıfır bilgi ispatı kanıtının doğrulanması.....                       | 99  |
| Şekil 3. 65. Oy kullanımı ve sıfır bilgi ispatı kanıtının blokzincire gönderilmesi. ....         | 100 |

|                                                                                          |     |
|------------------------------------------------------------------------------------------|-----|
| Şekil 3. 66. Oyun blokzincire kaydedilmesi için akıllı sözleşmeye doğrudan gönderilmesi. | 101 |
| Şekil 3. 67. Oyun blokzincire kaydedilmesi için relay servisine gönderilmesi. ....       | 101 |
| Şekil 3. 68. Akıllı sözleşme tarafından commitment değerinin doğrulanması.....           | 102 |
| Şekil 3. 69. Akıllı sözleşme tarafından nullifier değerinin kontrolü. ....               | 102 |
| Şekil 3. 70. Akıllı sözleşme tarafından sıfır bilgi ispatı kanıtının doğrulanması. ....  | 103 |
| Şekil 3. 71. Şifrelenmiş oy saklama yapısı.....                                          | 104 |
| Şekil 3. 72. Oyun kaydedilmesi için kullanılan örnek Solidity kodu.....                  | 104 |
| Şekil 3. 73. Birden fazla şifreli oy vektörünün homomorfik toplanma işlemi.....          | 105 |
| Şekil 3. 74. Homomorfik oy toplama işlemi. ....                                          | 106 |
| Şekil 3. 75. Seçim sonuçları için vektörün şifre çözümü. ....                            | 106 |
| Şekil 3. 76. Şifrelenmiş verinin çözülmesini sağlayan algoritma. ....                    | 107 |
| Şekil 3. 77. Seçim sonuçlarının kaydedilmesi işlemi.....                                 | 107 |
| Şekil 3. 78. Yetkili kimlik doğrulama sürecine ait akıllı sözleşme işlemleri.....        | 108 |
| Şekil 3. 79. Seçmen kimlik doğrulama sürecine ait akıllı sözleşme işlemleri. ....        | 108 |
| Şekil 3. 80. Seçmen yönetimi sürecine ait akıllı sözleşme işlemleri.....                 | 109 |
| Şekil 3. 81. Seçim yönetimi sürecine ait akıllı sözleşme işlemleri. ....                 | 110 |
| Şekil 3. 82. Yetkili yönetimi sürecine ait akıllı sözleşme işlemleri. ....               | 110 |
| Şekil 3. 83. Fakülte ve bölüm yönetimi sürecine ait akıllı sözleşme işlemleri. ....      | 111 |
| Şekil 3. 84. Oylama sürecine ait akıllı sözleşme işlemleri.....                          | 112 |
| Şekil 3. 85. Seçim işlemleri için Diamond Pattern dosya hiyerarşisi .....                | 113 |
| Şekil 3. 86. Diamond pattern için sıralama diyagramı (sequence diyagram).....            | 113 |
| Şekil 3. 87. Diamond pattern uygulamasında ortak bir depolama kullanımı.....             | 114 |
| Şekil 3. 88. Oy doğrulama sürecine ait akıllı sözleşme işlemleri. ....                   | 116 |

## TABLO LİSTESİ

### Sayfa No

|                                                                                                             |     |
|-------------------------------------------------------------------------------------------------------------|-----|
| <b>Tablo 2. 1.</b> Merkezi, merkezi olmayan ve dağıtık ağların karşılaştırılması.....                       | 11  |
| <b>Tablo 2. 2.</b> Simetrik ve asimetrik anahtar algoritmalarının temel farkları.....                       | 14  |
| <b>Tablo 2. 3.</b> Blokzinciri türlerinin temel farkları.....                                               | 18  |
| <b>Tablo 2. 4.</b> Blokzinciri evriminin tarihsel evrimi ve karşılaştırması. ....                           | 27  |
| <b>Tablo 2. 5.</b> Kâğıt tabanlı ve elektronik oylama sistemlerinin karşılaştırması.....                    | 35  |
| <b>Tablo 2. 6.</b> Merkezi ve merkezi olmayan oylama sistemlerinin karşılaştırılması. ....                  | 37  |
| <b>Tablo 3. 1.</b> Çalışmada kullanılan teknolojilerin tanımları ve kullanım amaçları.....                  | 45  |
| <b>Tablo 3. 2.</b> Paillier şifreleme algoritması için kullanılan rastgele değerler. ....                   | 86  |
| <b>Tablo 3. 3.</b> Circom devresinde kullanılan gizli (private) parametreler.....                           | 89  |
| <b>Tablo 3. 4.</b> Circom devresinde kullanılan açık (public) parametreler. ....                            | 89  |
| <b>Tablo 4. 1.</b> Gaz ve para birimi dönüşüm parametreleri (06 Haziran 2025).....                          | 122 |
| <b>Tablo 4. 2.</b> Akıllı sözleşmelerin dağıtılmasında oluşan maliyet. ....                                 | 123 |
| <b>Tablo 4. 3.</b> Akıllı sözleşmelerin dağıtılmasında seçmen eklenmesi için oluşan maliyet.....            | 124 |
| <b>Tablo 4. 4.</b> Yetkili yönetimi için oluşan sistemsel maliyet.....                                      | 124 |
| <b>Tablo 4. 5.</b> Seçmen yönetimi için oluşan sistemsel maliyet. ....                                      | 124 |
| <b>Tablo 4. 6.</b> Fakülte ve bölüm yönetimi için oluşan sistemsel maliyet.....                             | 125 |
| <b>Tablo 4. 7.</b> Seçim yönetimi için oluşan sistemsel maliyet.....                                        | 125 |
| <b>Tablo 4. 8.</b> Seçim işlemleri için oluşan sistemsel maliyet. ....                                      | 126 |
| <b>Tablo 4. 9.</b> Akıllı sözleşme tabanlı 32-bit Paillier ile oy kullanma performans analizi. ....         | 127 |
| <b>Tablo 4. 10.</b> JavaScript tabanlı 2048-bit Paillier ile oy kullanma performans analizi.....            | 128 |
| <b>Tablo 4. 11.</b> JavaScript tabanlı eşzamanlı 2048-bit Paillier ile oy kullanma performans analizi. .... | 129 |

|                                                                                                                                                           |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>Tablo 4. 12.</b> Poseidon hash fonksiyonunun oy kullanma performans analizi .....                                                                      | 130 |
| <b>Tablo 4. 13.</b> Poseidon hash fonksiyonu kullanılarak gerçekleştirilen eşzamanlı hash üretimi.<br>.....                                               | 131 |
| <b>Tablo 4. 14.</b> JavaScript ile kanıt üretiminin sistem üzerindeki analizi. ....                                                                       | 132 |
| <b>Tablo 4. 15.</b> JavaScript ile gerçekleştirilen eşzamanlı kanıt üretiminin sistem üzerindeki<br>analizi. ....                                         | 133 |
| <b>Tablo 4. 16.</b> JavaScript ile kanıt doğrulamanın sistem üzerindeki analizi.....                                                                      | 135 |
| <b>Tablo 4. 17.</b> JavaScript ile eşzamanlı kanıt doğrulamanın sistem üzerindeki analizi.....                                                            | 136 |
| <b>Tablo 4. 18.</b> Akıllı sözleşmeyle kanıt doğrulamanın sistem üzerindeki analizi. ....                                                                 | 138 |
| <b>Tablo 4. 19.</b> Akıllı sözleşmeyle eşzamanlı kanıt doğrulamanın sistem üzerindeki analizi. ..                                                         | 139 |
| <b>Tablo 4. 20.</b> JavaScript ve akıllı sözleşmenin kanıt doğrulamasının karşılaştırmalı analizi. ....                                                   | 141 |
| <b>Tablo 4. 21.</b> JavaScript ve akıllı sözleşmenin aracılığıyla gerçekleştirilen eşzamanlı kanıt<br>doğrulama süreçlerinin karşılaştırmalı analizi..... | 142 |
| <b>Tablo 4. 22.</b> Paillier şifreleme algoritmasının sistem üzerindeki analizi. ....                                                                     | 144 |
| <b>Tablo 4. 23.</b> Paillier şifreleme algoritmasının eşzamanlı sistem üzerindeki analizi. ....                                                           | 145 |

## SİMGE VE KISALTMA LİSTESİ

| Kısaltmalar     | Açıklama                                                        |
|-----------------|-----------------------------------------------------------------|
| <b>ZKP</b>      | : Zero-Knowledge Proof                                          |
| <b>PoW</b>      | : Proof-of-Work                                                 |
| <b>Dapp</b>     | : Decentralized Application                                     |
| <b>P2P</b>      | : Peer-to-Peer                                                  |
| <b>HTTP</b>     | : Hyper-Text Transfer Protocol                                  |
| <b>SMTP</b>     | : Simple Mail Transfer Protocol                                 |
| <b>FTP</b>      | : File Transfer Protocol                                        |
| <b>PoW</b>      | : Proof of Work                                                 |
| <b>PoS</b>      | : Proof of Stake                                                |
| <b>EVM</b>      | : Ethereum Virtual Machine                                      |
| <b>DeFi</b>     | : Decentralized Finance                                         |
| <b>NFT</b>      | : Non-Fungible Tokens                                           |
| <b>DAO</b>      | : Decentralized Autonomous Organization                         |
| <b>MPT</b>      | : Merkle Patricia Trie                                          |
| <b>AES</b>      | : Advanced Encryption Standard                                  |
| <b>DES</b>      | : Data Encryption Application                                   |
| <b>3DES</b>     | : Triple DES                                                    |
| <b>RSA</b>      | : Rivest-Shamir-Adleman                                         |
| <b>ECC</b>      | : Elliptic Curve Cryptography                                   |
| <b>ZK-SNARK</b> | : Zero-Knowledge Succinct Non-Interactive Argument of Knowledge |
| <b>ZK-STARK</b> | : Zero-Knowledge Scalable Transparent Argument of Knowledge     |
| <b>LDE</b>      | : Low-Degree Extension                                          |
| <b>ECDSA</b>    | : Elliptic Curve Digital Signature Algorithm                    |
| <b>SPA</b>      | : Single Page Application                                       |
| <b>ETH</b>      | : Ether                                                         |
| <b>USD</b>      | : United States Dollars                                         |
| <b>TL</b>       | : Türk Lirası                                                   |

## ÖZET

### [YÜKSEK LİSANS TEZİ]

#### [BLOKZİNCİR TABANLI MERKEZİ OLMAYAN ÜNİVERSİTE ÖLÇEKLİ OYLAMA SİSTEMİ ]

[Ahmet Can ÇİÇEK]

**İstanbul Üniversitesi-Cerrahpaşa**

**Lisansüstü Eğitim Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı**

**[Danışman : Prof. Dr. Muhammed Ali AYDIN]**

Demokratik toplumların temel unsurlarından biri, adil ve şeffaf seçim süreçleridir. Bu süreçlerin güvenilirliği ise doğrudan oylama sisteminin işleyişine bağlıdır. Oylama sistemi, oylamanın nasıl yapılacağını ve sonuçların nasıl değerlendirileceğini belirleyen kurallar ve yöntemler bütünüdür. Basit bir karardan, siyasi süreçlere kadar geniş bir kullanım alanına sahiptir. Oylama sistemlerinin daha güvenli, şeffaf, katılımcı ve erişilebilir hale getirilmesi amacıyla dijitalleşmenin sunduğu yeniliklerden yararlanılarak çeşitli çözümler geliştirilmiştir. Ancak, bu çözümler güvenilirlik, şeffaflık, doğrulanabilirlik ve gizlilik kriterleri açısından tartışma konusu olmuş ve çözülmesi zor teknik problemlere yol açmıştır. Bu nedenle, tez kapsamında blokzinciri teknolojisinin sunduğu yenilikçi özelliklerden yararlanarak mevcut sorunlara sürdürülebilir ve esnek bir çözüm geliştirilmesi hedeflenmektedir.

Ethereum blokzinciri platformu kullanılarak üniversite ölçeğinde merkezi olmayan, şeffaf ve güvenli bir oylama sistemi teorik ve pratik olarak geliştirilmiş ve bu sistem belirli metrikler ışığında maliyet ile performans açısından değerlendirilmiştir. Pratik çalışmalar kapsamında geliştirilen prototipte, kullanıcı kimliklerinin sistemde tanımlanabilmesi amacıyla MetaMask ve Ethereum uyumlu bir cüzdan oluşturma aracı geliştirilmiştir. Bu araç vasıtasıyla kullanıcılar birer açık anahtar (public key), özel anahtar (private key), anımsatıcı (mnemonic) ve Ethereum adresi elde edebilmektedir. Ethereum adresleri, oylama sistemi içerisinde kullanıcıları tanımlayan ama gerçek hayattaki kimlikleriyle ilişkilendirilemeyen bir değer olarak kullanılmaktadır. Kullanıcılar, özel anahtarlarını (private key) tarayıcı üzerindeki MetaMask içerisine dahil (import) ederek oylama sistemine giriş yapabilmektedir.

Seçmen gizliğinin cüzdan ile sağlanması, tek başına yeterli bir kriter değildir. Bu nedenle, kullanılan oy ile seçmen adresinin ilişkilendirilmesini engelleyecek mekanizmalara ihtiyaç duyulmaktadır. Bu kapsamda, oy gizliği için Paillier şifreleme algoritmasıyla birlikte sıfır

bilgi ispatı (ZKP) yönteminden yararlanılmış ve zk-SNARK protokolü tercih edilmiştir. Bu yöntemin tercih edilmesinin nedeni ise; düşük işlem maliyeti ve hızlı doğrulama sürecidir. Oylama sistemi içerisinde kimlik doğrulama, oy verme ve sonuçların doğrulanması süreçleri; cüzdan üretim aracı, MetaMask, Paillier şifreleme algoritması ve sıfır bilgi ispatının (ZKP) bir bütün olarak çalışmasıyla elde edilmektedir. |

Haziran 2025 , |188| sayfa.

**Anahtar kelimeler:** | blokzinciri, sıfır bilgi ispatı, oy, zk-snark, merkezi olmayan, paillier, şifreleme, , dağıtık, seçim, e-seçim |



## **ABSTRACT**

**[M.Sc. THESIS]**

### **[BLOCKCHAIN-BASED DECENTRALIZED UNIVERSITY-SCALE VOTING SYSTEM ]**

**[Ahmet Can ÇİÇEK]**

**İstanbul University-Cerrahpaşa**

**Institute of Graduate Studies**

**Department of Computer Engineering**

**Computer Engineering Programme**

**[Supervisor : Prof. Dr. Muhammed Ali AYDIN ]**

One fundamental component of democratic societies is fair and transparent election processes. The reliability of these processes is directly linked to the functioning of the voting system. A voting system encompasses rules and methods that determine how votes are cast and results are evaluated. Its applications range from simple decision-making to complex political processes. To improve the security, transparency, and accessibility of voting systems, various solutions have been developed utilizing digital innovations. However, these solutions have raised questions regarding their reliability, transparency, verifiability, and confidentiality, and have given rise to technical problems that are difficult to resolve. Therefore, this thesis aims to develop a sustainable and flexible solution by leveraging the innovative features offered by blockchain technology.

Using the Ethereum blockchain platform, we theoretically and practically developed a decentralized, transparent, and secure voting system for university-scale applications. Our prototype was evaluated in terms of cost-effectiveness and performance metrics. A MetaMask-compatible wallet creation tool was also developed to enable users to define their identities within the system using public keys, private keys, mnemonic phrases, and Ethereum addresses. Ethereum addresses are used to identify users within the voting system without linking them to their real-life identities. Users can enter the voting system by importing their private keys into MetaMask.

While a wallet-based solution provides some level of anonymity, it is not sufficient on its own. Therefore, mechanisms must be designed to prevent linking votes to user addresses. To achieve this, we employed the Paillier homomorphic encryption algorithm and zero-

knowledge proof (ZKP) methods, specifically zk-SNARKs. The reason for choosing this approach was its low computational cost and fast verification process. The identity verification, voting, and result validation processes within our system are achieved through a combination of wallet creation tools, MetaMask, Paillier encryption algorithms, and ZKPs. |

June 2025, |188| pages.

**Keywords:** |blockchain, zero-knowledge proof, vote, zk-snark, decentralized, paillier, encryption, distributed, election, e-voting|





## 1. GİRİŞ

Seim, tm demokratik kurum ve kuruluřların temelini oluřturan bir unsurdur. Bir veya birden fazla seenek arasından, bir veya birkaının tercih edilmesi iin gerekleřen sreci ifade eder. Oylama ise belirli bir ama iin bir araya gelen bireylerin, belirli bir kararı veya konuyu birlikte belirlemek amacıyla bařvurdıkları bir eylemdir. Oylama sistemi, oylamanın nasıl yapılacaėını ve sonuların nasıl deėerlendirileceėini belirleyen kurallar ve yntemler btndr. Kısaca, seim karar verme srecini ve oylama yapılan iřlemi ifade eder. Oylama iřleminin doėru ve gvenli bir řekilde gerekleřtirilmesi, seim srelerinin bařarısını doėrudan etkilemektedir. Tarihten gnmze demokrasi kavramının nem kazanmasıyla otokratik ynetim anlayıřı yerini demokrasiye bırakmıř ve seim sreci bu ynetim biiminin en temel yapı tařlarından birisi olmuřtur. Bu baėlamda demokratik yapıların geliřimi paralel olarak seim sistemlerinin geleceėini řekillendirmiřtir. Fakat demokratik yapılardan baėımsız olarak tarihte pek ok toplum ve uygarlıklar, seimleri bir ara olarak kullanmıřtır. Antik Yunan ve Roma İmparatorluėu'nda bunun ilk rnekleri grlebilir. rneėin, Antik Yunan'da M.. 5. yzyılda, tarihte Atina Demokrasisi olarak tanımlanan ve en erken demokrasi uygulaması olarak gsterilen bir ynetim biimi ortaya ıkmıřtır [1]. Bu ynetim biimiyle birlikte kararların alınmasında seimlerden yararlanılmıřtır. Bylece toplumdaki vatandaşlar sz hakkına sahip olmuř ve demokrasinin ilk rnekleri uygulanmaya bařlanmıřtır. Demokrasi kelimesi de kkeni itibariyle halk anlamına gelen Demos ve iktidar anlamına gelen kratos szcklerinden tretilmiřtir. Aynı yzyılda Roma İmparatorluėu'nda en yksek devlet grevlilerini semek iin yapılan Konsl Seimleri de tarihteki ilk seimlere rnek olarak verilmektedir. Bu sayede soylu ve soylu olmayanlar arasındaki ıkar farklılıklarının giderilmesi amalanmıřtır [1]. Bu baėlamda seimlerin ortak hareket edebilmek iin tarihte kullanılan aralardan biri olduėu grlmektedir. 1789 yılında ise, Fransız Devrimi, seimler ve demokratik ilkelerin dnya genelinde egemenlik kazanmasında nemli bir dnm noktası olmuřtur. Bu demokratik ilkelere sahip olan ynetim sistemlerinin en nemli gstergesi demokratik seimlerdir [2]. Bu demokratik seimlerin srdrlebilmesi iin de srecin bařarılı gerekleřtirilebilmesi ve oylama iřleminin gvenirliėinin olması gerekmektedir.

Teknolojideki geliřmeler, bireylerin ve toplumların yařam biimlerini her alanda etkilemiřtir. Dijitalleřme, iletiřimde hız ve bilgiye eriřim kolaylıėı insanların yařam

biçimlerinde, iletişim yöntemlerinde ve çalışma biçimlerinde bir dönüşüme neden olmuştur. Geleneksel sistemlerin, dijital hale dönüştürülmesiyle birlikte üretilen çıktıların daha hızlı, verimli ve tutarlı olması sağlanmıştır. Bu süreçlerden etkilenen en önemli parça seçim süreçleridir. Seçim süreçlerinin daha güvenli, şeffaf, katılımcı ve erişilebilir hale getirilmesi için dijitalleşmenin sunmuş olduğu yeniliklerden çözümler üretilmiş ve üretilmeye devam edilmektedir. Son 20 yıllık süreç incelendiğinde birçok ülkenin ve kurumun seçimlerinin dijital dönüşümü sağlanmış ve oylamaların dijital ortamlarda gerçekleştirildiği gözlemlenmiştir. Bu dijital seçim süreçleri için geliştirilen oylama sistemleri incelendiğinde, üretilen çözümlerin yetersiz kaldığı yetkililer tarafından tespit edilmiş ve bu yetersizliklerin aşılabilmesi için zaman içerisinde var olan çözümlere yeni geliştirmeler eklenmiştir. Tasarlanan sistemlerdeki yetersizlikler, mimari eksiklikler, demokratik kriterlerin olmaması ve gelişen teknolojinin ortaya çıkardığı yeni durumlara var olan oylama sistemlerin çözüm sunamaması genel sorunlar olarak görülmektedir. Bu sorunlara rağmen çoğu ülke veya kurum, seçim maliyetlerini düşürmek ve hızlı sonuç alınmasını sağlamak için dijital seçim sürecine ilgi göstermektedir ve elektronik oylama sistemlerini kullanmaya başlamıştır [3]. Fakat bu dijitalleşen seçim süreçleri, Web 2.0 kavramı altındaki teknolojileri kullanmaktadır. Bu sistemlerin genellikle merkezi bir yapıya sahip olması, sistemle ilgili şeffaflık ve güvenlik sorunlarını ortaya çıkarmaktadır. Fransa ve İsveç gibi ülkelerde yapılan dijital oylama sürecinde yaşanan teknik ve sosyal sorunlar böylesi durumlara birer örnektir [5]. Bu yüzden dijital oylama sistemleri kullanan ülkelerin bazıları tekrar oy pusulası, zarf ve sandık barındıran geleneksel oylama sistemine dönmüştür. Fakat oy pusulası, zarf ve sandık barındıran geleneksel oylama sistemindeki hız sorunları ve kandırılmaya elverişli insan faktörü sebebiyle başka sorunlar yaşanmaktadır. Geleneksel oylama sistemlerindeki ve var olan elektronik oylama sistemlerindeki sorunların tamamen çözülmemesi yeni teknolojilerle bir oylama sistemi oluşturulmasını tetikleyen bir unsurdur.

Blokszinciri (blockchain), seçim süreçlerinin dijital dünyada gerçekleştirilebilmesi için yenilikçi bir teknoloji altyapısı sunmaktadır. Blokszinciri ile geliştirilen bir oylama sisteminin en önemli amacı güvenilir, şeffaf ve hızlı bir çözüm ortaya koymasıdır. Blokszinciri teknolojisinin merkeziyetçi olmayan bir yapıya sahip olması, oy verilerinin manipüle edilmesini oldukça zorlaştırmaktadır. Ayrıca bu yapısı sayesinde herhangi bir merkezi otoritenin sisteme müdahale etmesi engellenmektedir.

## 1.1. KONU VE ÖNEMİ

Oylama sistemleri, bir araya gelen insanların ortak bir karara varabilmesi amacıyla kullanılan, kurum ve kuruluşların karar alma süreçlerinde önemli bir role sahip yöntemlerden birisi ve önemli bir araçtır. Günümüzde basit bir kararın alınmasından, siyasi bir karar alınmasına kadar geniş bir kullanım alanına sahiptir. Örneğin; aynı sitede birlikte yaşayan insanların çevre düzenlenmesinde bir karara varabilmesi amacıyla veya demokratik ülkelerin yönetimini belirlemek için oylama sistemleri yaygın olarak kullanılmaktadır. Bu bağlamda, oylama sistemleri toplumların bir arada yaşayabilmesi için önemli bir rol üstlenmektedir.

Dijitalleşmenin yaygınlaşmadığı dönemlerdeki seçimler geleneksel oylama sistemleri ile gerçekleştirilmiştir. Geleneksel olarak ifade edilen oylama sistemleri oy, zarf, pusula ve mühür gibi nesnelerin olduğu sistemleri ifade etmek için kullanılmaktadır [6]. Bu geleneksel sistemle seçim süreçlerini yöneten üniversiteler, şirketler ve devletler oylama işlemlerini sandık, pusula, mühür ve zarf gibi nesnelerle gerçekleştirmek zorundadır. Öte yandan, teknolojiadaki gelişmeler neticesinde dijitalleşme her alanı etkilemiş ve bu durumun da oylama sistemlerine etkileri olmuştur. Bu etkiler neticesinde geleneksel oylama sistemleri yerine dijital yöntemlerin oylama süreçlerini gerçekleştirmek için tercih edildiği görülmektedir [5, 7, 12]. Bu kapsamda, oylama sistemleri, dijitalleşmesinin ilk aşamasını oluşturan elektronik oy makinelerinden, günümüzde internet üzerinden yapılabilecek aşamaya evrilmiştir. Bu bağlamda, genel seçimini 2007 yılında internet üzerinden gerçekleştiren Estonya, internet vasıtasıyla elektronik oylamayı genel seçimler için kullanmıştır [5, 8, 9]. Estonya vatandaşları, halen pek çok oy kullanma işlemini mobil cihazları üzerinden gerçekleştirmektedir [9].

Estonya gibi elektronik ve internet tabanlı oylama sistemlerini uygulayan başka ülkeler de bulunmaktadır [3]. Bazı ülkeler, elektronik oy makineleriyle, bazıları da internet tabanlı çalışan mobil ve web platformlarını kullanan uygulamalarla bu süreci gerçekleştirmektedir. Gelişen teknolojiyle birlikte, bu alana yeni yatırım yapmak için çalışma başlatan ülkelerde her geçen gün artmaktadır. Fakat elektronik oylama sistemlerine geçiş, tartışmaları da beraberinde getirmiş, bunun neticesinde güvenlik ile ilgili yaşanan endişeler nedeniyle Fransa, İsviçre ve Norveç gibi bazı ülkeler başlattıkları projeleri iptal etmişlerdir [5, 9, 42]. Bir oylama sisteminde, güvenlik açıkları, manipülasyon ihtimali, şeffaflık eksikliği ve gizlilik sorunları sisteme olan güveni önemli ölçüde etkilemektedir. Bir ülkenin oylama sisteminin, dijital dönüşüm sürecinden geçmesi için toplumsal bilgi ve birikimin uygun şartlarda olması, bu sistemin uygulanma sürecine olan güvenin, sistemi uygulayanlar tarafından sağlanması gerekmektedir.

Bu tez, blokzinciri teknolojisinin getirdiği yeni özellikleri kullanarak tüm sorunlara yenilikçi, sürdürülebilir ve esnek bir çözüm sunmayı kapsamaktadır. Tezin konusu, Ethereum blokzinciri platformu kullanılarak üniversite ölçeğinde merkezi olmayan, şeffaf ve güvenli bir oylama sisteminin teorik ve pratik olarak geliştirilmesi, bu sistemin belirli metriklerle performans ve güvenlik açısından değerlendirilmesidir. Merkezi olmayan bir oylama sistemi geliştirebilmek için blokzinciri önemli bir altyapı sağlamaktadır. Blokzincirinin sağladığı özetleme algoritmaları, şifreleme teknikleri ve değiştirilemezlik özelliği, güvenlik ve veri bütünlüğü açısından avantajlar sağlamaktadır. Şeffaflık ve gizlilik endişesi oluşturan durumlara bu özellikleri sayesinde çözüm sunmaktadır [12].

Dijital oylama sistemleriyle ilgili yapılan çalışmalar genel olarak teorik modellemelerle ilgidir. Fakat, bu çalışmayla geliştirilen teorik modelleme uygun bir prototiple uygulamaya dönüştürülmektedir. Böylece, bu çalışma ile blokzinciri tabanlı bir oylama sisteminin geliştirilmesi, literatüre teorik ve pratik anlamda önemli katkılar sunmaktadır. Bu çalışma, Ethereum blokzinciri platformu sayesinde oylama işlemlerinin, akıllı sözleşmelerle daha otonom hale getirilmesini sağlamış ve insan faktörünün gerçekleştirdiği hataların ortadan kaldırılması konusunda önemli bir örnek ortaya koyulmasını sağlamaktadır. Ayrıca, bu çalışma aracılığıyla Ethereum platformu kullanılarak blokzinciri tabanlı bir oylama sisteminin uygulanabilirliğinin mümkün olduğu ortaya konmuştur.

Bu çalışmayla geliştirilen prototip, mevcut dijital oylama sistemleriyle karşılaştırmalı olarak değerlendirilmiş ve aralarındaki farklar ortaya konmuştur. Prototipin geliştirilme, kullanım ve test süreçlerinde elde edilen metrikler ışığında sistemin maliyet ve performans açısından değerlendirmesi literatür açısından önemli bilgiler sunmaktadır. Bu kapsamda, performans verileri, çalışma hızı, kullanılan gaz miktarı ve maliyet analizi bundan sonraki yapılacak yeni çalışmalara önemli veriler sunmaktadır.

Merkezi yapıya sahip dijital oylama sistemlerinden farklı olarak blokzinciri tabanlı geliştirilen bu prototip, blokzinciri teknolojisinin getirdiği tüm özellikleri kullanmaktadır. Oylar şifrelenmiş ve şifreli oylar üzerinden seçim sonuçlarının hesaplanması gerçekleştirilmiştir. Doğrulayıcı ve onaylayıcı olarak adlandırılan iki tarafın bulunduğu bir sistemde, bir tarafın elindeki veriyi (doğrulayıcı), başka bir tarafa (onaylayıcı), elindeki ifade hakkında herhangi bir bilgi vermeden kanıtlamasını sağlayan sıfır bilgi ispatından (ZKP) yararlanılmıştır [13]. Böylece şifreli oylar görülmeden doğrulanabilir hale gelmiştir. Bu yapıyla oylama verileri

blokszinciri üzerine şeffaf bir şekilde kaydedilebilirken, oy gizliliğinin korunması ve kullanıcı kimliğinin blokszinciri üzerinde anonim bir şekilde tutulması sağlanmıştır. Blokszincirinin dağıtık yapısı, oylama sisteminin merkezi bir yerden kontrol edilmesi sorununu çözerken, ZKP’de literatürde tartışılan oy gizliliği soruna çözüm olmaktadır. Homomorfik özellikli şifreleme algoritması ise, şifrelenmiş oylar üzerinden matematiksel işlem yapılmasına izin vererek seçim sonuçlarının gizlilik ilkesiyle açıklanmasına olanak sağlamaktadır.

Bu çalışmanın toplumsal açıdan oldukça yüksek önemi bulunmaktadır. Oylama sistemlerinin dijital hale gelmesi, kullanıcılara erişim kolaylığı sağlamakta ve böylece oylamanın geniş kitlelere yayılarak katılımın artırılmasını sağlamaktadır. Şeffaf, güvenilir, doğrulanabilir ve gizlilik ilkelerine bağlı bir sistem inşa etmek, toplumun sisteme olan güvenini ve demokrasiye olan inancını artırabilmektedir. Aynı zamanda iyi tasarlanmış bir sistem, sonuçların hızlı bir şekilde hesaplanmasını mümkün hale getirmektedir. Geleneksel sistemlerdeki gibi sonuçların açıklanması için gereken süre ortadan kalmaktadır. Bu hızlı aksiyon alma yetisi, verimli bir oylama sistemi oluşturulması açısından literatüre önemli bir katkı sunmaktadır.

## 1.2. ÖZGÜN DEĞERİ

Bu çalışmayla ilk olarak teorik bir model tasarlanmış ve daha sonra da bir prototip bu modele bağlı kalarak geliştirilmiştir. Teorik çalışma, Ethereum platformunun sunmuş olduğu bir alt yapı sayesinde somut bir uygulamaya dönüştürülmüştür. Bu bağlamdaki tüm çalışmalar, literatür taraması sırasında incelenmiş ve yapılan birçok çalışmanın sadece teorik bir modellemeye sınırlı kaldığı gözlemlenmiştir. Bu açıdan değerlendirildiğinde; bu çalışma, geliştirilen uygulama sayesinde diğer çalışmalardan ayrılarak literatürdeki boşlukları doldurmayı amaçlamaktadır.

Dijital oylama sistemleri için gizlilik ve doğrulanabilirlik en önemli iki kriterdir. Bu iki kriter merkezi bir oylama sisteminde genellikle endişe konularının da başında gelmektedir [14]. Geliştirilen tüm oylama sistemlerinde gizlilik ve doğrulanabilirlik dengesini doğru tasarlayabilmek zorluk olarak araştırmacıların karşısına çıkmaktadır. Yapılan çalışmalar incelendiğinde, gizlilik ve oy doğrulanabilirliği dengesinin günümüz koşullarına uygun olmadığı tespit edilmiştir. Bu çalışmayla, ZKP kullanarak gizlilik ve doğrulanabilirlik sorunlarının yenilikçi bir yaklaşımla çözülmesi amaçlanmıştır. Aynı zamanda oyların

şifrelenmesi ve şifrelenmiş oylar üzerinde matematiksel işlemlerin yapılabilmesi sistemin güvenlik seviyesini artıran bir unsur olacaktır. Bu modelde, ZKP ile seçmenlerin kimliği gizli kalarak oyların doğru bir şekilde doğrulanabilmesi sağlanmış olurken, şifreli oylar üzerinden sonuçların hesaplanabilmesi mümkün olacaktır. Bunun yanı sıra, seçmenlerin kimlikleri açığa çıkarılmadan oylarının sonuca doğru etki edip etmediği de kullanıcılar tarafından doğrulanabilmektedir. Böylece hem gizlilik hem de doğrulanabilirlik kriterlerinin dengesi korunarak ortaya bir çözüm konmuş olmaktadır.

Oylama sistemleri modellemeleri sırasında, gerçek kullanım senaryolarında sistemin ne kadar verimli çalıştığını tespit edebilmek mümkün değildir. Bir oylama sisteminin performans, ölçeklenebilirlik ve uygulanabilirlik açısından kapsamlı şekilde test edilmesi, literatürdeki önemli bir boşluğu doldurmaktadır. Bu çalışmayla, performans ve ölçeklenebilirlik testleri prototip üzerinde gerçekleştirilip temel metrikler üzerinden sistemin değerlendirilmesi sağlanmıştır. ZKP ve blokzinciri teknolojisi ile geliştirilen bir oylama sisteminin gerçek bir senaryoda verimli kullanılıp kullanılmayacağı ile ilgili önemli metrikler sunulmaktadır. Bu konuda ilerideki çalışmalara önemli veriler sunan bir çalışma olması amaçlanmıştır.

Yapılan çalışma kapsamında bir Ethereum tabanlı ve MetaMask uyumlu kişisel cüzdan geliştirilmiştir. Bu cüzdan sayesinde oylama sistemini kullanacak seçmenler, kendi kişisel cüzdanlarında özel (private) ve açık (public) anahtarlarını oluşturabileceklerdir. Seçmenler, açık anahtarlarını sistem yetkilileriyle paylaşarak, seçmen doğrulama ve ZKP ile oy kullanma sürecini gerçekleştirebileceklerdir. Bu kişisel cüzdan ile özel ve açık anahtarların oluşturulma süreci de sistemi daha merkezi yapıdan kurtaracak böylece literatürdeki ihmal edilen önemli bir boşluğu kapatacaktır.

### 1.3. AMACI VE HEDEFLERİ

Oylama sistemleri, tarihten günümüze farklı yöntemlerle görevlerini yerine getirmektedir. Kökenleri çok eski yıllara ait olan geleneksel oylama sistemleri demokratik süreçlerin vazgeçilmez araçlarından birisi olmuştur. Fakat güvenilirlik, şeffaflık, doğrulanabilirlik ve gizlilik kriterleri açısından tartışma konusuna ve çözülmesi zor teknik problemlere neden olmuştur. Öte yandan geleneksel oylama sistemleri, oylamayı yapacak kurum ya da kuruluşların içerisinden başlatılması sebebiyle merkeziyetçi bir yapıya sahiptir. Bu haliyle otoriteye bağlı bir parçadır ve kamusal güveni etkileyen unsurların başında

gelmektedir. Geleneksel oylama sistemleri, dijitalleşmenin etkisiyle elektronik oylama sistemlerine dönüşmektedir. Elektronik oylama sistemlerine dair çeşitli örnekler günümüzde mevcut olsa da, bu sistemlerin büyük bir kısmı güvenlik, şeffaflık, doğrulanabilirlik veya ölçeklenebilirlik gibi temel kriterleri tam anlamıyla karşılayamamaktadır. Merkezi yapıya sahip elektronik seçim sistemleri, güvenlik, gizlilik, doğrulanabilirlik ve şeffaflık gibi temel kriterleri karşılamakta yetersiz kalmakta; bu durum, kamusal güvenin tesisini de oldukça güçleştirmektedir. Böylesi sistemlerde manipülasyonu engellemek ve oy gizliliğini sağlayacak çözümler oldukça sınırlı ve yetersizdir.

Geleneksel oylama sistemlerini dijital hale getirebilmek için blokzinciri teknolojisi, akıllı sözleşmeler, ZKP, özetleme ve şifreleme algoritmaları kullanmak yaşanan problemlere yenilikçi bir çözüm sağlayacaktır. Blokzincir teknolojisinin sunduğu özetleme algoritmaları, şifreleme algoritmaları ve dağıtık altyapısı, yazılan verilerin değiştirilememesini sağlamak ve böylece gerekli güvenli ortam oluşturulmaktadır. ZKP ile oy gizliği ve doğrulanabilirliği problemlerine çözümler üretilebilmektedir. Oyları gizlemek için kullanılan şifreleme algoritmasının matematiksel işlemlere izin vermesi, gizli oylar üzerinden işlem yapılabilmesine olanak sağlamaktadır. Böylece bir oylama sisteminin sunması gereken şeffaflık, güvenilirlik, doğrulanabilirlik ve gizlilik gibi 4 önemli kriter sağlanmış olmaktadır.

Bu çalışma, blokzinciri tabanlı merkezi olmayan üniversite ölçekli oylama sistemi modellemesi, prototipi ve değerlendirmek için gerekli metrikleri ortaya koymaktadır. Yapılan çalışmayla şeffaflık, güvenilirlik, doğrulanabilirlik ve gizlilik kriterlerine uygun bir oylama sisteminin modellemesi ve prototipi efektif bir şekilde sunulacaktır. Ortaya konacak verilerle geleneksel oylama sistemlerinin ve bu sistemlerin dijitalleşme sürecinde ortaya çıkmış sorunlarına etkili çözümler sunulması hedeflenmiştir. Geliştirilen oylama sisteminin, artan kullanıcı sayısı ve yüksek işlem yoğunluğuna göre verdiği performans sonuçları yapılacak sonraki çalışmalar için önemli veri kaynağı olma özelliği gösterecektir. Bu bağlamda da yapılacak çalışmalara öncü olması bu çalışmanın yapılmasının temel amaçlarından birisidir. Yapılan çalışmanın topluma katkı sağlaması da amaçlar arasındadır. Bahsi geçen dört temel kriterin karşılanması, seçmenlerin güvenlik ve şeffaflık konusundaki endişelerinin giderilmesine katkı sağlayarak, demokratik süreçlere duyulan güvenin artmasına neden olabilir. Aynı zamanda, oylama sistemlerinin dijitalleşmesiyle geleneksel sistemlere göre oy kullanma ve sonuç açıklanma süresi kısılacığından seçmen katılımında artış gözlemlenebilecektir.

#### 1.4. KAPSAMI VE SINIRLAMALARI

Bu tez, blokzinciri teknolojiyle merkezi olmayan üniversite ölçeğinde bir oylama sisteminin tasarımı, prototip uygulamasını ve performans analizini kapsamaktadır. Ortaya konan çalışma, geleneksel oylama yöntemlerinin ve mevcuttaki elektronik oylama sistemlerinin çözüm sağlamakta zorlandığı şeffaflık, güvenilirlik, doğrulanabilirlik ve gizlilik isimli 4 önemli konuya blokzinciri, şifreleme algoritmaları ve sıfır bilgi ispatı kavramlarıyla bir çözüm yöntemi sunmaktadır. Geniş bir düğüm üzerinde çalışan bu oylama sistemiyle herhangi bir otoritenin kontrolünde olmayan bir oylama sistemi ortaya konulması amaçlanmıştır.

Çalışma üniversite ölçeğinde bir oylama sistemi olması sebebiyle belirli sınırlamalara sahiptir. Özellikle, ulusal bir oylamaya göre daha kısıtlı seçmen sayısına sahip olunması performans sonuçlarını olumlu yönde değiştirmektedir. Bu yüzden geniş çaplı çalışmalarda bu durum dikkate alınmalıdır. Yapılan çalışma, henüz herhangi bir kurum ya da kuruluş tarafından pilot uygulama kapsamında kullanılmadığından canlı performans değerlendirmeleri mevcut değildir. Bu kapsamda, değerlendirildiğinde, performans metriklerini ortaya koyabilmek için gerçekleştirilen testler ve kullanıcılar tamamen test ortamında oluşturulmuştur. Öte yandan, özel olarak herhangi bir kurum ya da kuruluş için geliştirilen bir oylama sistemi olmaması sebebiyle mevzuattaki seçim kanununu ve hukuki prosedürleri detaylı kapsamamaktadır.

#### 1.5. TEZİN YAPISI

Bu tez, blokzinciri tabanlı üniversite ölçekli bir oylama sistemi geliştirmeyi hedefleyerek, teorik bilginin yanı sıra güncel bir prototipin tasarlanmasını ve uygulanmasını içermektedir. Prototip performansının analiz edilmesi de kapsam dahilindedir. Yapılan çalışma, bir oylama sisteminin mimari tasarımından prototip geliştirilmesine ve analizine kadar olan tüm süreci kapsamaktadır. Bu süreçlerin açıklanabilmesi için tez detaylı bir şekilde düzenlenmiştir. Bu kapsamda, tez 6 bölümden oluşmaktadır:

1. Giriş: Çalışmanın konusu ve önemi hakkında bilgi vermektedir. Bu bağlamda tezin özgün tarafları, amacı ve hedefleri ayrıntılı olarak açıklanmaktadır.
2. Kavramsal Çerçeve: Çalışmanın temel çerçevesini oluşturan blokzinciri teknolojisinin ve bileşenlerinin teorik arka planı bu bölümde açıklanmaktadır.

3. Yöntem: Oylama sisteminin tasarlanması ve uygulanması için izlenen metodolojik yaklaşımlar bu bölümde açıklanmaktadır.
4. Bulgular: Sistemin çalışma performansını değerlendirebilmek amacıyla yapılan test ve analiz işlemlerinin sonuçlarını barındırmaktadır.
5. Tartışma: Test ve analiz sonuçlarında elde edilen verilerin, sistem açısından ne anlama geldiği bu bölümde açıklanmaktadır.
6. Sonuç ve Öneriler: Sistemin tasarlanması, geliştirilmesi ve test edilmesi neticesinde elde edilen verilerin özetini içermektedir ve genel çıkarımlar bulunmaktadır.



## 2. KAVRAMSAL ÇERÇEVE

Bu bölümde, merkezi, merkezi olmayan ve dağıtık ağlar, blokzincir teknolojisi, blokzincir tabanlı merkezi olmayan ve dağıtık uygulamalar, şifreleme ve özetleme fonksiyonları, sıfır bilgileri ispatı, kâğıt ve elektronik oylama sistemleri, güvenli elektronik oylama sistemleri için gereksinimler ve temel karakteristik özellikler detaylı olarak incelenmiştir.

### 2.1. MERKEZİ, MERKEZİ OLMAYAN ve DAĞITIK AĞLAR

Modern ağ uygulamalarında istemci-sunucu (client-server) ve eşler-arası (P2P) ağ mimarisi geniş kullanım alanına sahiptir [23, 24]. İstemci-sunucu mimarisinde, istemciler (clients) ve istemcilerden gelen isteklere cevap vermesi gereken sunucu (server) adı verilen bir ana bilgisayar bulunur. HTTP, FTP ve SMTP, istemci-sunucu mimarisini temel alan ağ protokollerinin güncel bir örneğidir. Ağa bağlı aktif bir cihaz veya bileşen düğüm (node) olarak ifade edilmektedir.

Geliştirilen uygulamanın ihtiyaçlarına göre istemci-sunucu mimarisinde birden fazla sunucu olacak şekilde tasarım gerçekleştirilebilmektedir. Amazon, Facebook ve Google gibi milyarlarca kullanıcıya hizmet sağlaması gereken şirketler kesintileri azaltmak ve daha verimli hizmet vermek amacıyla çok sayıda sunucuya sahip ağ mimarisi kullanırlar [11].

P2P ağ mimarisi hem istemci hem de sunucu gibi davranabilen eş (peer)'lerden oluşmaktadır. Buradaki eş kavramı, P2P ağlarda birlikte eşit bir şekilde çalışan düğümleri ifade etmek için kullanılmaktadır. Eşler birbirlerine bağlıdır ve karşılıklı olarak iletişim kurabilmektedir [10].

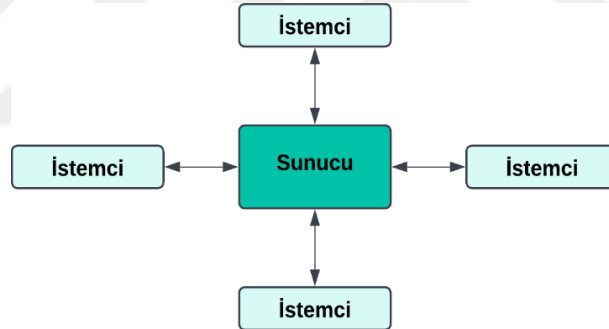
Bu bağlamda, tek bir ana düğüme bağlı istemcilerin olduğu yapılar merkezi ağlar, tek bir ana düğüm yerine kontrol ve veri yönetiminin birden fazla düğüme dağıtıldığı yapılar merkezi olmayan ağlar olarak tanımlanabilir. Merkezi bir otoriteye ihtiyaç duymadan, tüm eşlerin birbirine bağlı olduğu ve sorumluluk aldığı yapı dağıtık ağları oluşturmaktadır. Dağıtık ağlarda, her eş hem bir istemci hem de sunucu olma görevine sahiptir. Merkezi Ağlar, merkezi olmayan ağlar ve dağıtık ağlar Tablo 2.1.'de karşılaştırılmış olup daha sonraki başlıklarla detaylı bir şekilde açıklanmaktadır.

**Tablo 2. 1.** Merkezi, merkezi olmayan ve dağıtık ağların karşılaştırılması.

| Özellik            | Merkezi Ağ              | Merkezi Olmayan Ağ      | Dağıtık Ağ   |
|--------------------|-------------------------|-------------------------|--------------|
| Şeffaflık          | Düşük (Merkezi Otorite) | Düşük (Merkezi Otorite) | Yüksek       |
| Kontrol            | Merkezi Düğüm           | Birkaç Merkezi Düğüm    | Tüm Düğümler |
| Hata Toleransı     | Düşük                   | Orta                    | Yüksek       |
| Manipülasyon Riski | Yüksek                  | Orta                    | Düşük        |
| Karmaşıklık        | Düşük                   | Yüksek                  | Yüksek       |

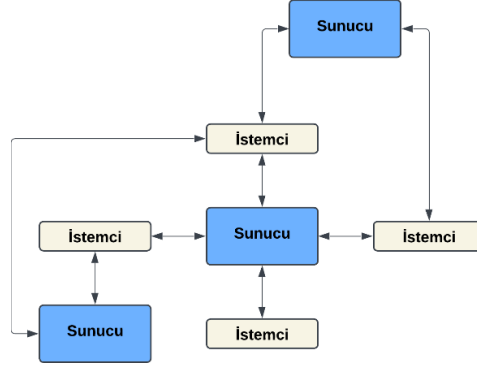
### 2.1.1. Merkezi Ağlar

Merkezi ağlar, iki veya daha fazla düğümün bir araya geldiği ve tüm kontrol sürecinin ana bir düğüm tarafından yapıldığı bir ağ türüdür. Bu ağ yapısında bir veya daha fazla istemci ve bir tane de sunucu bulunur. İstemci düğümleri arasında doğrudan bağlantı bulunmaz ve istemciler sadece sunucu adı verilen ana bir bilgisayarla iletişim kurarlar. Veri ve kaynak yönetimi tek bir merkezden kontrol edilir. İstemciler ana bilgisayarın kaynaklarına ve verilerine izin verilen ölçüde sahip olabilmektedir. Şekil 2.1.'de 4 tane istemci ve 1 tane sunucu düğümünden oluşan bir merkezi ağ örneği gösterilmektedir.

**Şekil 2. 1.** Merkezi ağ tasarımı.

### 2.1.2. Merkezi Olmayan Ağlar

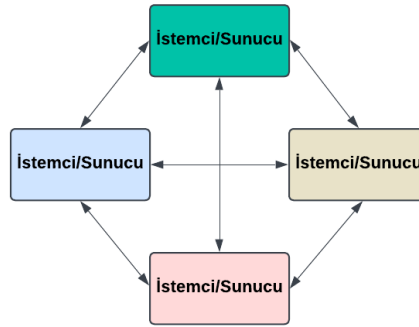
Merkezi olmayan ağlar, tek bir sunucu yerine daha fazla sunucunun çalıştığı ve istemcilerin bu sunucularla bağlantı kurduğu ağ yapılarıdır. Merkezi ağlar ile arasındaki farklardan birisi sunucu sayısındaki artıştır. Merkezi ve merkezi olmayan ağların çalışması için istemci düğümleri herhangi bir sorumluluk almaz. Sunucu adı verilen ana bilgisayarlar üzerinden, istemciler veri ve kaynak kullanımını gerçekleştirirler. İstemcilerin, veri yönetimi ve kaynak kullanımı konusunda yetkisinin olmaması bu yapıları da merkezi bir otoritenin kontrolü altında olduğunu göstermektedir. Şekil 2.2.'de birden fazla sunuculu bir merkezi olmayan ağ modeli gösterilmektedir. Özellikle kesintisiz, kaliteli ve sürdürülebilir hizmet vermek için bu yapı teknoloji firmaları tarafından tercih edilmektedir [10].



Şekil 2. 2. Merkezi olmayan ağ tasarımı.

### 2.1.3. Dağıtık Ağlar

Dağıtık ağlar, düğümlerin ortak bir amaç için çalıştığı ve birbirinden bağımsız olarak bulundukları ağ yapılarını ifade etmektedir. Veri yönetimi, kaynak paylaşımı ve işlem gücü ağ boyunca düğümler arasında paylaşılır ve tüm düğümler birbirleriyle karşılıklı iletişim kurabileceği doğrudan bir bağlantıya sahiptir. Ağ kurulurken ortak bir amaç belirlenir, böylece bu ortak amaç için tüm düğümler çalışmalarını gerçekleştirirler. Merkezi bir yapı olmadığı için kullanılan veriler ağ üzerinde paylaşılır. Her düğüm hem istemci hem de sunucu olabilme özelliğine sahiptir. Dağıtık bir ağ modelinin örneği Şekil 2.3.'de gösterilmektedir.



Şekil 2. 3. Dağıtık ağ tasarımı.

## 2.2. KRİPTOGRAFİK ALGORİTMALAR

Kriptografik algoritmalar, veri güvenliğini ve gizliliğini sağlamak için matematik biliminden ve bilgisayarların gücünden yararlanan yöntemler bütünüdür. Verilerin belirli formatta şifrenmesi veya maskelenmesi için kriptografik algoritmalarından yararlanır. Blokzinciri teknolojisi, şeffaf, güvenli, doğrulanabilir ve gizlilik kavramlarını kullandığı kriptografik algoritmaları vasıtasıyla sağlamaktadır. Öte yandan, blokzincirinin değiştirilmeye

ve kurcalanmaya karşı güçlü olması bu algoritmaların ne kadar dayanıklı olduğuyla doğru orantılıdır. Ağ teknolojileri, kriptografik algoritmalar sayesinde daha güvenli çalışmaktadır [54]. Kendi içerisinde farklı amaçlarla kullanılmak üzere çeşitli kategorilere ayrılmaktadır. Kriptografik algoritmalar, simetrik anahtar (symmetric-key), asimetrik anahtar (asymmetric-key), ve özetleme (hashing) fonksiyonları olarak üç ana kategoriye ayrılabilir [60].

### 2.2.1. Simetrik ve Asimetrik Anahtar Algoritmaları

Simetrik anahtar algoritmaları, şifreleme ve şifre çözme işlemlerini tek bir anahtar ile gerçekleştiren algoritmalarlardır. Verinin şifrelenmesi ve şifresinin çözülmesi için başlangıçta belirlenen tek bir anahtar kullanılmaktadır. Simetrik anahtar algoritmalarında, şifrelenecek düz metin veya veri (plaintext)  $P$ , şifrelenmiş metin veya veri (ciphertext)  $C$ , anahtar uzayı (key)  $K$ , şifreleme fonksiyonu  $E$ , şifre çözme fonksiyonu ise  $D$  ile temsil edilirse; Şifreleme,  $C=E_K(P)$  ve şifre çözme,  $P = D_K(C)$  formülleriyle ifade edilmektedir [48, 49]. AES (Advanced Encryption Standard), DES (Data Encryption Standard) ve 3DES, simetrik anahtar algoritmalarının bilinen örneklerindendir. AES algoritması 128, 192 ve 256-bit anahtar uzunluklarını desteklerken, DES 8-bit olan parite kontrolleri ile 64-bit anahtar uzunluğunu destekler [49]. 3DES ise DES algoritmasının üç kez çalıştırılması mantığına dayanmaktadır.

Asimetrik anahtar algoritmaları, şifreleme ve şifre çözme işlemlerinin bir çift anahtar ile gerçekleştirildiği kriptografik algoritmalarlardır. Bir çift anahtar, genel anahtar (public key) ve özel anahtar (private key) ikilisinden oluşmaktadır. Genel anahtar ve özel anahtar matematiksel olarak birbirine bağlıdır ve genel anahtarın üretilmesi için özel anahtara ihtiyaç duyulmaktadır. Genel anahtar, verinin şifrelenmesi için kullanılırken özel anahtar, şifrelenen verinin tekrar eski haline getirilmesi için şifre çözme adımında kullanılır. Özel anahtar (Private key), şifre çözme adımlarında kullanıldığından güvenli bir şekilde tutulması gerekmektedir. Asimetrik anahtar algoritmalarında, şifrelenecek düz metin veya veri (plaintext)  $P$ , şifrelenmiş metin veya veri (ciphertext)  $C$ , genel anahtar (public key)  $K_{\text{public}}$ , özel anahtar (private key)  $K_{\text{private}}$ , şifreleme fonksiyonu  $E$ , şifre çözme fonksiyonu ise  $D$  temsil edilirse; Şifreleme,  $C=E_{K_{\text{public}}}(P)$  ve şifre çözme,  $P=D_{K_{\text{private}}}(C)$  formülleriyle ifade edilmektedir [48, 49]. RSA (Rivest-Shamir-Adleman), ECC (Elliptic Curve Cryptography) ve Diffie-Helman Anahtar Değişimi algoritmaları asimetrik anahtar algoritmalarının bilinen örnekleridir. RSA 2048, 3072 ve 4096-bit, ECC 256, 384 ve 512-bit anahtar boyutu desteklemektedir [50].

Simetrik ve asimetrik anahtar algoritmalarının temel farkları, anahtar yapısı, karmaşıklık, güçlü ve zayıf yönleri açısından Tablo 2.2.'de sunulmuştur.

**Tablo 2. 2.** Simetrik ve asimetrik anahtar algoritmalarının temel farkları.

| Özellik        | Simetrik Şifreleme                                                               | Asimetrik Şifreleme                                                                  |
|----------------|----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Anahtar Yapısı | Şifreleme ve çözme işlemi, tek bir anahtar ile yapılır.                          | Şifreleme ve çözme işlemi, bir açık anahtar ve bir özel anahtar ile yapılır.         |
| Karmaşıklık    | Daha az karmaşık algoritmalarlardır.                                             | Daha karmaşık algoritmalarlardır.                                                    |
| Faydalar       | Hızlı algoritmalarlardır. Daha az işlem gücü gerektirir.                         | Güvenli anahtar paylaşımı sağlar.                                                    |
| Sınırlamalar   | Tek bir anahtar olması, anahtar paylaşımını gerektirdiğinden, saklanması zordur. | Büyük veri setlerinde düşük performans gösterir ve daha fazla işlem gücü gerektirir. |
| Örnek          | AES, DES, 3DES                                                                   | RSA, ECC, DSA                                                                        |

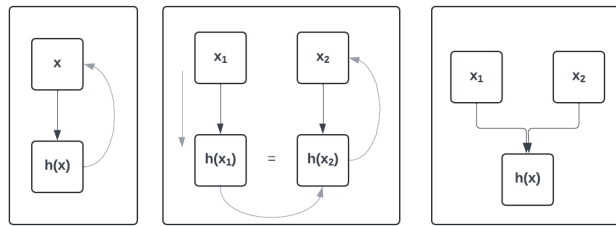
### 2.2.2. Özetleme Fonksiyonları

Özetleme fonksiyonları (Hash Functions), bir veriyi matematik işlemleri sonucunda sabit uzunlukta özet değere çeviren tek yönlü algoritmalarlardır. Bir verinin değiştirilip değiştirilmediğini kontrol etmek, verileri maskelemek ve doğrulamak için özetleme fonksiyonlarından yararlanılır. MD5, RIREMDB, SHA-1, SHA-2, Keccak-256 ve SHA-3 bilinen özetleme fonksiyonlarıdır. Bu fonksiyonları sağlaması gereken üç temel kural bulunmaktadır [51]:

1. **Ön Görüntü Direnci (Preimage Resistance):** Elde edilen bir özetleme değerinden, bu değeri üreten girdinin bulunmamasını ifade eder. Örneğin;  $H()$  özetleme fonksiyonuna  $x$  verisini verildiğinde  $y = H(x)$  ifadesi elde edilir. Burada  $y$  değerine uygun bir  $x$  bulmanın zor olması gerekmektedir.
2. **İkinci Ön Görüntü Direnci (Second-preimage resistance):** Bir özetleme fonksiyonuna verilen girdi sonucu bir değer elde edilir. Bu aynı özetleme değerine ulaşılacak başka bir girdinin üretilmemesi gerekmektedir. Burada ikinci ön görüntü direncini kırmak isteyen saldırgan, bir giriş değeri belirler ve bu giriş değeri ile elde edilen özetleme değeriyle aynı çıktıyı verebilecek başka bir giriş değeri bulmaya çalışmaktadır.
3. **Çakışma Direnci (Collision resistance):** Aynı hash değerini üreten iki farklı girdi bulmanın pratikte imkânsız olmasıdır. Örneğin;  $H()$  özetleme fonksiyonuna  $x$  verisini

verildiğinde  $y = H(x_1)$  ifadesi elde edilir ve  $y = H(x_2)$  ifadesinin olmaması gerekmektedir. Burada birbirinden farklı iki giriş ifadesiyle aynı özetleme değerine ulaşılmaya çalışılır.

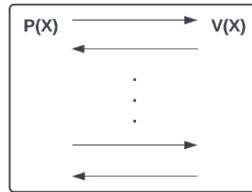
Şekil 2.4.'de ön görüntü, ikinci ön görüntü, çakışma direnci örnek şemayla gösterilmiştir [51]. Ön görüntü direnciyle üretilen özetleme değerinden giriş bilgisine ulaşılamaması, ikinci ön görüntü direnciyle üretilen özetleme değerine ulaşılacak farklı bir giriş değeri bulunamaması ve çakışma direnciyle iki farklı giriş değerinden aynı özetleme değerinin üretiliyor olmaması gerekmektedir.



**Şekil 2. 4.** Görüntü, ikinci ön görünüş ve çakışma direnci.

### 2.3. SIFIR BİLGİ İSPATI

Sıfır Bilgi İspatı (ZKP), 1985 yılında Shafi Goldwasser, Silvio Micali ve Charles Rackoff tarafından tanımlanmıştır [29]. Şekil 2.5.'de gösterildiği gibi iki aktörlü bir sistemde, bir tarafın bir bilgiyi bildiğine dair özel bir detay vermeden, karşı tarafa bu bilgiyi bildiğini kanıtlamasını sağlayan bir kriptografik tekniktir.



**Şekil 2. 5.** Sıfır bilgi ispatı için kanıtlayıcı ve doğrulayıcı gösterimi.

Bilgiyi bildiğini ispat etmek zorunda olan taraf kanıtlayıcı (prover) ve bunu doğrulayacak olan taraf doğrulayıcı (verifier) olarak tanımlanır. Kanıtlayıcı P ve doğrulayıcı da V harfleriyle ifade edilir ve her biri birer sistem aktörüdür. Sistem tasarımına göre aktör sayısında değişiklik yapılabilir. ZKP'nin sağlaması gereken üç temel özellik şunlardır [52, 53]:

1. Tamlik (Completeness): Doğrulayıcı, kanıtlayıcı tarafından gönderilecek her önerme için ikna edilmelidir. Eğer kanıtlayıcı gerekli bilgilere sahipse, doğrulayıcı bu duruma ikna olduğunu ve kanıtın doğruluğunu garanti etmelidir.
2. Sağlamlık (Soundness): Doğrulayıcı, yanlış bir önermeyle kandırılmamalı ve bu yanlış önerme için doğrulayıcı ikna etmek hesaplama açısından mümkün olmamalıdır.
3. Sıfır Bilgi (Zero-Knowledge): Kanıtlayıcı, önermenin doğru olduğunu ispat etmeye çalışırken, doğrulayıcı herhangi bir bilgi vermemelidir.

ZKP, etkileşimli (Interactive ZKP) ve etkileşimsiz (Non-Interactive ZKP) olarak iki türe ayrılır. Etkileşimli ZKP, kanıtlayıcı ve doğrulayıcı arasında doğrudan etkileşimin olduğu bir türdür. Doğrulayıcı, kanıtlayıcının bilgisi bildiğine ikna olabilmek için ona sorular sorar ve aldığı cevapları karşılaştırır. Etkileşimsiz ZKP, kanıtlayıcı ve doğrulayıcı arasında etkileşimin olmadığı bir türdür. Kanıtlayıcı, doğrulayıcıya bir önerme de bulunur ve doğrulayıcı da bu kanıta ikna olur veya olmaz. Etkileşimsiz ZKP'nin örneklerinden ikisi zk-SNARK ve zk-STARK'tır.

zk-SNARK, doğrulayıcı (verifier) tarafın, kanıtlayıcı (prover) tarafa bilginin doğru olduğunu ilgili bilgiyi vermeden ispat edebilmesi için kullanılan ZKP türüdür. Kanıtlayıcı tarafın karşı tarafa bir kanıt gönderilmesi için matematiksel devre (circuit) hazırlanması gerekmekte ve bu devre kanıt üretebilmek için kullanıcıdan alınan bilgilerle çalışmaktadır. Devre, alınan bilgiler doğrultusunda aritmetik işlemler sonucu bir çıktı üretmektedir. Örneğin, bir kullanıcı giriş değeri olarak istenilen bilgiyi bu devreye vererek bir kanıt (proof) üretir. Ürettiği bu kanıt, doğrulayıcıya (verifier) aktarılması sonucu, kanıtın geçerliliğinin kontrol edilebilmesi için aritmetik doğrulama sürecinden geçirilir. Burada amaç özel veriyi paylaşmak değil bu verinin belirli kurallara uyduğunu kanıtlayacak SNARK kanıtının doğrulayıcı tarafından onaylanmasını sağlamaktadır. Eğer kanıt doğruyla işlem tamamlanmıştır. zk-SNARK'da üretilen kanıt küçük boyutlu olduğundan doğrulama süresi çok kısadır fakat kuantum saldırılarına açık bir ZKP türüdür ve güvenli bir kurulum aşamasının gerçekleştirilmesi gerekmektedir [55]. zk-STARK'lara göre daha küçük ölçekli işlemler için tercih edilmekte ve daha az bant genişliği ihtiyacı duymaktadır.

zk-STARK, teoride aynı amaçla kullanılan fakat zk-SNARK'lara göre daha büyük hesaplamalar için verimli ve güvenli bir doğrulama imkânı sağlamaktadır. zk-STARK'ların

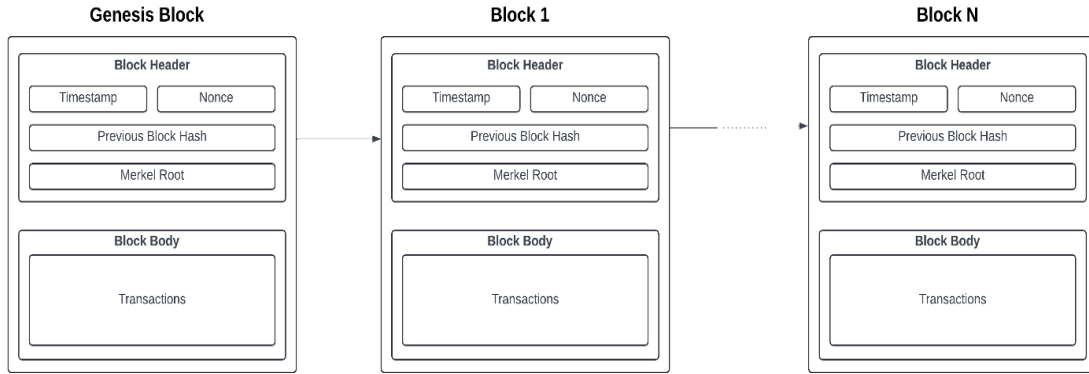
farkı, şeffaf olması ve bir kurulum gerektirmemesidir. Kanıt boyutunu küçültmek ve doğrulanabilirliğini artırmak için polinom kullanılır [55]. zk-STARK kullanılacağı zaman önce problem belirlenir. Örneğin; kullanıcının doğru parolayı bildiğinin kanıtlanmasının istendiği bir sistemde, sunucu parolayı özetlenmiş (hash) bir şekilde tutuyorsa; kullanıcının elindeki bilgiyle karşı taraftaki bilgiyi karşılaştırabilecek bir polinom ifadesi oluşturulması gerekmektedir. Aslında bunun polinom olarak ifade edilmesi oldukça zor olduğundan, orijinal hesaplama oldukça yakın bir polinom fonksiyonu bulma işleminin yapılması ve ilgili polinomunun kullanılması gerekmektedir. Burada matematiksel olarak polinom interpolasyonu ile fonksiyonun ifade edilmesi gerekmektedir. Polinom için veri uzayı oldukça düşük olması dolayısıyla, polinomları genişletme kısaca LDE yapılır. Çünkü polinomun derecesi düşükse, sahte kanıt üretmek isteyen saldırgan yanlış hesaplamalar yaparak sistemde zafiyete neden olabilmektedir. Bu aşamadan sonra kullanıcı, parolasının doğru olduğu ispat etmek için bir zk-STARK kanıtı üretebilir. zk-SNARK'lara göre büyük verileri daha hızlı işleyebilmekte ve büyük ölçekli işlemlerde daha verimli bir sonuç verebilmektedir.

## 2.4. BLOKZİNCİR

Blokzincir teknolojisi, Stuart Haber ve W. Scott Stornetta tarafından yayımlanan makalenin temelleri üzerine tasarlanmıştır [16]. Zaman damgası (timestamp) mekanizması ve verilerin değiştirilemezliği konusundaki yapılan çalışmalar blokzinciri teknolojisinin temel yapısını oluşturmaktadır. 2008 yılında ise Satoshi Nakamoto isimli kişi veya kişiler tarafından “Bitcoin: A Peer-to-Peer Electronic Cash System” isimli bir makale yayımlanmış ve bu makaleyle Stuart Haber ve W. Scott Stornetta’nın çalışmalarına referansta bulunarak Bitcoin isimli kripto para tanıtılmış ve kullanılmaya başlanmıştır [17, 25]. Duyurulduğu tarihten günümüze kadarki süreçte Bitcoin finansal ve teknoloji alanında tanınmış ve alt yapısını oluşturan blokzinciri teknolojisi geniş kullanım alanına sahip olmuştur [18, 42].

Blokzincir, işlemlerin güvenli ve şeffaf bir şekilde kayıt altında tutulmasını sağlamak için tasarlanmış merkezi olmayan ve dağıtık bir veri tabanı teknolojisidir. Merkezi bir yapıya sahip olmaması, kullanıcıların herhangi bir üçüncü taraf uygulamaya bağlı kalmaksızın işlemlerini bağımsız şekilde gerçekleştirebilmelerine olanak tanımaktadır. Blokzinciri, birbirine zincir şeklinde bağlı bloklardan oluşur ve bu her blok belirli sayıda transaction adı verilen kayıtlı işlem tutmaktadır. Blokların bir araya gelmesiyle oluşan zincir, P2P bir ağa bağlı düğümlerde tutulmaktadır. Şekil 2.6.’da Bitcoin’de kullanılan zincir yapısı görülebilir. Bitcoin

ve Ethereum’da kullanılan blokzinciri arasındaki farklar bu bölümde ayrıntılı olarak değerlendirilmiştir.



**Şekil 2. 6.** Bitcoin’de kullanılan blokzinciri modeli.

#### 2.4.1. Blokzincir Türleri

Blokzincir teknolojisi, sektörlerin ihtiyaçlarına göre değişiklik gösterecek bir şekilde tasarlanabilmektedir. Tarihine bakıldığında merkezi olmayan bir teknoloji olarak ortaya çıkmasının yanı sıra istenildiğinde merkezi bir otorite için kapalı bir ağ olarak kullanılabilir [15, 20, 25]. Blokzincir teknolojisiyle herkesin erişimine açık, özel veya belirli gruba ait olan ağlar oluşturulabilmektedir.

Dışarıdan erişimin istenmediği ve yüksek riskli verilerin kullanıldığı çeşitli sektörler bulunmaktadır. Bu sektörde faaliyet gösteren kurum ve kuruluşların kendi kurallarıyla tasarladıkları blokzinciri ağları bulunmaktadır [20]. Bu kapsamda, blokzinciri türleri farklı ihtiyaç ve kullanım senaryolarına göre üç farklı kategori altında toplanmaktadır [21, 22]. Bu üç kategorinin farklılıkları Tablo 2.3.’de ayrıntılı olarak yer almaktadır.

**Tablo 2. 3.** Blokzinciri türlerinin temel farkları.

| Kriter     | Açık Blokzinciri | Özel Blokzinciri              | Konsorsiyum Blokzinciri       |
|------------|------------------|-------------------------------|-------------------------------|
| Erişim     | Tüm Katılımcılar | Yetkilendirilmiş Kullanıcılar | Yetkilendirilmiş Katılımcılar |
| İzin       | İzinsiz          | İzinli                        | İzinli                        |
| Yönetim    | Merkezi Olmayan  | Merkezi                       | Yarı Merkezi                  |
| Verimlilik | Yavaş            | Hızlı                         | Hızlı                         |
| Güvenlik   | Yüksek           | Yüksek                        | Yüksek                        |

##### 2.4.1.1. Açık Blokzinciri (Public Blockchain)

Açık blokzinciri, herkesin izinsiz bir şekilde katılabileceği bir blokzinciri ağını ifade eder. Merkezi bir otorite olmadığı için herkes verileri indirebilir, doğrulayabilir ve yeni

bloklar ekleyebilmektedir. Tüm katılımcılar, açık blokzinciri sayesinde ağın şeffaflık ve güvenlik özelliklerinden yararlanabilmektedir. Herkesin kullanımına açık olması veri trafiğini artırmakta ve bu durumda ölçeklenebilirlik sorunlarının çıkmasına neden olmaktadır. Bitcoin bilinen ilk açık blokzinciri örneği olmakla birlikte Ethereum da bu türün örneklerindendir. Açık blokzincir ağlarına bloklar, Proof of Work (PoW) veya Proof of Stake (PoS) konsensüs mekanizmaları kullanılarak eklenir veya doğrulanır.

#### **2.4.1.2. Özel Blokzinciri (Private Blockchain)**

Özel blokzinciri, belirli kişiler ve gruplara açık olan bir türü ifade eder. Dağıtık bir yapıya sahip olması yanında belirli bir gruba ait ve o grup tarafından yönetilen blokzinciri ağlarıdır. Herkesin kullanımında olmadığı için kısmi bir merkezi olmayan ağ uygulamasıdır. Bu yüzden, şeffaflık ağıdaki herkes için söz konusu değildir. Genellikle daha az kullanıcıya hizmet verdiği için daha hızlı işlemler gerçekleştirilebilir.

#### **2.4.1.3. Konsorsiyum Blokzinciri (Consortium Blockchain)**

Konsorsiyum blokzinciri, kurum veya kuruluşlara ait ve yönetimleri de bu kurum ve kuruluşlar tarafından gerçekleştirilen bir blokzinciri ağını ifade eder. Ağa yalnızca ilgili kurum ve kuruluşlar tarafından yetkilendirilmiş katılımcılar erişim sağlayabilmektedir. Diğer blokzinciri türlerine göre kurum ve kuruluşlar için daha güvenlidir. Zincire eklenecek bloklar belirlenen düğümler tarafından gerçekleştirilebilir.

### **2.4.2. Blokzincir Özellikleri**

Blokzinciri teknolojisinin yaşanan sorunlara farklı yöntemlerle çözümler bulması temel karakteristik özelliklerini ifade etmektedir [9, 12, 18, 20, 22].

#### **2.4.2.1. Değişmezlik**

Blokzinciri teknolojisi, verilerin değiştirilmesini önleyecek mekanizmalara sahiptir. Her bloktaki tüm işlemler, Merkle Tree olarak adlandırılan bir veri yapısından geçirilir. Bunun sonucunda Merkle Root adı verilen tek bir hash değeri elde edilir ve bu değer ilgili bloğun içerisine yazılması sağlanır. Öte yandan, bir bloğun özetleme algoritmasından geçirilerek elde edilen hash bilgisi, kendisinden sonra gelecek bloğun içerisinde kullanılmakta ve ilgili bloğun hash bilgisinin oluşturulması için kullanılan alanlardan birisi olmaktadır. İşlemlerin, Merkle Tree veri yapısından geçirilerek tek bir hash değeri elde edilmesi, bloğun hash değerinin kendisinden sonra gelecek olan bloğa yazılması ve aynı zamanda zincirin merkezi bir sunucu yerine birbirinden farklı düğümlerde tutulması değişmezlik özelliğini oluşturmaktadır.

#### **2.4.2.2. Dağıtık ve Merkezi Olmayan Yapı**

Blokzinciri dağıtık bir veri tabanı teknolojisidir ve blokzincir üzerinde tutulacak veriler merkezi bir sunucu yerine ağdaki düğümler (nodes) üzerinde tutulmaktadır. Blokzincirindeki her düğüm, ağdaki tüm bilgilerin bir kopyasına sahiptir (Blokzincirinin türüne göre bazı istisnalar olabilir). Ağ üzerindeki düğümlerden biri veya birkaçında çeşitli sorunlar olduğu durumda, ağdaki diğer düğümler aktif olduğundan hata toleransı yüksektir.

#### **2.4.2.3. Şeffaflık**

Blokzinciri dağıtık bir veri tabanı teknolojisi olması sebebiyle, farklı noktada bulunan düğümler veri saklama görevini yerine getirirler. Veriler ve geçmişte yapılan tüm işlemler, ağa erişim izni bulunan tüm düğümler tarafından görülebilir. Bu durumda ağın şeffaf olmasını sağlamaktadır.

#### **2.4.2.4. Güvenlik**

Blokzinciri, verilerin değişmezlik özelliğini sağlamak için özetleme algoritmalarını kullanır. İşlemler, dijital imza ve asimetrik şifreleme algoritmaları vasıtasıyla doğrulanır. Kullanıcılar, bir özel anahtar (private key) ve açık anahtar (public key) çiftine sahiptir. Açık anahtar ile kullanıcıyla bağlantı kurulabilirken, özel anahtar ile bir işlemin imzalanma süreci tamamlanmış olur. Özel anahtar, işlemi imzalayacak kişiye aittir ve güvenli bir şekilde tutulması gerekmektedir.

#### **2.4.2.5. İzlenebilirlik**

Blokzinciri üzerinde bulunan her işlem bloklarda saklanmakta ve bu bloklar birbirine hash fonksiyonları vasıtasıyla bağlanmaktadır. Ağdaki tüm düğümler, bu bloklardan oluşan zinciri barındırmaktadır. Bu sebeple değişmezlik özelliği sağlanmakta ve geçmişteki tüm işlemler takip edilebilmektedir. Finansal işlemler ve tedarik zinciri gibi alanlarda bu özellik kritik bir öneme sahiptir [17, 20].

#### **2.4.3. Blok Yapısı**

Blokzinciri üzerine yazılan ilk blok “genesis block” olarak isimlendirilir. Şekil 2.6.’daki Bitcoin modelinde gösterildiği üzere bir blok iki bölümden oluşmaktadır [15, 18]. Bu iki blok header ve body olarak tanımlanmaktadır. Header bölümü bloğun meta verilerini ve önceki bloklarla bağlantısını tutar. Bu bölüm, blokların birbirine bağlanması için kritik önem taşımaktadır.

Bloğun body bölümü, transaction adı verilen ağda gerçekleşen işlemleri saklamak için kullanılır. Bir bloğun body bölümünde tutulacak olan transaction sayısı blokzincirinin altyapısına göre değişiklik göstermektedir. Örneğin; Bitcoin, header bölümü sabit uzunlukta 80 bytes ve her blok boyutu maksimum 1 MB sınıra sahip bir blokzinciridir [19]. Bu durumda her işlem için 250 bytes harcanması gerekirse, bir blok yaklaşık olarak 4000 transaction bulundurabilir.

Bloğun header bölümü, bir önceki bloğun hash değeri (previous block hash), zaman damgası (timestamp), Merkle Kökü (merkle root), nonce, zorluk seviyesi (difficulty target) ve versiyon bilgisi bileşenlerinden oluşur. Bir bloğun hash bilgisinin oluşturulabilmesi için bu değerlerin kullanılması gerekmektedir. Blokzinciri üzerindeki her bloğun hash değeri şu verilerin birleştirilmesiyle hesaplanır:

**Hash = hashFunction(merkle root, previous block hash, timestamp, target, nonce)**

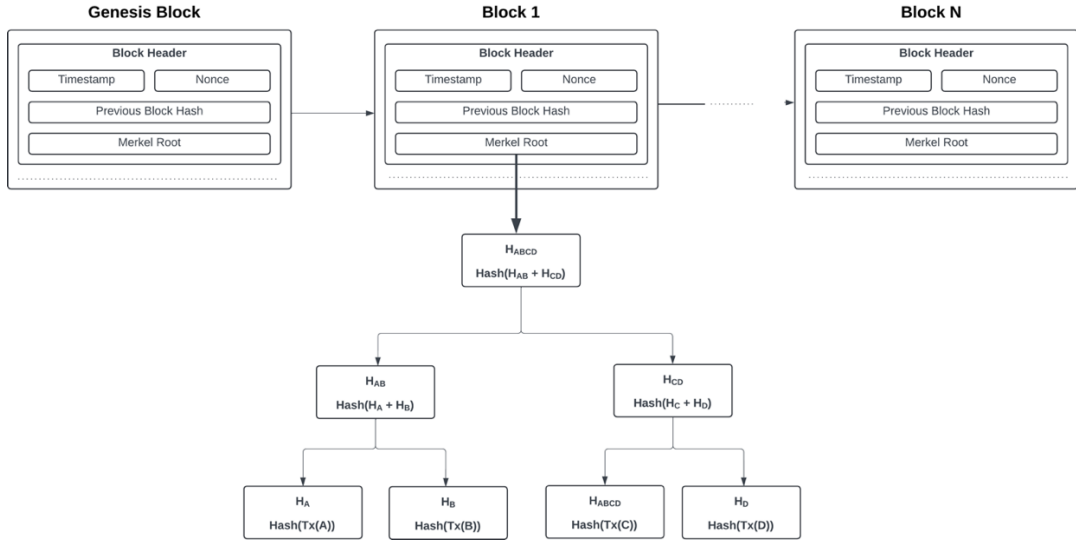
Kısaca,

**Hash = hashFunction (Header + Nonce)**

Yukarıdaki formül ile blokzincirine yazılacak olan bloğun hash değeri hesaplanmış olur. Bu değer, bir sonraki yazılacak olan blok için önceki blok hash değeri (previous block hash) bölümünü oluşturur. Böylece blokzinciri üzerindeki herhangi bir blokta yapılacak değişiklik, bu hash değerini etkileyeceğinden zincir üzerinde yapılan manipülatif bir değişiklik tespit edilebilir hale gelir.

#### **2.4.3.1. Merkle Kökü (Merkle Root)**

Merkle Kökü, tüm işlemlerin hash bilgilerinin birleşiminden oluşan bir kök hash değeridir. Bu değer, Merkle Tree denilen bir veri yapısı sayesinde hesaplanmaktadır. İlgili bloğun tüm işlemlerini (transaction) kapsadığı için blok üzerindeki bir işlemin değişmesi Merkle Kökü değerini de değiştireceğinden dolayı herhangi bir manipülatif davranış tespit edilebilmektedir. Şekil 2.7.'de bir bloğun transaction bilgilerinin, Merkle Tree veri yapısı sayesinde hesaplanması ve ilgili bloğun Merkle Root bölümünü oluşturması uygun bir şekilde modellenmiştir.



Şekil 2. 7. Merkle Tree ve Mekle Root modeli.

#### 2.4.3.2. Önceği Bloğun Hash Değeri (Previous Block Hash)

Bu alan, bir önceki bloğun hash değeridir. Önceki blokta herhangi bir bilgi değiştiğinde hash değeri değişecektir ve kendisinden sonraki blokta tutulan değerle eşleşme olmayacağından blokzincirinin bütünlüğü kırılacaktır. Bu değişiklikte blokzinciri üzerinde yapılan manipülatif durumlar tespit edilebilecektir. Zincirin bütünlüğünün sağlanması için kritik öneme sahiptir.

#### 2.4.3.3. Zaman Damgası (Timestamp)

Bir bloğun veya işlemin oluşturulma zamanını ifade eder. Unix zaman damgası formatında bloğa kaydedilir. Yeni bir bloğun timestamp değeri, önceli bloğun değerinden küçük olamayacağı için zaman bazlı yaşanacak problemlerin önüne geçilmiş olur. Bu değerler ile ağın blok üretim hızı tespit edilerek, blok üretim zorluluğun yeniden ayarlanması sağlanır. Madenciler, belirlenen hash değeri hedefini sağlamaları için nonce değeriyle birlikte kullanırlar.

#### 2.4.3.4. nBits (Zorluk Seviyesi – Difficulty Target)

Madenciler tarafından üretilen blokların üretim hızını kontrol etmek için kullanılan ve zorluk seviyesini (difficulty target) ifade eden bir değerdir. Bir bloğun, blokzincirine eklenebilmesi için hash bilgisinin belirli bir hedef değerinin altında olması gerekmektedir. Bu hedef değer, belirli bir sürede yeni bir bloğun blokzincirine eklenebilmesi için kullanılmaktadır. Örneğin; Bitcoin için yeni bir blok üretimi bu hedef sayesinde 10 dakika ve Ethereum için 12-14 saniye olur [30].

Target değeri, blok için üretilen hash değerinin, belirlenen değerin altında olması için gereken değeri ifade eder. Blok üretimini daha zor hale getirmek, kısaca süreyi uzatmak için target değerini düşük ve zorluğun düşürülmesi için de target değerinin daha yüksek olarak seçilmesi gerekir. Bu target değeri sayesinde blok oluşturma işlemi istenilen sabit aralıkta tutulabilir.

Difficulty Target, nBits alanında bulunan değer kullanılarak hesaplanmaktadır. Blok için üretilen hash bilgisinin bu hedef değerden düşük olması gerekmektedir.

#### **2.4.3.5. Nonce (*Number Only Used Once*)**

Proof of Work (Pow) tabanlı konsensüs mekanizmalarında, madencilerin blok için doğru hash değerini bulması için kullanılan rastgele bir sayıdır. Nonce değeri sürekli değiştirilerek test edilir ve doğru değer bulunmasıyla bloğun hash bilgisi üretilmekte ve zincire eklenmesi sağlanmaktadır. Bir madenci, blok hash değerini Difficulty Target değerinin altında oluşturabilmek için bu Nonce değerini kullanır.

Hash fonksiyonu deterministik olduğu için aynı blok aynı şekilde defalarca hash fonksiyonundan geçirilse dahi aynı değeri vermesi gerekmektedir. Fakat her seferinde farklı bir hash değeri elde etmek için bu Nonce değeri kullanılmakta ve böylece üretilen değerin değişmesi sağlanmaktadır.

#### **2.4.3.6. Versiyon Bilgisi**

Versiyon bilgisi, sistemin ve sistem içerisindeki protokollerin hangi sürümde olduğunu belirtmek için kullanılan bir bilgidir.

#### **2.4.4. Merkle Tree**

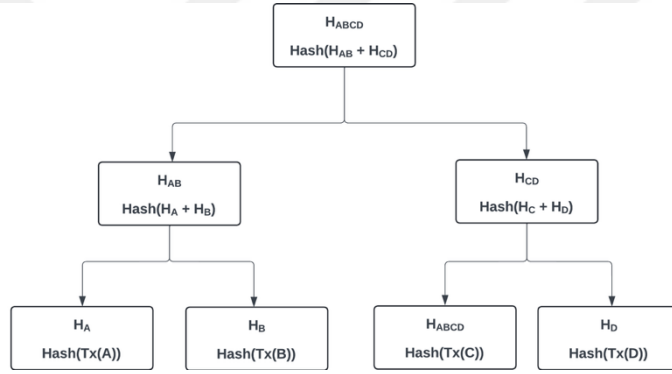
Merkle Ağacı (Merkle tree), Ralph Merkle tarafından 1979 yılında bulunmuş bir veri yapısıdır [31]. Kriptografide verilerin bütünlüğünü doğrulamak için kullanılmaktadır ve blokzinciri teknolojisinin temel elementlerinden birisidir. Blokzinciri teknolojisinde Şekil 2.8.'de görüleceği üzere işlemler (transaction) önce hash fonksiyonundan geçirilir. Daha sonrasında her işlem için elde edilen hash değeri ikili gruplar halinde toplanarak tekrar hash fonksiyonundan geçirilerek yeni hash değerleri üretilir. Bu durum tek bir kök (root) hash değeri elde edilene kadar devam eder [43, 58].

Merkle Ağacı ile elde edilecek kök hash değeri ağda bulunan bir işlemi doğrulamak için etkili bir çözüm yöntemi sunar. Örneğin; Şekil 2.8.'deki gibi Tx(A), Tx(B), Tx(C) ve Tx(D)

isimli 4 tane işlemin bulunduğu bir blok olsun. “Tx(A)” işleminin ilgili blokta olup olmadığını doğrulamak için aşağıdaki adımlar gerçekleştirilir [58]:

1. Tx(A)’nın hash değeri Hash(Tx(A)) ile hesaplanır.
2.  $H_B$  ve  $H_{CD}$  değerleri, doğrulamak yapmak için veri yapısına (Merkle Ağacı) sağlanır.
3.  $H_{AB} = \text{Hash}(H_A + H_B)$  değeri hesaplanır.
4.  $H_{ABCD} = \text{Hash}(H_{AB} + H_{CD})$  hesaplanır ve mevcut olan Merkle Root ile karşılaştırılır.

Bu işlemler sonucunda elde edilen  $H_{ABCD}$  değeri, bloğun header bölümündeki Merkle Root değeri ile eşleşiyorsa Tx(A) işleminin ilgili blokta olduğu ispat edilmiş olur.



**Şekil 2. 8.** Merkle Tree veri yapısının örnek modeli.

#### 2.4.5. Konsensüs Algoritmaları

Blokzincir teknolojisi, dağıtık bir veritabanı mekanizması olduğu için zinciri oluşturan blokların eklenebilmesi ve doğrulanabilmesi amacıyla ağda ortak bir görüş birliğine (konsensüs) varılması gerekmektedir. Bu nedenle dağıtık bir ağdaki tüm düğümlerin belirli algoritmalarla bağlı kalarak işlemlerini gerçekleştirmesi sağlanmalıdır.

Ağın ortak bir şekilde hareket edebilmesi için ağ gereksinimlerine göre çeşitli konsensüs algoritmaları bulunmaktadır. Örneğin; Bitcoin, Proof of Work (PoW) algoritmasını, Ethereum’da başlangıçta PoW ve Eylül 2022 itibarıyla de Proof of Stake (PoS) algoritmasını kullanmaktadır [32]. Bu iki algoritma dışında DPoS, ve PFBT gibi çeşitli algoritmalar da bulunmaktadır. Fakat bu tez kapsamında PoW ve PoS incelenmiştir.

#### 2.4.5.1. *Proof of Work (PoW)*

Blokzinciri teknolojisinde zincire yeni bir blok eklemek ve blokların doğrulama işlemlerini gerçekleştirmek için kullanılan konsensüs algoritmalarından birisidir. PoW, bloğun özetleme (hash) değerinin belirli kurallar dahilinde üretilmesi için çözülmesi gereken zor bir probleme dayanmaktadır. Madenciler arasında doğru çözüme ilk ulaşan, ağ tarafından blok ekleme hakkını elde eder ve Bitcoin gibi ağlarda belirli ücretle ödüllendirilir. Bir bloğun ağa eklenebilmesi için aynı zamanda diğer madenciler tarafından belirlenen kurallara göre doğrulama yapılması gerekmektedir. Bu durum da merkezi bir yapının bulunmasına gerek olmadan tüm madencilerin fikir birliğine ulaşmasını sağlar. Yüksek işlem gücü gerektirdiğinden ağın toplam hesaplama gücünün saldırganlar tarafından ele geçirilmesi zorlaşmaktadır. Bu durumun yanı sıra enerji verimliliği kötü bir algoritma olarak tanımlanmaktadır [7, 20, 34]. Aynı zamanda zor bir problemin çözülmesi vakit alacağından ağda gerçekleşen toplam işlem hızı düşüktür. PoW algoritmasına göre işlemler adım adım şu şekilde gerçekleşir [33, 35, 57]:

1. Yeni bloktaki yer alacak işlemler Merkle Ağacı veri yapısından geçirilir ve tek bir Merkle Root değeri oluşturulur.
2. Bir önceki bloğun özetleme (hash) değeri (previous block hash), zaman damgası (timestamp), Merkle Kökü (merkle root), nonce, zorluk seviyesi (difficulty target) ve versiyon bilgisi bileşenleri kullanılarak bir blok header bölümü oluşturulur.
3. Bloğun özetleme bilgisi oluşturularak, belirlenen kriterlere uyup uymadığı kontrol edilir. Bir madenci için buradaki amaç, blok özetleme değerini Difficulty Target değerinin altında oluşturabilmektir.
4. Eğer uygun bir özetleme bilgisi elde edildiyse, yeni blok başarıyla oluşturulmuş olup bu bilginin ağa duyurulması sağlanır. Eğer uygun bir özetleme bilgisi oluşturulmadıysa, nonce değeri değiştirilerek yeni bir özetleme değeri oluşturulmaya çalışılır. Nonce değeri belirli bir limite geldiğinde halen geçerli bir blok oluşturulamıyorsa, blok başlığını oluşturan diğer alanlar için özetleme değerinde değişikliğe neden olacak değişiklikler yapılır.

#### 2.4.5.2. *Proof of Stake (PoS)*

PoS, bir blokzinciri ağına blok eklemek ve doğrulamak için kullanılan bir konsensüs algoritmasıdır. Merkezi bir yapının bulunmasına gerek olmadan tüm madencilerin blokzinciri

üzerinde fikir birliğine ulaşmasını sağlar. PoS algoritmasında, PoW'dan farklı olarak madenciler yerine süreci kontrol etme görevini üstlenen doğrulayıcılar vardır. Aslında her iki rolün de amacı, işlemlerin doğrulanmasını, yeni blokların eklenmesini ve ağın güvenliğini sağlamaktır. Bu algoritmaya göre; blok oluşturma hakkına sahip olabilmek, sahip olunan pay (hisse, para) miktarına bağlıdır. Yeni blok ekleme ihtimalini yükseltebilmek için daha fazla hisse veya kripto paraya sahip olunması gerekmektedir. Bir blok oluşturmak üzere doğrulayıcıların (validator) rastgele seçilmesi amaçlansa da, daha fazla kripto para veya hisseye sahip olan katılımcıların seçilme olasılığı daha yüksektir. Burada aynı zamanda rastgelelik özelliğinin eklenmiş olması, varlıklı kişilerin sistemi ele geçirmesini önlemeyi amaçlar. Aynı zamanda seçilecek doğrulayıcı, belirlenen süreler içerisinde blok üretimini gerçekleştirmezse bir sonrakiyle devam edilir. Herhangi bir şekilde sahte işlem doğrulaması gerçekleştirmeye çalışan doğrulayıcı, ağ tarafından cezalandırılır. PoS algoritmasına göre işlemler adım adım şu şekilde gerçekleşir [33, 35, 56, 57]:

1. Doğrulayıcılar sahip oldukları hisse veya kripto paraya “stake” adı verilen kilitleme işlemini uygularlar. Bu teminat göstermek olarak düşünülebilir. Böylece ellerindeki değerler ile yeni blokları eklemek ve doğrulamak için şans elde ederler.
2. Doğrulayıcı rastgele bir şekilde belirlenir. Belirlenen doğrulayıcı yeni bloğu doğrulamak için yetkilendirilmektedir. Burada belirlenme ihtimalini artıran en önemli etken daha çok hisse veya kripto para sahibi olmaktır.
3. Seçilen doğrulayıcı, işlemlerin doğrulanması sürecini gerçekleştirir ve yeni bloğu zincire ekleyerek ağa duyurur.
4. Doğrulayıcı, ağ tarafından belirli bir miktarla ödüllendirilir. Eğer ağ kurallarını manipüle etmeye çalışırsa, bu durumdan dolayı ceza alabilir.

#### **2.4.6. Blokzincir Evrimi**

Bitcoin ile başlayan süreçten sonra farklı alanlarda da blokzinciri kullanılmaya başlanmıştır. Aynı zamanda, blokzinciri teknolojisinin yaygınlaşmasıyla birlikte ortaya çıkan sorunların giderilmesine yönelik çeşitli çözüm yöntemleri geliştirilmiş ve uygulamaya konulmuştur [36]. Tüm bu süreçler blokzincirinin tarihsel olarak belirli aşamalardan geçtiğini ifade etmektedir.

Tablo 2.4.'de gösterildiği gibi blokzinciri dört farklı aşamadan geçmiştir. Bitcoin'in ortaya çıkmasıyla başlayan süreç Blokzinciri 1.0 olarak ifade edilmektedir ve bu süreç 2008 ile 2013 yıllarını kapsamaktadır [25, 26]. Blokzinciri ile para transferinin veya dijital varlık saklama işlemlerinin mümkün olabilmesine olanak sağlayan bu süreç, kendinden sonraki tüm süreçlerin ortaya çıkmasına neden olmuştur. 2013 yılı ile 2015 yılı arasındaki akıllı sözleşmelerin kullanılmaya başlandığı aşamalar ise Blokzinciri 2.0 olarak tanımlanmaktadır ve bu süreci dağıtık uygulamaların kullanılabildiği, 2015 ile 2017 yılındaki süreci kapsayan Blokzinciri 3.0 izlemektedir [26, 29]. Blokzinciri 4.0 ise AI, IoT ve siber güvenlik kavramlarının değerlendirildiği ve bu kapsamda blokzinciri teknolojisine yeni özelliklerin ve uygulama alanlarının eklendiği dönemi ifade etmektedir [26, 28]. Bu dönemin halen kapanmadığı ve gelecekte de yeni entegrasyonların mümkün olabileceği bir dönemi kapsadığı düşünülmektedir [27]. Bu bölümde bu Blokzinciri 1.0, Blokzinciri 2.0, Blokzinciri 3.0 ve Blokzinciri 4.0 ayrıntılı olarak incelenmiştir ve Tablo 2.4.'de gösterildiği gibi karşılaştırılmıştır.

**Tablo 2. 4.** Blokzinciri evriminin tarihsel evrimi ve karşılaştırması.

| Kriter            | Blokzinciri 1.0  | Blokzinciri 2.0    | Blokzinciri 3.0  | Blokzinciri 4.0   |
|-------------------|------------------|--------------------|------------------|-------------------|
| Teknoloji         | Dağıtık Cüzdan   | Akıllı Sözleşmeler | Dağıtık Uygulama | Yapay Zeka ve IoT |
| Konsensüs         | PoW              | PoS, DPoS          | PoS, PoA, BFT    | DAG,              |
| Ölçeklenebilirlik | Kısmi            | Kısmi              | Ölçeklenebilir   | Yüksek            |
| Maliyet           | Yüksek Maliyetli | Maliyetli          | Az maliyetli     | Daha Az Maliyetli |
| Örnek             | Bitcoin          | Ethereum           | Solana, Algorand | IOTA              |

#### 2.4.6.1. Blokzincir 1.0

Blokzincir 1.0 olarak adlandırılan aşama, Bitcoin'in duyurulmasıyla ortaya çıkan ve 2008 yılından 2013 yılına kadarki süreci içine alan dönemi ifade etmektedir [25, 26]. Blokzinciriyle para transferiyle dijital varlık saklama işlemlerini kapsamaktadır ve kripto paraların yaygınlaşmasına neden olmuştur. Bu dönemin örneklerinden olan Bitcoin, anonim ve merkezi olmayan bir finansal araç kullanılmasını mümkün gösteren ilk örnektir [25].

#### 2.4.6.2. Blokzincir 2.0

Ethereum'un ilk kez 2013 yılında tanıtılmasıyla başlayan Blokzincir 2.0 dönemi, blokzinciri ağındaki işlemlerin otomatik bir şekilde gerçekleştirmesini sağlayan mekanizmaların kullanıldığı bir aşama olarak tanımlanmaktadır [37]. 2013 yılı ile 2015 yılı arasındaki süreci kapsar [26]. Ethereum'un sağladığı Ethereum Sanal Makinesi (EVM) üzerinde çalışan akıllı sözleşmelerle, geliştiriciler kendi uygulamalarını blokzinciri üzerinde dağıtabilmektedir. Bu

akıllı sözleşmeler, belirli kurallara bağlı olarak otomatik olarak çalışabilecek programlanabilir yapılardır.

#### **2.4.6.3. Blokzincir 3.0**

Blokzincir 3.0, 2015 ve 2017 yıllarını kapsayan merkezi olmayan dağıtık uygulamaların (dapp) geliştirilmesini içeren bir süreci ifade etmektedir [26, 29]. Dağıtık olarak geliştirilen uygulamalar, kullanıcıların ve uygulama geliştiricilerin, birbirine bağlandığı merkezi olmayan bir eşler arası ağda çalışmaktadır. Bu dağıtık uygulamaların daha ölçeklenebilir ve sürdürülebilir bir ağda çalışmasının hedeflenmesi bu sürecin bir amacıdır. Blokzincir 3.0 sürecinde sigorta, finans, ulaşım ve tedarik zinciri gibi alanlarda blokzincir teknolojisi kendisine aktif olarak kullanım alanı bulmuş ve yeni özellikler kazanmıştır. Dapp kavramı blokzincir 2.0 ile teknoloji dünyasında yer almaya başlamış fakat üçüncü nesille birlikte daha verimli, ölçeklenebilir ve güvenli bir alt yapıda çalışabilmesi için araştırma yapılan bir alan haline gelmiştir. Daha verimli ve ölçeklenebilir bir altyapıya sahip olabilmek için pek çok yeni konsensüs algoritmasından yararlanılmaktadır [26].

#### **2.4.6.4. Blokzincir 4.0**

Önceki blokzinciri aşamaları, endüstriyel hedefleri doğrudan sağlayamadığı için blokzincirinin, Endüstri 4.0'ın teknolojik temellerine yanıt vermesi için yapılan geliştirmeleri kapsayan süreci ifade eder. Blokzinciri teknolojisinin güncel kalabilmesi ve kullanılmaya devam edilebilmesi için sağlaması gereken yeni özellikler vardır ve bundan dolayı da geliştirilmeye açık bir alandır [26]. Ayrıca, blokzincirinin AI, IoT ve siber güvenlik alanlarında kullanılabilmesi için sağlaması gereken yeni özellikler bulunmaktadır [26, 28]. Bu yüzden bu süreç tamamlanmayan ve devam eden bir süreçtir [26, 27, 28].

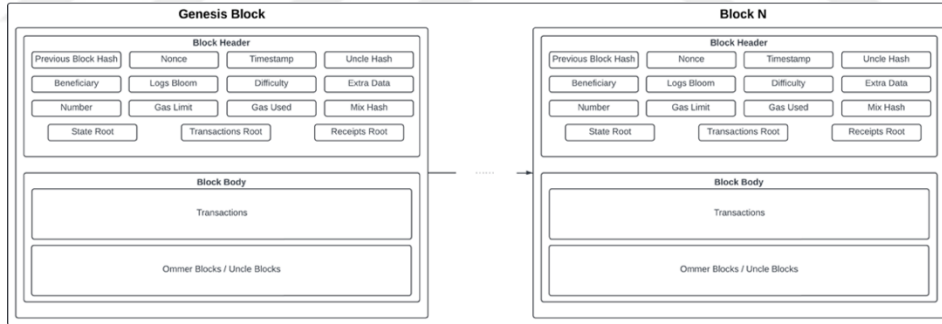
#### **2.4.7. Ethereum**

Ethereum, 2013 yılında Vitalik Buterin tarafından önerilmiş ve 2015 yılında piyasaya sürülmüş bir blokzinciri platformudur [37]. Bitcoin'den farklı olarak aynı zamanda programlanabilir bir blokzinciri platformu oluşturmak amaçlanmıştır. Önceden belirlenmiş kurallar ve koşullara göre çalışan programlar, akıllı sözleşme olarak tanımlanmaktadır [37]. Ethereum tabanlı platformlarda geliştirilen akıllı sözleşmeler için Solidity programlama dili kullanılır ve yerine kullanılabilecek alternatif diller de mevcuttur. Ethereum'un, DeFi (Decentralized Finance), NFT (Non-Fungible Tokens) ve DAO (Decentralized Autonomous Organizations) gibi kullanım alanları vardır. DeFi merkezi olmayan finans uygulamalarını,

NFT benzersiz dijital varlıkların oluşturulması ve ticaretini, DAO ise merkezi olmayan otonom organizasyonların oluşturulmasını ifade eden alanlardır [45, 47].

Akıllı sözleşmeler, Ethereum’da bytecode formatında ve State Trie adı verilen dinamik bir yapıda Ethereum hesabının bir parçası olarak saklanmaktadır ve ağdaki her düğümde bulunması gerekmektedir. Akıllı sözleşmelerin ve kullanıcı arayüzünün geliştirilmesi sonucu oluşan uygulamalar, merkezi olmayan dağıtık uygulamalar (dapp) olarak ifade edilmektedir. Ethereum platformu üzerinde geliştirilen akıllı sözleşmeler, Ethereum Sanal Makinesi (EVM) üzerinde çalışmaktadır. Dapp’ler ise EVM üzerinde çalışan bu akıllı sözleşmeleri kullanarak gerekli fonksiyonel özelliklerini kullanıcılara sunmaktadır.

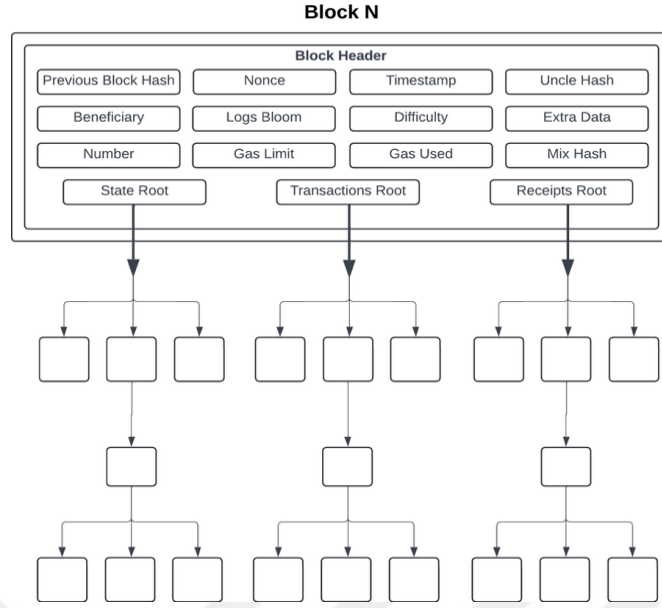
EVM üzerinde çalışan akıllı sözleşmelerin, dağıtılması ve yürütülmesi süreci için ağı hesaplama gücü kullanılacağından harcanması gereken bir enerji ve bunun sonucunda oluşacak bir maliyet mevcuttur. Bu hesaplama gücü için oluşan maliyete gaz ücreti (gas fee) denir. Bu gaz ücretinin birimi ise Ethereum platformunda Ether (ETH) olarak adlandırılır. Ethereum platformu üzerinde her işlem için minimum bir sabit ücret vardır. Akıllı sözleşmeler veya diğer tüm işlemler için belirli bir miktar Ether ödenmesi gerekmektedir.



**Şekil 2. 9.** Ethererum için kullanılan blok modeli.

Ethereum, ilk duyurulduğunda PoW konsensüs mekanizmasına sahipken, 2022 yılı itibariyle ağda yapılan güncelleme ile PoS konsensüs mekanizmasına geçiş yapılmıştır [38, 39]. Bu durum ağın ölçeklenebilirliği geliştirmiş ve enerji maliyetinin düşmesine neden olmuştur [39].

Bitcoin ile oluşturulan blokzinciri modelinde Şekil 2.6.’da gösterildiği gibi, header ve body isimli iki bölüm bulunmaktadır. Şekil 2.9.’da görüldüğü üzere, Ethereum platformunda yapı benzerlikler taşımakla birlikte bazı teknik farklılıklar da bulunmaktadır.



**Şekil 2. 10.** Ethereum blok yapısında header bölümü.

Bitcoin, bir blok işlendiğinde işlem geçmişi değiştirilemeyecek bir şekilde tasarlanmıştır. Ethereum platformunda ise, her işlem sonucu ağdaki hesapların veya akıllı sözleşmelerin durumu değiştirilebilmektedir. Bu değişiklik Account Trie veya State Trie olarak adlandırılan dinamik bir yapı sayesinde gerçekleştirilmektedir. Ağ üzerinde bir Ether transferi gerçekleştirildiğinde, gönderen ve alıcı hesap bakiyeleri güncellenmekte ve ardından State Trie yapısı içerisinde yeni bir kök özet (root hash) değeri oluşturulmaktadır. Bu özetleme değeri de State Root değerini temsil etmektedir.

Şekil 2.6.'da gösterilen Bitcoin blokzinciri yapısında, body bölümündeki transaction olarak adlandırılan işlemlerle, Merkle Tree ve özetleme fonksiyonları kullanılarak Merkle Root isimli tek bir özetleme değeri oluşturulması sağlanır ve bu Merkle Root değeri header bölümünün bir parçasını oluşturmaktadır [43]. Şekil 2.10.'da gösterildiği gibi Ethereum'da ise State Root, Transactions Root ve Receipts Root isimli üç farklı özetleme değeri bulunmaktadır [41, 58, 59].

State Root, ağdaki tüm hesapların ve akıllı sözleşmelerin güncel durumunu (state) ifade etmektedir. State tutulabilmesi için Merkle Tree ve Radix Trie (Patricia Trie) yapılarının birleşiminden oluşan Merkle Patricia Trie (MPT) adı verilen bir veri yapısı kullanılır [41, 46, 58]. Ethereum platformunda, her hesap ve akıllı sözleşme bir Ethereum adresine sahiptir ve bu adreslere ait durum bilgileri sistemde tutulmaktadır. MPT ile tüm hesap ve akıllı sözleşmelerin,

adres ve durumu bir anahtar-değer eşleştirmesiyle (key-value) hiyerarşik bir şekilde tutulabilmektedir [59]. Bu bilgilerin tutulduğu yapı State Trie olarak isimlendirilmektedir.

State Trie içerisinde Ethereum hesapları, kullanıcı (externally owned accounts) ve akıllı sözleşme hesapları (smart contract accounts) olarak iki türe ayrılır [45]. Kullanıcı hesapları, kullanıcılar tarafından kontrol edilen ve depolama alanı olmayan türleri ifade eder. Akıllı sözleşme hesapları ise, EVM tarafından çalıştırılan programlardır ve depolama alanları bulunması gerekmektedir [44].

Ethereum platformunda, State Root değerinin oluşturulabilmesi için MPT veri yapısı ile çalışan bir tane State Trie vardır. Hem kullanıcı hesapları hem de akıllı sözleşme hesaplarının adresleri State Trie içerisinde temsil edilirler. State Trie üzerinde bu hesapların tutulabilmesi için her hesabın, veri yapısında bir düğüm olarak tutulması gerekmektedir. Bu düğümler anahtar-değer (key-value) şeklinde tutulmalıdır. Burada anahtar (key) değerleri, hesap veya akıllı sözleşmelerin özetlenmiş Ethereum adresleridir. Kullanıcı hesapların ve akıllı sözleşme hesaplarının, değer (value) kümesinde nonce, balance, storageRoot ve codeHash bölümleri bulunmaktadır. Bu değer kümesinin, MPT üzerinde tutulabilmesi için özetlenmiş olması gerekmektedir. Bu yüzden Nonce, balance, storageRoot, ve codeHash alanları birleştirilerek tek bir özetleme değeri oluşturulur ve bu da değer kümesini ifade eder.

Bu kapsamda, State Root değerinin oluşturulabilmesi için, State Trie yapısında en alt düğümlerden başlanarak her seviye, özetleme (hash) fonksiyonlarıyla birleştirilir ve sonunda yeni bir kök özet değeri elde edilir. Bu süreç tek bir kök (root) özetleme değeri elde edene kadar devam eder. Son durumda elde edilen kök özetleme değeri ise State Root olarak tanımlanır ve kullanılır [46].

State Trie üzerindeki değer kümelerinde storageRoot isimli bir alan bulunmaktadır. Kullanıcı hesaplarını oluşturan düğümde bu bölüm boştur fakat akıllı sözleşme hesaplarını oluşturan değer kümesinde bu bölüm bir özetleme değeri saklamaktadır. storageRoot, akıllı sözleşmenin depolama alanında bulunan verilerin oluşturduğu bir özetleme değeridir. Akıllı sözleşme hesapları, depolama alanına sahip olduğundan MPT veri yapısını kullanan bir Storage Trie bulunmaktadır.

Akıllı sözleşmelerin depolama alanında saklanan veriler (değişkenler vb.), anahtar-değer ilişkisi içerisinde MPT veri yapısını kullanan bu Storage Trie üzerinde bir düğüm

oluşturacak şekilde tutulmaktadır. Tek bir storageRoot değerini oluşturulabilmesi için ilk önce en alt düğümlerden başlanarak her seviyedeki düğümlerin birleştirilmesi sonucu yeni bir özetleme değeri elde edilir. Bu süreç tek bir kök özetleme değeri elde edene kadar devam eder. Tek bir kök özetleme değeri oluşturulduğunda bu değer storageRoot olarak kabul edilir. Bu storageRoot ile bir akıllı sözleşmenin tüm depolama verileri doğrulanabilir hale gelir. Sözleşmeyle ilgili bir değişiklik olduğunda (değişkenin değer değişimi vb.), bu trie üzerinde bir güncelleme işlemi yapılır ve yeni bir storageRoot oluşturulmuş olur.

Kısaca, State Trie, tüm hesapların durum bilgileri tutan bir yapıdır. Storage Trie, sözleşmeye ait verileri tutar. Her akıllı sözleşme için ayrı bir Storage Trie oluşturulması gerekmektedir. Storage Trie ile oluşturulan storageRoot ilgili sözleşmenin State Trie üzerindeki bulunduğu düğümün değer kümesine bir alan olarak eklenir. MPT aracılığıyla elde edilen kök özetleme değeri, ilgili hesabın değer kümesinde yer alan storageRoot alanına kaydedilmektedir.

Transactions Root, Merkle Tree veri yapısı kullanılarak elde edilen bir alandır. Tüm işlemlere özetleme işlemi uygulanır. Sonra ikili gruplar halinde birleştirilerek tekrar özetleme işlemine tabi tutulur. Bu durum tek bir özetlenmiş değer (Transactions Root) elde edene kadar devam eder. Şekil 2.8.'de Merkle Tree çalışma mantığı gösterilmiştir.

Receipts Root, Merkle Tree veri yapısı ile çalışan ve her işlem sonucunda oluşan bilgilerin bir özetini tutan alanı ifade eder. Ethereum'da her işlem sonuçlandığında, bir işlem makbuzu (receipt) oluşturulmaktadır. Bu işlem makbuzlarının hepsine özetleme işlemi uygulanır. Sonra ikili gruplar halinde birleştirilerek tek bir özetleme değeri elde edilmesi sağlanır. Bunun sonucunda da tek bir özetlenmiş değer (Receipts Root) elde edebilmek için bu durum kök (root) oluşana kadar devam eder.

## 2.5. OYLAMA SİSTEMLERİ

Oylama, bir arada yaşayan insanların ortak bir karara varmak için tercihlerini ifade etmeleri amacıyla kullandıkları bir araçtır. Bu yüzden, tarih boyunca çeşitli nedenler için seçimler düzenlenmiş, bir karara varabilmek için oylama yapılmış ve insanlar tercihlerini bu şekilde ortaya koymuşlardır [1]. Bu kapsamda, seçimler bir süreci ifade ederken, oylama ise kararı ve konuyu belirlemek için yapılan işlemi tanımlamaktadır. Seçim süreçlerinde karara varabilmek içinse doğru bir oylama sisteminin geliştirilmesi gerekmektedir.

Seim srelerinde, oylamanın gerekleřtirilmesi iin zarf, pusula ve mhr gibi nesneler kullanılmaktadır. Fakat teknolojiadaki geliřmelerin neticesinde dijitalleřmenin hız kazanması, elektronik seimlerin kullanılmasına neden olmuřtur [3]. Gnmzde, niversiteler gibi kurum ve kuruluřlar; zarf, pusula ve mhrden oluřan nesnelerle kğıt tabanlı oylama sistemlerini veya dijitalleřmeyle birlikte gelen elektronik tabanlı oylama sistemlerini, eřitli kararları almak iin kullanmaktadır [12].

### **2.5.1. niversitelerde Kullanılan Oylama Sistemleri**

niversitelerde eřitli kararlar almak iin seim sreleri planlanmakta ve bu seimlere uygun oylama sistemleri tasarlanmaktadır. Bu seim sreleri, akademik ynetimin ve ğrenci topluluklarının belirlenmesi iin hazırlanmaktadır. ncelikle, seim srelerinin gerekleřtirilmesi amacıyla tasarlanan oylama sistemlerinde aktrlerin belirlenmesi gerekmektedir. Bu aktrler, bir ynetici veya başkan seimi iin adayı veya semeni ifade ettiğı gibi, herhangi bir adayın bulunmadığı ve yalnızca nceden belirlenmiř seeneklerin yer aldığı oylama senaryolarında, bu seenekler arasında oy kullanacak semenleri kapsayacak řekilde tanımlanabilmektedir.

niversitelerde, lkelerin yasalarına gre deėiřiklik olabilmekle birlikte genel olarak akademik ynetim, oylama ile belirlenmektedir. niversitelerin faaliyetlerine devam edebilmesi ve gerekli kararların alınması, seilecek ynetim ve ynetim kurulları tarafından alınan kararlarla saėlanmaktadır. Bu kapsamda, rektr seimi, dekan seimi, blm başkanı seimiyle senato ve ynetim kurulu kararları, oylama sistemleri ile gerekleřtirilmektedir.

Rektr seimleri, niversitenin ğretim yeleri tarafından yapılan oylama sonucu belirlenmektedir. Bir fakltenin yneticisi konumunda olan dekan seimiye ilgili fakltenin ğretim yeleri ve akademik personelin oy kullanması sonucu gerekleřmektedir. Blm başkanı, ilgili blmn akademik personeli tarafından yapılan oylama iřlemiyle belirlenmektedir. Bir niversitenin senato ve ynetim kurulu, akademik ve idari konularda alınacak herhangi bir kararı aralarında yaptıkları oylama ile gerekleřtirmektedir.

niversitelerde, ğrencilerin demokratik sreleri kullandığı eřitli seim sreleri gerekleřtirilmekte ve bu srelere uygun oylama sistemleri hazırlanmaktadır. ğrenci konseyi seimleri, blm temsilcisi seimi ve ğrenci kulb başkanlık seimleri bu srelere birer rnektir. Bu seimler iin uygun oylama sistemlerinin hazırlanması gerekmektedir.

Bu seçimlerin yanında, üniversiteler; öğrencilerin memnuniyetini ölçmek için anketler, yapılması düşünülen etkinlikler için karar almak ve düzenlenen sanatsal etkinliklerde jüri ve izleyicilerin sonucu belirlemesini sağlamak amacıyla oylamalar gerçekleştirir.

### **2.5.2. Kâğıt Tabanlı ve Elektronik Oylama Sistemleri**

Seçim süreçlerinin gerçekleştirilmesi için tarih boyunca çeşitli oylama sistemleri kullanılmıştır [1, 62]. Bu kapsamda, ilk seçim örnekleri incelendiğinde, oylama işlemlerinde taş, kırık çömlek parçaları, kleroterion (kura makinesi) ve parşömen gibi araçların kullanıldığı tespit edilmiştir [63]. Demokratik süreçlerini geliştirmek isteyen ülkelerin, güvenilir ve düzenli bir seçim sürecine sahip olabilmek amacıyla belirlediği kurallar sebebiyle söz konusu araçlar zamanla yerini daha gelişmiş veya uygun alternatif araçlara bırakmıştır.

Seçimlerin daha güvenli ve düzenli hale gelmesi için tarihte, oy pusulası, mühür, zarf ve sandık gibi araçların kullanıldığı kâğıt tabanlı oylama sistemleri geliştirilmiştir. İlk olarak 19. yüzyılda kullanılmaya başlayan sistem, halen birçok kurum veya kuruluştaki temel bir oylama yöntemi olarak tercih edilmektedir [64]. Bunun yanı sıra, gelişen teknoloji çeşitli alanlarda değişime neden olmuştur. Bu durum dijitalleşmenin hızını artırmış, çeşitli alanda dijital çözümler geliştirilmesini sağlamış ve seçim süreçlerinde kullanılan oylama sistemlerinin değişmesine neden olmuştur. Bu değişim neticesinde, 19. yüzyılda mekanik oylama makinaları ve 20. yüzyılda delikli kart sistemleri, oylama işlemlerini gerçekleştirmek için kullanılmıştır [65]. Bu süreci, 20. yüzyılda dokunmatik ekranlı cihazların oylama işlemi için kullanımı izlemiştir [67].

21. yüzyıl içerisinde teknolojinin geldiği nokta itibarıyla Estonya ülkesi, internet üzerinden gerçekleştirilebilen elektronik oylama sistemlerini kullanmaya başlamıştır [66]. Seçim süreçlerindeki maliyetlerin azaltılması, erişim kolaylığı ve hızlı sonuç alabilme yetenekleri, elektronik oylama sistemlerinin kullanım alanı bulmasına neden olmaktadır.

Kâğıt tabanlı ve elektronik oylama sistemleri, kurum ve kuruluşlarda kullanım alanına sahiptir [68]. Bu kapsamda, Tablo 2.5.'de gösterildiği gibi her sistemin olumlu ve olumsuz yönleri bulunmaktadır. Geliştirilecek sistemlerde bu konulara dikkat edilmesi büyük önem taşımaktadır.

**Tablo 2. 5.** Kâğıt tabanlı ve elektronik oylama sistemlerinin karşılaştırması.

| Kriter             | Kâğıt Tabanlı Oylama Sistemleri                                                    | Elektronik Oylama Sistemleri                                                                            |
|--------------------|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Güvenlik           | Fiziksel ortamdaki değişkenlere bağlıdır.                                          | Siber güvenlik kriterlerine bağlıdır.                                                                   |
| Süre               | Katılım sayısına bağlı olarak oy kullanma ve sonuçlanma uzun sürebilmektedir.      | Hızlı oy kullanma ve otomatik sayım imkânı ile süre kısabilmektedir.                                    |
| Maliyet            | Katılım sayısına bağlı olarak gözlemci, matbaa ve depolama maliyeti bulunmaktadır. | Teknolojik altyapı, teknik personel ve bakım maliyetleri bulunmaktadır.                                 |
| Erişilebilirlik    | Fiziksel olarak katılım gerektirmektedir.                                          | Akıllı telefon, bilgisayar veya oy makineleriyle internet üzerinden katılım gerçekleştirilebilmektedir. |
| Manipülasyon Riski | İnsan faktörüne bağlıdır.                                                          | Teknoloji tasarımına ve siber güvenlik önemlerine bağlıdır.                                             |

#### 2.5.2.1. Kâğıt Tabanlı Oylama Sistemleri

Kâğıt tabanlı oylama sistemleri, seçmenlerin zarf, pusula, sandık ve mühür gibi araçları kullanarak, oy kullanma işlemlerini gerçekleştirdikleri yöntemi ifade etmektedir. Seçmenler kâğıt tabanlı oylama sistemlerinde, fiziksel olarak oylamanın yapıldığı yerde bulunmak ve kimliklerini kişisel kimlik kartı, pasaport vb. nesneyle ispat etmek zorundadır.

Her seçmene, seçim sürecinde oylama işlemini gerçekleştirebilmek için tüm adayların olduğu oy pusulası verilmektedir. Seçmen, oy pusulasında oy vermek istediği ilgili aday için ayrılan alanı mühür gibi bir araçla işaretlemektedir. Kapalı bir zarfa bu oy pusulasını yerleştirerek kapalı bir sandığa pusulayı atmaktadır. Oy kullanma işlemi bittiğinde, ilgili sandıklar sayım süreci için yetkililer tarafından açılmakta ve sayım şeffaf bir şekilde gerçekleştirilmektedir. Kurum ve kuruluşlar arasında çeşitli yönetmelik farklılıkları olmakla birlikte genel olarak kâğıt tabanlı oylama sistemleri bu şekilde gerçekleştirilmektedir. Seçimin fiziksel olarak ve insanların etkisiyle gerçekleştirilmesi sebebiyle bazı olumlu ve olumsuz yönler bulunmaktadır.

Kâğıt tabanlı oylama sistemleri, büyük bir teknolojik altyapı gereksinimine ihtiyaç duymadan kullanılabilir. Bu yüzden standart ve basit prosedürlere sahiptir. Tüm seçim süreci seçmen, gözlemci ve yetkililer tarafından şeffaf bir şekilde gerçekleştirildiğinde manipülasyon imkânı zorlaşmaktadır. Yeterli oy pusulası olmadığında, yetkililer ve gözlemciler tarafından eksiklikler hızlıca telafi edilebilmektedir.

Bu sistemlerde, katılımcı sayısına bağlı olarak seçimin gerçekleştirilmesi ve sonuçlanması uzun sürebilmektedir. İnsan faktörünün söz konusu olduğu durumlarda hataların olduğu gözlemlenmekte, kötü niyetli gözlemciler ve yetkililerin seçimi manipüle etmesi söz konusu olabilmektedir. Katılımcı sayısı arttıkça, hazırlanması gereken kâğıt sayısı maliyete olumsuz etki etmekte ve seçim sonucunda tüm her şeyin saklanması yüksek depolama maliyetine neden olmaktadır [6]. Öte yandan, sıfır atık ilkesine bağlı hedefler, yüksek kâğıt kullanımı sebebiyle gerçekleştirilememektedir.

#### **2.5.2.2. Elektronik Oylama Sistemleri**

Elektronik oylama sistemleri, seçim süreçlerinin bir parçası olan oylama işlemlerinin dijital araçlarla gerçekleştirilmesidir. Bu kapsamda, oy verme, sayım ve doğrulanma süreçleri dijital araçların yardımıyla gerçekleştirilmektedir. Oylama işleminin gerçekleştirilmesi için belirli merkezler belirlenmekte ve bu merkezlere, seçmenler ulaşarak makinalar (kiosk) aracılığıyla oy verme sürecini tamamlayabilmektedirler. Bu makinalar, kimlik tespiti için kullanıcıdan kimlik kartı veya bir giriş bilgisi talep etmektedir. Bunun yanı sıra, sadece internet ortamından çalışan elektronik oylama sistemleri de tasarlanabilmektedir. Bu yaklaşımda, kullanıcılar herhangi bir merkeze fiziksel olarak ulaşmadan, sadece cep telefonu ve bilgisayarlar aracılığıyla oy verebilirler. Oylama işleminin dijital ortamda gerçekleşmesinin olumlu ve olumsuz yönleriyle birlikte aşılması gereken çeşitli zorlukları bulunmaktadır.

Elektronik sistemlerde oylamanın gerçekleşme ve sonuçların açıklanma süresi kısalmaktadır. Öte yandan, insan faktörü ortadan kalktığı için buna bağlı hatalarda ortadan kalkmaktadır [6]. Ayrıca, internet ortamında yapılabilen oylamalar, seçmenlere kolaylık sağlayacağından katılım oranında artış olması mümkündür. Bunun yanı sıra, internet ortamında yapılan oylamalar sayesinde engelli ve uzak konumda bulunan seçmenlerin seçimlere ulaşmasının önündeki engeller ortadan kalkmaktadır. Oylama işleminin, kâğıt tabanlı bir sistem yerine elektronik ortamda gerçekleştirilmesi kâğıt, baskı, nakliye ve depolama açısından tasarruf yapılmasını sağlamaktadır [69].

Güvenlik ve şeffaflık endişesi sebebiyle seçmenler, elektronik oylama sistemlerini kabul etmekte zorlanabilirler. Ayrıca, bu sistemleri kullanacak seçmenin, dijital okuryazarlığının yüksek olması gerektiğinden bir eğitim süreci düzenlenmelidir. Oylama sırasında yaşanabilecek teknik problemler seçim sürecinde aksama ve ertelemelere neden olmaktadır. Bu duruma kırsal bölgelerin altyapı yetersizliği de eklendiğinde, ortaya çözülmesi zor problemler çıkmaktadır.

Elektronik oylama sistemlerinde, seçimin büyüklüğüne bağlı olarak teknolojik altyapının, donanımların ve yazılımsal araçların ölçeklenebilir olması gerekmektedir. Bu sistemleri geliştirecek ve bakımını yapabilecek yeterli teknik personel ihtiyacı olacağından, kurum ve kuruluşlar için ek maliyet gerekmektedir [6]. Kullanılan teknolojilerin güvenilir olması ve sistemin siber saldırılara karşı dirençli olması önemlidir. Veri güvenliğinin sağlanması temel bir zorunluluktur. Bu konuda yapılacak çalışmalar belirli standartlara ve yönetmeliklere göre düzenlenmelidir. Öte yandan, seçmen ve oy gizliliğinin sağlanması gerekmektedir. Böylesi sistemlerde, oy gizliliği korunurken, sonuçların doğrulanmasını sağlayan mekanizmalar kurmak ve seçmenlerle siyasi partilerin sisteme güvenmesini sağlamak zor bir dengedir. Fransa ve İsveç gibi ülkelerin dijital oylama süreçlerinde, çeşitli sebeplerle teknik ve sosyal sorunlar ortaya çıkmıştır [5].

### 2.5.3. Merkezi ve Merkezi Olmayan Oylama Sistemleri

Teknolojideki gelişmeler neticesinde dijitalleşme hız kazanmış ve bu durumda elektronik oylama sistemlerinin ortaya çıkmasına neden olmuştur [3, 12, 66]. Elektronik oylamalar, merkezi ve merkezi olmayan sistemler olarak iki kategoriye ayrılmaktadır. Bu iki kategori; oy kullanma, toplama, sonuçların açıklanması ve doğrulanması süreçlerinde birbirinden farklı yapısal iş akış şemasına sahiptir. Tablo 2.6.'da gösterildiği gibi iki kategorinin de olumlu ve olumsuz yönleri bulunmaktadır.

**Tablo 2. 6.** Merkezi ve merkezi olmayan oylama sistemlerinin karşılaştırılması.

| Kriter             | Merkezi Oylama Sistemleri                                                                      | Merkezi Olmayan Oylama Sistemleri                                                                                               |
|--------------------|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Güvenlik           | Merkezi bir kurum veya birden çok kuruma bağlıdır. Güvenlik kurumun aldığı önlemlere bağlıdır. | Dağıtık yapısı sayesinde tek bir otoriteye bağlı değildir. Geliştiricilerin ve ağın aldığı ortak güvenlik önlemlerine bağlıdır. |
| Manipülasyon Riski | Merkezi müdahale ile manipülasyon yapılabilir.                                                 | Tek bir otoritenin olmaması ve kayıtların değiştirilememesi manipülasyon ihtimalini düşürmektedir.                              |
| Maliyet            | Geliştirme ve bakım maliyetleri düşüktür.                                                      | Geliştirme, bakım ve teknik personel masrafları yüksektir.                                                                      |
| Şeffaflık          | Merkezi bir otoritenin kontrolünde olduğu için verilere erişim kısıtlıdır.                     | Veriler tüm katılımcılara açıktır ve doğrulama yapılabilir.                                                                     |

#### 2.5.3.1. Merkezi Oylama Sistemleri

Merkezi oylama sistemlerinde, oylama yönetimi ve veri işleme yetkisi tek bir merkez veya sınırlı sayıda merkez tarafından gerçekleştirilmektedir. Yetki sahibi kurum veya

kurumlar; sistemin kurulması, seçmen ve yönetici hesaplarının belirlenmesi, oy kullanımının gerçekleştirilmesi ve sonuçların ilanına kadarki tüm süreci tek başına gerçekleştirmektedir.

#### **2.5.3.2. Merkezi Olmayan Oylama Sistemleri**

Merkezi olmayan oylama sistemlerinde, oylama yönetimi dağıtık bir yapı kullanılarak gerçekleştirilir. Tek bir merkezi yapı olmadığı için ağ üzerindeki tüm katılımcı düğümler (nodes) sistemin bir parçasıdır. Tüm seçim süreci ve veri işleme yetkisi ağdaki bu düğümler arasında eş zamanlı olarak gerçekleştirilir.

Blokzinciri teknolojisi, merkezi olmayan dağıtık oylama sistemleri için altyapı sağlayan teknolojilerden birisidir. Blokzinciri teknolojisiyle çalışan merkezi olmayan oylama sistemlerinde; ağdaki her kullanıcı, oylama işlemlerinin şeffaflığını ve bütünlüğü kontrol edebilmektedir. Blokzinciri üzerinde tutulan tüm veriler değiştirilemediğinden, geçmiş kayıtlarla ilgili endişeler ortadan kalkmaktadır.

#### **2.5.4. Güvenli Elektronik Oylama Sistemleri için Gereksinimler**

Kurum veya kuruluşların, elektronik oylama sistemlerini, geleneksel oylama sistemlerine bir alternatif olarak değerlendirdiği gözlemlenmektedir [66]. Geleneksel oylama sistemlerine göre hız, erişilebilirlik, kâğıt tasarrufu ve nakliye açısından avantajları bulunmaktadır [69]. Fakat, elektronik ortamda gerçekleştirilen oylama işlemleri için sağlanması gereken temel gereksinimler bulunmaktadır [7, 70, 71, 72]. Elektronik oylama sistemlerinin kurum veya kuruluşlarda kullanılabilmesi için yasal düzenlemelere göre düzenlenmesi gerekmektedir. Kurum ve kuruluşların ihtiyaçlarına bağlı olarak yeni gereksinimler ortaya çıkabilse de, aşağıda belirtilen temel unsurlar, demokratik ve şeffaf bir oylama sisteminin sağlanabilmesi için vazgeçilmezdir.

##### **2.5.4.1. Gizlilik ve Anonimlik (Privacy and Anonymity)**

Bir oylama sisteminde seçmenlerin oy tercihleri gizli kalmalıdır. Gizli kalmaması, seçmenlerin üzerinde baskı oluşturmakta ve demokratik bir seçim sürecinin gerçekleşmemesine neden olmaktadır. Aynı zamanda, seçmenin elektronik ortamdaki kimliğiyle gerçek kimliği arasında bir izolasyon olması gerekmektedir. Bu durum kişinin kimliğinin gizlenmesine ve oy tercihinin açığa çıkmamasına yardımcı olur.

##### **2.5.4.2. Adalet (Fairness)**

Oylama işlemi sırasında herhangi bir seçmenin veya yöneticinin, seçim sonuçlarını etkileyebilecek özel bir yetkisinin olmaması gerekmektedir. Oylama sırasında hiçbir özel

kullanıcı seçim sonuçlarını görememelidir ve seçimle ilgili sürecin sonlandırılmasıyla sonuçlar demokratik şartlara uygun şekilde ilan edilmelidir.

#### **2.5.4.3. Tekrar Kullanılamazlık (Unreusability)**

Oylama sistemlerinde, her seçmenin bir kez oy kullanma hakkı olmalıdır. Böylece, seçim bütünlüğü korunacak ve adil bir seçim süreci gerçekleştirilecektir. Bu nedenle, bir seçmenin tekrar oy kullanamaması için uygun mekanizmaların geliştirilmiş olması gerekmektedir. Bu gereksinim sağlanmadığında, seçim sonuçları seçmenler, yöneticiler veya saldırganlar tarafından manipülasyona açık hale gelmektedir. Bu durum da seçim sürecinin kamuoyu tarafından olumsuz olarak değerlendirilmesine neden olmaktadır.

#### **2.5.4.4. Seçmen Uygunluğu (Eligibility)**

Oylama işlemlerinin doğru yapılabilmesi için; seçimlerde, sadece seçmen olarak kayıtlı kişilerin oy kullanılması sağlanmalıdır. Bunun için uygun kimlik doğrulama (authentication) ve yetkilendirme (authorization) mekanizmaları geliştirilmelidir.

#### **2.5.4.5. Geçerlilik (Soundness)**

Sadece oylama sisteminin belirtmiş olduğu kurallar dahilinde kullanılan oylar geçerli kabul edilmelidir. Kurallara aykırı kullanılan oylar, sistem tarafından reddedilmelidir ve gerekli uyarı mekanizmalarıyla teknik personel ve yöneticilerin bilgilendirilmesi sağlanmalıdır. Sistemin, manipülasyona sebep olabilecek oy kullanma işlemlerini filtrelemesi gerekmektedir.

### 3. YÖNTEM

Blokzinciri teknolojisiyle merkezi olmayan üniversite ölçekli bir oylama sisteminin tasarlanması ve uygulanması için izlenen metodolojik yaklaşımlar ayrıntılı olarak bu bölümde açıklanmaktadır. Kullanılan yöntemler, sistemin güvenliğini, sürdürülebilirliğini, seçmen gizliliğini ve genel performans kriterlerini sağlamak amacıyla belirlenmiştir. Bu bağlamda, sistem aktörleri ve gereksinimleri, kullanılan teknoloji ve araçlar, mimari tasarım ve kullanılan kriptografik teknikler ayrıntılı olarak açıklanmaktadır.

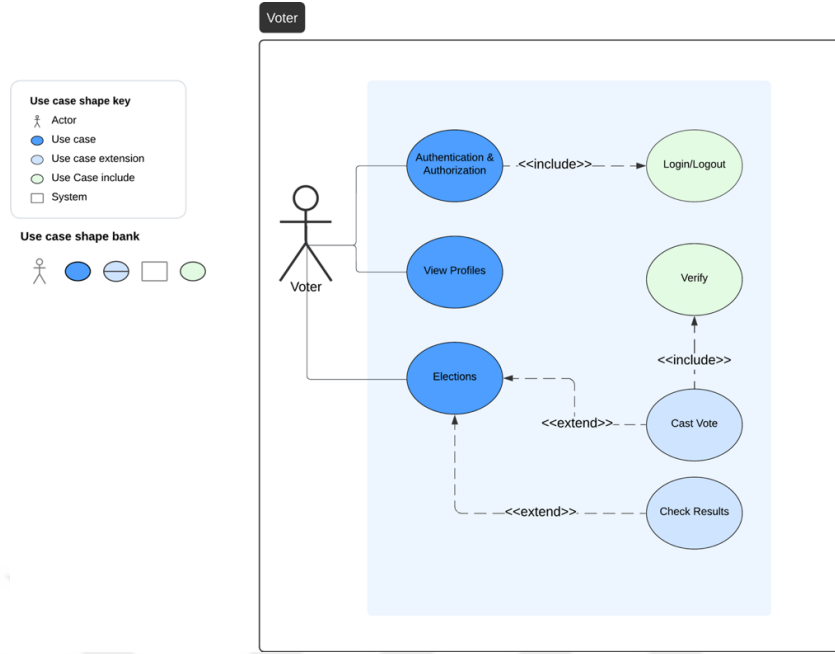
Oylama süreçlerinde şeffaflık, güvenilirlik, doğrulanabilirlik ve gizlilik önemli kriterlerdir ve bu kriterler arasında denge kurulması gerekmektedir. Bir seçmenin oy verme sürecinde kimliğinin gizliliği korunurken, kullanmış olduğu oyun şeffaf bir şekilde ispatlanması, bu dengenin doğru bir şekilde çalıştığının göstergesidir. Bu dengenin doğru kurabilmesi mimari ve teknolojik seçimlerin doğru belirlenmesine bağlıdır. Geliştirilen prototip kapsamında yapılan tüm teknik seçimler, bu dengeyi gözeterek belirlenmiştir.

#### 3.1. SİSTEMİN AKTÖRLERİ

Oylama sisteminde, sürecin adil, güvenli ve şeffaf yürütülebilmesi için farklı aktörler bulunmaktadır. Her bir aktör, oylama sisteminde farklı yetkilere ve sorumluluklara sahiptir. Teknolojik kısıtlamalar ve kriptografik güvenlik önlemleriyle aktörlerin sistemi manipüle etmeleri engellenmektedir. Bu kapsamda, bölüm içerisinde her bir aktörün yetkileri, sınırları ve diğer aktörlerle bağlantıları, oylama sisteminin tamamının nasıl çalıştığını kapsayacak bir şekilde açıklanacaktır.

##### 3.1.1. Seçmen (Voter)

Oylama sisteminde, oy kullanma hakkında sahip olan aktörü ifade etmektedir. Sistem üzerinde her seçmenin kimliği, Ethereum adresleriyle ifade edilmektedir. Seçmenlerin, oylama sistemi üzerinde yapabilecekleri işlemler Şekil 3.1.'deki diyagramda görülebileceği üzere üç başlık altında toplanır ve her biri ayrı ayrı değerlendirilir.



**Şekil 3. 1.** Seçmen aktörü için kullanım durumu (use case) diyagramı.

Bir seçmen, ilk olarak web uygulamasına giriş ve çıkış yapabilmektedir. Bu oylama sisteminde geliştirilen oturum yönetimiyle sağlanmaktadır. Bu oturum yönetimini kullanarak giriş ve çıkış işlemlerinin gerçekleştirilebilmesi için seçmenlerin tarayıcıları üzerindeki MetaMask eklentisine, Ethereum adresleriyle ilişkili özel anahtarlarını (private key) bağlamaları gerekmektedir. Bu işlemi gerçekleştirdiklerinde, oylama sisteminin arayüzünü temsil eden web uygulamasına MetaMask ile giriş yapabilmektedirler.

Seçmenler; oylama sisteminde, kişisel profilleriyle ilgili bilgileri bu web uygulaması aracılığıyla görüntüleyebilmektedirler. Herhangi bir yanlışlık görülmesi durumunda, seçim yöneticileriyle bağlantı kurulabilir ve seçmenin üniversite içerisindeki rolü (öğrenci, akademisyen, memur), fakülte ve departman bilgileri değiştirilebilmektedir. Seçmenler, bir seçim süreci başlatılırken son güncel bilgilere göre oy kullanma hakkına sahip olacaklardır. Aktif seçim için herhangi bir değişiklik yapılmasına izin verilmemektedir.

Seçmenin oy verme, oy doğrulama ve sonuçları görebilmesi web uygulaması aracılığıyla sağlanmaktadır. MetaMask ile web uygulamasına giriş yapıldıktan sonra seçmenin üniversite, fakülte veya departmanlarıyla ilişkili seçimleri görebilmesi mümkündür. Gerekli seçim zamanı içerisinde ise oy kullanma işlemini gerçekleştirebilir ve oyunun sisteme başarıyla eklendiğini doğrulayabilmektedir. Oy kullandıktan sonra içeriğinin görülmesi, oy gizliliğini korumak için mümkün değildir.

### 3.1.2. Oylama Yetkilisi (Election Authority)

Oylama yetkilisi olarak tanımlanan aktör, seçim, seçmen, yetkili, fakülte ve departman yönetimi haklarına sahip aktörleri ifade etmektedir. Oylama sisteminde, seçim sürecinin başarılı bir şekilde takip edilebilmesi için bağımsız olarak seçilen kuruldur. Şekil 3.2.'deki diyagramdan görülebileceği üzere kurul, sistem üzerinde altı farklı kategoride işlem gerçekleştirebilmektedir.

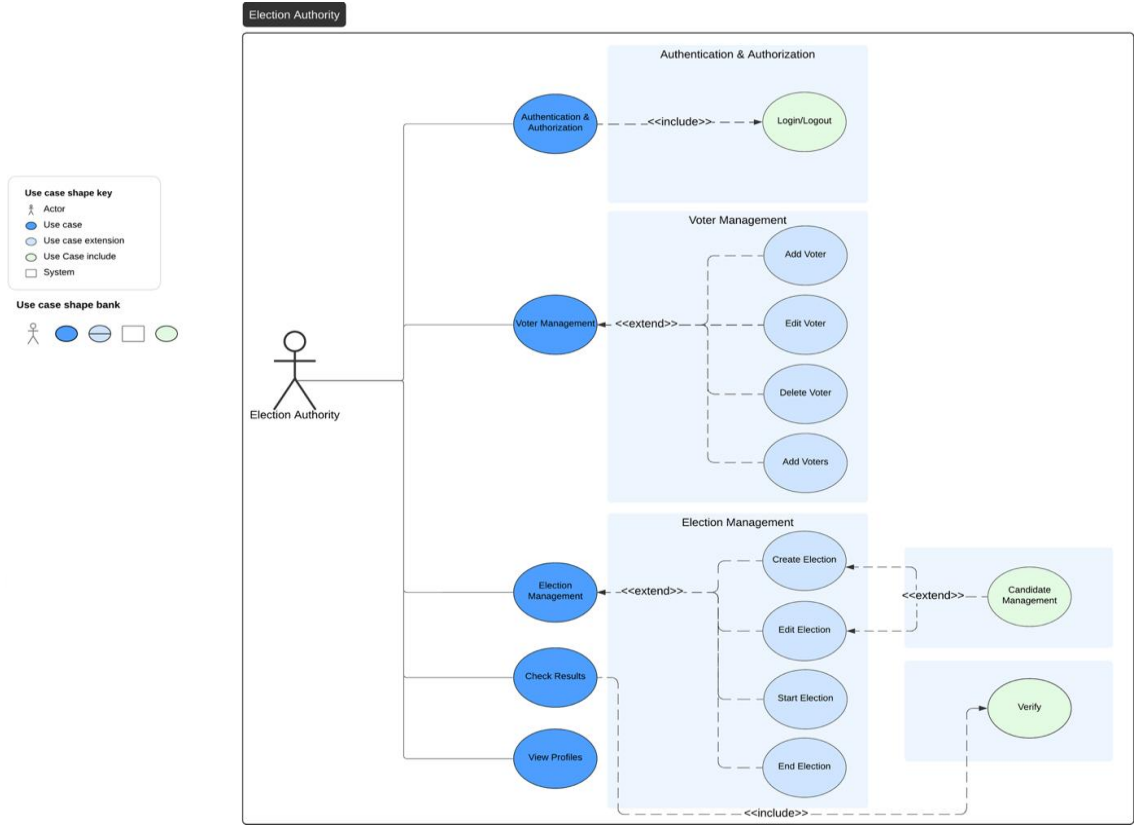
Yetkililerin sistemi kontrol edebilmesi için yetkili paneline giriş ve çıkış yapabilmesi gerekmektedir. Oylama sistemi ilk çalıştırıldığında, ayar dosyalarında belirtilen yetkililer sistem içerisinde otomatik olarak kayıtlı olacaktır. Aktif olan yetkililer tarafından yeni yetkili eklenebilmektedir. Her yetkili, aynı seçmenler gibi Ethereum adresleriyle sistem içerisinde tanımlanmaktadır. Her yetkilinin, giriş ve çıkış işlemini gerçekleştirebilmesi için Ethereum adresiyle uyumlu özel anahtarını (private key) tarayıcıları üzerindeki MetaMask uzantısına eklemesi gerekmektedir. Bu işlem neticesinde panele giriş yaparken MetaMask üzerinden onay işlemini gerçekleştirecek ve ilgili panele erişebileceklerdir.

MetaMask ile yetkili paneline eriştikten sonra, yetkililer sistem içerisindeki bilgilerini görebilmektedirler. Bu bilgilerde herhangi bir yanlışlık olması durumunda teknik personellerle görüşülerek hatanın giderilmesi sağlanmalıdır. Bir yetkiliye ait Ethereum adresiyle, herhangi bir seçmenin olması mümkün değildir.

Yetkililer, seçmen yönetimi bölümünden seçmenlerle ilgili ekleme, düzenleme ve silme işlemi gerçekleştirebilmektedirler. Burada her seçmen Ethereum adresi, rolü (öğrenci, akademisyen, memur), fakülte ve departman bilgisiyle ifade edilmektedir. Kişisel hayattaki kimlik bilgileriyle ilgili detaylar bulunmamaktadır.

Oylama sürecinin yönetilebilmesi için seçim yönetimi bölümü bulunmaktadır. Bu bölüm içerisinde yetkili seçim oluşturabilir, düzenleyebilir, başlatabilir ve bitirebilmektedir. Seçim süreci kapsamında adayların yönetimi de bu bölüm içerisinden gerçekleştirilmektedir. Seçim başlatma işlemi sırasında ilgili seçim için blokzincir üzerine anlık seçmen listelerinin görüntüsü kaydedilmektedir.

Sonuçlandırılan seçim sonuçları yetkili panelinden görüntülenebilmektedir. Seçim sonuçların doğruluğun kontrol edilmesiyle ilgili süreçler bu panel üzerinden yapılabilmektedir.



Şekil 3. 2. Oylama yetkilisi için kullanım senaryosu (use case) diyagramı.

### 3.1.3. Teknik Personel (Operator)

Oylama sisteminin kurulumunu gerçekleştirecek aktörleri ifade etmektedir. Teknik personeller bağımsız kişilerden oluşturulmalıdır. Geliştirilen servislerdeki teknolojik kısıtlamalar ve kriptografik güvenlik önlemleri, sistemin manipüle edilmesinin engellenmesinde kritik öneme sahip olmasına rağmen; kurulumun denetlenebilir bağımsız takımlar tarafından gerçekleştirilmesi gerekmektedir. Dağıtık bir blokzinciri uygulamasının (dapp), kurulumlarını yapacak kişiler bu takımdaki teknik personellerdir.

## 3.2. SİSTEM GEREKSİNİMLERİ

Blokzinciri tabanlı merkezi olmayan üniversite ölçekli oylama sistemi için temel gereksinimler belirlenmiş, bu temel gereksinimler neticesinde teknoloji ve mimari tercihlere karar verilmiştir. Bu kapsamda, oylama sisteminin sağlaması gereken gereksinimler şunlardır:

1. Oy Gizliliği: Oyların, herhangi bir aktör tarafından görülmemesi gerekmektedir. Sistem içerisinden veya dışarısından bir kişi, kullanılan oyun içeriği hakkında bir bilgiye ulaşamamalıdır.

2. Mükerrer Oy Koruması: Bir seçmen, katıldığı seçimde kendisine tanımlanan oy hakkından daha fazla oy kullanmamalıdır. Sistemde, herkesin bir oy kullanma hakkı olmalıdır ve daha fazla oy kullanılması mümkün olmamalıdır.
3. Doğrulanabilirlik: Kullanılan oyun, seçmen ve yetkili tarafından doğrulanabilir olması gerekmektedir. Bu oyun geçerli olduğunun bir göstergesidir.
4. Oy Bütünlüğü: Kullanılan oyların değiştirilmesi ve silinmesi sistem bütünlüğünü açısından güvenlik sorunu oluşturduğundan, oylamadaki hiçbir işlem değiştirilememelidir. Geçmişe ait veriler blokzinciri üzerinden görüntülenebilmelidir.
5. Şeffaflık: Sistemin tüm süreci gözlenebilir bir şekilde herkese açık olmalıdır. Seçmen ve yetkililer ağ trafiğini takip edebilmelidir.
6. Erişilebilirlik: Seçmenlerin ve yetkililerin, geliştirilen sisteme farklı cihazlardan erişebilmesi gerekmektedir.
7. Ölçeklenebilirlik: Oylama sistemi üzerinde gerçekleştirilen testler sonucunda metrikler sunulması gerekmektedir. Bu metriklerin, sistemin gerektiğinde geliştirilebileceğini göstermesi önemli bir göstergedir.

### 3.3. KULLANILAN TEKNOLOJİ VE ARAÇLAR

Bu tez kapsamında şeffaf, güvenilir, doğrulanabilir ve gizlilik ilkelerine bağlı bir sistem inşa edebilmek amacıyla güncel teknolojik ve kriptografik yaklaşımlar kullanılmıştır. Geliştirilen blokzinciri tabanlı oylama sistemi, mevcuttaki geleneksel oylama sistemlerinin çözüm sağlamakta zorlandığı konulara yenilikçi bir yaklaşım sunmaktadır. Bu yenilikçi yaklaşımın sunulmasında kullanılan teknolojilerin tanımları ve kullanım amaçları Tablo 3.1.'de gösterilmektedir.

Oylama işlemlerinin doğru, değiştirilemez ve dağıtık bir şekilde çalışabilmesini sağlamak için Ethereum blokzinciri doğru bir altyapı sunmaktadır. Bu nedenle, blokzinciri platformu olarak tercih edilmiştir. Bu blokzinciri üzerinde Solidity programlama diliyle geliştirilen akıllı sözleşmeler (smart contract) çalışmaktadır ve böylece seçim süreçlerinin otomatik bir şekilde gerçekleştirilebilmesi sağlanmaktadır.

Oy gizliliğini sağlamak ve böylece seçmenlerin bağımsız bir biçimde oy kullanabilmelerine imkân verebilmek amacıyla sıfır bilgi ispatı (Zero-Knowledge Proof) devreleri tasarlanmıştır. Bu devrelerin içerisinde, hesaplama verimliliği ve blokzinciri üzerindeki işlem maliyetini azaltmak amacıyla Poseidon özetleme algoritması kullanılmıştır.

Sıfır bilgi ispatının yanı sıra seçmen ve oy gizliliğini daha güvenli hale getirebilmek amacıyla Paillier homomorfik şifreleme algoritması (Paillier Homomorphic Encryption Algorithm) kullanılmıştır. Böylece kullanılan oylar şifrelenmiş ve oylar üçüncü taraf kişilerden gizlenmiştir. Şifrelenmiş veriler üzerinde matematiksel işlemlerin gerçekleştirilmesine olanak tanıdığı için, oylar açığa çıkmadan şifreli verilerle hesaplamalar yapılabilmektedir.

Sıfır bilgi ispatı sayesinde oy içeriği paylaşılmadan oylamanın doğrulanması ve Paillier şifreleme algoritmasıyla da oyların görülmenden hesaplanabilmesi mümkün hale gelmiştir. Böylece sistem içerisinde güvenlik, gizlilik, şeffaflık ve doğrulanabilirlik kriterleri, dengeli ve sürdürülebilir şekilde bir arada sağlanabilmektedir.

**Tablo 3. 1.** Çalışmada kullanılan teknolojilerin tanımları ve kullanım amaçları.

| Teknoloji                         | Açıklama                         | Kullanım Amacı                                                                                                                                                                                                          |
|-----------------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Blokzinciri Altyapısı             | Ethereum                         | Dağıtık blokzinciri uygulamasının geliştirilebilmesi için tercih edilmiştir.                                                                                                                                            |
| Blokzinciri Aracı                 | Hardhat                          | Ethereum platformu üzerinde geliştirme ve test yapılabilmesi için kullanılmıştır.                                                                                                                                       |
| Programlama Dili                  | Solidity, Java, Bash, JavaScript | Solidity, akıllı sözleşmelerin geliştirilmesi için kullanılmıştır. Java ile cüzdan üretici uygulaması geliştirilmiştir. Bash ile geliştirilen script sayesinde servislerin otomatik olarak dağıtılması sağlanmıştır.    |
| Framework                         | OpenZeppelin, Express            | OpenZeppelin, güvenli ve verimli akıllı sözleşme geliştirmek amacıyla tercih edilen bir kütüphanedir.                                                                                                                   |
| Kriptografik ve Güvenlik Araçları | Circom, Paillier, Poseidon Hash  | Circom ile sıfır bilgi ispatı için devre oluşturulmuştur. Paillier ile verilerin şifrelenmesi sağlanmıştır. Poseidon hash fonksiyonu, devre ve verilerin bütünlüğünü sağlamak amacıyla güvenlik için tercih edilmiştir. |
| Kripto Cüzdan                     | MetaMask                         | Ethereum platformundaki oturum işlemlerinin gerçekleştirilmesi için cüzdan aracı olarak kullanılmıştır.                                                                                                                 |

|                            |                              |                                                                                                                                                                                       |
|----------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Web Geliştirme Kütüphanesi | JavaScript, React, Ethers.js | Seçmen ve yetkili paneli arayüzlerinin geliştirmesi için tercih edilmiştir. Ethers.js, Ethereum blokzinciri ile kullanıcı arayüzü arasında iletişimi sağlamıştır.                     |
| JavaScript Kütüphanesi     | SnarkJS                      | Circom ile yazılan devrenin derlenmesini, web uygulamasında kullanılacak devre dosyalarının oluşturulmasını ve sıfır bilgi ispatlarının üretilmesini sağlamak için tercih edilmiştir. |
| UI/UX                      | Material UI (MUI)            | Material UI, arayüz için hazır butonlar, formlar ve menüler vb. gibi bileşenler sunar.                                                                                                |

### 3.3.1. Ethereum

Dağıtık uygulamalar (dapp) geliştirebilmek amacıyla kullanılan, açık kaynak kodlu ve akıllı sözleşme desteği sağlayan bir blokzinciri platformudur. Geliştirilen oylama sisteminde, işlemlerin değiştirilemez, denetlenebilir ve dağıtılabilir bir biçimde gerçekleştirilebilmesi amacıyla blokzinciri platformu olarak tercih edilmiştir. Tüm süreç, Ethereum blokzinciri üzerinde gerçekleştirildiğinden şeffaf bir oylama sistemi elde etmek amaçlanmıştır.

### 3.3.2. Solidity

Ethereum üzerinde çalışacak olan akıllı sözleşmelerin (smart contracts) geliştirilmesi için kullanılan bir programlama dilidir. Oylama sistemine ait tüm mantıksal iş akışının kodlanması solidity programlama dili ile gerçekleştirilmiştir.

### 3.3.3. Hardhat

Ethereum platformu üzerinde geliştirme ve test yapılmasına imkân sağlayan bir araçtır. Geliştirilen oylama sisteminde, akıllı sözleşmelerin derlenmesi, dağıtılması ve testlerinin gerçekleştirilmesi Hardhat altyapısıyla sağlanmıştır. Tüm süreçlerde önemli bir altyapı simülasyonu olarak kullanılmıştır. Hardhat ile çeşitli eklentiler kullanılmıştır; bu sayede oylama sisteminin maliyeti, performans metrikleri ve güvenlik açıkları tespit edilebilmiştir.

### 3.3.4. React.js

React.js, Facebook tarafından geliştirilen ve kullanıcı arayüzü oluşturmak için kullanılan bir JavaScript kütüphanesidir. Seçmenlerin ve yetkililerin sisteme erişimini sağlayan web uygulaması React.js kullanılarak geliştirilmiştir.

### 3.3.5. JavaScript

JavaScript, arayüz ve sunucu tarafında kullanılabilen bir programlama dilidir. Seçmenlerin ve yetkililerin sisteme erişimini sağlayan arayüzlerin geliştirilmesi sırasında JavaScript kullanılmıştır. Akıllı sözleşmeler ile etkileşim, Ether.js isimli JavaScript kütüphanesiyle gerçekleştirilmektedir.

### 3.3.6. Material UI

Material UI (MUI), React tabanlı projelerde kullanılan bir bileşen kütüphanesidir. Arayüzün geliştirilmesinde hazır bileşenler (butonlar, formlar, menüler vb.) sağlamaktadır. Bu bileşenler kullanılarak; seçmen ve yetkililerin, sistemdeki kritik süreçleri düzenli ve erişilebilir bir arayüzle gerçekleştirmesi sağlanmıştır.

### 3.3.7. Ethers.js

Ethereum tabanlı oylama sistemi ile kullanıcı arayüzün iletişim kurabilmesi için Ethers.js kütüphanelerinin kullanılması gerekmektedir. JavaScript tabanlı geliştirilen uygulamanın, Ethereum blokzinciri ile doğrudan iletişim kurması için kullanılmıştır.

### 3.3.8. Express

Express, Node.js üzerinde çalışan bir web sunucusu framework'dür. Oylama sisteminde, seçmenlerin gaz ücretinin, sistem tarafından karşılanması için Express kullanılarak relay servisi geliştirilmiştir. Relay servisi, seçmenin oy verme aşamasında, Ethereum platformu ile kullanıcı arayüzü arasında bir köprü görevi görmektedir.

### 3.3.9. Java

Seçmenlerin ve yetkililerin, oylama sistemine giriş ve çıkış yapabilmeleri için tarayıcı üzerinde MetaMask uzantısı kullanmaları gerekmektedir. Bu nedenle seçmen ve yetkililerin MetaMask uyumlu cüzdanlarının olması gerekmektedir. Cüzdanların oluşturulabilmesi için Java programlama diliyle bir cüzdan üretici geliştirilmiştir. Bu araç sayesinde kullanıcılar MetaMask üzerinde kullanabilecekleri özel anahtar (private key), açık anahtar (public key), Ethereum adresi (Ethereum address) ve anımsatıcı (mnemonic) üretebilmektedir.

### 3.3.10. Circom

Sıfır bilgi ispatı (Zero-Knowledge Proof) üretmek ve doğrulamak için kullanılan özel bir devre tanımlama dilidir. Oylama sisteminde, seçmenlerin kimlikleri doğrulanabilirken; oylarının içeriğinin gizli kalabilmesi için Circom ile geliştirilen devre kullanılmıştır. Böylece

bir seçmenin oy verdiği doğrulanabilmekte ama oyun içeriğinin görülebilmesi mümkün olmamaktadır. Oylama sisteminde Circom devreleri, şifreli oy verilerini işlememektedir. Circom, bu veriler üzerinde tanımlı kriptografik işlemlerin geçerliliğini ispatlamak amacıyla kullanılmaktadır. Circom, Paillier şifreleme ile oluşturulan gizli oyların doğrulanabilirliğini sağlayan kriptografik bir köprü rolünü üstlenmektedir.

### 3.3.11. CircomlibJS

CircomlibJS, Circom devrelerinde kullanılan kriptografik yapıları bir kütüphane olarak sunan ve JavaScript tarafında da aynısını kullanmayı sağlayan bir kütüphanedir. Circom devresinde; hash fonksiyonu kullanılarak bir değer elde edildiyse, arayüz tarafında da aynı hash değerinin üretilmesi için kullanılması gerekmektedir. Circom ile geliştirilen devrelerin, JavaScript ile geliştirilen bir uygulamada kullanılabilmesi için gereklidir.

### 3.3.12. SnarkJS

SnarkJS, JavaScript tabanlı bir kütüphanedir. Circom devresinin derlenmesini ve sıfır bilgi ispatının üretilmesini sağlamaktadır. Seçmenin oy kullanması sonucu; SnarkJS kullanılarak sıfır bilgi ispatı kanıtları üretilmekte ve bu kanıtlar akıllı sözleşmelere iletilmektedir. Bu kanıtlar doğrultusunda işlemler doğrulanabilmektedir ve blokzincir üzerinde güvenli bir şekilde kayıtlı tutulabilmektedir.

### 3.3.13. Paillier Şifreleme Algoritması

Paillier şifreleme algoritması, sistemde seçmenlerin oylarını gizlemek amacıyla kullanılan asimetrik bir algoritmadır. Bu algoritma sayesinde, toplamsal homomorfik özellik kullanılarak oylar açılmadan, şifreli oylar üzerinden seçim sonuçları hesaplanabilmektedir. Böylece oy gizliliği korunarak şeffaf bir oy sayım süreci gerçekleştirilebilmektedir.

### 3.3.14. Relay Mimarisi (İşlem Ücreti Aracısı)

Ethereum platformu üzerinde herhangi bir işlem gerçekleştirebilmek için o işleme özel belirlenen ve gaz olarak ifade edilen bir ücretin ödenmesi gerekmektedir. Bu gaz ücretinin birimi Ethereum platformunda Ether ve onun alt birimi Gwei'dir.

Seçmenlerden oy kullanma işlemi sırasında herhangi bir işlem ücreti alınmaması için relay mimarisi uygulanmıştır. Seçmenler, oylarını kendileri kullanmakta fakat gaz ücretini bir relay servisi karşılamakta ve blokzincirine iletmektedir. Böylece, seçmenlerin cüzdanında Ether tutulmasına gerek kalmamaktadır. Bu relay servisinin barındıracağı Ether miktarı ve bu

miktarın hangi hesaptan çekileceği, sistemin kurulumu sırasındaki ayar dosyalarından belirlenmektedir.

### **3.3.15. Diamod Pattern (EIP-2535)**

Ethereum platformu üzerinde geliştirilen akıllı sözleşmeler belirli kısıtlamalar barındırmaktadır. Bu kısıtlamalardan birisi de her bir sözleşmenin maksimum boyutunun 24 KB (24,576 bytes) olması zorunluluğudur. Seçim süreçlerini ilgilendiren her işlevin tek bir akıllı sözleşmenin içerisinde bulunması, bu byte sınırının aşılmasına neden olabilmektedir. Seçim süreçlerini yöneten sözleşmelerin geliştirilmesi aşamasında bu sınır aşıldığından sorunlar yaşanmıştır. Bu problemin ortadan kaldırılması ve sistemin geliştirmeye açık kalabilmesi için Diamond Pattern (EIP-2535) kullanılmıştır. Bu tasarım deseninin kullanılmasıyla, seçim sürecinin her işlevi (oylama, aday yönetimi, veri ve sonuç açıklama) ayrı ayrı “facet” modüllerinde tanımlanmış ve merkezi bir proxy akıllı sözleşmesi üzerinden erişilebilir kılınmıştır.

### **3.3.16. Poseidon Hash Fonksiyonu**

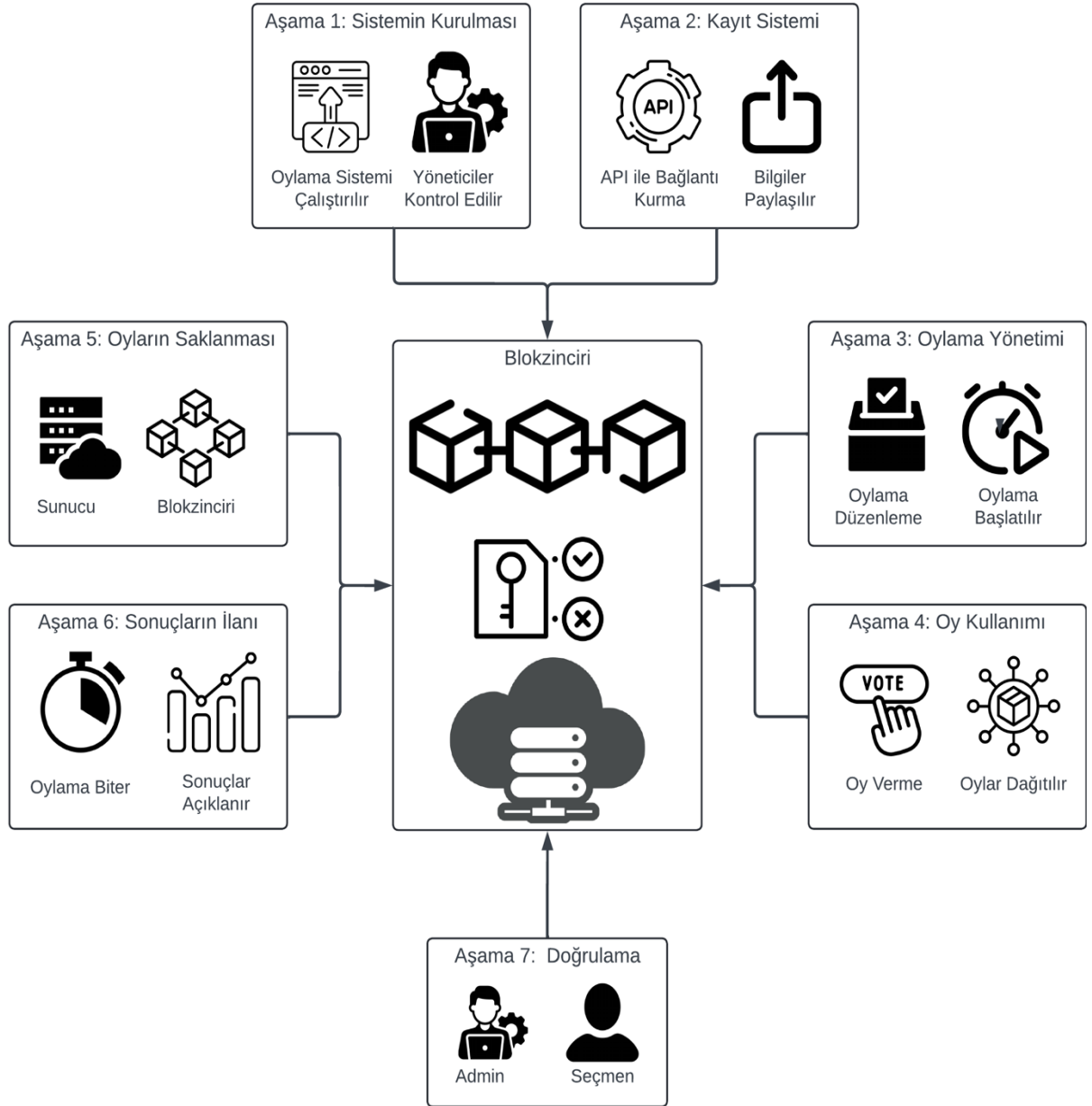
Sıfır bilgi ispatlarının (Zero-Knowledge Proof) kullanabilmesi için geliştirilen devrede Poseidon hash fonksiyonu kullanılmıştır. Poseidon hash fonksiyonu kullanılarak veriler maskelenmiş ve bu maskelenmiş veriler üzerinden doğrulama işlemleri gerçekleştirilmiştir.

### **3.3.17. OpenZeppelin**

OpenZeppelin, güvenli ve verimli akıllı sözleşme geliştirmek için kullanılan açık kaynaklı bir kütüphanedir. Zaman ve maliyet tasarrufu sağlamak, güvenli bir uygulama oluşturmak ve belirlenmiş standartlarda oylama sistemi ortaya çıkartabilmek için tercih edilmiştir.

## **3.4. OYLAMA SİSTEMİNİN YAŞAM DÖNGÜSÜ**

Bu tez kapsamında üniversite çapında merkezi olmayan bir oylama sistemi oluşturabilmek amacıyla teorik ve pratik çalışmalar gerçekleştirilmiştir. Bu pratik çalışmaların sonucunda ortaya çeşitli servislerden oluşan bir uygulama sunulmuştur. Uygulamanın doğru bir şekilde çalışabilmesi için tüm servislerin kurulması, verilerin aktarılması, oylama ve doğrulama mekanizmasının doğru bir şekilde çalıştırılması gerekmektedir. Bu kapsamda bahsedilen işlemlerin tamamı, oylama sisteminin yaşam döngüsünü oluşturmaktadır. Bu yaşam döngüsü Şekil 3.3.’de gösterildiği gibi yedi aşamayla ifade edilmektedir.



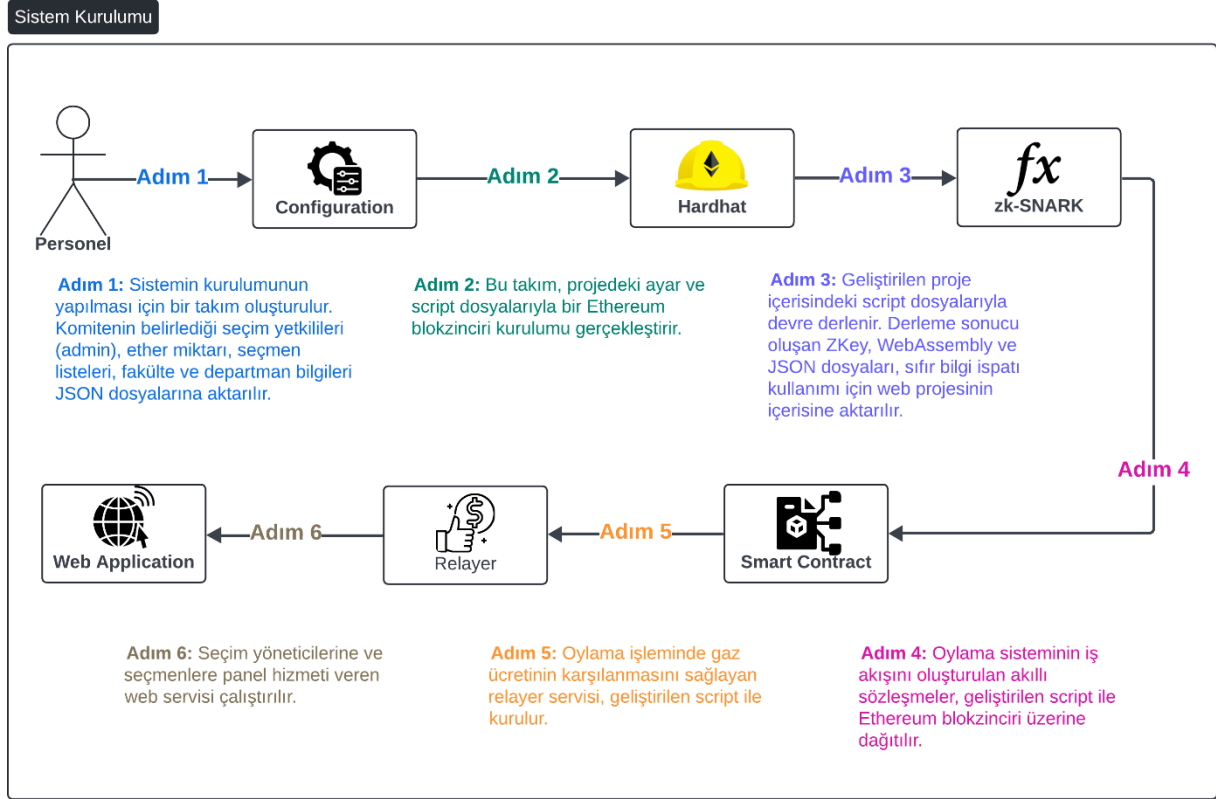
Şekil 3. 3. Oylama sisteminin aşamalarını gösteren model.

### 3.4.1. Aşama 1: Sistemin Kurulması

Oylama sisteminin aktif hale gelmesi için gereken süreçler bütünü ifade etmektedir. Bu süreçte, merkezi olmayan güvenli bir uygulama oluşturulabilmesi için yetkin ve tarafsız bilirkişilerin sistemin kurulmasına dahil olması gerekmektedir.

Teorik çalışmalar sonucunda pratik bir uygulama olarak geliştirilen platform; server, relay, web uygulaması ve aritmetik devreden oluşmaktadır. Bu dört servisin, başarılı bir şekilde kurulabilmesi için doğru sırayla çalıştırılması ve birbirleriyle gerekli adreslerle

bağlanmaları gerekmektedir. Bu süreç, Şekil 3.4.'deki gibi gösterilir ve her bir adım ayrı ayrı değerlendirilir:



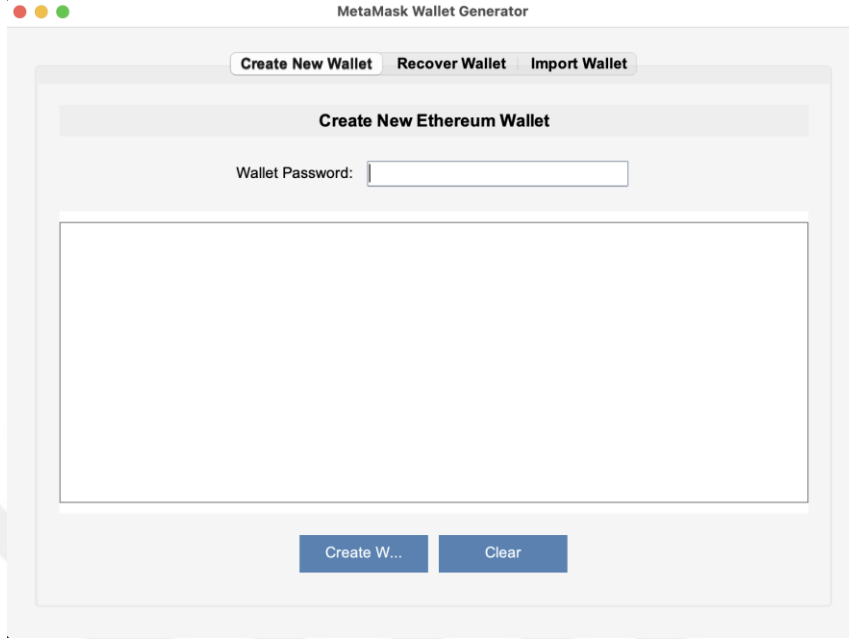
**Şekil 3. 4.** Oylama sisteminin kurulumunu gösteren diyagram.

#### 3.4.1.1. Cüzdan Sağlayıcısının Kurulumu

Oylama sisteminde, seçim yetkililerinin ve seçmenlerin oturumlarını yönetip oy kullanmalarını sağlamak için bir kimlik bilgisine ihtiyaç vardır. Ethereum platformunun kullanılması sebebiyle; her seçim yetkilisinin ve seçmenin kimliğini belirten bir Ethereum adresi oluşturulması amaçlanmıştır. Böylece yetkililer ve seçmenler, oturumlarını yönetmek için bu adresleri kullanacaktır.

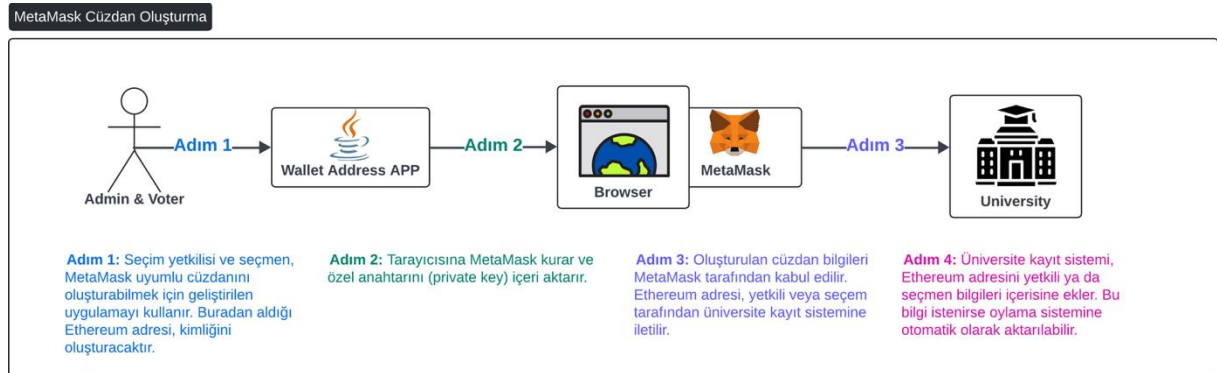
Ethereum adresleriyle, oturum yönetimlerini gerçekleştirebilmek için MetaMask cüzdanı kullanılması uygun görülmüştür. MetaMask, harici bir cüzdanın özel anahtarının (private key) içeriğe aktarılması yoluyla ilgili cüzdanın yönetimini mümkün kılmaktadır. Tarayıcı üzerinde çalışması sebebiyle; web tabanlı geliştirilen uygulamalar içerisinde oturum işlemleri için de kullanılabilir. MetaMask, oturum işlemlerinde kullanılacağı için seçim yetkililerinin ve seçmenlerin MetaMask uyumlu birer Ethereum

cüzdanına ihtiyaçları vardır. Cüzdanların, dışarıda üretilmesi istenmediğinden; Şekil 3.5.'de gösterilen güvenli bir cüzdan üretici geliştirilmiştir.



Şekil 3. 5. Java ile geliştirilen MetaMask uyumlu Ethereum cüzdan üretici program.

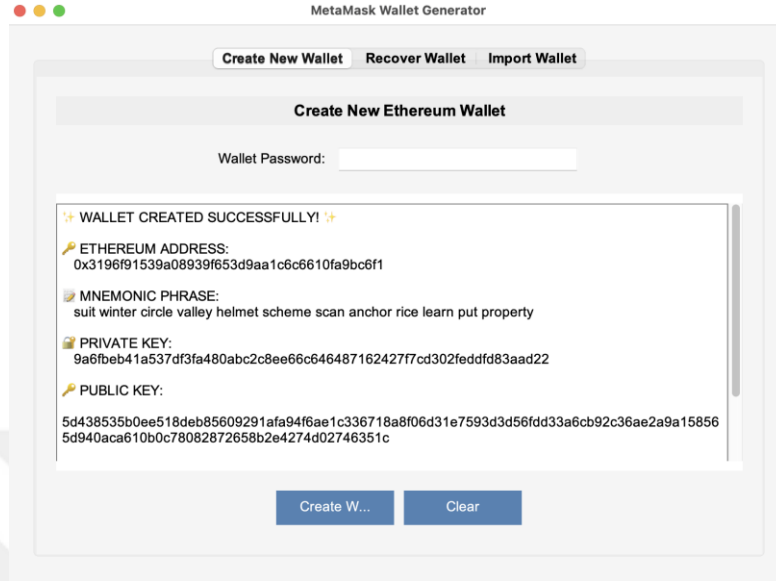
Seçmenlerin ve yetkililerin, oylama sistemine kaydolabilmesi için kendi kişisel bilgisayarlarında bu uygulamayı çalıştırıp, istedikleri parolayla üretilen cüzdan için genel anahtar (public key), özel anahtar (private key), mnemonic ve Ethereum adresini ifade eden bilgileri almaları gerekmektedir. Bu süreç Şekil 3.6.'da adım adım gösterilmiştir.



Şekil 3. 6. Seçmenlerin ve seçim otoritesi için cüzdan üretilmesi ve sisteme aktarılması.

Üretilen Ethereum adresleri, yetkililerin ve seçmenlerin kimliklerini oylama sisteminde temsil etmektedir ve gerçek hayattaki kimlikleriyle herhangi bir bağlantısı bulunmamaktadır. Ethereum adresleriyle seçmenler arasında bir bağlantı kurulsa bile sistemde güvenlik problemi

olmayacaktır. Çünkü oylama gizliliği başka çözümlerle sağlanmaktadır. Şekil 3.7.'de üretilen örnek cüzdan bilgisi görülebilir.



Şekil 3. 7. MetaMask uyumlu Ethereum cüzdan üretilmesi.

#### 3.4.1.2. Konfigürasyon Dosyalarının Hazırlanması

Sistemin kurulumu sırasında ayar dosyalarının hazırlanması gerekmektedir. Akıllı sözleşmeleri oluşturan server bölümü, sistemin iş akışını oluşturduğu için bu ayar dosyalarına ihtiyaç duymaktadır. Akıllı sözleşmelerin dağıtılması sırasında seçim yetkililerinin belirlenmesi, fakülte ve departmanların hazırlanmasıyla birlikte seçmenlerin kaydedilmesi bu ayar dosyalarıyla sağlanabilmektedir. Bu ayar dosyaları, JSON formatında hazırlanmaktadır ve akıllı sözleşmelerin kurulumunda gerekli bilgilerin otomatik olarak eklenmesini sağlamaktadır. Bu yöntemle; seçim yetkililerinin, işlemler için tek tek zaman harcamasına gerek kalmamış ve insan faktöründen kaynaklanan hataların engellenmesi amaçlanmıştır.

Şekil 3.8.'de seçim yetkililerinin belirlenmesini sağlayan örnek admins.json dosyası görülebilir. Bu sadece başlangıç aşamasındaki yetkililerdir. Geliştirilen panelden daha fazla seçim yetkilisi eklenebilmektedir.

```
{
  "initialAdmins": [
    "0x7596b4577370988d2dbce9c0cc3ac932871fe500",
    "0x301c19302b4e797aff38e02d3dafd466cdeaf996"
  ]
}
```

Şekil 3. 8. Başlangıçta seçim otoritelerin belirlenmesini sağlayan JSON dosyası.

Fakültelerin oluşturulmasını sağlayan faculties.json dosyasının bir örneği Şekil 3.9.'da görülebilir.

```
[
  { "id": 1, "name": "Engineering" },
  { "id": 2, "name": "Science" }
]
```

**Şekil 3. 9.** Başlangıçta fakültelerin belirlenmesini sağlayan JSON dosyası.

Fakülterle bağlı bölümlerin oluşturulmasını sağlayan departments.json dosyasının bir örneği Şekil 3.10.'da görülebilir. Bu ayar dosyasıyla veriler kaydedilirken, bölümlerin bağlı olduğu fakültelerin kayıtlı olması gerekmektedir.

```
[
  { "id": 50, "name": "Computer Science", "facultyId": 1 },
  {
    "id": 51,
    "name": "Electrical Engineering",
    "facultyId": 1
  },
  { "id": 52, "name": "Mathematics", "facultyId": 2 }
]
```

**Şekil 3. 10.** Başlangıçta bölümlerin belirlenmesini sağlayan JSON dosyası.

Başlangıç aşamasında seçmenlerin oluşturulmasını sağlayan users.json dosyasının bir örneği Şekil 3.11.'de görülebilmektedir. Bu ayar dosyasıyla veriler kaydedilirken, seçmenlerin bağlı bulundukları fakülte ve departmanın, sistem içerisinde kayıtlı olup olmadığı kontrol edilmektedir.

```
{
  "users": [
    {
      "address": "0x260d0d6426667d037bf36e16399b56be5776f56f",
      "facultyId": 1,
      "departmentId": 50,
      "role": "Academic",
      "isActive": true,
      "isRegistered": true
    },
    {
      "address": "0x2150148fcd4bc32c692e7ea88c7fd0cae926d9c9",
      "facultyId": 1,
      "departmentId": 51,
      "role": "Student",
      "isActive": true,
      "isRegistered": true
    }
  ]
}
```

**Şekil 3. 11.** Başlangıçta seçmenlerin belirlenmesini sağlayan JSON dosyası.

### 3.4.1.3. Blokzinciri Ortamının Kurulumu

Geliştirilen oylama sistemi blokzinciri tabanlı olduğu için bir blokzincir platformunun kurulması gerekmektedir. Bu tez kapsamında Ethereum kullanıldığı için seçilecek araç bu kritere bağlı kalınarak belirlenmiştir.

```
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
=====
Account #0: 0x07E86c32C1E8e5820a7D75E8A2612caA3F637FF (9000 ETH)

Account #1: 0xBaf89Ab1D73AA262bB2b76331Bd142aE9b6bfe7d (5000 ETH)

Account #2: 0x25b2dF679C5ee6E0542eefA56676E3c20DaE2a2A (5000 ETH)

Account #3: 0x39C116A036A226ff4d2ee74d3224EDB637A9fB58 (5000 ETH)

eth_blockNumber (4)
```

Şekil 3. 12. Hardhat ile blokzinciri ağının oluşturulması.

Bu bağlamda, akıllı sözleşmelerin geliştirilmesi ve test edilebilmesi için Hardhat ortamının kurulu olması gerekmektedir. Sistem içerisinde bulunan Hardhat ayar dosyalarıyla Şekil 3.12.'deki gibi Ethereum platformu oluşturulabilmektedir. Bu işlem sonucunda akıllı sözleşmeler, test edilir, derlenir ve dağıtılır.

### 3.4.1.4. Aritmetik Devrenin Kurulumu

Oy gizliliğini sağlamak için Circom dili ile geliştirilen devrenin derlenmesi gerekmektedir. Bu derlenme işlemi için JavaScript araçlarından SnarkJS kullanılmaktadır. Bu kütüphaneyle devrenin derlenmesi sonucu gerekli anahtar (zkey), web assembly (wasm), json ve akıllı sözleşme dosyası (verifier.sol) elde edilmektedir. Şekil 3.13.'de örnek bir derleme sonucu görülebilir. Elde edilen akıllı sözleşme dosyası (verifier.sol), server olarak ifade edilen servisin içerisinde sıfır bilgi ispatı doğrulaması için kullanılmaktadır. Anahtar (zkey), web assembly (wasm) ve JSON dosyaları ise web uygulamasında kullanılacaktır. Kullanıcının oy kullanması aşamasında sıfır bilgi ispatı kanıtı oluşturmak için kullanılacak olan gerekli dosyalardır.

```
3. Groth16 setup yapılıyor...
[INFO] snarkJS: Reading r1cs
[INFO] snarkJS: Reading tauG1
[INFO] snarkJS: Reading tauG2
[INFO] snarkJS: Reading alphatauG1
[INFO] snarkJS: Reading betatauG1
[INFO] snarkJS: Circuit hash:
98254f2d 48eaf244 32dd1b05 11be5a1c
d9298ca8 ee6dae9e db42f6eb 726e1abf
de5e982f 7e8d4c25 feff81aa 4376ab73
de31bd3f 036a4d67 bb0aed74 df0769aa
6. Doğrulama anahtarı dışa aktarılıyor...
[INFO] snarkJS: EXPORT VERIFICATION KEY STARTED
[INFO] snarkJS: > Detected protocol: groth16
[INFO] snarkJS: EXPORT VERIFICATION KEY FINISHED
7. Solidity doğrulayıcı dışa aktarılıyor...
[INFO] snarkJS: EXPORT VERIFICATION KEY STARTED
[INFO] snarkJS: > Detected protocol: groth16
[INFO] snarkJS: EXPORT VERIFICATION KEY FINISHED
Derleme tamamlandı!
Çıktı dosyaları: circom/build dizininde
```

Şekil 3. 13. Circom ile geliştirilen devrenin derlenmesi.

### 3.4.1.5. Akıllı Sözleşmelerin Dağıtımı

Server servisi, akıllı sözleşmeleri ve bu sözleşmelerin Ethereum ağına dağıtılması için gerekli konfigürasyon dosyalarını barındırmaktadır. Bu konfigürasyon dosyalarıyla kurulan Ethereum ağına, akıllı sözleşmeler dağıtılmakta ve bu akıllı sözleşmelerin adresleri diğer servislerde (web, relayer) kullanılmak üzere bir JSON dosyasına aktarılmaktadır. Solidity diliyle geliştirilen bu akıllı sözleşmeler, oylama sisteminin iş katmanını oluşturmaktadır. Tercih edilen geliştirme ortamı Hardhat olduğu için gerekli ayar dosyaları buna göre belirlenmiştir. Özellikle Hardhat test ağında çalışıldığı için yetkili hesaplarına ve relayer servisine belirli bir miktar ETH verilmektedir. Bunun için kurulum aşamasında sözleşmeler dağıtılırken ETH sahibi olan geçici hesaplar oluşturulmaktadır. Bu oluşturulan geçici hesapların ETH bakiyeleri, yetkili hesaplarına ve relayer servisine aktarılmaktadır. Şekil 3.14.'de akıllı sözleşmelerin ağda dağıtılması sonrası oluşan sözleşme adresleri gösterilmektedir.

```

Deployed Contract Addresses:
"AdminManager": "0xBE2c16376c944eCB4Df0B36a1d8d557948Fa5711"
"VoterManager": "0x45C0a33d255Ef893eC50BEeFef279826752d059"
"FacultyManager": "0x3733cF3d0d198B62f2411e08C71ac1a717E9479b"
"ElectionManager": "0x78B43051FD5A002e9840a041E3aAE1B042C36369"
"Voting": "0x626AFAa650fE7d9A28885591B0Db573DCA35F1d"
"Groth16Verifier": "0xBA8A4249D7eA94A1C01a33A35b572FE2c0DD42c7"
"Diamond": "0xDF203452242a7210017091ba65a677e8cD8ae14A"

Diamond Facets:
"DiamondCutFacet": "0x93b9B6400a53CFFfADC550BEeFeE13de52C3e03e"
"DiamondLoupeFacet": "0x6c4EFCD8b031B42a20B794160144edff3f640b58"
"ElectionBaseFacet": "0xa46Fb3b8cC3A5B7A8199Dd32177E2CC3803b3422"
"ElectionVoterFacet": "0x40440931085f8d1670733Ac16da67402Ab0C7382"
"ElectionCandidateFacet": "0xA61AA364454B7547A4e16cbc1C90F6d9f892BE86"
"ElectionStateFacet": "0x4bf7123592Fa7da6E8e69fbf5a4AAe72329c89cB"
"ElectionStatsFacet": "0xa9385E64C76f34C936A262d87BBFE3f3D5737B10"
Deployment successful!
{
  "network": "localhost",
  "adminManager": "0xBE2c16376c944eCB4Df0B36a1d8d557948Fa5711",
  "facultyManager": "0x3733cF3d0d198B62f2411e08C71ac1a717E9479b",
  "voterManager": "0x45C0a33d255Ef893eC50BEeFef279826752d059",
  "diamond": "0xDF203452242a7210017091ba65a677e8cD8ae14A",
  "votingContract": "0x626AFAa650fE7d9A28885591B0Db573DCA35F1d",
  "verifier": "0xBA8A4249D7eA94A1C01a33A35b572FE2c0DD42c7",
  "paillierCrypto": "0x05EB23603c3Ec9a58621DCE3a40566AF56052188",
  "encryptedVoting": "0x15F0970B6bA9e7589AF79c22A9289C47E16f4B55",
  "facets": {
    "diamondCut": "0x93b9B6400a53CFFfADC550BEeFeE13de52C3e03e",
    "diamondLoupe": "0x6c4EFCD8b031B42a20B794160144edff3f640b58",
    "electionBase": "0xa46Fb3b8cC3A5B7A8199Dd32177E2CC3803b3422",
    "electionVoter": "0x40440931085f8d1670733Ac16da67402Ab0C7382",
    "electionCandidate": "0xA61AA364454B7547A4e16cbc1C90F6d9f892BE86",
    "electionState": "0x4bf7123592Fa7da6E8e69fbf5a4AAe72329c89cB",
    "electionStats": "0xa9385E64C76f34C936A262d87BBFE3f3D5737B10"
  }
}
Deployment process completed!
All processes completed successfully!

```

Şekil 3. 14. Akıllı sözleşmelerin derlenmesi ve dağıtılması.

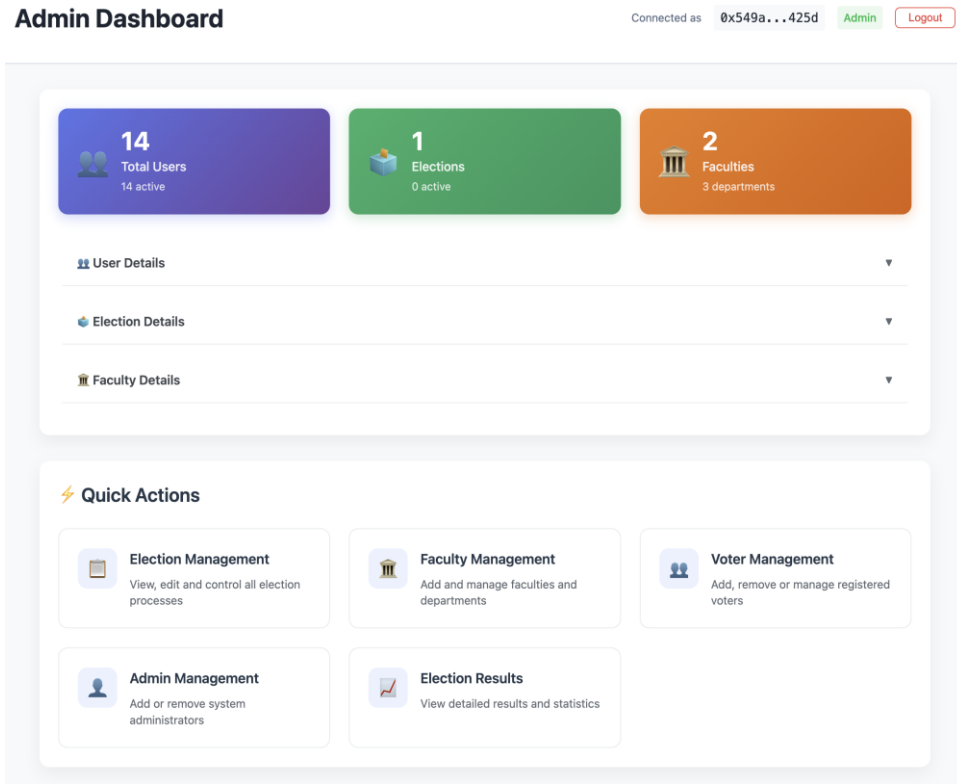
### 3.4.1.6. Relayer Servisinin Kurulumu ve Yapılandırması

Oylama sistemi, Hardhat geliştirme ortamında Ethereum platformuyla çalıştığı için ağda yapılacak her işlemin bir maliyeti bulunmaktadır. Yetkililer veya seçmenler, ağ üzerinde bir işlem yapması gerektiğinde bu gaz maliyetini kendi cüzdanlarından ödemesi gerekmektedir. Seçmenlerden, oy kullanmak amacıyla ücret istenmesi doğru olmadığı için maliyetin oylama sistemi tarafından karşılanması uygun görülmüştür. Relayer olarak tanımlanan servis,

kullanıcıların kendi cüzdan bakiyeleri yerine oylama sisteminin bakiyelerini kullanmalarına olanak sağlamak için kurulmaktadır. Kullanıcının ağda gerçekleştireceği ücretli işlemlerde; istek önce relay olarak adlandırılan bu servise gelmekte ve bu servis tarafından gerekli maliyetin karşılanarak server servisine işlemin iletilmesi sağlanmaktadır. Bu servisin doğru çalışabilmesi için Hardhat ile çalıştırılan Ethereum platformu içerisinde, ETH bakiyeli geçici hesaplara ihtiyaç duyulmaktadır. Aynı zamanda, akıllı sözleşmelerin adreslerini tutan JSON dosyasıyla oylama işleminden sorumlu akıllı sözleşmelerin ABI bilgisini barındıran JSON dosyasının, servis içerisinde bulunması gerekmektedir.

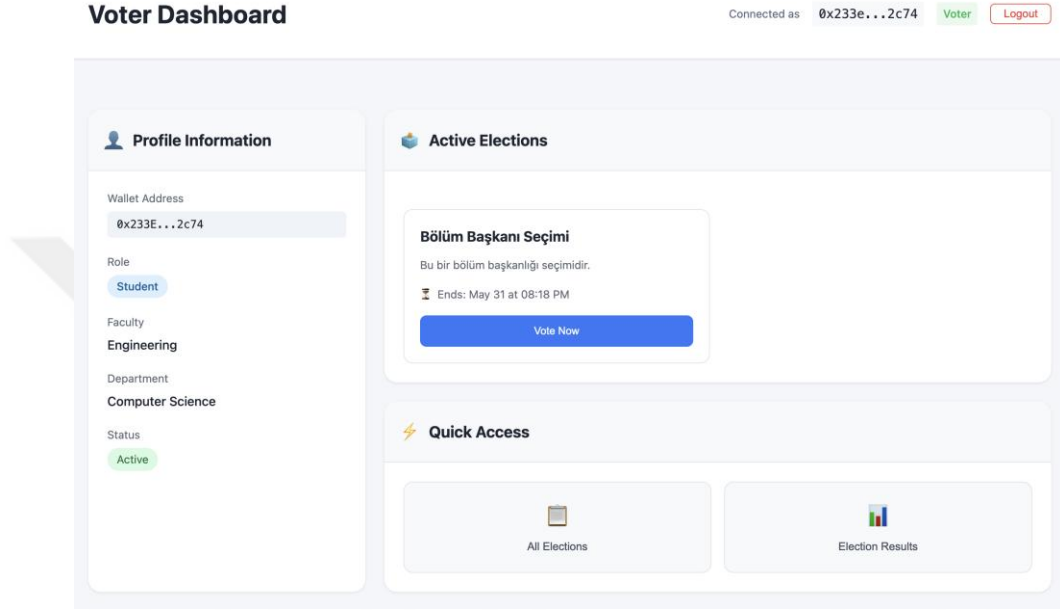
#### 3.4.1.7. Kullanıcı Arayüzünün Kurulumu

Web uygulaması olarak adlandırılan bu servis, React.js kullanılarak geliştirilmiştir. Oylama sisteminin kullanıcıyla etkileşime girmesi için geliştirilen bir web uygulamasıdır. Bu web uygulamasıyla, seçim yetkilileri ve seçmenlere modern bir arayüz sunmak amaçlanmıştır. Bu arayüzün kurulmasıyla seçim yetkilileri ve seçmenler, oylama sistemini bir web arayüzüyle kullanabilmektedir. Bu uygulama içerisinde seçim yetkilileri ve seçmenler için iki ayrı bölüm bulunmaktadır. Seçmenlerle ilgili olan bölüm seçmen paneli ve yetkililerle ilgili olan bölüm yetkili paneli olarak adlandırılmaktadır.



Şekil 3. 15. Seçim yetkilisi paneli ana sayfa arayüzü.

Seim yetkilileri, kendilerine ayrılan bölümden yönetim işlemlerini gerçekleştirebilmektedir. Seim yetkilisi arayüzü Şekil 3.15.'de gösterilmektedir. Seçmenler; kendilerine ayrılan bölümden seçimleri görüntülemek, katılmak, sonuçları görüntülemek ve doğrulamak için gerekli işlemleri gerçekleştirebilmektedir ve arayüz Şekil 3.16.'da gösterilmektedir.



Şekil 3. 16. Seçmen paneli ana sayfa arayüzü.

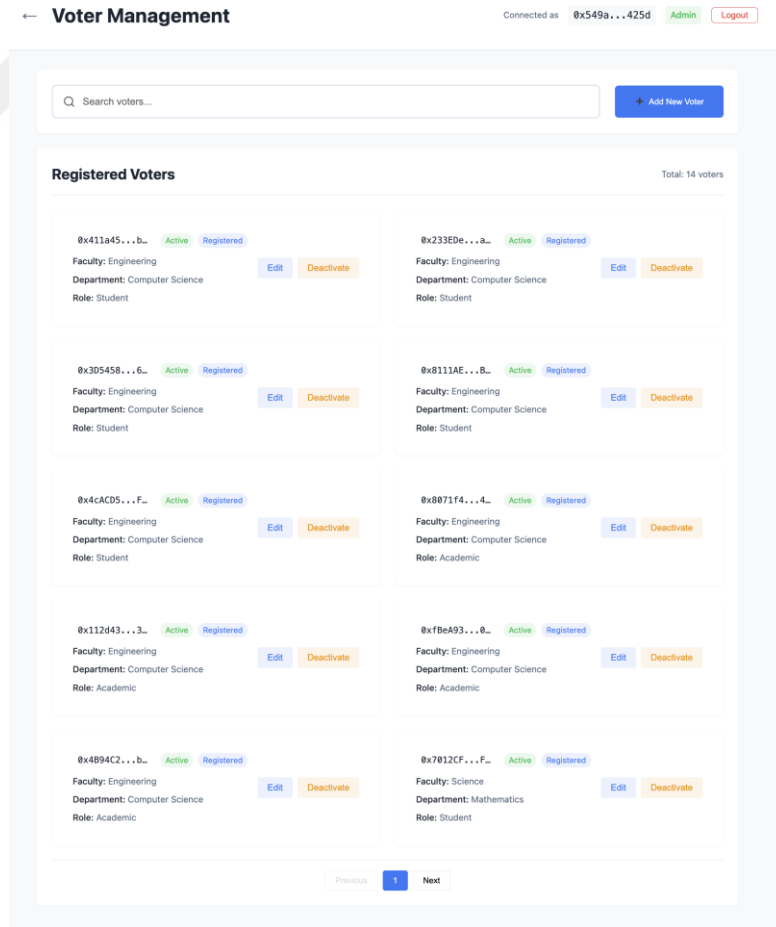
Web uygulamasının doğru bir şekilde çalışabilmesi için öncelikle devrenin derlenmiş dosyalarına, Hardhat ortamının bilgilerine ve sözleşme adresleri barındıran json dosyasına ihtiyaç duyulmaktadır. Önceki aşamalardan elde edilen bu bilgiler doğrultusunda React uygulaması çalıştırılabilmekte ve ilgili arayüzlere ulaşılabilir.

### 3.4.2. Aşama 2: Kayıt Sistemi

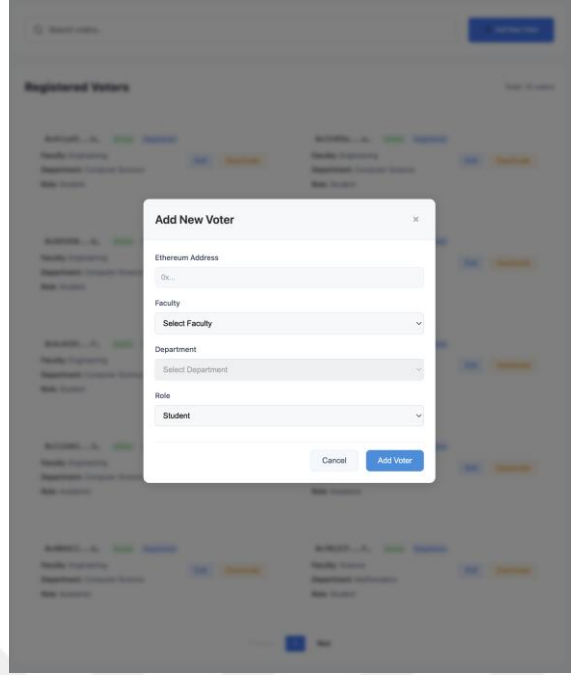
Kurulum aşamasında hazırlanmış olan JSON dosyalarıyla yetkili, fakülte, departman ve seçmen bilgilerinin sisteme eklenmesi sağlanmaktadır. Bu kapsamda, üniversitenin akademik kayıt sisteminden bilgiler JSON dosyasına aktararak oy sistemine dahil edilebilmektedir. Burada kritik ve önemli bir nokta bulunmaktadır. Kullanıcıların seçmen olarak oylama sisteminde yer alabilmesi için MetaMask uyumlu bir Ethereum cüzdanına sahip olması gerekmektedir. Bu cüzdan, geliştirilen wallet isimli bir masaüstü uygulaması vasıtasıyla oluşturulabilmektedir.

Her seçmen, bilgisayarında cüzdan oluşturucu programı çalıştırarak gerekli bilgilerin oluşturulmasını (public key, private key, mnemonic ve ethereum address) sağlayabilmektedir. Oluşturulan bu bilgiler içerisindeki Ethereum adresi, o seçmeni ifade edecektir ama kişisel bilgisini göstermeyecektir. Burada her seçmen, kendi bilgisayarında bu bilgileri oluşturduktan sonra üniversitenin akademik kayıt sistemine veya ilgili personeline, Ethereum adresini verecektir. Akademik kayıt sisteminden bu bilgiler JSON dosyalarına aktarılacaktır. Böylece oylama sistemi kurulurken; bu bilgilerin, toplu olarak eklenebilmesi sağlanacaktır.

Yönetici hesapları, seçmenlerin adreslerini Şekil 3.17.'de gösterilen arayüz vasıtasıyla Şekil 3.18.'deki bilgileri girerek yönetici panelinden ekleyebilir. Bu seçim yetkililerinin, seçmen yönetimi bölümünü kullanmasıyla gerçekleştirilmektedir. Burada sisteme girişi yapan yetkili; Ethereum adresinin hangi seçmene ait olduğunu, üniversitenin akademik kayıt sistemi üzerinden görebilir. Fakat seçmenin Ethereum adresi bilinse bile; oyunun içeriğinin görülebilmesi, kullanılan kriptografik teknikler ile engellenmiştir.

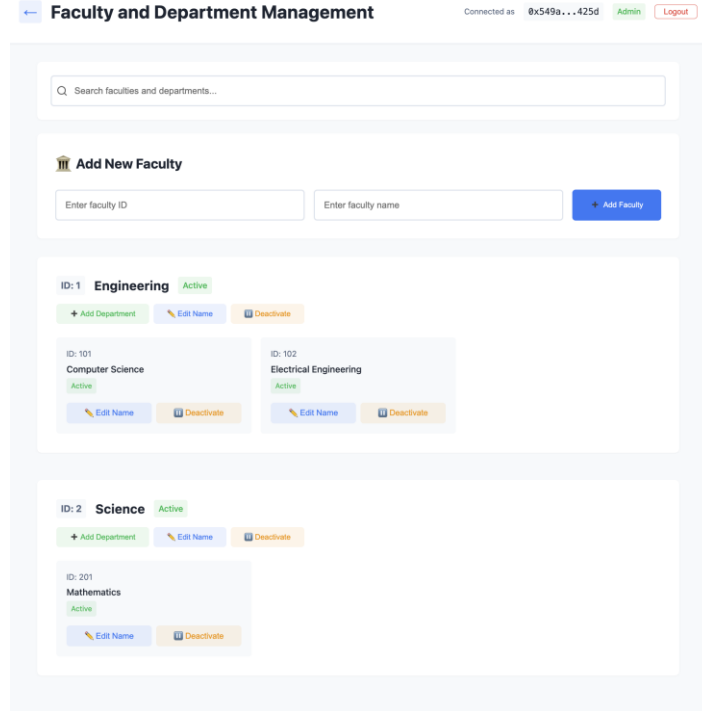


Şekil 3. 17. Seçim yetkilisi panelinde seçmen yönetimi arayüzü.



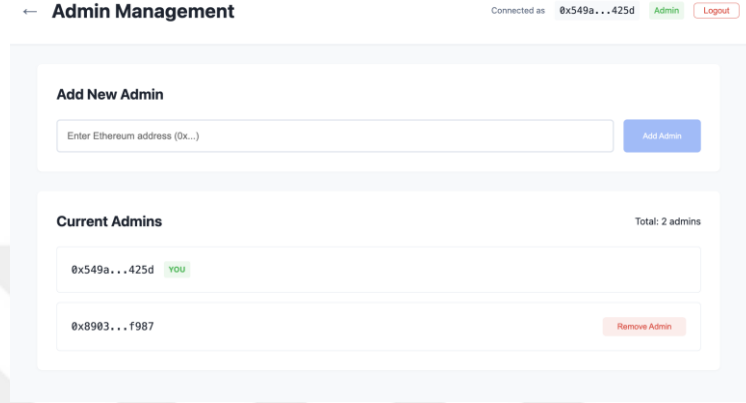
**Şekil 3. 18.** Seçim yetkilisi panelinde seçmen ekleme işlemi.

Seçim yetkilisi paneli üzerinde, fakülte ve bölüm işlemleri için Şekil 3.19.'da gösterilen arayüz bulunmaktadır. Bu bölüm kullanılarak; oylama sistemini kullanan üniversitenin fakülte ve bölüm bilgileriyle ilgili değişiklikler yapılabilmektedir.



**Şekil 3. 19.** Seçim yetkilisi paneli fakülte ve bölüm yönetimi arayüzü.

Kayıtlı olan yetkililer, yönetim işlemlerini gerçekleştirmek için yeni yetkililer ekleyebilmekte ve var olanları silebilmektedirler. Fakat sistemde en az iki yetkili olması gerektiğinden, silme işlemi sırasında yetkili sayısı kontrolü yapılmakta ve bunun sonucunda silme izni verilmektedir. Şekil 3.20.'de seçim yetkilileri yönetimi için kullanılan arayüz gösterilmektedir.



**Şekil 3. 20.** Seçim yetkilisi paneli yetkili yönetimi arayüzü.

### 3.4.3. Aşama 3: Seçim Yönetimi

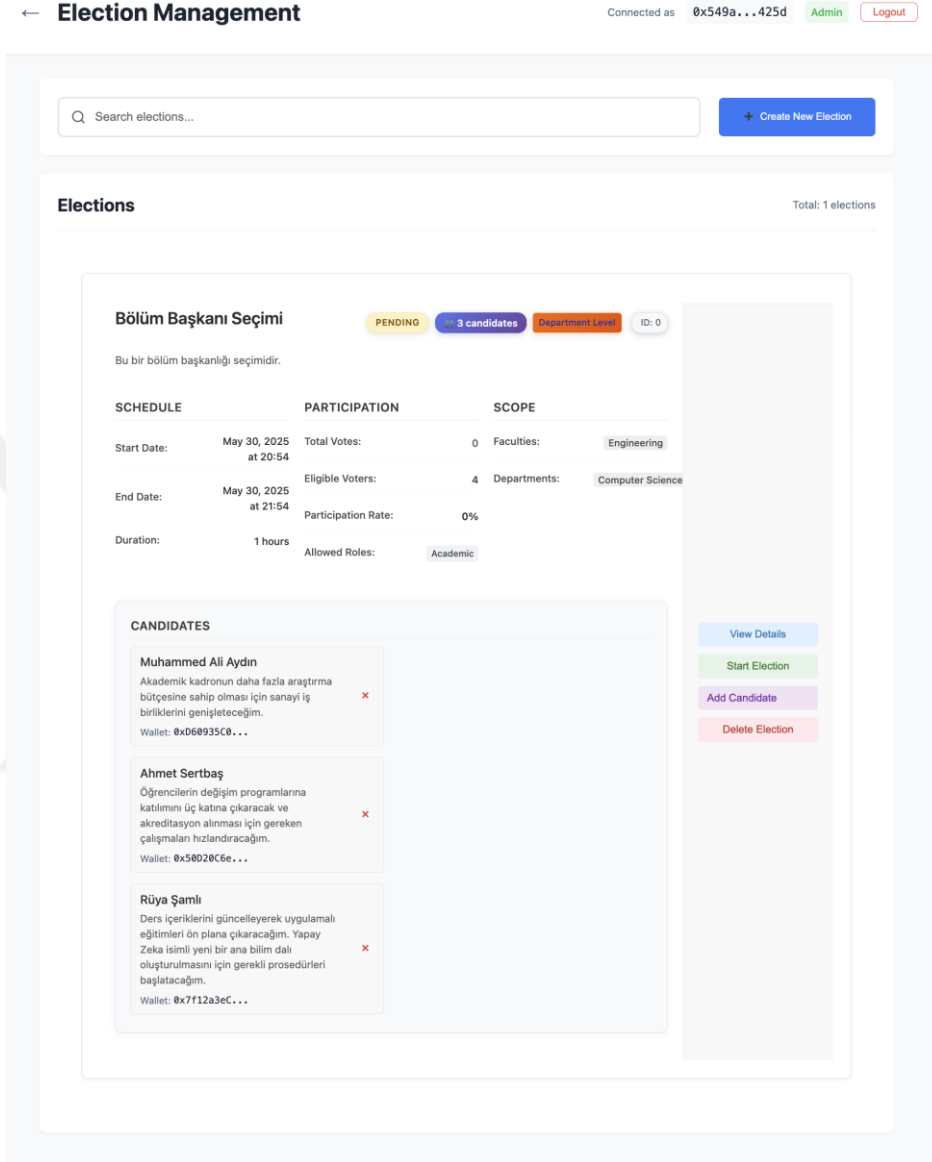
Seçim yönetimi, oylamanın gerçekleştirilmesi için seçim süreçlerinin yönetilmesini sağlayan aşamaları kapsamaktadır. Oylama sistemindeki tüm yetkililer, seçim yönetimine dahil olabilmektedir. Seçim yönetimi için kullanılan arayüz Şekil 3.21.'de gösterilmektedir. Seçim yönetimi sayfasında bir seçimin içeriği görüntülemek istenildiğinde ise Şekil 3.23.'de gösterilen arayüz kullanılmaktadır.

Seçim oluşturabilmek için gerekli bilgilerin girilmesi ve bu bilgiler doğrultusunda çeşitli iş akışının sağlanması gerekmektedir. Bu nedenle, seçim oluşturabilmek amacıyla Şekil 3.22.'de gösterilen arayüz üzerinden sekiz farklı alanın doldurulması sağlanır.

Bir yetkilinin, seçim oluşturabilmesi için öncelikle seçim başlığını ve açıklamasını girmesi gerekmektedir. Bu bilgiler, seçmenlerin hangi seçim için oy kullanacaklarını bilmesi için yardımcı olacaktır.

Seçimin gerçekleştirileceği zaman aralıklarının belirtilmesi gerekmektedir. Başlangıç ve bitiş tarihleri gün, ay, yıl, saat ve dakika cinsinden ifade edilir. Burada önemli bir detay bulunmaktadır. Seçimin başlangıç tarihi bulunulan zamandan beş dakika önceye kadar

belirlenebilirken, bitiş tarihi başlangıç tarihinden en geç 2 gün sonra ayarlanabilmektedir. Seçimin bitiş tarihi, seçimin başlangıç tarihiyle aynı veya daha erken seçilememektedir.



**Şekil 3. 21.** Seçim yetkilisi paneli seçim yönetimi arayüzü.

Geliştirilen oylama sistemi üniversite ölçeğinde olduğu için seçime katılacak kişiler öğrenciler, akademisyenler ve üniversite personelleridir. Yetkilinin, ilgili seçimde kimlerin oy kullanabileceğine karar vermesi gerekmektedir.

Yetkili, seçimin gerçekleştirileceği fakülteleri seçmesi gerekmektedir. Herhangi bir fakülte seçilmemesi durumunda, seçim üniversitenin tüm fakültelerini kapsayacak şekilde gerçekleştirilecektir. Fakülte seçimleri sonrası ilgili fakülteye bağlı bölümlerin seçilmesi

gereklidir. Fakülte seçimi sonrası herhangi bir bölüm seçilmezse fakülteye bağlı tüm bölümler bu seçime dahil edilmektedir.

← Create New Election

Connected as @x549a...425d Admin Logout

### Create New Election

**Basic Information**

Title  
Bölüm Başkanı Seçimi

Description  
Bu bir bölüm başkanlığı seçimidir.

Start Date & Time  
30.05.2025, 20:54

End Date & Time  
30.05.2025, 21:54

**Eligible Voters**

Allowed Roles  
☐ Student ☒ Academic ☐ Staff

Faculties  
Leave empty for university-wide elections  
☒ Engineering ☐ Science

Departments  
Leave empty for faculty-wide elections  
Engineering  
☒ Computer Science ☐ Electrical Engineering

**Candidates**

Candidate #1  
Name: Muhammed Ali Aydın  
Description: Akademik kadronun daha fazla araştırma bütçesine sahip olması için sanayi ile birliklerini genişleteceğin.  
Wallet Address: 0xd80935cd618ee0771819bbbf4ade8d6d46

Candidate #2  
Name: Ahmet Serbag  
Description: Öğrencilerin değişim programlarına katılımını üç katına çıkararak ve akreditasyon alınması için  
Wallet Address: 0x5020c6e99ccodee69e90234504e9a267e4

+ Add Candidate

Cancel Create Election

Şekil 3. 22. Seçim yetkilisi paneli seçim ekleme arayüzü.

Son bölüm olarak aday yönetimi bulunmaktadır. Yetkili, seçimde yer alacak aday bilgilerini bu bölüm aracılığıyla ekleyecektir. Burada aday eklenirken adayı tanıtan isim, Ethereum adresi ve adayın açıklaması eklenmelidir. Ayrıca dikkat edilmesi gereken bir ayrıntı bulunmaktadır. Seçim kuralları gereği adayın ismi yerine Ethereum adresi kullanılabilir.

Böylece kişilerden bağımsız sadece olaylara yönelik seçim yapılması amaçlanmıştır. Bu durum adaya ait ön yargı yerine adaya ait vaatlere odaklanılan seçimler veya mülakatların gerçekleştirilebilmesine imkân vermektedir.

Seçimin başlangıç ve bitiş tarihleri belli olsa da yetkili tarafından seçim başlatılsın butonuna basılmadan seçim aktif olmamaktadır. Seçim başlatılıp aktif olmadığı için bu aşamada seçimin silinebilmesi ve aday eklemeye çıkarma işlemlerinin yapılabilmesi mümkündür. Seçimi başlatma butonuna basıldığında seçim ile ilgili tüm veriler kontrol edilmektedir. Aynı zamanda seçimin düzenlendiği fakülte veya departmanlara seçmen eklendiye veya çıkarıldıysa gerekli güncellemeler ilgili seçim için gerçekleştirilmektedir ve yeni seçmen görüntüsü hazırlanmaktadır. Bu aşamada belirlenen seçmenler ilgili seçim için oy kullanabilme hakkına sahip olmakta ve daha sonra sistemden pasif duruma getirilmiş olsalar bile aktif olan seçimden çıkarılamamaktadır. Seçimlerle ilgili detayların görüntülenebilmesi için Şekil 3.23.'de gösterilen arayüz kullanılmaktadır.

**Election Details** Connected as 0x549a...425d Admin Logout

**Bölüm Başkanı Seçimi** PENDING 0 votes 4 eligible 0% participation

**Election Information**

Description: Bu bir bölüm başkanlığı seçimidir.

Start Date: May 30, 2025 at 20:54

End Date: May 30, 2025 at 21:54

Duration: 1 hours

**Scope**

Allowed Roles: Academic

Faculties: Engineering

Departments: Computer Science (Engineering)

**Candidates** + Add Candidate

| Candidate                 | Wallet                                 | Address |
|---------------------------|----------------------------------------|---------|
| <b>Muhammed Ali Aydın</b> | 0xd60935C0a618ee771819bb8BfA4ADE8fd6F  | 4689    |
| <b>Ahmet Sertbaş</b>      | 0x58D28C6e59CccDee69E90234504e9A267ef  | 4168    |
| <b>Rüya Şamlı</b>         | 0x7f12a3eC47DBDcB2d0f5E1E67EfB3462976D | 63C5    |

**Vote Results**

Vote Results

Vote results will be available when the election is closed.

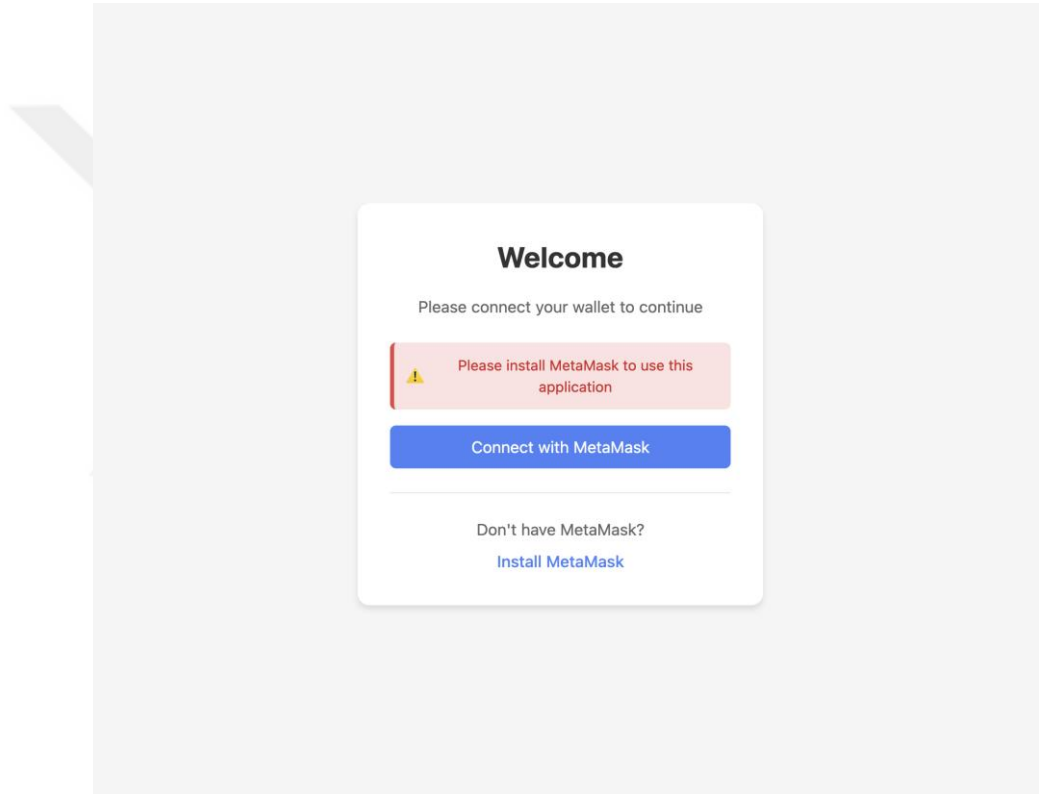
Current Status: Pending

Back to Elections Start Election Delete Election

Şekil 3. 23. Seçim yetkili paneli seçim detayları arayüzü.

#### 3.4.4. Aşama 4: Oy Kullanımı

Seçmenlerin, oylama sistemini kullanabilecekleri bir panel bulunmaktadır. Bir seçmenin bu panele ulaşabilmesi için öncelikle seçmen kaydının, oylama sistemi içerisinde bulunması gerekmektedir. Bu bilgiler kapsamında seçmen, MetaMask kurulumu gerçekleştirmeli ve cüzdan bilgilerini MetaMask içerisine aktarmalıdır. Bu işlemler tamamlandığında, Şekil 3.24.'de gösterilen MetaMask bağlantı butonu ile giriş yapabilmektedir. Bu bağlantı, seçmenin cüzdanını doğrulama sürecinden geçirmekte ve oylama sisteminde bu cüzdana ait adres bulunması durumunda seçmeni ilgili panele yönlendirmektedir.

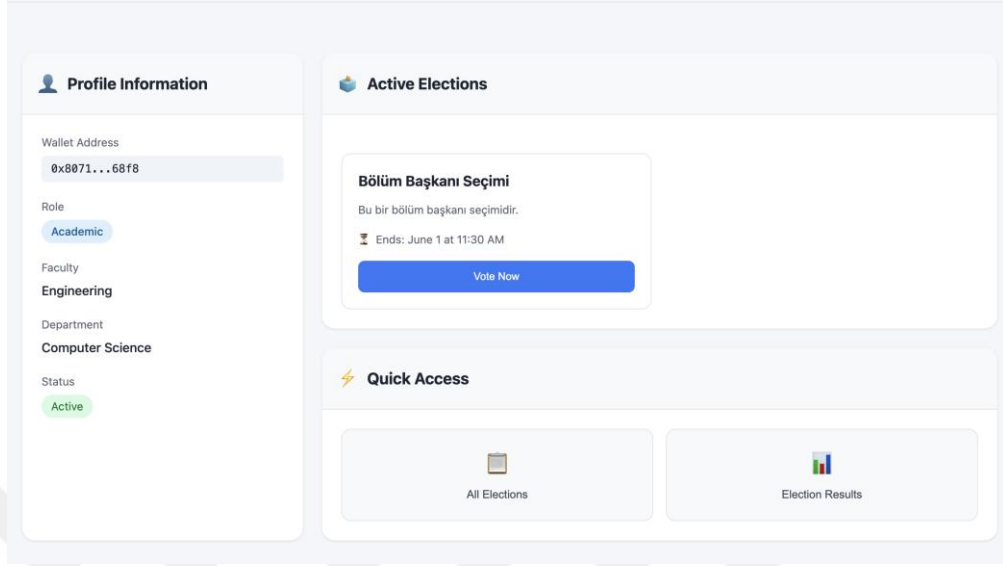


**Şekil 3. 24.** Seçmen ve seçim yetkilisi giriş arayüzü.

Seçmen, bilgileriyle giriş yaptıktan sonra Şekil 3.25.'de gösterildiği gibi seçimleri görebilmektedir. Seçmen panelinde görüntülenen seçim listeleri, seçmene göre hazırlanmaktadır. Seçmenin dahil olduğu rol (öğrenci, akademisyen veya memur), fakülte ve departman bilgisi dikkate alınmaktadır. Bu bilgiler ışığında seçmenin katılabileceği bir seçim varsa listelenmektedir. Seçmen, bir seçimde oy kullanmak istediğinde ise; oy kullanma zamanının, seçimin başlangıç ve bitiş tarihi arasında olması gerekmektedir.

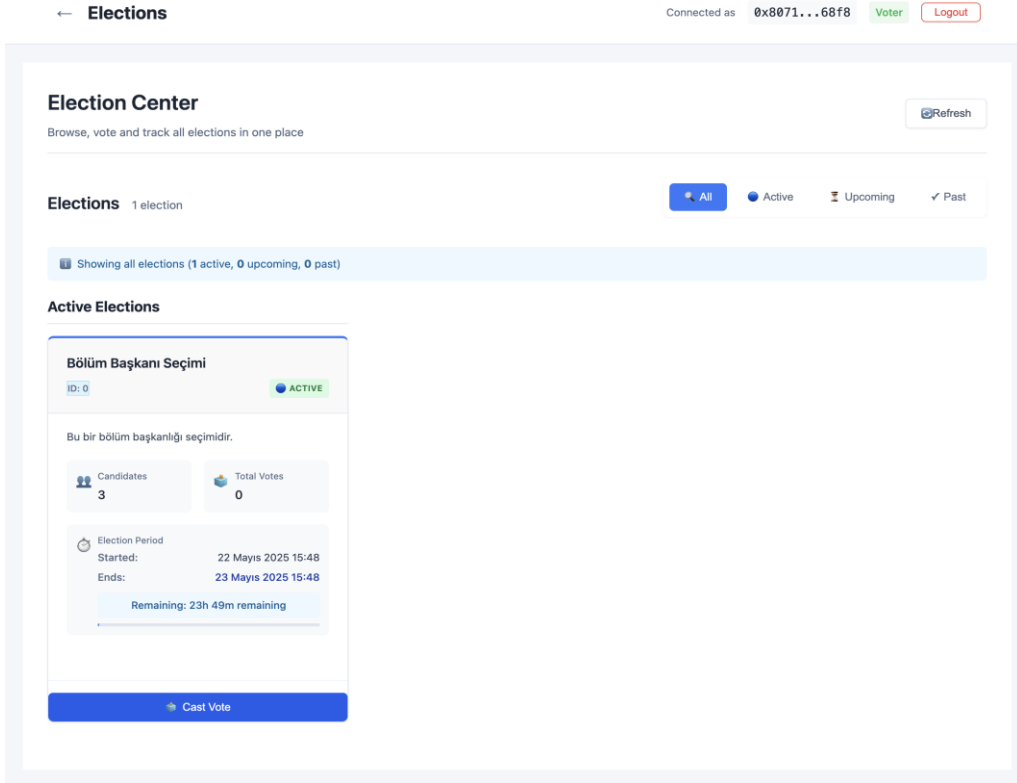
## Voter Dashboard

Connected as 0x8071...68f8 Voter Logout



Şekil 3. 25. Seçmen paneli ana sayfasında aktif seçimlerin görüntülenmesi.

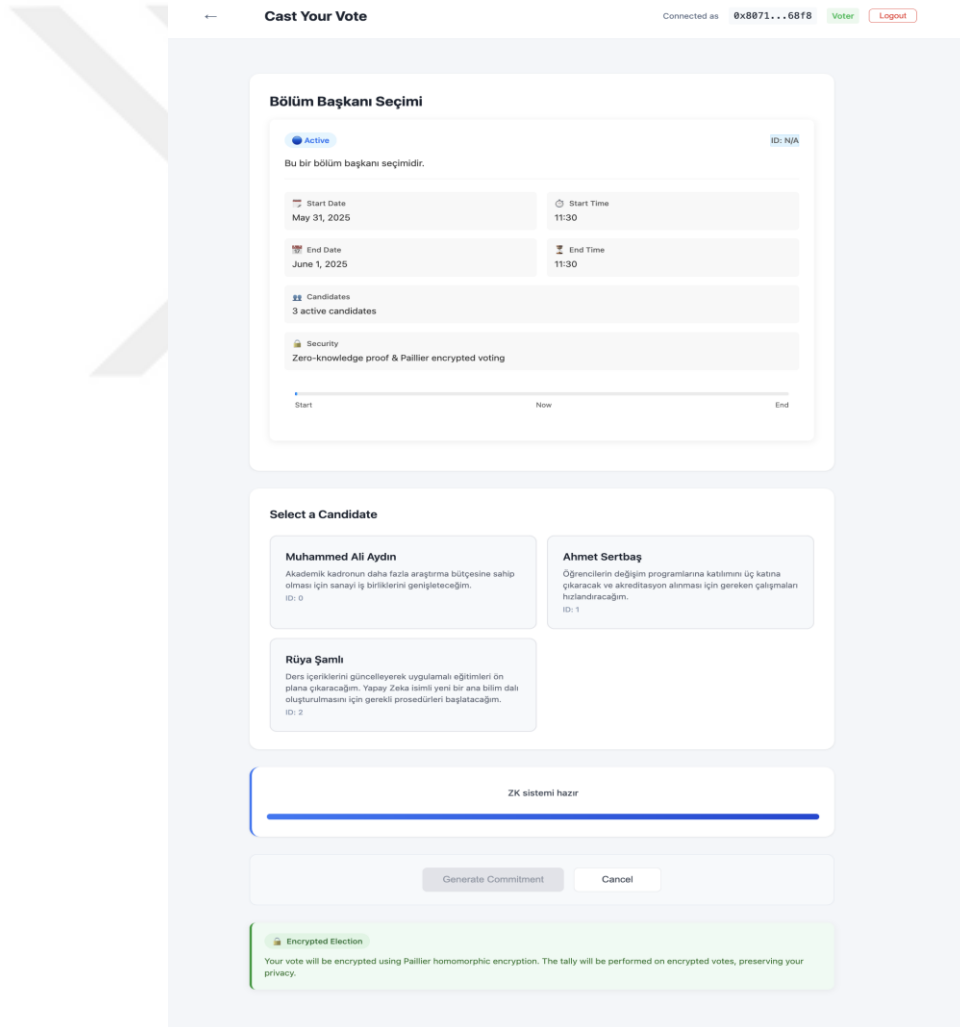
Bunun yanı sıra; seçmen, gelecek ve aktif seçimleri görmek için tüm seçimlerin olduğu Şekil 3.26.'daki arayüzü kullanabilmektedir.



Şekil 3. 26. Seçmen paneli üzerinden seçim listesinin görüntülenmesi.

Seçimde kullanıcı, aktif olan seçimlerde oyunu kullanmak için ilgili seçim pusulasının bulunduğu sayfayı kullanmaktadır. Bu sayfanın bir örneği Şekil 3.27.'de gösterilmektedir.

Kullanıcı mevcut seçeneklerden veya adaylardan birini seçer. İlgili seçenek seçildikten sonra commitment (taahhüt) butonuna basarak bir commitment ve nullifier üretilmesi sağlanır. Bu commitment ve nullifier değerinin üretilmesi sonucu oy kullanma butonu aktif olur. Seçmen bu butona bastığında şifreleme algoritmasıyla seçmenin oyu şifrelenir ve SnarkJS yardımıyla Circom diliyle geliştirilen devre kullanılarak bir zk-SNARK kanıtı (proof) oluşturulur. Oluşturulan parametreler akıllı sözleşmeye iletilir ve doğrulama işlemleri tekrar gerçekleştirilerek süreç tamamlanır.



Şekil 3. 27. Seçmen paneli üzerinden oy kullanma arayüzü.

Bu süreç tamamlandığında seçmene Şekil 3.28.'de gösterilen arayüz sunulmaktadır. Seçmen bu arayüz üzerinden, oyunu doğrulamak için gereken bilgileri kaydedebilmektedir.

← Verify Your Vote
Connected as 0x8071...68f8 Voter Logout


**Vote Successfully Cast!**  
Your vote has been securely recorded on the blockchain with zero-knowledge privacy protection.
Verified

**✓ Your Vote Verification**
Verified on Blockchain


Your vote was cast with:  
**Zero-Knowledge Privacy Protection**  
Your candidate choice is encrypted and private

 Election ID  
0

 Vote Time  
May 31, 2025  
11:40

 Block Number  
23

 Transaction  
0x584348...4593b8

**Cryptographic Verification**

Commitment Hash  
0x1f435af386656d...0x1f435af386656d2434de41572fac057c2524f3079df66c7330322f0f8d9db1f4

Nullifier  
4354121764322757...

Encryption  
Paillier Homomorphic

Proof System  
Groth16 ZK-SNARK

Hide Proof Data

**Zero-Knowledge Proof Data**  
**✓ ZKP Data Available**

```

{
  "proof": {
    "a": [
      "17044126020644254718482350741151335899917776557701638407818006346660142551422",
      "14036612052079890430549204378788299009293149993943539624309868693145562681743"
    ],
    "b": [
      "11407999416292507331568003242191853566619902086915430453061022071047364761859",
      "12903663958141948777179410928202708909838173500005785694417483519944878937984"
    ]
  },
  "public_inputs": [
    "14140704782013232515179303416943798357941887127715427552906995591242684674548",
    "0",
    "3",
    "1748580200",
    "1748766680",
    "1748583000",
    "8616460799",
    "8616460800",
    "72728516115775366074"
  ]
}

```

**Public Inputs**

```

[
  "14140704782013232515179303416943798357941887127715427552906995591242684674548",
  "0",
  "3",
  "1748580200",
  "1748766680",
  "1748583000",
  "8616460799",
  "8616460800",
  "72728516115775366074"
]

```

**Tip:** Save your vote proof for future reference and independent verification.

Verify Vote
Save Vote Proof

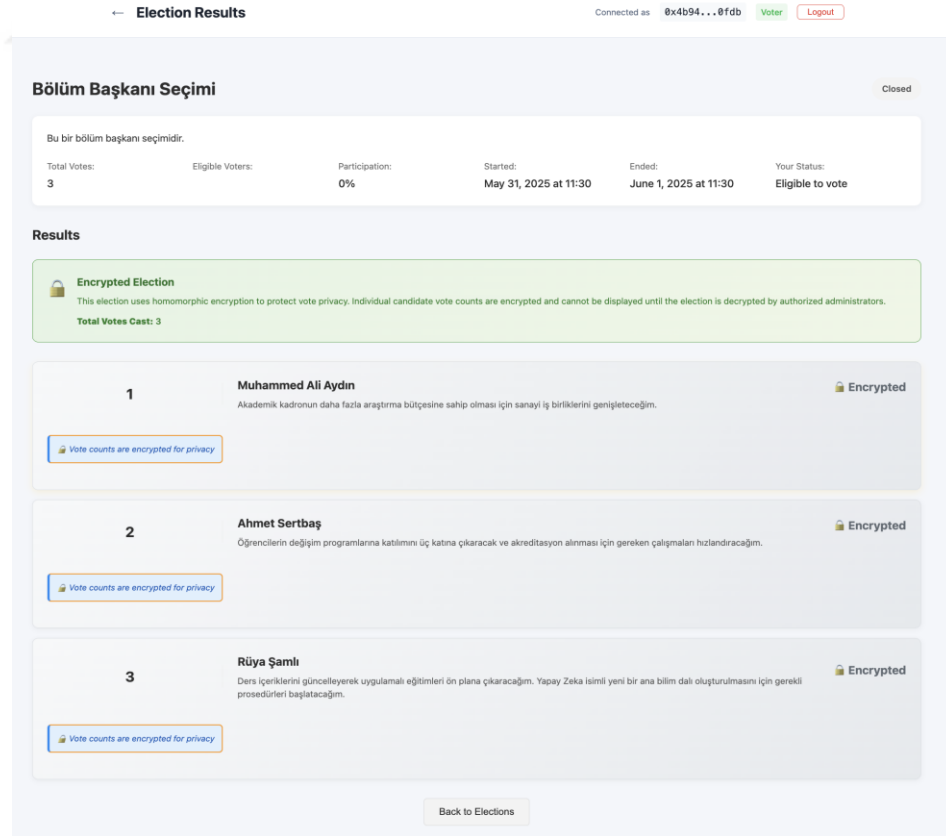
Şekil 3. 28. Seçmen paneli başarılı oy kullanma arayüzü.

### 3.4.5. Aşama 5: Oyların Saklanması

Geliştirilen oylama sisteminde, oylar blokzincir üzerine yazılmadan önce Paillier homomorfik şifreleme algoritmasıyla şifrelenir. Şifrelenmiş oylar, blokzincir üzerinde açık olarak saklanır fakat şifrelenmiş olduğu için oyun içeriği gizli kalmaktadır. Oyunun geçerli olduğunun ispatı için sıfır bilgi ispatı kullanılır. Circom dili ile geliştirilen devre, kullanılan oyun geçerliliğini doğrulamayı sağlar. Böylece oyun içeriği görüntülenmeden geçerli olduğu ispatlanmış olur. Kullanılan her oy için benzersiz bir nullifier ve commitment değeri üretilmektedir. Seçmenin mükerrer oy kullanmaması için nullifier değeri blokzincire kaydedilir. Commitment değeri ise, seçmen için oyun varlığını ve içeriğini kilitler, değiştirilmesini engeller ve oy kullanma işlemi sonucunda blokzincire kaydedilir.

### 3.4.6. Aşama 6: Sonuçların İlanı

Seçim süresi bittiğinde oy kullanma işlemine, sistem tarafından izin verilmez. Fakat seçimin kapatılması ve tamamlanması işlemini seçim yetkilileri gerçekleştirebilmektedir. Bu aşamada seçimin kapatılması sonuçların ilanı için yeterli değildir ve bu aşamada sonuçlar hiçbir aktör tarafından görülemez.



Şekil 3. 29. Seçmen paneli seçim sonuçları arayüzü.

Bir seçmen, kapatılmış veya sonlandırılmış seçimin sonuçlarını görmek istediğinde Şekil 3.29.'daki arayüz ile karşılaşacaktır. Seçim yetkilisi ise; seçim sonuçlarını görmek için seçim detay sayfasını açtığında, Şekil 3.30.'daki arayüzle karşılaşacaktır. Bu durum seçim sonuçlarının ilan edilmemesinden kaynaklıdır.

← Election Details

Connected as 0x549a...425d Admin Logout

Bölüm Başkanı Seçimi

CLOSED 3 votes 4 eligible 75% participation

Election Information

Description:

Bu bir bölüm başkanı seçimidir.

Start Date:

May 31, 2025 at 11:30

End Date:

June 1, 2025 at 11:30

Duration:

24 hours

Scope

Allowed Roles:

Academic

Faculties:

Engineering

Departments:

Computer Science (Engineering)

Candidates

Muhammed Ali Aydın

Akademik kadronun daha fazla araştırma bütçesine sahip olması için sanayi iş birliklerini genişleteceğim.

Wallet

0xD60935C0a618ee0771819bb88fA4ADE8fd6F

Address:

46B9

Ahmet Sertbaş

Öğrencilerin değişim programlarına katılımını üç katına çıkaracak ve akreditasyon alınması için gereken çalışmaları hızlandıracam.

Wallet

0x5020C6e59CccDee69E90234504e9A267ef

Address:

416B

Rüya Şamlı

Ders içeriklerini güncelleyerek uygulamalı eğitimleri ön plana çıkaracağım. Yapay Zeka isimli yeni bir ana bilim dalı oluşturulmasını için gerekli prosedürleri başlatacağım.

Wallet

0x7f12a3eC470B0cB20f5E1E67Efb3462976D

Address:

63C5

Vote Results

Encrypted Election

Decryption Method:

Per-Candidate Results (Full Transparency)

Full Transparency Mode

Reveals each candidate's vote count using homomorphic aggregation. Individual votes remain private, but results are fully transparent.

Complete election results

Winner determination

Candidate rankings

Individual votes still private

Candidate performance visible

Decrypt Election (Full Results)

Aggregated Encrypted Election

This election uses aggregated encryption for maximum privacy:

Homomorphic Aggregation:

Votes are summed while encrypted

Privacy Preservation:

Individual choices never revealed

Total Only:

Only aggregate vote count can be decrypted

Permanent Protection:

Candidate-specific votes remain encrypted forever

No vote results available yet.

Results will appear here once votes are cast and the election is closed.

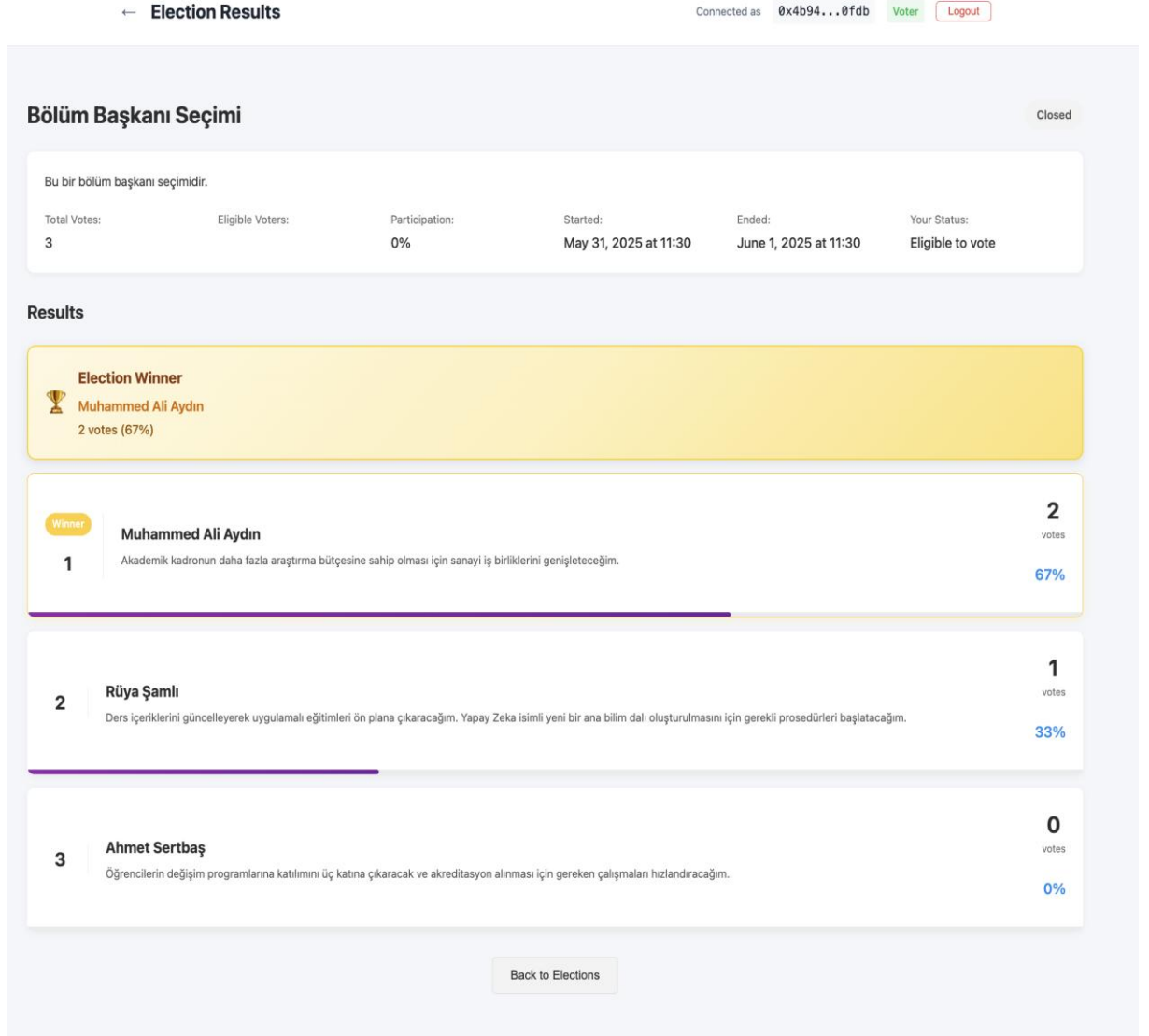
Back to Elections

Verify Election

Complete Election

Şekil 3. 30. Seçim yetkilisi paneli seçim detayları arayüzü.

Seçim sonuçlarının görülebilmesi için seçim yetkilisinin, Şekil 3.30.'da gösterilen arayüz üzerinden Decrypt Election butonuna basması ve MetaMask üzerinden onay vermesi gerekmektedir. Bu süreç sonucunda, seçim sonuçlarının açıklanabilmesi için homomorfik toplama işlemi gerçekleştirilir ve ardından toplam sonucun şifresinin çözülmesi sağlanır.



**Şekil 3. 31.** Seçmen paneli seçim sonucu arayüzü.

Seçim sonuçlarının açıklanmasıyla seçmen, Şekil 3.31.'de gösterilen arayüz üzerinden seçim sonuçları hakkında bilgi alabilmektedir. Seçim yetkilisi ise, Şekil 3.32.'de gösterilen seçim detay sayfasıyla sonuçları görüntüleyebilmektedir.

← Election Details
Connected as 0x549a...425d Admin Logout

### Bölüm Başkanı Seçimi

CLOSED
3 votes
4 eligible
75% participation

#### Election Information

Description: Bu bir bölüm başkanı seçimidir.

Start Date: May 31, 2025 at 11:30

End Date: June 1, 2025 at 11:30

Duration: 24 hours

#### Scope

Allowed Roles: Academic

Faculties: Engineering

Departments: Computer Science (Engineering)

#### Candidates

##### Muhammed Ali Aydın

Akademik kadronun daha fazla araştırma bütçesine sahip olması için sanayi iş birliklerini genişleteceğim.

Wallet: 0xD60935C0a618ee0771819bb8BfA4ADE8fd6F  
Address: 46B9

##### Ahmet Sertbaş

Öğrencilerin değişim programlarına katılımını üç katına çıkaracak ve akreditasyon alınması için gereken çalışmaları hızlandıracam.

Wallet: 0x50D20C6e59CcccDee69E90234504e9A267ef  
Address: 4168

##### Rüya Şamlı

Ders içeriklerini güncelleyerek uygulamalı eğitimleri ön plana çıkaracağım. Yapay Zeka isimli yeni bir ana bilim dalı oluşturulmasını için gerekli prosedürleri başlatacağım.

Wallet: 0x7f12a3eC47DB0cB200f5E1E67EfB3462976D  
Address: 63C5

#### Vote Results

Results Available

##### Vote Results

##### Election Winner

### Muhammed Ali Aydın

2 votes (66.7%)

Total Votes: 3

Candidates: 3

Election Type: Standard

##### #1 Winner Muhammed Ali Aydın

Votes: 2 Percentage: 66.7%

##### #2 Rüya Şamlı

Votes: 1 Percentage: 33.3%

##### #3 Ahmet Sertbaş

Votes: 0 Percentage: 0.0%

Back to Elections
Verify Election
Complete Election

Şekil 3. 32. Seçim yetkilisi panel seçim sonuçları arayüzü.

Bütün bu süreçlerle birlikte seçim sonuçlarının ve sıfır bilgi ispatı kanıtlarının blokzincir üzerine yazılması, ağ üzerinde değiştirilemez kayıtlar oluşturulmasını sağlamaktadır. Seçmen ve yetkililerin bu kayıtları kullanarak seçim sonuçlarını doğrulayabilmesi gerekmektedir. Bu

nedenle, seçmenlerin ve seçim yetkililerinin doğrulama işlemlerini yapabilecekleri arayüzler hazırlanmıştır. Seçmenler, Şekil 3.33.'de gösterilen arayüz üzerinden doğrulama işlemlerini gerçekleştirebilmektedir.

← **Vote Verification**
Connected as 0x4b94...0fdb Voter Logout

**Bölüm Başkanı Seçimi**

|              |   |         |        |              |   |
|--------------|---|---------|--------|--------------|---|
| Election ID: | 0 | Status: | Closed | Total Votes: | 3 |
|--------------|---|---------|--------|--------------|---|

**My Vote Status**

✓ You have voted in this election

|                   |                                                                    |
|-------------------|--------------------------------------------------------------------|
| Transaction Hash: | 0x96cee93ec5d2ec26190bef699ebe38fffd220fdfe7cb79ccd81662952e614ca5 |
| Block Number:     | 25                                                                 |
| Commitment:       | 0x1634eb52095d2a2ade44996e6d1c4bfd79279cf5956c962a1cb0933094fc83a9 |

**Verify My Vote**

Verify your vote using stored commitment data:

Verify My Vote

Or verify manually by entering your commitment:

Enter your vote commitment (0x...)

Verify

**Election Integrity**

✓ Secure

|                    |         |
|--------------------|---------|
| Total Votes:       | 3       |
| Nullifier Count:   | 3       |
| Counts Match:      | ✓ Valid |
| Nullifiers Unique: | ✓ Valid |

**Cryptographic Proof Verification (Zero-Knowledge)**

Enter ZK Proof Data

**Blockchain Verification**

Verify on Blockchain

**Independent Verification**

Perform Independent Verification

Back to Elections

Şekil 3. 33. Seçmen paneli seçim doğrulama arayüzü.

Seçim yetkilileriye, Şekil 3.34.'de gösterilen sayfa üzerinden seçimle ilgili tüm doğrulama bilgilerine ulaşabilmektedir.

← Election Verification
Connected as 0x549a...425d Admin Logout

### Bölüm Başkanı Seçimi

Election ID: 0 Status: Closed Total Votes: 3 Candidates: 3

#### Overall Audit Summary

Valid

|              |   |                  |   |             |       |                    |       |
|--------------|---|------------------|---|-------------|-------|--------------------|-------|
| Total Votes: | 3 | Nullifier Count: | 3 | Math Valid: | Valid | Nullifiers Unique: | Valid |
|--------------|---|------------------|---|-------------|-------|--------------------|-------|

Audit Hash:

0x0e17dfe74bb96c01b042e0eb28d911232b824eaf39fd15a262a21968ff1bfc87

#### Integrity Verification

Valid

|                      |     |
|----------------------|-----|
| Contract Vote Count: | 3   |
| Nullifier Count:     | 3   |
| Counts Match:        | Yes |

Election Hash:

0x2d4090c190a234f46ed87abb2ba  
d6162481c26540829f3a5fef12743a  
a87ad5

#### Mathematical Verification

Valid

|                   |     |
|-------------------|-----|
| Calculated Total: | 3   |
| Stored Total:     | 3   |
| Totals Match:     | Yes |
| Candidate Count:  | 3   |

Math Hash:

0xc875613611a47928eb9d87d9c08d  
b0d0b23d728e80e82ee6b7746e205e  
c8cb23

#### Nullifier Verification

Valid

|                   |     |
|-------------------|-----|
| Total Nullifiers: | 3   |
| Duplicate Count:  | 0   |
| All Unique:       | Yes |

Nullifier Hash:

0x87f4666eee97506a21d01bf44212  
e3efa9e250a8d31ea16358aa147e8e  
10bdc5

#### Blockchain Event Verification

Valid

|                         |     |
|-------------------------|-----|
| Events from Blockchain: | 3   |
| Events from Contract:   | 3   |
| Events Match:           | Yes |

Refresh Verification Export Audit Report Back to Elections

Şekil 3. 34. Seçim yetkilisi paneli seçim kontrolü arayüzü.

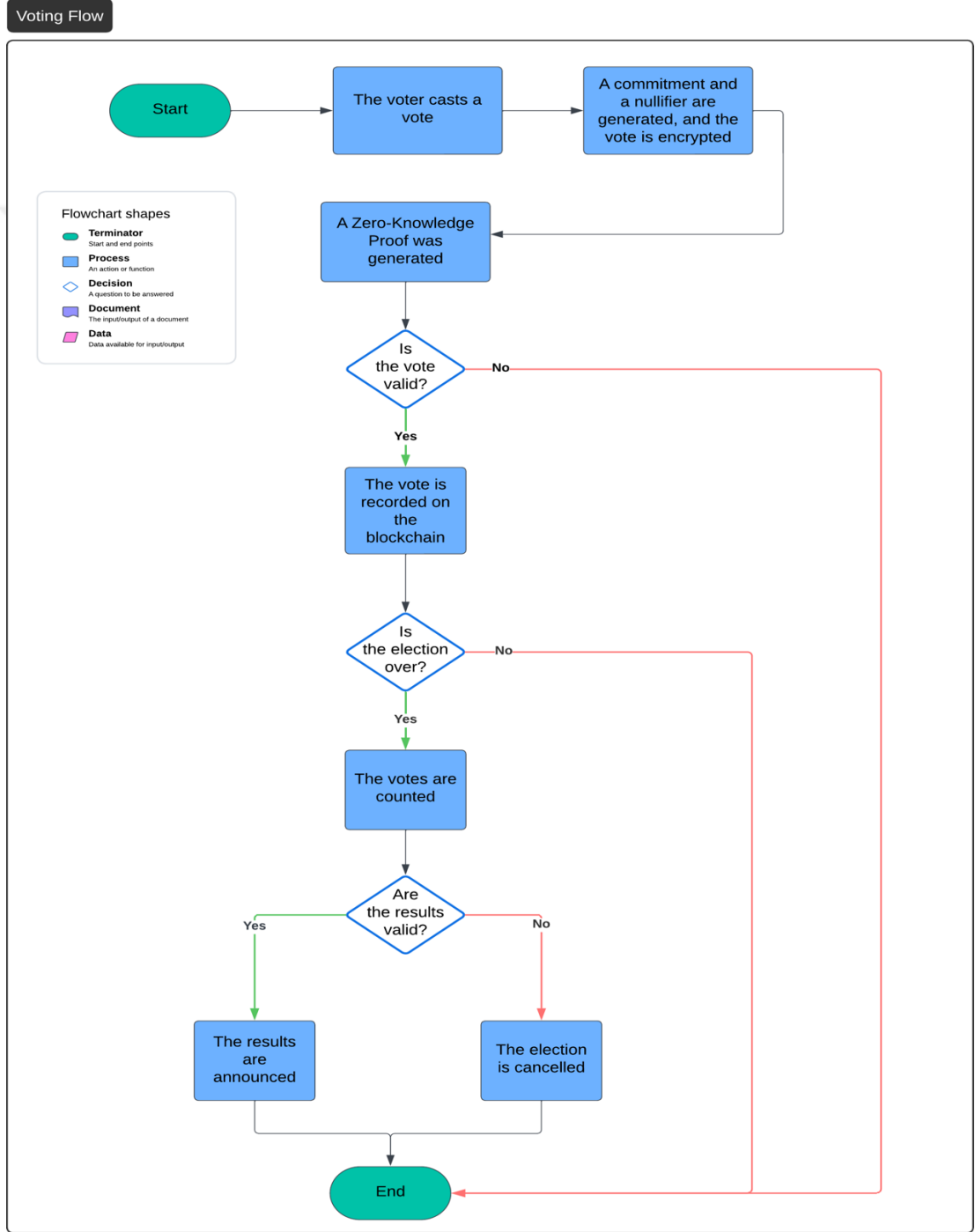
### 3.5. OYLAMA SÜRECİ VE İŞ AKIŞI

Oylama işlemi, sistemin en kritik bölümüdür ve genel olarak şu aşamalardan oluşmaktadır:

1. Seçmen Kimlik Doğrulaması: Seçmen, tarayıcı kullanarak MetaMask cüzdanıyla kimliğini oylama sistemi üzerinden doğrular. Oylama sistemi, seçmenin kayıt durumunu ve oy kullanma yetkisini kontrol eder.
2. Seçime Katılma: Seçmen, aktif ve süresi dolmamış seçimler içerisinde istediği seçime katılabilir.
3. Aday Seçimi: Seçmen, seçim pusulası için açılan sayfadan istediği adayı seçer ve arayüz tarafından geçici olarak kayıt işlemi gerçekleştirilir.
4. Commitment Oluşturma: Arayüz tarafında nonce, secret, randomness değerleri oluşturulur. Bu değerlerle, seçilen adayın indeks değeri birleştirilerek commitment oluşturulur.
5. Nullifier Oluşturma: Secret değeri ve seçimin indeks değeri ile birlikte nullifier değeri oluşturulur. Nullifier değeri, bir seçmenin birden fazla oy kullanmasını engellemek için kullanılır.
6. Oyun Şifrelenmesi: Seçmen tarafından kullanılan oyun Paillier şifreleme algoritması ile şifrelenmesi gerçekleştirilir.
7. Sıfır Bilgi İspatı Kanıtı Oluşturma: Arayüz tarafından ZKP için geliştirilen devre dosyalarına gerekli parametrelerin iletilmesi sağlanır. SnarkJS isimli JavaScript kütüphanesi kullanılarak kanıt oluşturulur. Bunun sonucunda oluşturulan kanıt, akıllı sözleşmeye gönderilmek için uygun formata dönüştürülür. Kanıt oluşturma sürecinde, gizli parametre olarak oyun açık hali kullanılır; ancak üretilen kanıt, oyunun içeriğine dair herhangi bir anlamlı bilgi barındırmaz.
8. Akıllı Sözleşme İşlemleri: Oluşturulan kanıt, kanıt oluşturulması sırasında verilen açık parametreler, şifreli oy, commitment ve nullifier değerleri akıllı sözleşmeye gönderilir. Akıllı sözleşme, gönderilmiş olan kanıtı doğrular ve geçerliyle oyu kaydeder. İşlem

başarılı ile sonuçlandığında, kullanıcıya kullanmış olduğu oyu doğrulamak için seçenek sunulur.

Bu süreçlerin tamamı, seçim sürecinin gerçekleştirilmesi ve oylamanın başarıyla sonuçlanması için gereklidir. Seçmen tarafından yapılan oylama işleminin nasıl gerçekleştirildiği, Şekil 3.35.'deki akış diyagramında gösterilmiştir.



Şekil 3. 35. Oylama işleminin akış diyagramı.

### 3.6. SİSTEM MİMARİSİ

Bu çalışma kapsamında oluşturulmuş olan mimari, güvenlik, şeffaflık ve gizlilik ilkelerine bağlı kalarak seçmenlerin şifreli ve doğrulanabilir bir şekilde oy kullanmalarına olanak sağlamak için tasarlanmıştır. Tasarım aşamasındaki tüm tercihler, blokzinciri tabanlı merkezi olmayan üniversite ölçekli bir oylama sistemine uygun olarak belirlenmiştir.

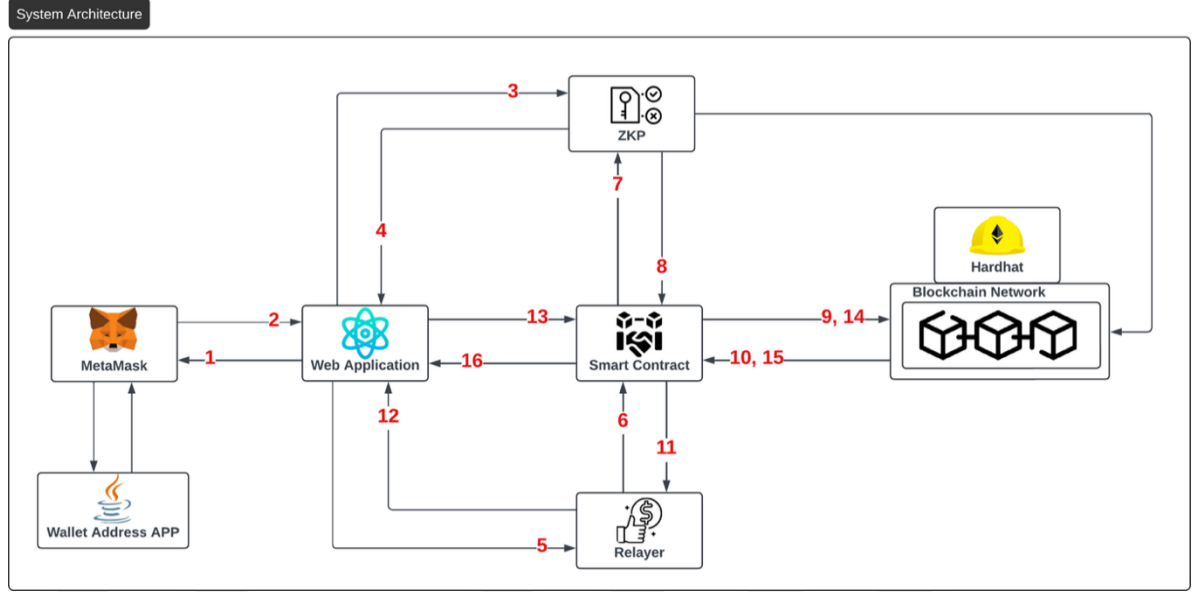
**Kullanıcı Arayüzü:** Bu bölüm, seçim yetkililerinin ve seçmenlerin kullandıkları arayüzü kapsamaktadır. React kullanılarak SPA (Single Page Application) yapısında bir web uygulaması olarak hizmet vermektedir. Bu bölüm kapsamında seçim yetkilileri ve seçmenler olmak üzere iki panel arayüzü oluşturulmuştur.

**Relayer Servisi:** Akıllı sözleşmeler ile kullanıcı arayüzü arasında bir köprü oluşturarak, seçmenlerin Ethereum ağı üzerinde oy kullanma işlemi gerçekleştirirken, cüzdanlarından gaz ücreti ödemek zorunda kalmamaları için geliştirilmiştir. Bütün gaz maliyeti bu servis tarafından karşılanmaktadır.

**Akıllı Sözleşmeler:** Ethereum platformu üzerinde çalışan akıllı sözleşmeler, sistemin mantıksal iş akışını yürütmektedir. Seçim yetkilileri tarafından seçim, yetkili, seçmen ve fakülte yönetiminin yapılmasını sağlamakla birlikte; seçmenlerin, seçim süreçlerine katılması için gereken iş süreçlerinin tamamını kapsamaktadır. Sıfır bilgi ispatı için doğrulama yapılması, şifreli oyların saklanması ve nullifier kontrol işlemleri bu bölüm içerisinde gerçekleştirilir.

**Kriptografik Teknikler:** Seçmen arayüzünde üretilen ve akıllı sözleşmeler üzerinde doğrulaması yapılabilen tüm kriptografik işlemleri kapsamaktadır. Sistem içerisindeki aşamalarda gerçekleştirilen şifreli veri, commitment, nullifier ve sıfır bilgi ispatı kanıtı oluşturma işlemlerini kapsar.

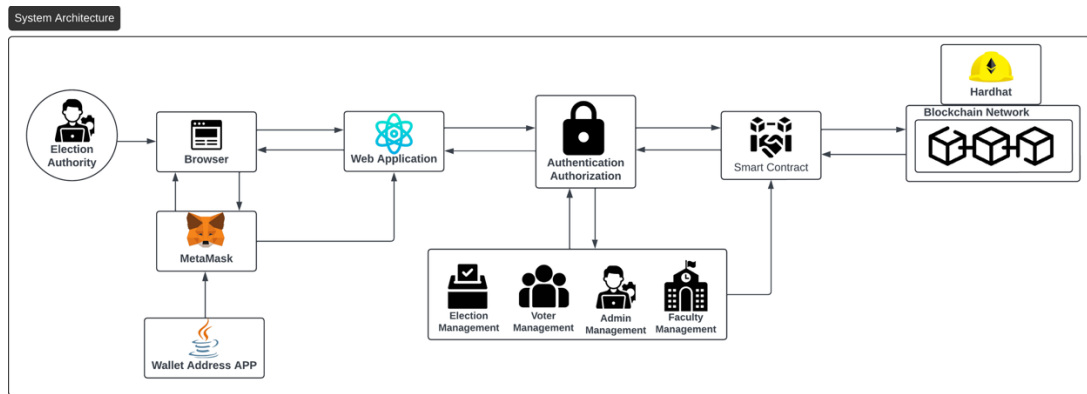
**Sıfır Bilgi İspatı Devresi:** Sıfır bilgi ispatı kanıtı üretilmesi işlemi için devrenin geliştirilmesi, derlenmesi ve dağıtılmasıyla ilgili süreçler için hizmet vermektedir.



Şekil 3. 36. Oylama sisteminin sistem mimarisi.

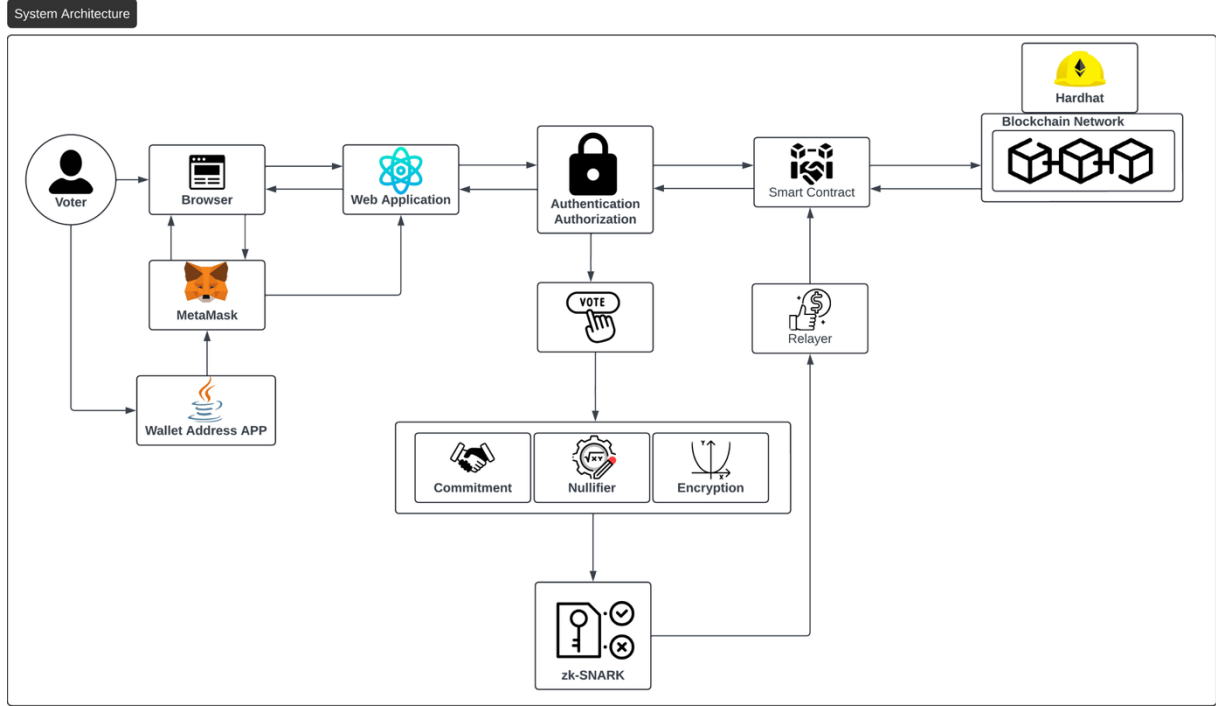
Bu tercihler neticesinde Şekil 3.36.'da gösterilen bileşenlerin bir araya getirilmesiyle blokzinciri tabanlı üniversite kapsamında gerçekleştirilen bir oylama sistemi mimarisi oluşturulabilmektedir. Ayar dosyalarıyla ve ağ üzerinde gerçekleştirilecek değişikliklerle yapı tamamen merkezi olmayan dağıtık bir sistem olarak hazırlanabilmektedir. Bu bileşenlerin her biri farklı noktalarda birbirleriyle çalışmaktadır. Sistemin her aşamasında bir bütün olarak değil, gerektiğinde birbirleriyle bağlantı kuracak şekilde tasarlanmıştır. Bu sebeple seçim yetkilileri ve seçmenler için ortaya iki farklı şema çıkartılabilmektedir.

Bir seçim yetkilisinin, sistemi kontrol etmesiyle ilgili süreçleri içeren mimari tasarım Şekil 3.37.'de gösterilmektedir. Yetkililerin süreçleri denetleme hakları bulunmasına rağmen, seçim sonuçlarına müdahale etmesi mümkün değildir.



Şekil 3. 37. Seçim otoritesi süreçlerine ait sistem mimarisi diyagramı.

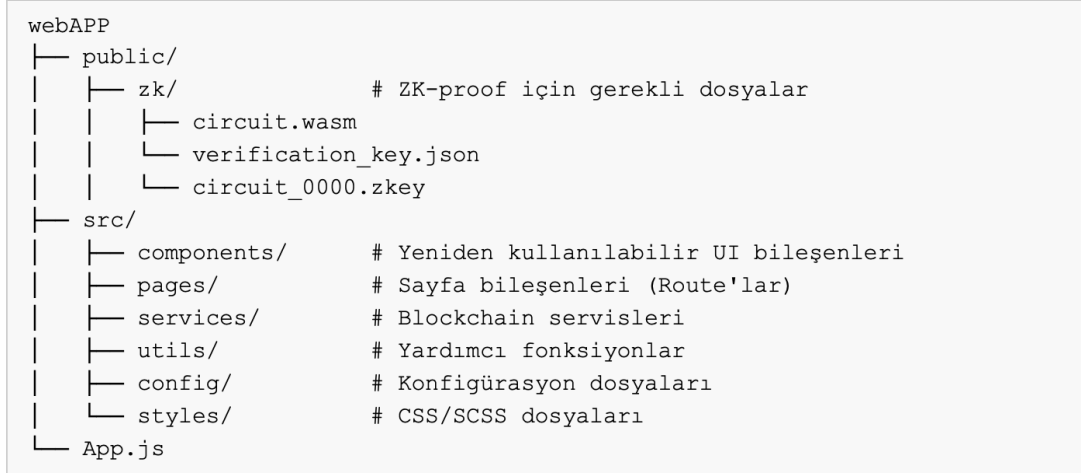
Şekil 3.38.'de gösterilen şema, bir seçmenin oy kullanma sürecinin mimarisini göstermektedir. Oy kullanma işlemi, tüm sistem bileşenlerinin kullanıldığı bir bölümdür. Oyun başarılı bir şekilde kullanılması ve doğrulanması için sistemde bulunan tüm bileşenlerin uyumlu bir şekilde çalışması gerekmektedir.



Şekil 3. 38. Oy kullanma sürecine ait sistem mimarisi diyagramı.

### 3.7. KULLANICI ARAYÜZÜ

Seçmenler ve yetkililerin kullandıkları arayüz React.js kullanılarak SPA (Single Page Application) yapısında bir web uygulaması olarak geliştirilmiştir. Geliştirme aşamasında kullanılan teknolojiler React.js, Ether.js, SnarkJS ve Material UI'dır. Circom dili ile geliştirilen devrenin; JavaScript programlama diliyle SnarkJS kütüphanesi kullanılarak derlenmesi sonucu WASM, JSON ve ZKEY uzantılı dosyalar oluşmaktadır. Bu dosyaların, seçmenin tarayıcısı üzerinde yüklenebilmesi için kullanıcı arayüzündeki public klasöründe bulunması gerekmektedir. Bu dosyalar, istemci tarayıcısı üzerinde yüklenir ve arayüz tarafından sıfır bilgi ispatı kanıtının oluşturulması için kullanılır. Şekil 3.39.'da kullanıcı arayüzün yapısı gösterilmiştir. Public içerisindeki zk klasörü devrenin derlenme dosyalarını ve src klasörü de geliştirilen uygulamanın dosyalarını barındırmaktadır.



**Şekil 3. 39.** Arayüzün yapısını gösteren şema.

### 3.8. RELAYER SERVİSİ

Oylama sisteminde bulunan ve Meta-Transaction Relayers olarak isimlendirilen relayer servisi, seçmenlerle akıllı sözleşmeler arasında bir katman olarak çalışmaktadır. Relayer servisi, Node.js tabanlı bir Express uygulaması olarak geliştirilmiştir. Böylece seçmenlerin oy kullanması sırasında ödemesi gereken gaz ücreti üçüncü taraf bir aracı (relayer) tarafından karşılanmaktadır ve işlemler bu aracı vasıtasıyla akıllı sözleşmelere aktarılmaktadır. Relayer servisi, işlemlerin aşağıdaki nedenlerle daha verimli bir şekilde gerçekleştirilmesini sağlar:

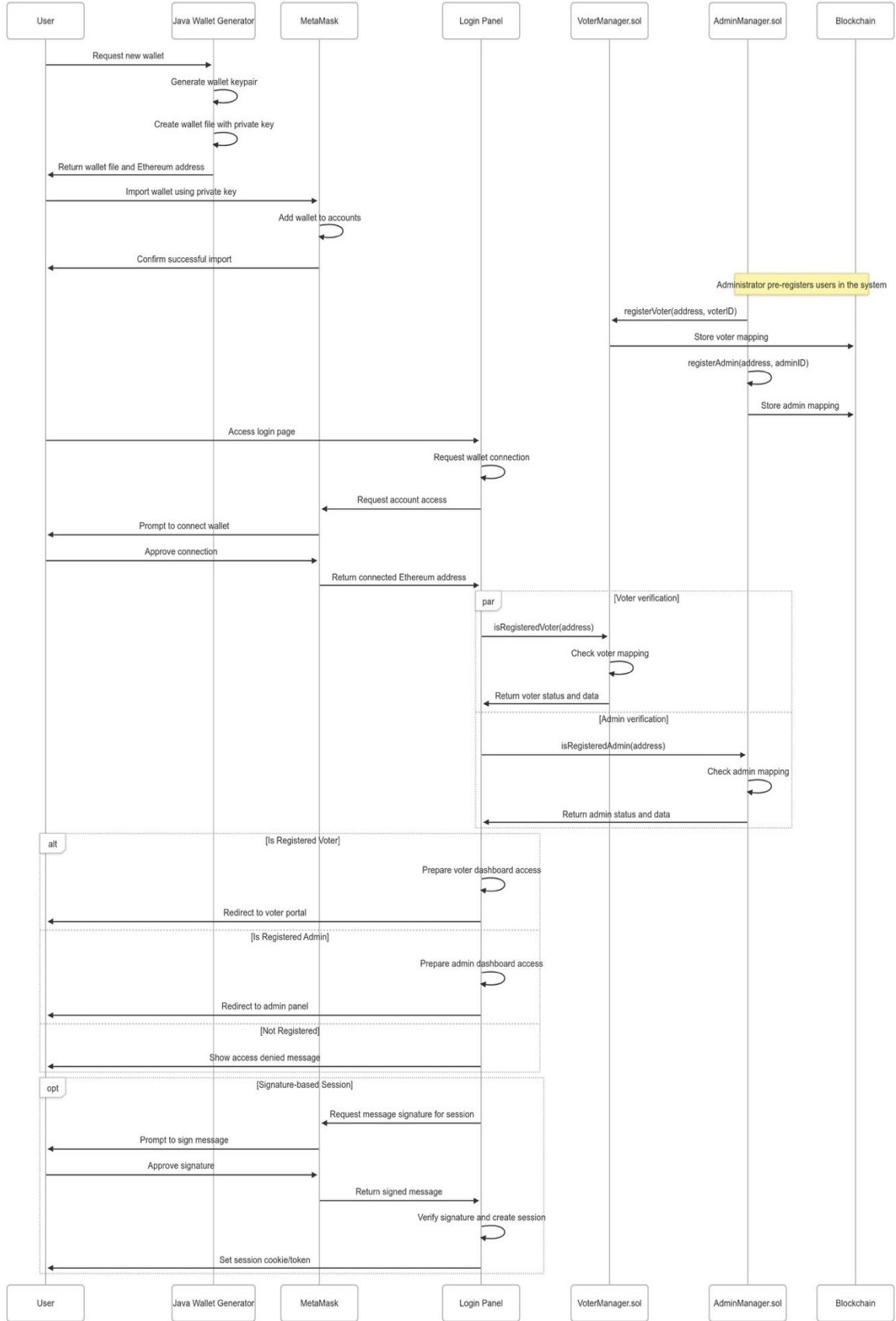
1. Gaz Yönetimi: Meta-Transaction Relayers ile seçmenlerin oy kullanma sürecinde oluşan gaz ücreti, üçüncü taraf servisle (relayer) ödenerek gerçekleştirilir.
2. Yeniden Deneme Mekanizması: Doğru bir şekilde gerçekleşmeyen işlemleri otomatize ederek tekrar gerçekleşmesi için gereken süreci başlatır. Böylece seçmene herhangi bir hata verilmeden süreç tamamlanmış olur. Burada doğru bir eşik (threshold) değeri belirlenmesi önemlidir.

### 3.9. KİMLİK DOĞRULAMASI

Kimlik doğrulaması, oylama sistemini oluşturan en önemli bileşenlerden birisidir. Seçim yetkililerinin ve seçmenlerin, oylama sistemine giriş yapabilmeleri için kimlik doğrulama sürecinden geçmeleri gerekmektedir. Bu doğrulama sürecinin, seçmenin kişisel kimlik bilgilerinden izole bir şekilde gerçekleştirilmesi önem taşımaktadır. Seçmenlerin gerçek

hayattaki kimlikleriyle bu süreç arasında bir bağlantı kurulamaması seçim güvenilirliğini artırmaktadır. Bu sürecin başarıyla gerçekleştirilebilmesi için birden fazla katman birlikte çalışmaktadır. Katmanların birlikte çalışmasını gösteren diyagram Şekil 3.40.'da bulunmaktadır.

Geliştirilen MetaMask uyumlu cüzdan üretici Java uygulamasıyla seçmen veya yetkililerin, birer cüzdan üretmesi ve bu üretilen cüzdan bilgilerinden elde edilen Ethereum adresinin, oylama sisteminde kayıtlı olması gerekmektedir. Cüzdan bilgilerinin içerisinde bulunan özel anahtarın (private key), tarayıcıları üzerindeki MetaMask uzantısına dahil edilmesiyle (import) seçmen veya yönetici sisteme giriş yapma isteğinde bulunabilmektedir. Seçmen veya yönetici, giriş sayfası (LoginPanel) üzerinden MetaMask cüzdanlarıyla istekte bulunulduğunda, MetaMask ile akıllı sözleşmelerin iletişim kurması sağlanmaktadır. Bu istek MetaMask, web uygulaması ve akıllı sözleşmeler arasında gerçekleştirilir. Bu iletişim kapsamında VoterManager ve AdminManager isimli iki akıllı sözleşmeyle, istek gönderen cüzdan sahibinin giriş izninin olup olmadığı belirlenmektedir. Bunun sonucunda istek gönderen kişinin seçmen veya seçim yetkilisi olması durumuna göre gerekli sayfa yönlendirmeleri gerçekleştirilmektedir.



Şekil 3. 40. Kimlik doğrulama sürecine ait sıralama diyagramı (sequence diagram).

### 3.10. KRİPTOGRAFİK TEKNİKLER

Bu bölümde gerçekleştirilen işlemler, seçmenlerin oyunu gizlemek ve doğrulamak için kullanılan yöntemlerden oluşmaktadır. Şekil 3.38.'de gösterilen oylama şeması üzerindeki kriptografik yapılar ayrıntılı olarak açıklanmaktadır.

#### 3.10.1.1. Commitment Oluşturma (Taahhüt Oluşturma)

Oyun değiştirilememesi ve açığa çıkartılamaması amacıyla kullanılan commitment oluşturma işlemi bir çeşit taahhüt işlemidir. Bu yöntemle, seçmenin oyu hash fonksiyonundan geçirilerek bir commitment değeri oluşturulur. Seçmenin oyunda herhangi bir değişiklik yapıldığında yeni üretilecek olan commitment, hash fonksiyonlarının özelliği gereği önceki oyla üretilen commitment değeriyle eşleşmeyecektir. Bu nedenle seçmenin oyu değiştiğinde commitment değerinden tespitler yapılabilecektir. Bu durumla bağlayıcılık (binding) ve gizlilik (hiding) ilkelerinin sağlanması amaçlanmıştır. Commitment oluşturulması aşaması şu şekilde yapılır:

İlk Özetleme (First Hash): Seçmenin oy bilgisi ve rastgele bir nonce değerinden ilk özetleme değeri oluşturulur.

$$hash1 = Poseidon(vote, nonce)$$

İkinci Özetleme (Second Hash): İlk özetleme sonucu, secret değeri ve randomness (ek rastgelelik) değerinden oluşturulur.

$$hash2 = Poseidon(hash1, secret, randomness)$$

Burada:

1. vote: Seçmenin belirlediği adayın indeks (ID) değeri
2. nonce: Oyun gizliliği için rastgele değer
3. secret: Seçmen için üretilen gizli değer
4. randomness: Commitment için rastgele değer

Aşağıdaki şekilde tek bir satırla ifade edilebilir:

$$commitment = Poseidon(Poseidon(vote, nonce), secret, randomness)$$

### 3.10.1.2. Nullifier Oluşturma

Bir seçmenin tekrar oy kullanmasını, kriptografik yöntemle engellemek için kullanılan bir değerdir. Seçmen oy kullanırken, ilgili seçim ve seçmenle bağlantı kurularak bir nullifier değeri oluşturulacaktır ve oy kullanma işlemi neticesinde bu değer blokzincire yazılacaktır. Seçmen aynı seçim için tekrar oy kullanmaya çalışırsa üretilecek olan nullifier değerinin blokzincirde bulunup bulunmadığı kontrol edilecektir. Nullifier kontrolü aşamasında çift katmanlı özetleme algoritması kullanılmakta ve şu adımlardan oluşmaktadır:

İlk Özetleme (First Hash): Seçmenin secret değeri ve seçim ID değeri alınarak ilk özetleme değeri oluşturulur.

$$hash1 = Poseidon(secret, electionId)$$

İkinci Özetleme (Second Hash): İlk özetleme değeri, tekrar özetleme değeri ile özetlenir.

$$nullifier = Poseidon(hash1)$$

Aşağıdaki şekilde tek bir satırla ifade edilebilir:

$$nullifier = Poseidon(Poseidon(secret, electionId))$$

### 3.10.1.3. Paillier Şifreleme Algoritması

Paillier şifreleme algoritması, seçmenlerin oylarını gizlemek amacıyla kullanılmaktadır. Sahip olduğu homomorfik özelliğinden dolayı şifreli oylar toplanabilmekte ve şifreli tek bir sonuç elde edilebilmektedir. Bu şifreli tek bir sonucun, başlangıçta belirlenen gizli anahtar kullanılarak açılmasıyla seçim sonuçları ilan edilebilmektedir. Paillier şifreleme algoritması için anahtar uzunluğuna karar verilmesi, anahtarın güvenli bir yerde saklanması ve uygun matematiksel formüllerin belirlenmesi kritik önem taşımaktadır. Sistemde kullanılan Paillier şifreleme algoritması için seçilen parametreler şu şekildedir:

#### Anahtar Boyutu Seçimi

2048-bit (256-byte) anahtar boyutu seçilir.

#### Anahtar Üretimi

p ve q büyük asal sayılar olmak üzere

$$n = p \times q$$

$$g = n + 1$$

$$\lambda = \text{lcm}(p - 1, q - 1)$$

$$\mu = L(g^\lambda \bmod n^2)^{-1} \bmod n$$

$$L(x) = \frac{x - 1}{n}$$

### Mesaj Seçimi

$$m = \text{verilen oy}$$

### Şifreleme İşlemi

$$c = g^m x r^n \bmod n^2$$

$0 < r < n$  ve  $\text{grc}(r, n) = 1$  şartını sağlayan rastgele bir sayıdır.

### Şifre Çözme

$$m = L(c^\lambda \bmod n^2) x \mu \bmod n$$

### Homomorfik Toplama

$$E(m_1) \cdot E(m_2) \equiv E(m_1 + m_2) \pmod{n^2}$$

Sistemdeki her oy, aday sayısı uzunluğunda bir vektör olarak temsil edilmektedir. Bu yüzden 10 adayın olduğu bir seçimde, her oy 10 elemanlı bir vektördür. Başlangıçta vektör Şekil 3.41.'deki gibi gösterilmektedir. Vektör üzerindeki indeksler, her bir aday için kullanılan oyu tanımlamaktadır. Örneğin; 0. indeks 1. adayı, 9. indeks 10. adayı temsil etmektedir.

$$[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

**Şekil 3. 41.** Şifrelenmemiş 10 elemanlı vektör gösterimi.

Seçmen, 0-9 indeks değerine sahip 10 aday arasından bir tercih yapmaktadır. Bu tercih neticesinde oy kullanılan adayın, vektördeki indeks değerinin değişmesi gerekmektedir. Seçilen aday için 1 değeri ve diğer adaylar için 0 değeri kullanılmaktadır.

$[Enc(0) \ Enc(0) \ Enc(0) \ Enc(0) \ Enc(1) \ Enc(0) \ Enc(0) \ Enc(0) \ Enc(0) \ Enc(0)]$

**Şekil 3. 42.** Paillier şifreleme algoritmasıyla şifrelenecek 10 elemanlı vektör gösterimi.

Oyun açığa çıkmaması için vektördeki her eleman, 0 veya 1 rakamının Paillier şifreleme algoritmasıyla şifrelenmiş değerini tutması gerekmektedir. Bu işlem için vektör, şifreleme işlemine tabi tutulur. Şekil 3.42.'de şifreleme işlemi uygulanan vektör gösterimi bulunmaktadır. Bu işlemlerin nasıl gerçekleştiği aşağıda basit bir örnekle gösterilmektedir.

Şifreleme algoritması için gerekli olan iki asal sayı seçilir.

$$p = 11$$

$$q = 13$$

Güvenliğin sağlanması amacıyla p ve q sayıları için büyük bir asal sayı seçilmelidir. Örnek gösterim için p = 11 ve q = 13 tercih edilmiştir. Her indeks değeri için ise farklı rastgele sayıların kullanılması gerekmektedir. Bunun için Tablo 3.2.'de bulunan değerler kullanılmıştır.

**Tablo 3. 2.** Paillier şifreleme algoritması için kullanılan rastgele değerler.

| Sıra            | Rastgele Değer |
|-----------------|----------------|
| r <sub>1</sub>  | 5              |
| r <sub>2</sub>  | 7              |
| r <sub>3</sub>  | 9              |
| r <sub>4</sub>  | 3              |
| r <sub>5</sub>  | 2              |
| r <sub>6</sub>  | 8              |
| r <sub>7</sub>  | 4              |
| r <sub>8</sub>  | 6              |
| r <sub>9</sub>  | 1              |
| R <sub>10</sub> | 10             |

Bu değerler ve yukarıda Paillier şifreleme algoritması için verilen hesaplamalar yapılarak yeni şifrelenmiş değerler elde edilir. Şifreleme işlemi sonrası Şekil 3.43.'de gösterilen 10 elemanlı yeni vektör elde edilir.

$[102 \ 124 \ 176 \ 143 \ 192 \ 110 \ 134 \ 150 \ 189 \ 140]$

**Şekil 3. 43.** Paillier şifreleme algoritmasıyla şifrelenmiş 10 elemanlı vektör gösterimi.

#### 3.10.1.4. Sıfır Bilgi İspatı

Seçmenlerin, kriterlere uygun bir şekilde oy kullandığını, oyu göstermeden ispat edebilmek için Groth16 protokolüne dayalı bir zk-SNARK sistemi kullanılmıştır. Bu sistemde sıfır bilgi ispatı kullanabilmek için Circom 2.0 kullanılarak bir devre geliştirilmiştir. Geliştirilen devrenin içerisinde şu kontroller gerçekleştirilmektedir:

1. Seçmenin oy kullanma sırasında sistem tarafından üretilmiş olduğu commitment değerinin doğruluğu kontrol edilmektedir.
2. Seçmenin, seçimde kayıtlı bir adaya oy verip vermediğin kontrolü gerçekleştirilmektedir.

$$0 \leq vote \leq candidateCount$$

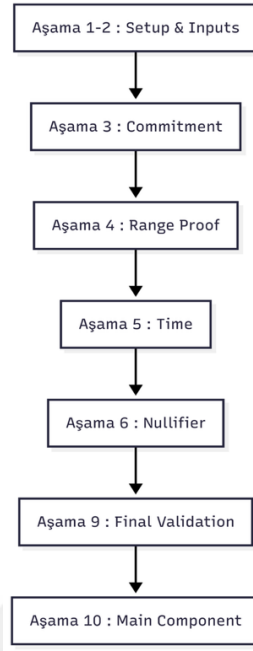
3. Oyun kullanıldığı zamanın, seçimin başlangıcıyla bitiş zamanı arasındaki ilişki kontrol edilmektedir.

$$startTime \leq currentTime \leq endTime$$

4. Seçmenin oy kullanma sırasında sistem tarafından üretilmiş olan nullifier değerinin doğruluğu kontrol edilmektedir.
5. Tüm kontrollerin başarılı olma durumu “ve” kapısıyla (and gate) doğrulanmaktadır. Başarılı ise valid=1 çıktısı ve başarısız ise valid=0 çıktısı verilmektedir.

Bu devre içerisinde şifrelenmiş verinin doğrulanmasını yapabilmek için şifreleme algoritmasının anahtar uzunluğunun küçültülmesi gerekmektedir. Çünkü Circom devreleri BN254 eğrisi üzerinde çalışır ve büyük sayılarla sınırlı işlem yapabilme yeteneğine sahiptir. Sistemde 2048-bit anahtar uzunluğu kullanıldığı için devre içerisinde şifreleme kontrolü eklenmemiştir.

Sıfır bilgi ispatı kanıtının oluşturulması ve doğrulanması süreçleri birbirinden farklılık göstermektedir. Bu süreçleri anlamak için devrenin incelenmesi gerekmektedir. Geliştirilen devre Şekil 3.44.’de gösterilen aşamalardan oluşmaktadır.



**Şekil 3. 44.** Circom devresinde kullanılan aşamalar.

Devrede giriş (input) ve çıkış (output) parametreleri bulunmaktadır. Giriş parametreleri, gizli (private) ve açık (public) olarak iki gruba ayrılmaktadır. Gizli parametreleri sadece kanıt üreten kişinin bilmesi gerekmekte iken, açık parametreleri herkes görebilmekte ve açık parametreler blokzinciri üzerinde saklanmaktadır. Çıkış parametreleri ise devrenin hesaplama işlemi sonrası elde ettiği sonuçları temsil etmektedir. Giriş ve çıkış parametreleri sıfır bilgi ispatı için temel bir güvenlik sağlamaktadır.

```

1 template ZKVotingCircuit() {
2   // ===== PRIVATE INPUTS (Gizli Girdiler) =====
3   signal input vote;           // Kullanıcının seçtiği aday indeksi (0, 1, 2)
4   signal input secret;         // Kullanıcının gizli anahtarı
5   signal input nonce;          // Commitment için rastgele değer
6   signal input randomness;      // Commitment için ek rastgele değer
7
8   // ===== PUBLIC INPUTS (Genel Girdiler) =====
9   signal input commitment;     // Oy taahhüdü için hash
10  signal input electionId;      // Seçim kimlik numarası
11  signal input candidateCount;  // Toplam aday sayısı
12  signal input startTime;       // Seçim başlangıç zamanı
13  signal input endTime;         // Seçim bitiş zamanı
14  signal input currentTime;     // Mevcut zaman damgası
15  signal input encryptedC;      // Şifrelenmiş oy değeri
16  signal input nullifierHash;   // Çift oy önleme için hash
17
18  // ===== OUTPUTS (Çıktılar) =====
19  signal output valid;           // Oy geçerli mi?
20  signal output nullifierOut;    // Hesaplanan nullifier
  
```

**Şekil 3. 45.** Circom devresinin açık ve gizlik parametreleri.

Geliştirilen devrede kullanılan giriş ve çıkış parametreleri için örnek kodlar Şekil 3.45.'de gösterilmektedir. Devrede kullanılan gizli parametreler Tablo 3.3.'de detaylı bir şekilde açıklanmaktadır.

**Tablo 3. 3.** Circom devresinde kullanılan gizli (private) parametreler.

| Girdi      | Türü   | Açıklama                                                     |
|------------|--------|--------------------------------------------------------------|
| vote       | BigInt | Seçmenin oy kullanmak istediği adayın indeks değeridir.      |
| nonce      | BigInt | Oy gizliliği için kullanılan rastgele bir değerdir.          |
| secret     | BigInt | Seçmenin oy kullanma sırasında belirlenen gizli anahtardır.  |
| randomness | BigInt | Commitment oluşturmak için kullanılan rastgele bir değerdir. |

Devrede kullanılan açık parametreler Tablo 3.4.'de gösterilmektedir. Bu parametreler, kanıtın doğrulanması için akıllı sözleşmelere iletilecektir ve blokzinciri üzerinde kayıtlı kalacaktır.

**Tablo 3. 4.** Circom devresinde kullanılan açık (public) parametreler.

| Girdi          | Türü   | Açıklama                                                                                     |
|----------------|--------|----------------------------------------------------------------------------------------------|
| commitment     | BigInt | Seçmenin oyunu gizlemek ve değiştirilmesini engellemek için kullanılan commitment değeridir. |
| electionId     | BigInt | Seçmenin oy kullandığı seçimin indeks değeridir.                                             |
| candidateCount | BigInt | Oy kullanılan seçimdeki aday sayısıdır.                                                      |
| startTime      | BigInt | Seçimin başlangıç zamanını ifade eder. (Unix timestamp)                                      |
| endTime        | BigInt | Seçimin bitiş zamanını ifade eder. (Unix timestamp)                                          |

Devre kontrolleri içerisinde taahhüt (commitment) doğrulama bölümü bulunmaktadır. Bu bölüm içerisinde çift katmanlı özetleme (hash) bulunması güvenliği artırarak kaba kuvvet (brute-force) saldırılarını zorlaştırır. Şekil 3.46.'da bu işlemin nasıl gerçekleştiği gösterilmektedir. Burada amaç; verilen parametrelerle devre tarafından bir commitment hesaplaması yapılması ve parametre olarak verilmiş olan commitment değeriyle bu değer karşılaştırılmasıdır.

```

1 // ===== 1. COMMITMENT VERIFICATION =====
2 // Çift katmanlı commitment hesaplama
3 // İlk katman: hash(vote, nonce)
4 component innerHasher = Poseidon(2);
5 innerHasher.inputs[0] <= vote;
6 innerHasher.inputs[1] <= nonce;
7
8 // İkinci katman: hash(innerHash, secret, randomness)
9 component outerHasher = Poseidon(3);
10 outerHasher.inputs[0] <= innerHasher.out;
11 outerHasher.inputs[1] <= secret;
12 outerHasher.inputs[2] <= randomness;
13
14 // Hesaplanan commitment ile verilen commitment eşit mi?
15 component commitmentCheck = IsEqual();
16 commitmentCheck.in[0] <= outerHasher.out;
17 commitmentCheck.in[1] <= commitment;

```

**Şekil 3. 46.** Circom devresindeki commitment kontrolü.

Kullanılan oyun geçerli bir adaya verilir verilmediğinin kontrolü yapılmaktadır. Böylece seçmen tarafından geçersiz bir adayın seçilmesi önlenmektedir. Şekil 3.47.'de bu işlemin gerçekleştirilme adımları gösterilmektedir.

```

1 // ===== 2. RANGE PROOF =====
2 // Oy geçerli aralıkta mı? (0 <= vote < candidateCount)
3 component rangeCheck = LessThan(8);
4 rangeCheck.in[0] <= vote;
5 rangeCheck.in[1] <= candidateCount;
6
7 // Aday sayısı pozitif mi?
8 component candidatePositive = GreaterThan(8);
9 candidatePositive.in[0] <= candidateCount;
10 candidatePositive.in[1] <= 0;

```

**Şekil 3. 47.** Circom devresindeki oy ve aday kontrolü.

Oyun seçim süresi içinde kullanılması gerekmektedir. Bu nedenle zaman kontrolleri Şekil 3.48.’de gösterildiği gibi devre içerisinde yapılmaktadır.

```

1 // ===== 3. TIME VALIDATION =====
2 // Zaman aralığı kontrolü: startTime <= currentTime < endTime
3 component timeStart = GreaterEqThan(32);
4 timeStart.in[0] <= currentTime;
5 timeStart.in[1] <= startTime;
6
7 component timeEnd = LessThan(32);
8 timeEnd.in[0] <= currentTime;
9 timeEnd.in[1] <= endTime;
10
11 // Zaman sırası doğru mu? (endTime > startTime)
12 component timeOrder = GreaterThan(32);
13 timeOrder.in[0] <= endTime;
14 timeOrder.in[1] <= startTime;

```

**Şekil 3. 48.** Circom devresindeki zaman doğrulaması.

Mükerrer oy kullanımının engellenmesi amacıyla hash fonksiyonlarından yararlanılmaktadır. Devre içerisinde bunun için benzersiz üretilen nullifier hash değeri kullanılmaktadır. Nullifier üretim ve doğrulama işlemi Şekil 3.49.’da gösterildiği şekilde yapılmaktadır. Devreye verilen parametrelerle devre tarafından bir nullifier hash değeri üretilmektedir. Üretilen bu değer devreye parametre olarak verilen nullifier değeriyle karşılaştırarak doğrulama sürecinin tamamlanması sağlanmaktadır.

```

1 // ===== 4. NULLIFIER GENERATION & VERIFICATION =====
2 // Nullifier hesaplama
3 // İlk katman: hash(secret, electionId)
4 component nullifierInnerHasher = Poseidon(2);
5 nullifierInnerHasher.inputs[0] <= secret;
6 nullifierInnerHasher.inputs[1] <= electionId;
7
8 // İkinci katman: hash(innerHash)
9 component nullifierOuterHasher = Poseidon(1);
10 nullifierOuterHasher.inputs[0] <= nullifierInnerHasher.out;
11
12 // Hesaplanan nullifier ile verilen nullifier eşit mi?
13 component nullifierCheck = IsEqual();
14 nullifierCheck.in[0] <= nullifierOuterHasher.out;
15 nullifierCheck.in[1] <= nullifierHash;
16
17 // Nullifier çıktısı
18 nullifierOut <= nullifierOuterHasher.out;

```

**Şekil 3. 49.** Circom devresindeki nullifier üretimi ve doğrulaması.

Tüm doğrulama süreçlerinin birleştirilerek tek bir sonucun oluşturulması gerekmektedir. Bu sonuca göre tüm süreç başarılı veya başarısız olmuş olacaktır. Bunun için Şekil 3.50.'de gösterilen işlemler gerçekleştirilmektedir.

```

1 // ===== 7. FINAL VALIDATION =====
2 // Tüm kontroller geçti mi?
3
4 // İlk grup: Commitment ve range kontrolleri
5 component and1 = AND();
6 and1.a <= commitmentCheck.out;
7 and1.b <= rangeCheck.out;
8
9 // İkinci grup: Aday sayısı ve zaman kontrolleri
10 component and2 = AND();
11 and2.a <= candidatePositive.out;
12 and2.b <= timeStart.out;
13
14 component and3 = AND();
15 and3.a <= timeEnd.out;
16 and3.b <= timeOrder.out;
17
18 // Üçüncü grup: Nullifier ve şifreleme kontrolleri
19 component and4 = AND();
20 and4.a <= nullifierCheck.out;
21 and4.b <= encryptionCheck.out;
22
23 // Tüm grupları birleştir
24 component finalAnd1 = AND();
25 finalAnd1.a <= and1.out;
26 finalAnd1.b <= and2.out;
27
28 component finalAnd2 = AND();
29 finalAnd2.a <= and3.out;
30 finalAnd2.b <= and4.out;
31
32 component finalAnd3 = AND();
33 finalAnd3.a <= finalAnd1.out;
34 finalAnd3.b <= finalAnd2.out;
35
36 // Son geçerlilik çıktısı
37 valid <= finalAnd3.out;

```

**Şekil 3. 50.** Circom devresinin tüm kontrollerinin sonucu.

Devrenin ana bileşen tanımlaması Şekil 3.51.'de gösterilmektedir. Bu tanımlamayla devre derlenebilir hale getirilmektedir ve açık giriş değerleri tanımlanmaktadır. Bu listede olmayan giriş değerleri gizli parametrelerdir.

```

1 // Ana bileşen - public inputs tanımlaması ile
2 component main {public [commitment, electionId, candidateCount, startTime, endTime,
  currentTime, nullifierHash]} = ZKVotingCircuit();

```

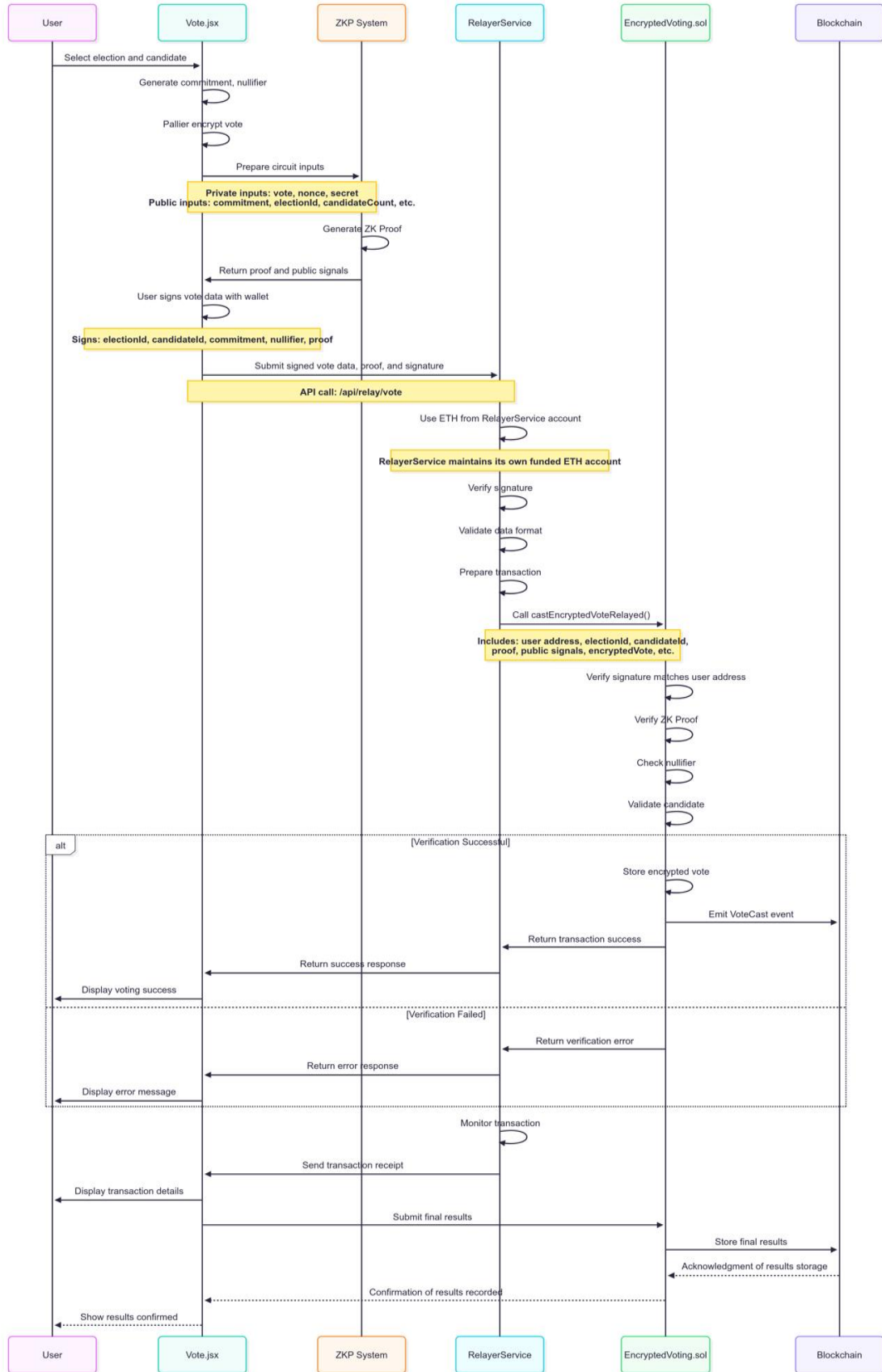
**Şekil 3. 51.** Circom devresinin ana bileşen tanımlanması.

### 3.11. KRİPTOGRAFİK TEKNİKLERİN KULLANIMI

Bu bölüm içerisinde, kriptografik tekniklerin uygulama içerisindeki kullanımı ve işleyişi anlatılmaktadır.

#### 3.11.1. Oy Kullanma İşlemi

Sistem içerisinde kriptografik tekniklerin kullanım amacı oy gizliliğini sağlamak, seçmenlerin doğrulayabilecekleri bir oylamayı oluşturmak ve güvenilir bir altyapı kurabilmektir. Bu oy kullanma işlemi Şekil 3.52.'de gösterilen şekilde gerçekleştirilmektedir.



Şekil 3. 52. Oylama işlemini gösteren sıralama diyagramı (sequence diagram).

### 3.11.1.1. Seçime Katılma

Seçmen, kendisine göre düzenlenmiş panelden seçimleri görüntüleyebilmekte ve oy vermek istediği seçime arayüz üzerinden ulaşabilmektedir. Oyun verileceği sayfa, seçimle ilgili detaylar ve oy pusulası barındırmaktadır.

### 3.11.1.2. Aday Seçimi

Seçmen, oy vermek istediği adayı arayüz üzerinden seçebilmektedir. Seçim gerçekleştirildikten sonra ilgili aday sadece arayüz üzerinden seçilmekte ve seçim geçici olarak kaydedilmektedir. Bu aşamada oy verme işlemi blokzinciri üzerinde tutulmamaktadır.

### 3.11.1.3. Commitment Oluşturulması

Seçmen, aday seçimini gerçekleştirdikten sonra arayüz üzerinde “Generate Commitment” isimli bir buton aktif olmaktadır. Bu butona basılması sonucu ilk önce commitment oluşturma süreci başlatılmış olur.

```

1 async function generateCommitment(vote, nonce, secret, randomness) {
2   // Poseidon hash fonksiyonunu yükle
3   const { buildPoseidon } = await import('circomlibjs');
4   const poseidon = await buildPoseidon();
5
6   // BigInt
7   const voteValue = BigInt(vote);
8   const nonceValue = BigInt('0x' + nonce);
9   const secretValue = BigInt('0x' + secret);
10  const randomnessValue = BigInt('0x' + randomness);
11
12  // İki katmanlı hash hesaplama
13  // İlk katman: vote ve nonce için hash üret
14  const innerHash = poseidon([voteValue, nonceValue]);
15  const innerHashResult = poseidon.F.toString(innerHash);
16
17  // İkinci katman: İlk hash, secret ve randomness için hash üret
18  const outerHash = poseidon([
19    BigInt(innerHashResult),
20    secretValue,
21    randomnessValue
22  ]);
23
24  // Sonuç commitment değeri
25  return poseidon.F.toString(outerHash);
26 }

```

**Şekil 3. 53.** Commitment oluşturulmasına yönelik örnek JavaScript fonksiyonu.

Commitment oluşturulmasını temsil eden örnek JavaScript fonksiyonu Şekil 3.53.’de gösterilmektedir. Geliştirilen uygulama içerisinde bu bölüm karmaşık olduğundan burada temsili bir gösterim yapılmaktadır. Şekil 3.53.’da geliştirilen fonksiyonun, oy kullanma sayfasındaki kullanımı Şekil 3.54.’deki JavaScript fonksiyonuyla temsili olarak gösterilmektedir. Burada ilk önce commitment üretimi için gereken rastgele değerler üretilmekte ve daha sonra Şekil 3.53.’deki commitment üretim fonksiyonu çağrılmaktadır. Bu fonksiyona parametrelerin doğru bir şekilde iletilmesiyle oluşturulacak değer doğru bir commitment değerini temsil edecektir.

```

1 const generateVoteCommitment = async () => {
2
3   // Rastgele degerler olustur
4   const voteNonce = getRandomHex(8);
5   const voteRandomness = getRandomHex(8);
6   const secret = generateDeterministicSecret(address);
7
8
9   // Commitment olustur
10  const commitmentValue = await generateCommitment(
11    selectedCandidate,
12    voteNonce,
13    secret,
14    voteRandomness
15  );
16
17  setCommitment(commitmentValue);
18
19 };

```

Şekil 3. 54. Commitment üreten JavaScript fonksiyonu örneği

#### 3.11.1.4. Nullifier Oluşturulması

Commitment oluşturma sürecinin devamında, seçmenin çift oy kullanmasının önüne geçmek için nullifier oluşturulma süreci başlatılmaktadır. Şekil 3.55.'de bu işlem için kullanılan temsili JavaScript fonksiyonu gösterilmektedir.

```

1 const generateNullifier = async (secret, electionId) => {
2   // İlk katman: secret ve electionId değerleri
3   const innerHash = await poseidon([BigInt('0x${secret}'), BigInt(electionId)]);
4
5   // İkinci katman: iç hash değerini tekrar
6   const nullifier = await poseidon([innerHash]);
7
8   return nullifier.toString();
9 };

```

Şekil 3. 55. Nullifier oluşturulmasına yönelik örnek JavaScript fonksiyonu.

Şekil 3.55.'deki fonksiyon, oy kullanma sayfasında Şekil 3.56.'da gösterildiği gibi kullanılmaktadır. Parametrelerin fonksiyona doğru gönderilmesi neticesinde geçerli bir nullifier değeri elde edilecektir.

```

1 // Nullifier üret (seçim ID'siyle)
2 let nullifier;
3 try {
4   nullifier = await generateNullifier(secret, electionId);
5   console.log('Generated nullifier:', nullifier);
6 } catch (nullifierError) {
7   console.error('Error generating nullifier:', nullifierError);
8   toast.error('Failed to generate nullifier: ${nullifierError.message}');

```

Şekil 3. 56. Nullifier oluşturan JavaScript kodu örneği.

#### 3.11.1.5. Oyun Şifrelenmesi

Commitment ve nullifier oluşturma sürecinin başarıyla sonuçlanması neticesinde, Paillier şifreleme algoritmasıyla oyun şifrelenmesi için gerekli sürecin başlatılması

gerekmektedir. Bu süreç içerisinde şifreleme işlemi için kullanılan fonksiyonlar Şekil 3.57.'de gösterilmektedir. Şifreleme işlemi yapılabilmesi için sistemde oylama işleminin temsil edildiği formatın bilinmesi gerekmektedir. Sistem üzerindeki her oy, aday sayısı uzunluğunda bir vektör olarak temsil edilmektedir. Bu durumda 10 adayın bulunduğu bir seçimde 10 elemanlı bir vektör oluşturulmaktadır. Her indeks sırasıyla bir adayı temsil etmektedir. Bir seçmenin oy verdiği adayın indeks değeri şifrelenmiş olarak 1 değerini ve oy vermediği diğer adayların indeks değeri şifrelenmiş 0 değerini taşımaktadır. Her adayın indeks değerine yazılacak ifade şifrelenirken; her bir işlem için rastgele oluşturulan  $r$  değeri kullanılmaktadır.

```

1 // Frontend: Paillier şifreleme
2 function encrypt(message, publicKey, r = null) {
3   const { n, g } = publicKey;
4   const n2 = n * n;
5
6   // Rastgele r değeri (yoksa üret)
7   r = r || generateRandomBigInt(n);
8
9   // c = (g^m * r^n) mod n^2
10  const gm = modExp(g, BigInt(message), n2);
11  const rn = modExp(r, n, n2);
12  const c = modMul(gm, rn, n2);
13
14  return { c, r };
15 }
16
17 // Modüler us alma (BigInt optimized)
18 function modExp(base, exp, mod) {
19   if (mod === 1n) return 0n;
20
21   let result = 1n;
22   base = base % mod;
23
24   while (exp > 0n) {
25     if (exp % 2n === 1n) {
26       result = (result * base) % mod;
27     }
28     base = (base * base) % mod;
29     exp = exp >> 1n;
30   }
31
32   return result;
33 }
34
35 // Modüler carpma
36 function modMul(a, b, mod) {
37   return ((a % mod) * (b % mod)) % mod;
38 }

```

Şekil 3. 57. Paillier algoritmasıyla şifreli oy üreten JavaScript fonksiyonu.

Şifreleme fonksiyonun kullanımı için gerekli parametre değerlerinin verilmesi gerekmektedir. Örnek kullanım Şekil 3.58.'de gösterilmektedir.

```

1 // Örnek kullanım
2 const publicKey = {
3   n: BigInt("XXXXXXXX"),
4   g: BigInt("YYYYYYYY")
5 };
6
7 // Oy Şifreleme
8 const vote = 1; // Aday ID
9 const encrypted = encrypt(vote, publicKey);
10 console.log("Encrypted vote:", encrypted.c.toString());

```

Şekil 3. 58. Şifreleme işlemini gerçekleştiren JavaScript örneği.

Bu şifreleme işleminin gerçekleşmesinde kullanılan vektörün oluşturulması temsili olarak Şekil 3.59.'da gösterilmektedir. Vektör oluştururken dikkat edilmesi gereken noktalar

bulunmaktadır. Vektör üzerinde, seçmenin oy kullandığı veya kullanmadığı adaylar için şifreli değerler bulunduğundan, bu değerlerin açığa çıkmaması gerekmektedir. Bu nedenle, vektör oluşturulurken her adayın şifreli oy değeriyle rastgele değerin toplanması sağlanmaktadır.

```

1 // Tum adaylar icin sifreli oylar hazirlama - Vector Encryption
2 export const prepareEncryptedVotesForCandidates = async (
3   inputs,
4   publicKey,
5   selectedCandidateIndex,
6   candidateCount,
7   progressCallback = null
8 ) => {
9
10  const encryptedVotes = [];
11
12  for (let i = 0; i < candidateCount; i++) {
13    // Secilan aday icin 1, digerleri icin 0
14    const voteValue = (i === selectedCandidateIndex) ? 1 : 0;
15
16    // Her aday icin farkli rastgele deger
17    const randomness = inputs.paillierR + BigInt(i + 1);
18
19    if (progressCallback) {
20      const progress = 0.1 + (0.8 * (i + 1) / candidateCount);
21      progressCallback({
22        status: 'Candidate ${i + 1}/${candidateCount} encryption...',
23        progress
24      });
25    }
26
27    // Paillier sifreleme
28    const encryptedVote = await encrypt2048Bit(
29      voteValue,
30      publicKey,
31      randomness
32    );
33
34    // Format ve store
35    const formattedVote = {
36      candidateId: i + 1,
37      voteValue: voteValue,
38      encryptedValue: encryptedVote.c,
39      randomness: encryptedVote.r,
40      data: formatCiphertext(encryptedVote).data,
41      hash: formatCiphertext(encryptedVote).hash
42    };
43
44    encryptedVotes.push(formattedVote);
45  }
46
47  return encryptedVotes;
48 };

```

Şekil 3. 59. Oy vektörünü üreten JavaScript fonksiyonu.

Bu süreçlerin tamamlanmasıyla birlikte oyun hazırlanma aşaması tamamlanacaktır. Bunun sonucunda arayüz üzerinde seçmene “Cast Vote” isimli buton gösterilecektir.

#### 3.11.1.6. Sıfır Bilgi İspatı Kanıtının Oluşturması

“Cast Vote” isimli butona basıldığında sıfır bilgi ispatı için kanıt üretim süreci başlatılacaktır. Bu sürecin başarıyla gerçekleştirilebilmesi için daha öncesinde Circom devresinin derlenmesi ve derleme sonucunda oluşturulan dosyaların web uygulaması içerisinde bulunması gerekmektedir. Bu dosyalar Şekil 3.60.’da gösterilmektedir.

```

web/public/zk/
├── ZKVotingCircuit.wasm      # Devre WASM dosyası
├── ZKVotingCircuit_0000.zkey # Proving key
└── verification_key.json    # Doğrulama anahtarı

```

Şekil 3. 60. Circom devresinin derlenmesi sonrası oluşturulan dosyalar.

Kanıtın oluşturulması aşamasında ilgili JavaScript dosyasında Şekil 3.61.’deki bağımlılıkların kullanılması gerekmektedir.

```
import * as snarkjs from 'snarkjs';
import { poseidon } from 'circomlib';
```

**Şekil 3. 61.** Kanıt oluşturma süreci JavaScript bağımlılıkları.

Bir sıfır bilgi ispatı kanıtının oluşturulabilmesi için açık ve gizli giriş değerleri kullanılır. Bu değerlerin oluşturulabilmesi için gerekli fonksiyonların önceden hazırlanmış olması gerekmektedir. Bu nedenle, önceki aşamalarda bu fonksiyonlarla ilgili işlemler tamamlanmaktadır. Böylece kanıt oluşturmak için kullanılacak gizli ve açık giriş değerlerinin hazırlanması süreci tamamlanabilmektedir. Kanıt üretme süreci için öncelikle commitment ve nullifier değerlerinin elde edilmesi için işlemler gerçekleştirilmektedir. Bu süreç sona erdiğinde devrenin tüm açık ve gizli giriş değerleri bir araya getirilmektedir. Şekil 3.62.'de gösterildiği gibi bu aşamada 4 gizli ve 6 açık giriş değeri bulunmaktadır. Daha sonrasında devre girdilerinin hazırlandığı ana fonksiyona değerlerin döndürülmesi işlemi gerçekleştirilmektedir. Bu süreçteki işlemlerin örnek kodu Şekil 3.62.'de gösterilmektedir. Buradaki örnek referans amaçlıdır.

```
1 const prepareCircuitInputs = async (voteData, electionData) => {
2   const currentTime = Math.floor(Date.now() / 1000);
3
4   // Commitment ve nullifier hesaplama
5   const commitment = await generateCommitment(
6     voteData.candidateId, voteData.nonce,
7     voteData.secret, voteData.randomness
8   );
9
10  const nullifier = await generateNullifier(
11    voteData.secret, electionData.electionId
12  );
13
14  // FLAT STRUCTURE
15  const circuitInputs = {
16    // Private inputs (4 total)
17    vote: BigInt(voteData.candidateId),
18    nonce: BigInt('0x' + voteData.nonce),
19    secret: BigInt('0x' + voteData.secret),
20    randomness: BigInt('0x' + voteData.randomness),
21
22    // Public inputs (6 total)
23    commitment: BigInt(commitment),
24    electionId: BigInt(electionData.electionId),
25    candidateCount: BigInt(electionData.candidateCount),
26    startTime: BigInt(electionData.startTime),
27    endTime: BigInt(electionData.endTime),
28    currentTime: BigInt(currentTime)
29  };
30
31  return {
32    circuitInputs,
33    metadata: {
34      commitment: commitment.toString(),
35      nullifier: nullifier.toString(),
36      timestamp: currentTime
37    }
38  };
39 };
```

**Şekil 3. 62.** Devrenin giriş değerlerinin hazır hale getirilmesi.

Giriş değerlerinin oluşturulması kanıt üretme işlemi için önemlidir. Giriş değerlerinin oluşturulması kanıt üretme sürecinin ilk aşamasını oluşturmaktadır. Tüm süreç Şekil 3.63.'de ayrıntılı olarak örnek çalışma üzerinden gösterilmektedir.

```

1 async function generateProof(voteData, electionData) {
2   console.log("Starting ZK-Proof flow...");
3
4   // 1. Circuit inputs hazırla
5   console.log("Preparing circuit inputs...");
6   const { circuitInputs, metadata } = await prepareCircuitInputs(voteData,
7     electionData);
8
9   // 2. Proof oluşturun
10  console.log("Generating proof...");
11  const proofGenerationTime = performance.now();
12  const { proof, publicSignals } = await snarkjs.groth16.fullProve(
13    circuitInputs,
14    '/zk/ZkBallotPaillierCircuit.wasm',
15    '/zk/ZkBallotPaillierCircuit_0000.zkey'
16  );
17  console.log('Proof generated in ${performance.now() - proofGenerationTime}ms');
18
19  // 3. Verification
20  console.log("Verifying proof locally...");
21  const verificationKey = await fetch('/zk/verification_key.json')
22    .then(res => res.json());
23
24  const isValid = await snarkjs.groth16.verify(
25    verificationKey,
26    publicSignals,
27    proof
28  );
29
30  if (!isValid) {
31    throw new Error("Local proof verification failed");
32  }
33
34  // 4. Contract formatına donustur
35  console.log("Formatting proof for contract submission...");
36  const formattedProof = {
37    a: [proof.pi_a[0].toString(), proof.pi_a[1].toString()],
38    b: [
39      [proof.pi_b[0][1].toString(), proof.pi_b[0][0].toString()],
40      [proof.pi_b[1][1].toString(), proof.pi_b[1][0].toString()]
41    ],
42    c: [proof.pi_c[0].toString(), proof.pi_c[1].toString()]
43  };
44
45  return {
46    proof: formattedProof,
47    publicSignals: publicSignals.map(s => s.toString()),
48    commitment: metadata.commitment,
49    nullifier: metadata.nullifier,
50    isValid
51  };
52 }

```

**Şekil 3. 63.** Sıfır bilgi ispatı kanıtı oluşturulmasına yönelik JavaScript fonksiyon örneği.

Bu süreç yukarıda anlatılan adımlarla birlikte toplam beş aşamadan oluşmaktadır:

1. Devre Girdilerinin Hazırlanması: Devrenin kanıt üretmek için kullanacağı tüm girdilerin üretilmesi ve bu girdilerin doğru formata hazır hale getirilmesi aşamasıdır.
2. Witness Hesaplama: Devrenin hesaplama işlevselliğini çalıştırmak için gerçekleştirilen bir aşamadır ve kanıt oluşturmanın temelidir. Üretilecek kanıtın buradaki temel kurallara uygunluğu kontrol edilecektir.
3. Kanıt Oluştur: Gerekli giriş bilgileriyle sıfır bilgi ispatı kanıtının üretilmesini kapsamaktadır.

4. Kanıtın Doğrulanması: Üretilen kanıtın geçerlilik kontrolünün web arayüzü üzerinden yapılmasıdır. Bu sadece kanıtın doğru üretilip üretilmediği kontrol etmek içindir.
5. Kanıt Formatının Değiştirilmesi: Üretilen kanıtın doğru formata dönüştürülmesinin gerçekleştirildiği aşamadır.

Kanıtın doğrulanması aşamasında kullanılan örnek çalışma Şekil 3.64.'de gösterilmektedir. Akıllı sözleşme üzerinde doğrulama işlemi yapılmadan önce JavaScript ile yapılan doğrulama işlemini kapsamaktadır. Böylece kanıt, akıllı sözleşmeye gönderilmeden önce daha az maliyetle bir doğrulama işlemine tabi tutulur ve gereksiz gaz kullanımından kaçınılır. Kanıtın web arayüzü üzerinden doğrulanması yapılmak istendiğinde kullanılacak ilk işlem bu fonksiyonu kullanmaktır.

```

1 class ZKProofService {
2   constructor() {
3     // Verification key - public klasorunden yukleniyor
4     this.verificationKey = null;
5     this.loadVerificationKey();
6   }
7
8   async loadVerificationKey() {
9     const response = await fetch('/zk/verification_key.json');
10    this.verificationKey = await response.json();
11    console.log('Verification key loaded successfully');
12  }
13
14  async verifyProof(proof, publicSignals) {
15    // Girdileri normalize et
16    const normalizedSignals = publicSignals.map(signal =>
17      typeof signal === 'string' ? signal : signal.toString()
18    );
19
20    // SnarkJS formatına proof donustur
21    const snarkjsProof = {
22      pi_a: [proof.pi_a[0], proof.pi_a[1], "1"],
23      pi_b: [
24        [proof.pi_b[0][1], proof.pi_b[0][0]],
25        [proof.pi_b[1][1], proof.pi_b[1][0]],
26        ["1", "0"]
27      ],
28      pi_c: [proof.pi_c[0], proof.pi_c[1], "1"]
29    };
30
31    // SnarkJS ile proof dogrula
32    const isValid = await snarkjs.groth16.verify(
33      this.verificationKey,
34      normalizedSignals,
35      snarkjsProof
36    );
37
38    return {
39      isValid,
40      timestamp: Date.now(),
41      proofHash: this.calculateProofHash(proof, publicSignals)
42    };
43  }
44
45  calculateProofHash(proof, publicSignals) {
46    const data = JSON.stringify({ proof, publicSignals });
47    return ethers.keccak256(ethers.toUtf8Bytes(data));
48  }
49 }

```

**Şekil 3. 64.** JavaScript ile sıfır bilgi ispatı kanıtının doğrulanması.

Bir seçim için oy kullanıldığında kanıt üretme işlemi için Şekil 3.65.'de gösterilen örnek kod yapısı işletilir. Burada oy ver butonuna basıldığında işleyecek süreç basitleştirilerek gösterilmektedir.

```

1 // Kullanım Örneği
2 async function handleVoteSubmission() {
3   try {
4     // Vote data
5     const voteData = {
6       candidateId: 1,
7       nonce: "123456", // hex string (0x olmadan)
8       secret: "789012", // hex string (0x olmadan)
9       randomness: "345678" // hex string (0x olmadan)
10    };
11
12    // Election data
13    const electionData = {
14      electionId: 123,
15      candidateCount: 5,
16      startTime: new Date('2025-06-01T10:00:00Z'),
17      endTime: new Date('2025-06-02T18:00:00Z')
18    };
19
20    // Proof oluştur
21    const proofData = await generateProof(voteData, electionData);
22
23    console.log("Ready for blockchain submission:", {
24      proof: proofData.proof,
25      publicSignals: proofData.publicSignals,
26      commitment: proofData.commitment,
27      nullifier: proofData.nullifier
28    });
29
30    // Bu proof blokzincire gönderiliyor
31    return proofData;
32  } catch (error) {
33    console.error("Vote submission failed:", error);
34  }
35 }
36

```

Şekil 3. 65. Oy kullanımı ve sıfır bilgi ispatı kanıtının blokzincire gönderilmesi.

#### 3.11.1.7. Akıllı Sözleşme Etkileşimi

Akıllı sözleşme etkileşimi, oyun şifrelenmesi ve kanıtın üretilmesi sonucunda oluşan bilgilerin akıllı sözleşmeye iletilme sürecini ifade etmektedir. Gerekli parametrelerin akıllı sözleşmeye iletilmesi için web uygulaması üzerinden seçmen kimliğinin doğrulanması gerekmektedir.

Prototip geliştirilirken ilk önce doğrudan arayüzle akıllı sözleşme bağlantısı yapılmıştır. Tüm işlemler iki servis arasında gerçekleştirilmiştir. Bu çalışmanın başarılı olması neticesinde relay servisi geliştirilmiş ve arayüzle akıllı sözleşme arasında çalışacak şekilde hazırlanmıştır. Arayüz ve akıllı sözleşmenin doğrudan iletişim kurması sonucu çalışan sistemde; ortaya çıkan gaz maliyeti, seçmenin cüzdanındaki bakiye ile ödenmiştir. Bu bakiye, seçim başlatılırken seçmene oy kullanması için gönderilen Ether ile oluşturulmuştur. Oyun, doğrudan akıllı sözleşmeye gönderilmesinde kullanılan örnek JavaScript kodu Şekil 3.66.'da görülebilir.

```

1 const tx = await encryptedContract.castEncryptedVote(
2   electionId,
3   proof.a,
4   proof.b,
5   proof.c,
6   input,
7   encryptedVote,
8   commitment,
9   nullifier
10 );
11
12 const receipt = await tx.wait();
13 console.log(receipt);
14
15 return {
16   success: true,
17   transactionHash: receipt.transactionHash,
18   blockNumber: receipt.blockNumber
19 }

```

**Şekil 3. 66.** Oyun blokzincire kaydedilmesi için akıllı sözleşmeye doğrudan gönderilmesi.

Sistemin ikinci aşamasında, gaz maliyetinin sistem tarafından karşılanması amaçlandığından relayer servisi geliştirilmiştir. Bu aşamada, oy önce relayer servisine gönderilmiş ve oradan da akıllı sözleşmeye iletilmiştir. Relayer servisi, arayüz ve akıllı sözleşme arasında bir köprü görevi üstlenmiştir. Oyun, relayer servisine gönderilmesinde kullanılan örnek JavaScript kodu Şekil 3.67.’de görülebilir.

```

1 // İmza doğrula
2 const message = `Vectorial Vote - ElectionID: ${voteData.electionId}, Commitment: ${
3   parsedEncryptedVote.commitment}, Nullifier: ${parsedEncryptedVote.nullifier},
4   Voter: ${userAddress.toLowerCase()}`;
5
6 const recoveredAddress = ethers.verifyMessage(message, signature);
7
8 if (recoveredAddress.toLowerCase() !== userAddress.toLowerCase()) {
9   return res.status(401).json({ error: 'Invalid signature' });
10 }
11
12 // İşlemi gönder
13 const tx = await contract.castVectorialVoteOnBehalf(
14   ethers.toBigInt(voteData.electionId),
15   a, b, c,
16   inputs,
17   encryptedVotes,
18   commitment,
19   nullifier,
20   userAddress,
21   signature
22 );
23
24 const receipt = await tx.wait();
25
26 res.json({
27   success: true,
28   transactionHash: receipt.hash
29 });

```

**Şekil 3. 67.** Oyun blokzincire kaydedilmesi için relayer servisine gönderilmesi.

### 3.11.2. Oy Doğrulama Süreci

Önceki adımların tamamlanmasıyla birlikte akıllı sözleşmeye; seçim bilgileri, sıfır bilgi ispatı kanıtı ve kanıt için gerekli açık parametreler iletilir. Bunun sonucunda oyun doğrulama süreci başlatılır.

### 3.11.2.1. Commitment Kontrolü

İlk önce gönderilen commitment değerinin, Şekil 3.68.'de gösterilen solidity kodu ile kontrolü sağlanmaktadır. Bu ifade yalnızca örnek bir gösterimdir; oylama sistemi içerisinde çeşitli güncellemeler ve farklılıklar mevcuttur.

```

1 contract VotingSystem {
2   // Kullanıcı commitmentları
3   mapping(uint256 => mapping(address => bytes32)) private voteCommitments;
4
5   // Commitment kaydetme fonksiyonu
6   function storeCommitment(
7     uint256 electionId,
8     bytes32 commitment
9   ) public {
10    require(!hasVoted[electionId][msg.sender], "Already voted");
11    voteCommitments[electionId][msg.sender] = commitment;
12  }
13
14  // ZK-proof doğrulamasında commitment kontrolü
15  function verifyCommitment(
16    uint256 electionId,
17    address voter,
18    bytes32 commitment,
19    uint256[] calldata publicInputs
20  ) internal view returns (bool) {
21    // publicInputs[0] ZK-proof'tan gelen commitment değeri
22    bytes32 proofCommitment = bytes32(publicInputs[0]);
23
24    // Kaydedilen commitment ile karşılaştır
25    return voteCommitments[electionId][voter] == proofCommitment;
26  }
27 }

```

Şekil 3. 68. Akıllı sözleşme tarafından commitment değerinin doğrulanması.

### 3.11.2.2. Nullifier Kontrolü

İkinci aşama olarak nullifier değerinin, Şekil 3.69.'da temsili olarak gösterilen solidity kodu ile kontrolü sağlanmaktadır.

```

1 // Solidity: Nullifier kontrolü
2 contract VotingSystem {
3   // Kullanılmış nullifier değerleri
4   mapping(uint256 => mapping(uint256 => bool)) public usedNullifiers;
5
6   // Nullifier kontrolü ve kayıt
7   function processNullifier(
8     uint256 electionId,
9     uint256 nullifier
10  ) internal returns (bool) {
11    // Nullifier daha önce kullanılmış mı kontrol et
12    if (usedNullifiers[electionId][nullifier]) {
13      return false;
14    }
15
16    // Yeni nullifier'ı kaydet
17    usedNullifiers[electionId][nullifier] = true;
18    return true;
19  }
20 }

```

Şekil 3. 69. Akıllı sözleşme tarafından nullifier değerinin kontrolü.

### 3.11.2.3. Sıfır Bilgi İspatı Kanıtının Doğrulanması

Üçüncü aşama olarak sıfır bilgi ispatı kanıtının doğrulanması için süreç başlatılmaktadır. Bu işlem için akıllı sözleşmeye, kanıtın ve gerekli açık parametrelerin gönderilmesi gerekmektedir. Ayrıca gönderilen kanıt, belirlenen formatta iletilmelidir. zk-SNARK kanıtları, akıllı sözleşmeye üç ana array olarak ulaştırılmaktadır. Server servisinde,

doğrulama işlemlerinden sorumlu olan verifier.sol adlı bir akıllı sözleşme yer almaktadır. Bu sözleşme, Circom devresinin derlenmesiyle oluşturulmuş olup, üretilen kanıtların akıllı sözleşme üzerinde doğrulanmasında kullanılmaktadır. Şekil 3.70.'de sıfır bilgi ispatı kanıtının doğrulanma süreci örnek Solidity kodu ile gösterilmektedir.

```

1 contract VotingSystem {
2
3     function verifyEncryptedVote(
4         uint256 electionId,
5         uint[2] memory a,
6         uint[2][2] memory b,
7         uint[2] memory c,
8         uint[6] memory input,
9         string[] memory encryptedVoteVector,
10        bytes32 commitment,
11        uint256 nullifier,
12        address voter,
13        bytes memory signature
14    ) public returns (bool) {
15
16        // 3. ZK kanıtını doğrula
17        bool isValid = false;
18        try verifier.verifyProof(a, b, c, input) returns (bool result) {
19            isValid = result;
20            if (!isValid) {
21                revert ProofVerificationFailed();
22            }
23        } catch {
24            revert ProofVerificationFailed();
25        }
26
27        return true;
28    }
29
30 }

```

**Şekil 3. 70.** Akıllı sözleşme tarafından sıfır bilgi ispatı kanıtının doğrulanması.

### 3.11.3. Oyun Saklanması

Doğrulama işleminin gerçekleştirilmesi sonrasında oyların blokzincire kaydedilmesi gerekmektedir. Oyların saklanma süreci şu aşamalardan oluşmaktadır:

1. Şifreli Oyun Kaydedilmesi: Oluşturulan oy vektörlerinin kaydedilmesi sağlanır.
2. Nullifier Kaydedilmesi: Oluşturulan nullifier değerinin kaydedilmesi sağlanır.
3. Oy Sayısı Güncelleme: Kullanılan oy sayısının güncellenmesi sağlanır.

#### 3.11.3.1. Oy Saklama Yapısı

Oylama işleminden sorumlu akıllı sözleşme aracılığıyla Şekil 3.71.'de gösterilen kayıtlar tutulmaktadır. Bu kayıtlar oy saklama yapısını temsil etmektedir. Prototip çalışması olduğu için oyların bireysel olarak şifresinin çözülmesi için çalışmalar da gerçekleştirilmekte ve buna uygun yapılarda tutulmaktadır.

```

1 // EncryptedVoting.sol kontratında
2 // electionId => voter => encryptedVoteVector (string array)
3 mapping(uint256 => mapping(address => string[])) private voterEncryptedVectors;
4
5
6 // electionId => candidate count for vector length validation
7 mapping(uint256 => uint256) private electionCandidateCount;
8
9
10 // Aday bazında deşifrelenmiş sonuçlar
11 mapping(uint256 => uint256[]) public decryptedResults;
12
13 // Toplam deşifrelenmiş oy sayısı
14 mapping(uint256 => uint256) public decryptedTotalVotes;
15
16 // Seçimin deşifre edilip edilmediği
17 mapping(uint256 => bool) public electionDecrypted;

```

Şekil 3. 71. Şifrelenmiş oy saklama yapısı.

### 3.11.3.2. Oy Saklama İşlemi

Oy saklama sürecinde yapılan tüm işlemlerin örnek kodu Şekil 3.72.'de gösterilmektedir.

```

1 function castEncryptedVote(
2     uint256 electionId,
3     uint[2] memory a,
4     uint[2][2] memory b,
5     uint[2] memory c,
6     uint[6] memory input,
7     string[] memory encryptedVoteVector,
8     bytes32 commitment,
9     uint256 nullifier,
10    address voter,
11    bytes memory signature
12 ) public returns (bool) {
13
14     // Signature doğrulama
15     address recoveredSigner = recoverSignerFromMessage(message, signature);
16     require(recoveredSigner == voter, "Invalid signature");
17
18     // Temel kontroller
19     if (hasVoted[electionId][voter]) {
20         revert AlreadyVoted();
21     }
22     if (usedNullifiers[electionId][nullifier]) {
23         revert NullifierAlreadyUsed(nullifier);
24     }
25
26     // ZK Proof doğrulama
27     bool proofValid = verifier.verifyProof(a, b, c, input);
28     if (!proofValid) {
29         revert ProofVerificationFailed();
30     }
31
32     // Vektor uzunluğu kontrolü
33     uint256 expectedLength = electionCandidateCount[electionId];
34     if (encryptedVoteVector.length != expectedLength) {
35         revert InvalidVoteVector(
36             electionId,
37             expectedLength,
38             encryptedVoteVector.length
39         );
40     }
41
42     // Şifreli vektörü kaydet
43     voterEncryptedVectors[electionId][voter] = encryptedVoteVector;
44
45     // Durum güncellemeleri
46     hasVoted[electionId][voter] = true;
47     usedNullifiers[electionId][nullifier] = true;
48     electionNullifiers[electionId].push(nullifier);
49     totalVotesInElection[electionId]++;
50     voteCommitments[electionId][voter] = commitment;
51
52     emit VectorialVoteCasted(
53         electionId,
54         voter,
55         commitment,
56         nullifier,
57         encryptedVoteVector.length
58     );
59     return true;
60 }

```

Şekil 3. 72. Oyun kaydedilmesi için kullanılan örnek Solidity kodu.

### 3.11.4. Seçim Sonuçlarının Hesaplanması

Seçim sonuçlarının hesaplanmasını anlayabilmek için oyların saklanmasını incelemek gerekmektedir. Oylar, akıllı sözleşmeler vasıtasıyla şifrelenmiş bir şekilde tutulur. Seçim sonuçları açıklanacağı zaman, vektörel olarak şifrelenmiş oyların tamamı alınarak homomorfik toplama işlemi gerçekleştirilir. Bu işlem sonucunda şifreli toplam bir vektör oluşmaktadır. Bu vektörün üzerindeki değerlerin şifresi çözülerek sonuçların açıklanması sağlanmaktadır. Bu işlem ağ dışında (off-chain) gerçekleştirilir. Çünkü, akıllı sözleşme üzerinde 2048 bitlik işlemler gerçekleştirildiğinde, gaz kullanım limiti aşılmaktadır.

#### 3.11.4.1. Homomorfik Oy Toplama

Bir seçmenin oyu şifrelenmiş bir şekilde tutulmaktadır. Seçim sonuçlarının açıklanması gerektiğinde oylar, homomorfik olarak toplanmaktadır. 10 adayın bulunduğu bir seçimde, her oy 10 elemanlı bir vektör olarak temsil edilmektedir. Bu vektörde, 0. indeks 1. adayı ifade eder ve benzer şekilde her indeks ilgili adaya karşılık gelir. Tüm oylarda, aynı indekslerin homomorfik toplanmasıyla her aday için şifreli oy toplamı elde edilir. Sonuç olarak oluşan 10 elemanlı şifreli vektör, adayların almış oldukları oy sayılarının şifreli halini temsil eder. Dolayısıyla, adaylara ait şifreli oyların toplandığı homomorfik işlem adımı Şekil 3.73.'de ayrıntılı olarak gösterilmektedir. Bu sayede, birden fazla şifrelenmiş oy vektörü üzerinde toplama işlemi gerçekleştirilerek toplu oy sayımı yapılabilir.

```

1  /**
2   * Birden fazla şifreli oy vektörünü toplar
3   * @param {Array} encryptedVectors - Şifreli oy vektörleri
4   * @param {Object} publicKey - Public key
5   * @returns {Array} Toplanmış şifreli vektör
6   */
7  function homomorphicSum(encryptedVectors, publicKey) {
8      const { n2 } = publicKey;
9      const vectorLength = encryptedVectors[0].length;
10     const voterCount = encryptedVectors.length;
11
12     const sumVector = [];
13
14     // Her aday pozisyonu için ayrı ayrı toplama
15     for (let candidateIndex = 0; candidateIndex < vectorLength; candidateIndex++) {
16
17         // İlk seçmenin oyuyla başla
18         let encryptedSum = BigInt(encryptedVectors[0][candidateIndex]);
19
20         // Diğer seçmenlerin oylarını ekle
21         for (let voterIndex = 1; voterIndex < voterCount; voterIndex++) {
22             const encryptedVote = BigInt(encryptedVectors[voterIndex][candidateIndex]);
23             encryptedSum = homomorphicAdd(encryptedSum, encryptedVote, n2);
24         }
25
26         sumVector.push(encryptedSum.toString());
27     }
28
29     return sumVector;
30 }
31
32 
```

Şekil 3. 73. Birden fazla şifreli oy vektörünün homomorfik toplanma işlemi.

Şekil 3.74.'de homomorfik oy toplama işleminin nasıl gerçekleştirildiğini gösteren örnek bir JavaScript kodu sunulmaktadır.

```

1 /**
2  * İki şifreli değeri homomorfik olarak toplar
3  * @param {BigInt} ciphertext1 - ilk şifreli değer
4  * @param {BigInt} ciphertext2 - ikinci şifreli değer
5  * @param {BigInt} modul
6  * @returns {BigInt} Toplam şifreli değer
7  */
8 function homomorphicAdd(ciphertext1, ciphertext2, n2) {
9     return (ciphertext1 * ciphertext2) % n2;
10 }

```

Şekil 3. 74. Homomorfik oy toplama işlemi.

#### 3.11.4.2. Şifre Çözme ve Sonuç Hesaplama

Bu aşama, aday bazında vektörel olarak saklanan şifreli seçim sonuçlarının aday bazında çözülmesi için yapılması gereken işlemleri kapsamaktadır. Böylece her adayın aldığı oy sayının ortaya çıkartılması sağlanmaktadır. Gizli anahtar (private key), oyların açılabilmesi için gereklidir. Yapılan işlemlerin örnek kodları Şekil 3.75.'de gösterilmektedir.

```

1 /**
2  * Şifreli vektörü desifreler
3  * Her indeks bir adayın toplam oyunu temsil eder
4  */
5 decryptVector(encryptedSums) {
6     try {
7         const results = [];
8
9         for (let i = 0; i < encryptedSums.length; i++) {
10             const encryptedSum = BigInt(encryptedSums[i]);
11
12             // Paillier desifreleme: m = L(c^λ mod n^2) * μ mod n
13             const decrypted = this.decrypt(encryptedSum);
14             const voteCount = Number(decrypted);
15
16             results.push(voteCount);
17         }
18
19         return results;
20     } catch (error) {
21         console.error('Vector decryption failed:', error);
22         throw error;
23     }
24 }
25 }

```

Şekil 3. 75. Seçim sonuçları için vektörün şifre çözümü.

Bu işlemler sırasında kullanılan şifre çözme fonksiyonu örneği Şekil 3.76.'da gösterilmektedir.

```

1  /**
2   * Paillier Sifre Cozme Algoritması
3   * Formula:  $m = L(c^\lambda \bmod n^2) * \mu \bmod n$ 
4   * @param {BigInt} ciphertext - Şifreli metin (BigInt formatında)
5   * @param {Object} privateKey - Private key {lambda, mu, n}
6   * @returns {number} Desifrelenmiş mesaj (JavaScript Number)
7   */
8  decrypt(ciphertext, privateKey = null) {
9    // Private key parametrelerini al
10   const { lambda, mu, n } = privateKey || this.keys.privateKey;
11   const n2 = n * n; //  $n^2$  hesapla
12
13   // ADIM 1:  $c^\lambda \bmod n^2$  hesapla
14   const u = this.modPow(ciphertext, lambda, n2);
15
16   // ADIM 2:  $L(u)$  fonksiyonunu uygula
17   const Lu = this.L(u, n);
18
19   // Kontrol
20   if (Lu === 0n)
21     return null;
22
23   // ADIM 3:  $L(u) * \mu \bmod n$  hesapla
24   const message = (Lu * mu) % n;
25
26   // BigInt'i Number'a çevir
27   const result = Number(message);
28   return result;
29 }

```

Şekil 3. 76. Şifrelenmiş verinin çözülmesini sağlayan algoritma.

#### 3.11.4.3. Sonuçların Saklanması

Homomorfik toplama ve toplama işleminin sonucunda oluşan değerlerin şifresinin çözülmesiyle seçim sonuçları açıklanabilmektedir. Seçim sonuçları açıklandıktan sonra kaydedilmesi gerekmektedir. Bu kaydetme işlemi, hesaplama işleminin tekrar yapılmasına olan ihtiyacı ortadan kaldıracak ve sisteme olan yükü azaltacaktır. Bu işlem için Şekil 3.77.'de gösterilen örnek JavaScript fonksiyonu kullanılmaktadır.

```

1  // EncryptedVoting.sol
2  contract EncryptedVoting {
3    /**
4     * @dev Off-chain desifrelenmiş sonuçları kaydet
5     */
6    function storeDecryptedResults(
7      uint256 electionId,
8      uint256[] memory results
9    ) external onlyAuthorizedAdmin {
10     require(!electionDecrypted[electionId], "Already decrypted");
11
12     // Sonuçları kaydet
13     decryptedResults[electionId] = results;
14
15     // Toplam oy sayısını hesapla
16     uint256 totalVotes = 0;
17     for (uint256 i = 0; i < results.length; i++) {
18       totalVotes += results[i];
19     }
20     decryptedTotalVotes[electionId] = totalVotes;
21     electionDecrypted[electionId] = true;
22
23     emit ElectionDecrypted(electionId, results, totalVotes);
24   }
25 }

```

Şekil 3. 77. Seçim sonuçlarının kaydedilmesi işlemi.

### 3.12. AKILLI SÖZLEŞMELER

Akıllı sözleşmeler, sistemin iş akışını oluşturmaktadır. Blokzinciri tabanlı oylama sisteminin merkezi yapıdan bağımsız olabilmesi için akıllı sözleşmelerin doğru bir şekilde çalışması gerekmektedir. Seçmen ve yetkili kimliklerinin doğrulanması, oyların saklanması ve ispatlanmasıyla birlikte seçim sonuçlarının ilan edilmesi süreci akıllı sözleşmeler tarafından

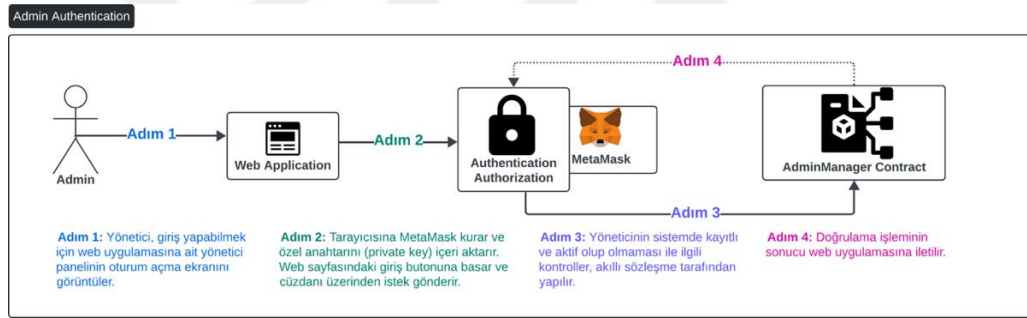
gerçekleştirilmektedir. Akıllı sözleşmeler, kimlik ve yetkilendirme, yetkili işlemleri, oylama süreci ve sonuçların doğrulanmasından oluşan beş ana bölümden oluşmaktadır.

### 3.12.1. Kimlik Doğrulama ve Yetkilendirme

Kimlik ve yetkilendirme, seçmenlerin veya yetkililerin sistemi kullanabilmesi için gerçekleştirilen, doğrulama ve yetkilendirme süreçlerini ifade etmektedir. Akıllı sözleşmeler, sisteme giriş yapmak isteyen seçmenlerin veya yetkililerin kimliklerini denetleyerek bu süreci gerçekleştirirler. Sözleşmeleri kullanan iki aktör olmasından dolayı doğrulama ve yetkilendirme süreçleri iki bölüme ayrılmaktadır.

#### 3.12.1.1. Seçim Yetkilisi Doğrulama ve Yetkilendirme

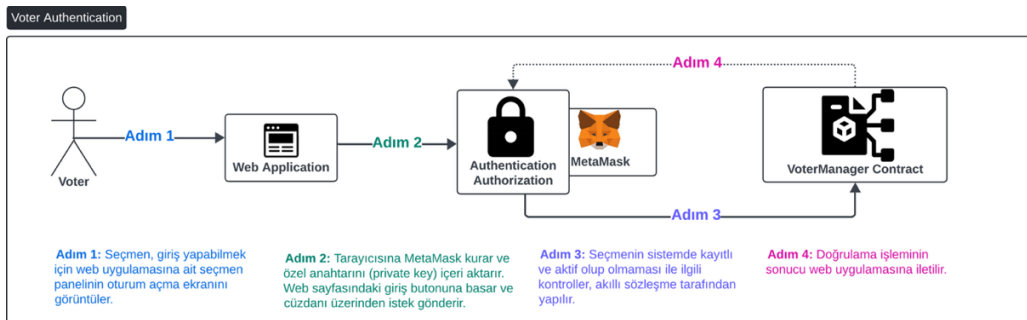
Seçim yetkilileri için kimlik doğrulama ve yetkilendirme süreçlerini kapsamaktadır. Yetkililerin kontrolünü gerçekleştiren AdminManager isimli akıllı sözleşmedir. Şekil 3.78.'de sürecin işleyici gösterilmektedir.



Şekil 3. 78. Yetkili kimlik doğrulama sürecine ait akıllı sözleşme işlemleri.

#### 3.12.1.2. Seçmen Doğrulama ve Yetkilendirme

Seçmenler için kimlik doğrulama ve yetkilendirme süreçlerini kapsamaktadır. Seçmenlerin kontrolünü VoterManager isimli akıllı sözleşme gerçekleştirmektedir. Bu süreç Şekil 3.79.'da gösterilen şekilde işlemektedir.

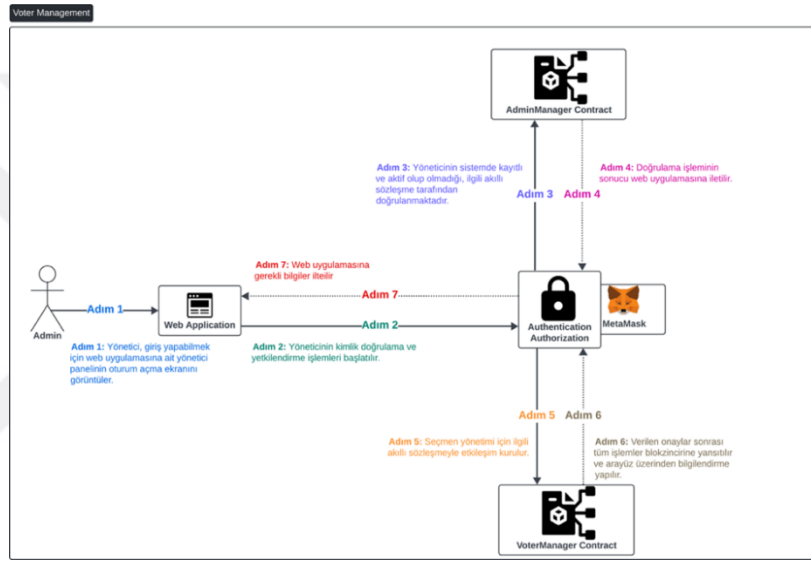


Şekil 3. 79. Seçmen kimlik doğrulama sürecine ait akıllı sözleşme işlemleri.

### 3.12.2. Yetkili İşlemleri

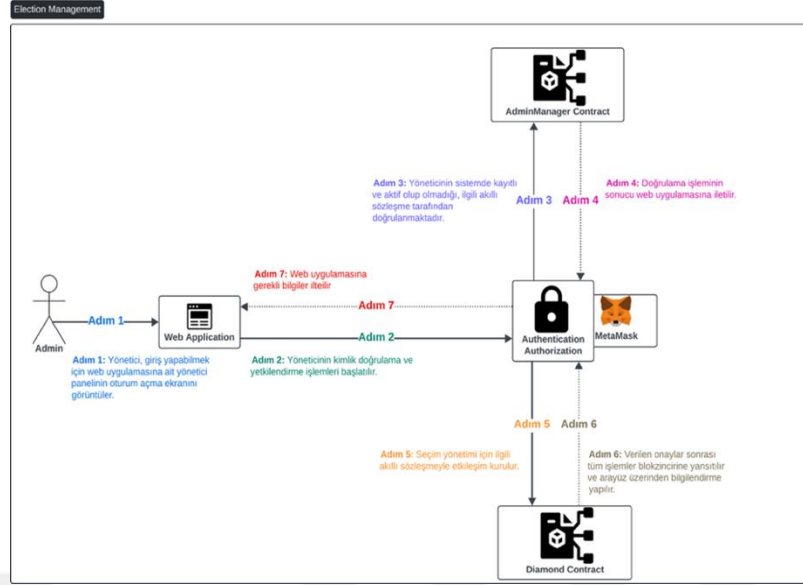
Seçim yetkililerinin oylama sisteminde gerçekleştirebilecekleri tüm sistem içi iş süreçleri yetkili işlemleri olarak tanımlanabilir. Yetkililer, seçmen yönetimi, yetkili yönetimi, seçim yönetimiyle fakülte ve bölüm yönetimleri görevlerini üstlenmektedir.

Seçmen yönetimi işlemleri, VoterManager isimli akıllı sözleşme tarafından gerçekleştirilmektedir. Kimliği doğrulanmış bir yetkili, seçmen yönetimi için akıllı sözleşmeyi kullanabilmektedir. Şekil 3.80.'de seçmen yönetimi işlemlerinin akıllı sözleşmelerle gerçekleştirilme süreci gösterilmektedir.



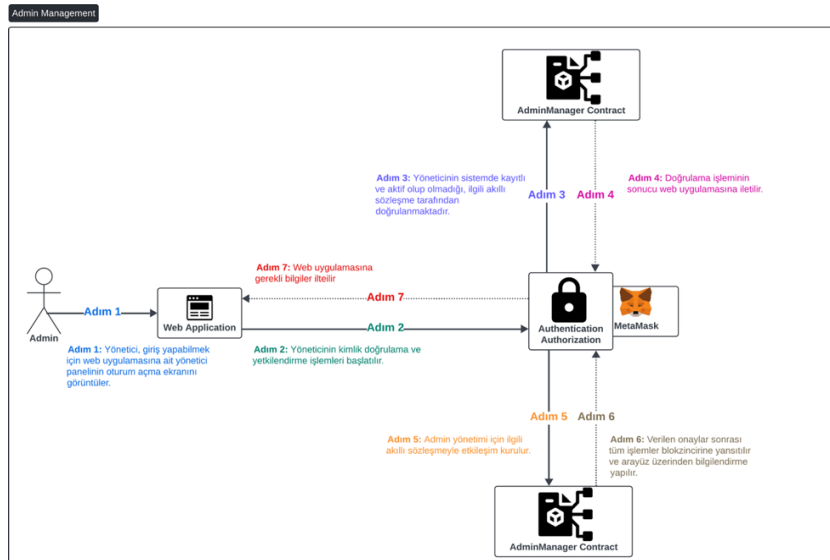
Şekil 3. 80. Seçmen yönetimi sürecine ait akıllı sözleşme işlemleri.

Seçim yönetimi işlemleri Diamond isimli akıllı sözleşme üzerinden gerçekleştirilmektedir. Diamond sözleşmesi, Diamond Pattern uygulaması sonucu ortaya çıkmıştır. Şekil 3.81.'de seçim yönetimi işlemlerinin gerçekleştirilme süreci gösterilmektedir.



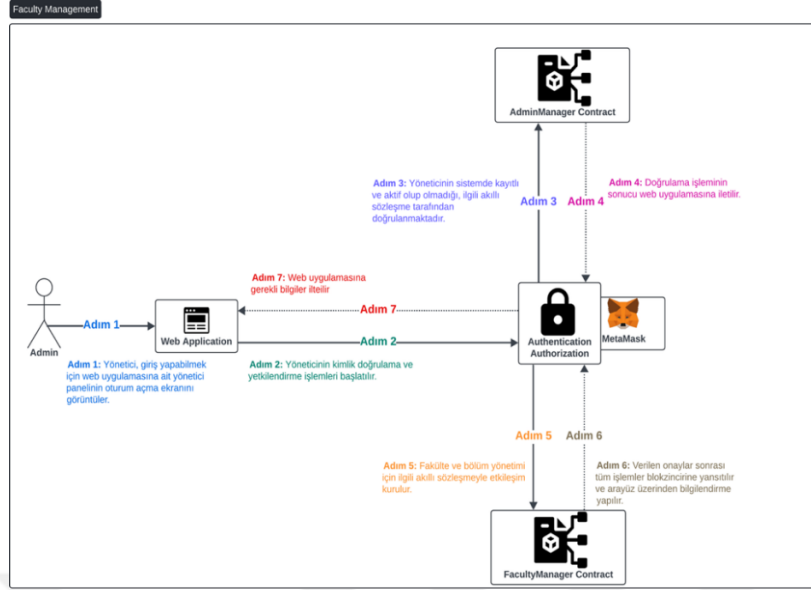
**Şekil 3. 81.** Seçim yönetimi sürecine ait akıllı sözleşme işlemleri.

Yetkili yönetimi işlemlerinin gerçekleştirilmesi için AdminManager isimli akıllı sözleşme kullanılmaktadır. Bu akıllı sözleşme, aynı zamanda yetkili oturum işlemlerini de gerçekleştirmektedir. Şekil 3.82.'de yetkili yönetimi işlemlerinin gerçekleştirilme süreci gösterilmektedir.



**Şekil 3. 82.** Yetkili yönetimi sürecine ait akıllı sözleşme işlemleri.

Fakülte ve bölüm yönetimi işlemleri FacultyManager isimli akıllı sözleşmeyle gerçekleştirilmektedir. Fakülte ve bölüm yönetimi işlemlerinin akıllı sözleşmelerle gerçekleştirilme süreci Şekil 3.83.'de gösterilmektedir.

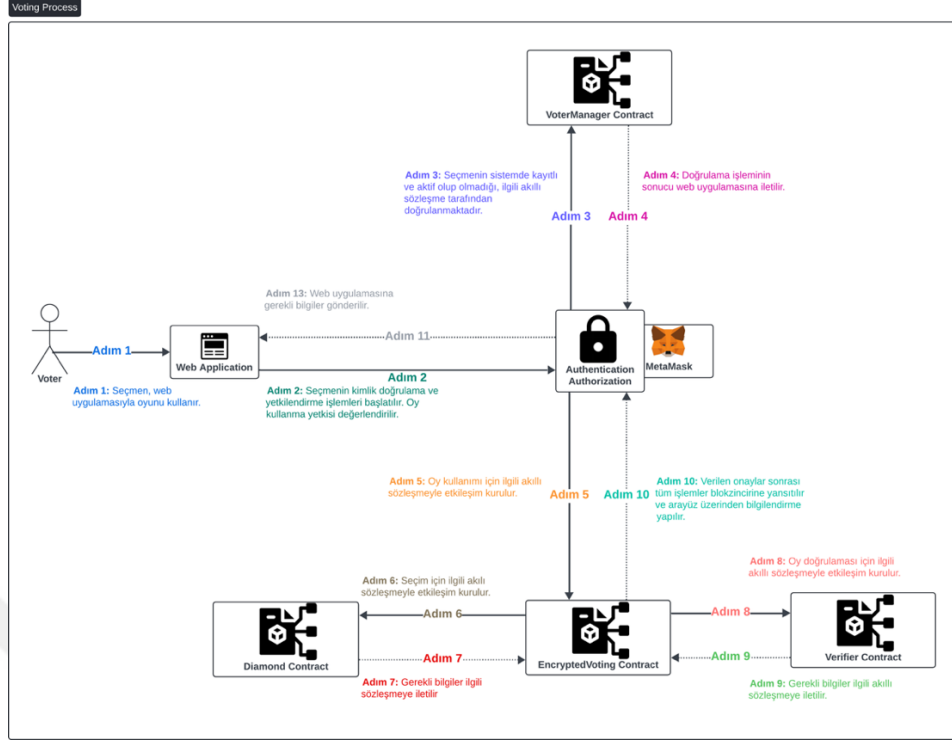


**Şekil 3. 83.** Fakülte ve bölüm yönetimi sürecine ait akıllı sözleşme işlemleri.

### 3.12.3. Oylama Süreci

Oylama işleminin gerçekleştirilebilmesi için çeşitli akıllı sözleşmelerin ve sıfır bilgi ispatı devresinin etkileşimli bir şekilde çalışması gerekmektedir. Bu işlemlerde, VoterManager, EncryptedVoting, verifier ve Diamond isimli akıllı sözleşmeler önemli görevler üstlenmektedir. Bu verifier.sol isimli akıllı sözleşme, Circom diliyle geliştirilen sıfır bilgi ispatı devresinin derlenmesi sonrası oluşmaktadır. Verifier sözleşmesinin, server servisi içerisindeki diğer sözleşmelerin içerisine aktarılması gerekmektedir. Verifier, oy için üretilen kanıtın doğrulanabilirliğini kontrol etmek için kullanılan sözleşmedir.

Seçim süreçleriyle ilgili işlemler Diamond akıllı sözleşmesi üzerinden gerçekleştirilmektedir. Bu sözleşme, Diamond Pattern uygulanarak oluşturulmuştur. Diamond Pattern kullanılarak tüm seçim işlemleri Facet isimli akıllı sözleşmelere bölünmüştür. Seçim süreçlerinde, oy kullanımı ve doğrulaması amacıyla EncryptedVoting isimli akıllı sözleşme kullanılmaktadır. EncryptedVoting sözleşmesinin seçimlerle ilgili bilgilere ulaşması gerektiğinde Diamond sözleşmesiyle iletişim kurması gerekmektedir. Oylar, aday bazında vektörel olarak şifreli bir şekilde EncryptedVoting kullanılarak saklanmaktadır. Bir seçimde oylamanın akıllı sözleşmelerle gerçekleştirilmesi ve tamamlanması Şekil 3.84.'de gösterilmektedir.



Şekil 3. 84. Oylama sürecine ait akıllı sözleşme işlemleri.

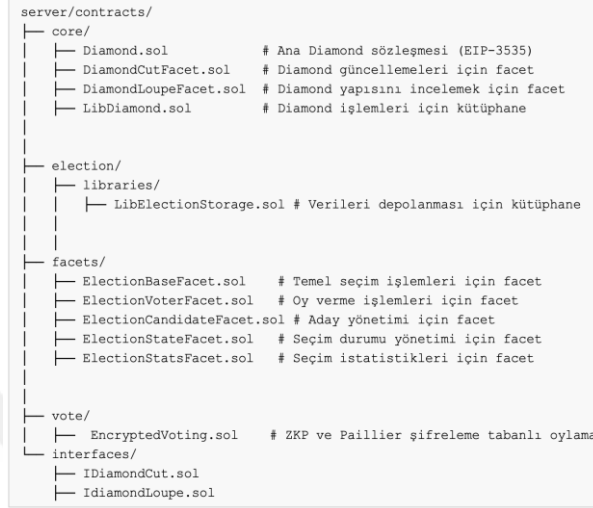
### 3.12.4. Diamod Pattern (EIP-2535)

Ethereum platformunda bir akıllı sözleşmenin boyutu 24,576 bytes'tır. Diamond pattern (EIP-2535), bu boyut limitlerini aşmak ve güncellenebilir akıllı sözleşmeler oluşturabilmek için geliştirilmiş bir standarttır. Diamond, bir akıllı sözleşmenin değiştirilemez olma özelliğinin esnetilmesi için çözümler sunmaktadır. Bu standart, geliştiricilere şu özellikleri sağlamaktadır:

1. Tek Sözleşme Adresi: Birden fazla akıllı sözleşmeden oluşan bir yapıda tüm işlevsellik tek bir akıllı sözleşme adresi üzerinden sağlanmaktadır.
2. Modülerlik: Farklı görevler için "facet" adı verilen ayrı akıllı sözleşmeler kullanılmaktadır.
3. Ortak Depolama Alanı: Tüm "facet" akıllı sözleşmeleri ortak bir depolama alanını paylaşmaktadır.
4. Çalışma Zamanında Yükseltme: Akıllı sözleşmelerin işlevleri çalışma zamanında eklenebilmekte, değiştirilebilmekte veya kaldırılabilir.

5. Fonksiyon Çakışmalarının Önlenmesi: Her fonksiyon ilgili facet sözleşmesine ait olduğundan, aynı isme sahip fonksiyonların çakışması önlenmektedir.

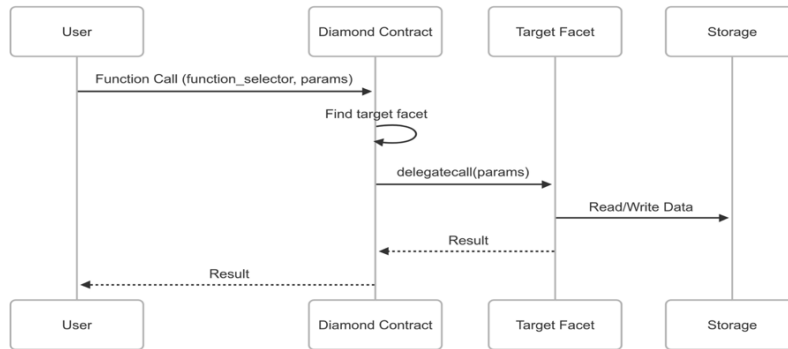
Sistemde diamond pattern Şekil 3.85.'de gösterilen temel dosyalar ile kullanılmaktadır.



Şekil 3. 85. Seçim işlemleri için Diamond Pattern dosya hiyerarşisi

#### 3.12.4.1. Diamond.sol

Diamond, merkezi giriş noktası ve ana yapıyı oluşturan sözleşmedir. Arayüz uygulamaları bu adresle etkileşime girmektedir. Tüm istekler diamond içerisindeki delegasyon tabanlı fallback fonksiyonu üzerinden facet sözleşmelerine iletilmektedir. Diamond pattern çalışması Şekil 3.86.'da gösterilmektedir.



Şekil 3. 86. Diamond pattern için sıralama diyagramı (sequence diyagram).

#### 3.12.4.2. LibDiamond.sol

Diamond Pattern için temel işlevler ve veri yapılarını sağlamaktadır. İçerisinde diamond sözleşmesinin durumunu depolayan bir yapı, diamond verilerine erişim imkânı sağlayan bir

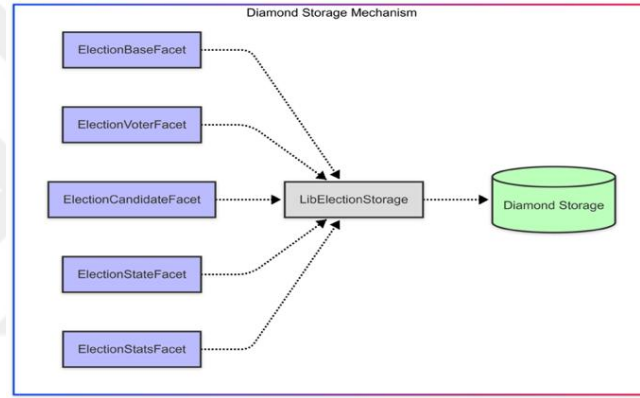
fonksiyonla facet ekleme, deęiřtirme ve kaldırma işlemleri için temel fonksiyonlar bulunmaktadır.

**DiamondStorage:** Diamond sözleşmesinin durumunu depolayan yapıdır. Veri saklamak için kullanılan özel bir yapıdır. Facet'ler bu yapıya erişerek çakışmadan verileri kullanabilirler.

**DiamondCut:** Facet eklemek, deęiřtirmek ve kaldırmak için kullanılan fonksiyondur.

#### 3.12.4.3. *LibElectionStorage.sol*

Diamond pattern kullanıldığında, tüm facet sözleşmeleri ortak bir depolama alanını kullanırlar. Bu depolama alanı LibElectionStorage.sol isimli kütüphanedir. Ortak bir depolama kullanımı Şekil 3.87.'de gösterilmektedir.



Şekil 3. 87. Diamond pattern uygulamasında ortak bir depolama kullanımı.

#### 3.12.4.4. *DiamondCutFacet.sol*

DiamondCutFacet, Diamond akıllı sözleşmesinin yapısını deęiřtirmek için kullanılan bir facet sözleşmesidir. Böylece, Diamond sözleşmesine yeni fonksiyonlar eklenebilir, mevcut fonksiyonlar deęiřtirilebilir ve kaldırılabilir. Sisteme her yeni özellik ayrı bir facet sözleşme olarak eklendiğinden, bu facet sözleşmelerinin DiamondCutFacet ile Diamond'a eklenmesi sağlanır.

#### 3.12.4.5. *DiamondLoupeFacet.sol*

DiamondLoupeFacet, Diamond akıllı sözleşmesinin yapısını incelemek için kullanılan bir facet akıllı sözleşmesidir. Bu facet sayesinde, Diamond sözleşmesinin hangi fonksiyonlara sahip olduęu bilgisi öğrenilebilir.

#### **3.12.4.6. ElectionBaseFacet.sol**

Temel seçim işlemleri için işlevler sağlayan facet akıllı sözleşmesidir. Yeni bir seçim oluşturmak, mevcut seçimi silmek, seçimlerin varlığını ve aktifliğini kontrol etmek ve seçim detaylarını öğrenmek amacıyla kullanılır.

#### **3.12.4.7. ElectionVoterFacet.sol**

Seçimler için seçmen yönetimi işlemlerini gerçekleştiren facet akıllı sözleşmesidir. Seçim sürecinin tüm aşamalarındaki seçmen yönetimi bu akıllı sözleşmeyle gerçekleştirilmektedir.

#### **3.12.4.8. ElectionCandidateFacet.sol**

Seçimler için aday yönetimi işlemlerini gerçekleştirmektedir. Seçim sürecinin tüm aşamalarında aday yönetimi için gereken işlemler bu akıllı sözleşmeyle gerçekleştirilmektedir.

#### **3.12.4.9. ElectionStateFacet.sol**

Seçimler için durum yönetimi işlevleri sağlayan facet akıllı sözleşmesidir. Seçimlerin başlatılması, bitirilmesi, seçmen listelerinin hazırlanması için gereken adımlar, aktiflik ve pasiflik kontrolleri bu akıllı sözleşmeye gerçekleştirilmektedir.

#### **3.12.4.10. ElectionStatsFacet.sol**

Seçimlerle ilgili istatistiklerin açıklanması için gerekli işlevleri sağlayan facet akıllı sözleşmesidir.

### **3.12.5. Snapshot Sistemi**

Seçmenlerin güvenli, doğrulanabilir ve performans kriterlerini sağlayacak şekilde sisteme dahil edilmesi büyük önem taşımaktadır. Bu çalışmada, seçmen listesinin doğruluğunu sağlamak ve aynı zamanda işlem maliyetini düşürmek amacıyla Snapshot tabanlı seçmen kaydı ve Lazy Loading mimarisi uygulanmıştır. Bu sistem, seçmenlerin uygunluğunu dinamik olarak belirli bir zaman noktasına (snapshot) göre kontrol eder.

Snapshot sisteminin temelinde, seçim sürecinin belirli bir zaman damgası ile başlatılması yer alır. Bu zaman damgası sistemin referans noktasıdır. Seçmen uygunluğuna ilişkin tüm kontroller, bu zaman damgası öncesi duruma göre yapılır. Seçim başlatıldığında:

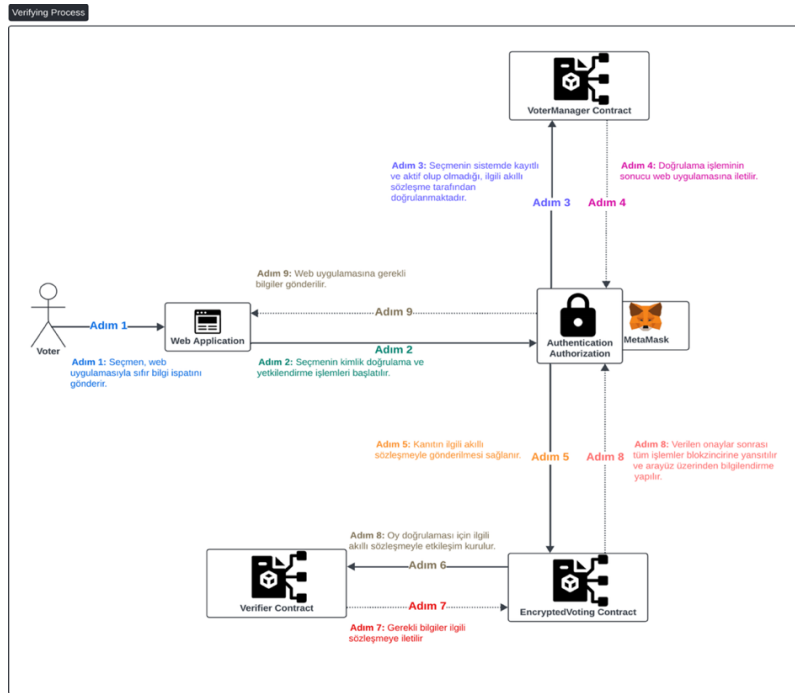
- Sistemdeki kullanıcılar arasında uygunluk filtrelemesi yapılır (örneğin kayıt durumu, kimlik doğrulaması, rol durumu, fakülte ve bölüm uygunluğu.).

- Bu filtreleme sonucunda elde edilen seçmen listesi bir veri yapısı olarak **snapshot** haline getirilir.
- Snapshot listesi, sunucu tarafında hazır bekletilir.
- Seçmen oy kullanmaya geldiğinde, sisteme giriş yapan kullanıcının kimliği bu listeye göre kontrol edilir.

Bu yapı sayesinde, seçim süreci boyunca sisteme sonradan kaydolun, kimlik değiştiren veya silinen kullanıcılar seçmen listesine etki edemez. Snapshot listesi, özellikle büyük ölçekli sistemlerde tamamı belleğe yüklenmeden, yalnızca ihtiyaç anında seçmen bazlı olarak çağrılır (lazy loading). Bu sayede hem sunucu hem istemci tarafında bellek ve işlem tasarrufu sağlanmaktadır.

### 3.12.6. Oy Doğrulama İşlemleri

Seçim süreçlerinde, seçmene oy kullanması sonrası sıfır bilgi ispatı kanıtı verilmektedir. Bu kanıt sayesinde oy hem seçmen hem de yetkililer tarafından doğrulanabilmektedir. Aynı zamanda blokzincire kanıtların yazılıyor olması, ağ üzerindeki herkesin oyu doğrulayabildiği fakat içeriğini göremediği verimli bir sistemin oluşturulmasını sağlamaktadır. Seçmenin oy doğrulama süreci Şekil 3.88.'de gösterilmektedir.



Şekil 3. 88. Oy doğrulama sürecine ait akıllı sözleşme işlemleri.

### 3.13. GELİŞTİRME VE TEST ORTAMI

Geliştirilen sistemin entegrasyonu, testleri ve performans analizleri, Apple Silicon tabanlı bir donanım ortamında gerçekleştirilmiştir. Testlerin tümü aynı ortamda çalıştırılmıştır. Kullanılan donanım ve yazılım ortamı aşağıda özetlenmiştir:

- **Bilgisayar:** Apple MacBook Pro
- **İşlemci (CPU):** Apple M4 Pro, 12 çekirdek (8 performans + 4 verimlilik)
- **Grafik İşlemci (GPU):** 16 çekirdekli entegre Apple GPU
- **Bellek (RAM):** 24 GB (Unified Memory Architecture)
- **İşletim Sistemi:** macOS Sequoia 15.5
- **Tarayıcı:** Brave ve Chrome
- **Blokzinciri Geliştirme Araçları:**
  - Node.js v22.14.0
  - Hardhat v2.22.19
  - Solidity v0.8.28
  - Circom v0.5.46
  - CircomlibJS v0.1.7
  - Circomlib v2.0.5
  - SnarkJS v0.7.5
  - Ethers.js v6.8.0
  - React v19.0.0
  - Express v4.18.2
  - Java SE 17

Bu geliştirme ve test ortamında, yüksek işlem gücü ve hızlı veri işleme kapasitesine ulaşılmıştır. Bu ortamda, blokzincir tabanlı merkezi olmayan üniversite ölçekli oylama sistemi sorunsuz bir şekilde çalıştırılmıştır. Bunun yanı sıra, karmaşık test ve analiz işlemleri için gerekli hesaplamalar, mevcut kaynaklar ve araçlar kullanılarak gerçekleştirilmiştir.

### 3.14. TEST YÖNTEMİ

Bu bölümde, geliştirilen blokzinciri tabanlı merkezi olmayan üniversite ölçekli oylama sisteminin çeşitli alt bileşenlerine yönelik gerçekleştirilen testlerin kapsamı ve uygulama biçimleri açıklanmaktadır. Bu çalışmada, geliştirilen sistemin performans, doğruluk ve maliyet açısından değerlendirilmesi amacıyla test senaryoları oluşturulmuş ve sıralı bir şekilde uygulanmıştır. Amaç, sistemin bileşen düzeyindeki işlevleriyle birlikte genel çalışma yapısının, belirli performans ve maliyet metrikleri çerçevesinde analiz edilmesidir. Sistem performansını değerlendirmek amacıyla aşağıdaki temel bileşenler üzerinde test ve analiz işlemleri gerçekleştirilmiştir.

- Oy Kullanma İşlemi
- Hash Üretimi (Commitment ve Nullifier)
- Sıfır Bilgi İspatı Kanıt Üretimi ve Doğrulaması
- Oy Şifreleme İşlemi
- Eşzamanlı Testler

Bunlara ek olarak, sistemin blokzincir işlem maliyetlerinin değerlendirilmesine yönelik, gaz kullanımı ve hesaplama maliyetlerini analiz eden işlem maliyeti testleri gerçekleştirilmiştir. Her test işlemi, farklı seçmen sayıları ve senaryolarla tekrarlanmış; elde edilen metrikler karşılaştırmalı olarak analiz edilmiştir. Bu test işlemlerinden yalnızca maliyet testleri (gaz tüketimi) tarayıcı ortamında MetaMask kullanılarak gerçekleştirilmiştir. Diğer tüm performans testleri, daha kontrollü ve ölçülebilir bir yapı oluşturabilmek için doğrudan JavaScript tabanlı test fonksiyonlarıyla gerçekleştirilmiştir. Her bir bileşen için uygulanan test süreçleri aşağıda açıklanmıştır:

- **Oy Kullanma Testi:** Baştan sona bir seçmenin oy verme sürecinde gerçekleştirdiği tüm teknik adımları kapsamaktadır. Bu test, bir seçmenin sistemde baştan sona gerçekleştireceği oy kullanma sürecini bir bütün olarak test etmek üzere tasarlanmıştır. Gerçekleştirilen tüm işlemler, bir seçmenin oy kullanma sürecini başlatmasından itibaren, oyunun sistemde başarılı bir şekilde gönderilmesine kadar olan süreci içermektedir. Testler, hem sıralı hem de çoklu seçmen içeren eşzamanlı senaryolarla (örneğin 10, 100, 500, 1.000 seçmen) gerçekleştirilmiş ve işlem süresi, sistem yanıtı ve işlem yoğunluğu açısından karşılaştırmalı analizler yapılmıştır. Oy kullanma testi, aşağıda adım adım belirtilen aşamalardan oluşmakta olup; bu aşamaların tamamı, oy kullanma sürecine ilişkin performans değerlendirmesine dahil edilmiştir.
  - **Nullifier ve Commitment Üretimi:** Seçmenin kimliğini anonimleştirmek amacıyla Poseidon hash fonksiyonu ile benzersiz nullifier ve commitment değerleri oluşturulur.
  - **ZKP Kanıt Üretimi:** Nullifier, oy ve commitment gibi bilgileri içeren sıfır bilgi ispatı kanıtı (Zero-Knowledge Proof) üretilir. Circom devresi ve SnarkJS aracıyla Groth16 protokolü kullanılarak kanıt oluşturulur.
  - **Oy Şifreleme:** Seçmenin tercihi Paillier homomorfik şifreleme algoritmasıyla, vektör yapısında gizlenir.
  - **ZKP Kanıt Doğrulama:** Üretilen sıfır bilgi ispatı kanıtı hem JavaScript hem de akıllı sözleşme tabanlı doğrulanır.
  - **Veri Gönderimi ve Blokzincire Kaydedilmesi:** Üretilen kanıt, hash'ler ve şifreli oy, Ethereum test ağı üzerinde çalışan akıllı sözleşmeye gönderilir. İşlemin blokzincire yazılması sağlanır.
- **Hash Üretimi:** Seçmenin oy kullanma sürecinde üretilen commitment ve nullifier değerlerinin üretim hızını test edebilmek amacıyla, Poseidon hash fonksiyonu kullanılarak hash değerlerinin oluşturulma süresi değerlendirilmiştir.
- **Kanıt Üretimi:** Circom devreleri kullanılarak SnarkJS aracılığıyla sıfır bilgi kanıtları oluşturulmuştur. Bu işlem, oy kullanma süreci içerisinde gerçekleştirilmiş olup, gerçekçi koşullarda kanıt üretim performansının ölçülmesi amaçlanmıştır.

- **Kanıt Doğrulama:** Sistemde kanıt doğrulama işlemi hem JavaScript ile hem de akıllı sözleşmeler aracılığıyla gerçekleştirilmektedir. Bu nedenle, JavaScript ortamında gerçekleştirilen doğrulama süresi ile akıllı sözleşme üzerinde yapılan doğrulama süreci ayrı ayrı test edilmiş ve karşılaştırmalı olarak değerlendirilmiştir.
- **Oy Şifreleme:** Paillier şifreleme algoritması kullanılarak, seçmen oylarının vektör tabanlı şifreleme süresi ölçülmüştür. Ayrıca, şifreleme algoritmasında kullanılan anahtar boyutunun değiştirilmesinin performans üzerindeki etkisi analiz edilmiştir.
- **Eşzamanlı Testler:** Oy kullanımı, hash üretimi, sıfır bilgi ispatı (ZKP) kanıtlarının oluşturulması ve doğrulanması ile Paillier şifreleme algoritmasıyla gerçekleştirilen oy şifreleme işlemleri, Node.js'in WorkerThread modülü kullanılarak yürütülmüştür. Bu kapsamda, seri ve eşzamanlı çalıştırılan testlerle işlem süreleri karşılaştırmalı olarak analiz edilmiştir. JavaScript Worker Thread yapısı doğası gereği eşzamanlı (concurrent) işlem yürütmeyi amaçlayan bir modeldir. Worker Thread'ler, bağımsız görevlerin aynı zaman dilimi içerisinde işlenmesini sağlar ancak bu işlemlerin fiziksel olarak aynı anda paralel yürütülüp yürütülmediği donanım ve işletim sistemi kaynaklarına bağlıdır. Çok çekirdekli işlemciler sayesinde, Worker Thread'ler farklı çekirdeklerde gerçek paralel işlem yapabilmekte ve böylece paralel yürütme gerçekleşmektedir. Ancak, bu paralelliğin donanım koşullarına bağlı olduğu ve farklı ortamlarda eşzamanlı (concurrent) olarak çalışabileceği göz önünde bulundurulmalıdır.

## 4. BULGULAR

Bu çalışma kapsamında blokzinciri tabanlı merkezi olmayan üniversite ölçekli bir oylama sistemi geliştirilmiştir. Katılımcı sayısının artırılması ve güvenli bir seçim sürecinin gerçekleştirilebilmesi için sistemin bütün parçalarının doğru bir şekilde çalışması gerekmektedir. Bu bağlamda, bütün parçaların çalışmasını değerlendirebilmek amacıyla çeşitli test ve analiz işlemleri gerçekleştirilmiştir. Bu bölüm, gerçekleştirilen test ve analiz işlemleri sonucunda elde edilen bulguları barındırmaktadır. Bulgular, işlem maliyeti, performans, güvenilirlik ve ölçeklenebilirlik açısından kritik veriler barındırmaktadır. Sistem, sıfır bilgi ispatı, şifreleme ve özetleme algoritmaları kullanılarak geliştirildiği için her verinin detaylı bir şekilde değerlendirilmesi gerekmektedir.

Sistem, üniversite ortamında kullanılmak amacıyla bir prototip olarak geliştirilmiştir. Kullanılabilir olduğunu gösterebilmek amacıyla farklı rollere sahip çok sayıda seçmenle oylama süreci için test ve analiz senaryoları hazırlanmıştır. Bu bağlamda, sistemin değerlendirilmesi için işlem maliyeti (gas fee), işlem süreci, sistem ve kriptografik tekniklerin performansı, saniyedeki işlem sayısı (TPS) gibi ölçütler dikkate alınmıştır. Bu değerlendirmeler, sistemin güçlü ve zayıf yönlerini ortaya çıkarmaktadır. Böylece, oylama sisteminin kullanmış olduğu teknoloji ve tekniklerin, daha büyük ölçekli oylama sistemlerinde uygulanabilirliği değerlendirilebilecektir.

### 4.1. MALİYET ANALİZİ

Maliyet analizi, geliştirilen sistemlerin uygulanabilirliğini tespit edebilmek ve kaynakları verimli kullanabilmek için önemli bir yöntemdir. Geliştirilen oylama sisteminin ortaya çıkartacağı maliyetlerin hesaplanması; üniversitelerin bu teknolojiye geçiş kararı alabilmesi için veri sağlayacak ve benzeri alanlarda yapılacak çalışmalara bir referans olacaktır. Gerçekleştirilen maliyet analizi, aynı zamanda sistemin sürdürülebilirliği hakkında bilgiler sunacaktır ve bu alanda yapılacak yatırımlar için örnek bir çalışma olarak kullanılabilecektir.

Ethereum ağında gerçekleşen her işlem için gaz ücreti (gas fee) denilen bir miktarın karşılanması gerekmektedir. Yerel bilgisayar üzerinde oluşturulan test ortamlarında bu ücretler geçici hesaplardan karşılanmaktadır. Fakat oylama sisteminin test ağları yerine ana ağlarda

çalışacağı varsayılmaktadır. Bu nedenle, ana ağlarda gerçekleştirilecek işlemler için gerekli gaz ücretinin ödenmesi gerekmektedir.

Ethereum tabanlı blokzinciri ağında oluşan maliyetler; ETH, USD ve TL olarak hesaplanabilmektedir. ETH ve Gwei, Ethereum blokzinciri üzerindeki işlemler için kullanılan birim değerlerdir. Ether, Ethereum blokzincirinin para birimidir. Gwei, Ether'in küçük bir alt birimidir ve gaz ücretini (gas fee) belirtmek için kullanılmaktadır.

Ether, Gwei, USD ve TL ilişkisi ve oluşan gaz maliyetinin hesaplanması şu şekildedir:

$$1 \text{ Ether} = 1.000.000.000 \text{ Gwei} \text{ (1 milyar Gwei)}$$

ETH cinsinden gaz maliyeti:

$$\text{Kullanılan Gaz} \times \text{Gaz Fiyatı (Gwei)} \times \left( \frac{1 \text{ ETH}}{1.000.000.000 \text{ Gwei}} \right)$$

USD cinsinden gaz maliyeti:

$$\text{ETH Maliyeti} \times \frac{\text{ETH}}{\text{USD}}$$

TL cinsinden gaz maliyeti:

$$\text{USD Maliyeti} \times \frac{\text{USD}}{\text{TRY}}$$

Ethereum ağlarında her gazın bir gwei türünden fiyatı bulunmaktadır. Bu fiyatın değeri; ağın yoğunluğu, işlem hızı ve harcanacak güce göre değişiklik göstermektedir. Maliyet hesaplamasında kullanılacak gaz ve para birimi dönüşüm parametreleri Tablo 4.1.'de gösterilmektedir.

**Tablo 4. 1.** Gaz ve para birimi dönüşüm parametreleri (06 Haziran 2025).

| Parametre   | Değer                |
|-------------|----------------------|
| Gaz Fiyatı  | 1 Gaz = 1,52 Gwei    |
| ETH Fiyatı  | 1 ETH = 2,482,47 USD |
| USD/TR Kuru | 1 USD = 39,28 TL     |

#### 4.1.1. Sistem Dağıtım ve Başlangıç Maliyetleri

Ethereum tabanlı geliştirilen uygulamalarda, akıllı sözleşmelerin derlenmesi ve ağ üzerine dağıtılması sürecinde belirli bir maliyet (gaz ücreti) oluşmaktadır. Ethereum ağında sözleşmeyi dağıtanlar tarafından bu maliyetin karşılanması gerekmektedir. Dağıtım maliyetleri belirlenirken kullanılan birim parametreleri Tablo 4.1.'de gösterilmektedir.

Akıllı sözleşmelerin dağıtım sırasında kullanılan gaz miktarı ve bunun sonucunda oluşan maliyet Tablo 4.2.'de gösterilmektedir. Sisteme 2 yetkili, 3 fakülte ve 6 bölüm varsayılan olarak eklenmiş ve dağıtım maliyetlerine dahil edilmiştir.

**Tablo 4. 2.** Akıllı sözleşmelerin dağıtılmasında oluşan maliyet.

| Akıllı Sözleşme        | Kullanılan Gaz    | Gaz Maliyeti (ETH)   | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) |
|------------------------|-------------------|----------------------|--------------------|-------------------|
| AdminManager           | 496.469           | 0,00075463288        | 1,87               | 73,59             |
| VoterManager           | 3.072.710         | 0,00467051920        | 11,59              | 455,43            |
| FacultyManager         | 2.421.468         | 0,00368063136        | 9,14               | 358,90            |
| EncryptedVoting        | 2.873.421         | 0,00436759992        | 10,84              | 425,89            |
| Verifier               | 580.211           | 0,00088192072        | 2,19               | 86,00             |
| Diamond                | 2.473.391         | 0,00375955432        | 9,33               | 366,60            |
| DiamondCutFacet        | 1.119.208         | 0,00170119616        | 4,22               | 165,89            |
| DiamondLoupeFacet      | 430.473           | 0,00065431896        | 1,62               | 63,80             |
| ElectionBaseFacet      | 1.948.097         | 0,00296110744        | 7,35               | 288,74            |
| ElectionCandidateFacet | 1.296.061         | 0,00197001272        | 4,89               | 192,10            |
| ElectionStateFacet     | 2.568.207         | 0,00390367464        | 9,69               | 380,65            |
| ElectionStatsFacet     | 742.992           | 0,00112934784        | 2,80               | 110,12            |
| ElectionVoterFacet     | 1.416.938         | 0,00215374576        | 5,35               | 210,01            |
| <b>Toplam</b>          | <b>21.505.674</b> | <b>0,03268862448</b> | <b>81,15</b>       | <b>3187,51</b>    |

Akıllı sözleşmelerin dağıtılması sırasında seçmen ekleme işlemi yapıldığında ortaya çıkacak maliyet Tablo 4.3.'de gösterilmektedir. Gaz kullanım limitine takılmamak için seçmenler grup (batch) olarak isimlendirilen parçalar halinde eklenmektedir. Her parçada 100 seçmen eklenecek şekilde geliştirme yapılmıştır. 500 seçmenin eklenmesi için 5 adım gerekirken 1000 seçmenin eklenmesi için 10 adım gerekmektedir. Seçmen eklenmesi için oluşacak maliyetin sözleşme dağıtım maliyetinin üzerine ekleneceği dikkate alınmalıdır. Bu işlemler, yalnızca sistemin kurulumu sırasında oluşan maliyetleri ifade etmektedir. Sistem aktif olduktan sonra eklenecek seçmenler için oluşacak maliyet, bu değerlerden farklı olacaktır.

**Tablo 4. 3.** Akıllı sözleşmelerin dağıtılmasında seçmen eklenmesi için oluşan maliyet.

| Seçmen Sayısı | Batch Boyutu | Batch Sayısı | Kullanılan Gaz | Gaz Maliyeti (ETH) | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) | Kişi Başına Düşen Gaz Miktarı |
|---------------|--------------|--------------|----------------|--------------------|--------------------|-------------------|-------------------------------|
| 1             | 100          | 1            | 189.978        | 0,00028876656      | 0,72               | 28,16             | 189.978                       |
| 2             | 100          | 1            | 333.920        | 0,00050755840      | 1,26               | 49,49             | 189.979                       |
| 5             | 100          | 1            | 765.741        | 0,00116392632      | 2,89               | 113,50            | 153.148                       |
| 10            | 100          | 1            | 1.485.513      | 0,00225797976      | 5,61               | 220,18            | 148.551                       |
| 20            | 100          | 1            | 2.925.043      | 0,00444606536      | 11,04              | 433,54            | 189.982                       |
| 50            | 100          | 1            | 7.244.502      | 0,01101164304      | 27,34              | 1.073,76          | 144.890                       |
| 100           | 100          | 1            | 14.446.035     | 0,02195797320      | 54,51              | 2.141,15          | 144.460                       |
| 500           | 100          | 5            | 72.230.043     | 0,10978966536      | 272,55             | 10.705,75         | 144.460                       |
| 1.000         | 100          | 10           | 144.460.146    | 0,21957942192      | 545,10             | 21.411,50         | 144.460                       |

#### 4.1.2. Yetkili Tabanlı İşlem Maliyetleri

Sistem yetkililerinin, oylama sistemi üzerinde gerçekleştirdiği her işlem için ağda belirli bir miktar gaz harcanacaktır. Bu harcanan gaz miktarının karşılanması gerekmektedir. Yetkililerin gerçekleştirebildiği her işlemin maliyet hesaplaması bu bölüm içerisinde bulunmaktadır.

##### 4.1.2.1. Yetkili Yönetimi

Yetkililerinin, sistem üzerinde yetkili yönetimi yapabilecekleri bir bölüm bulunmaktadır. Bu bölüm içerisinde gerçekleştirdikleri işlemlere ilişkin karşılanması gereken maliyetler Tablo 4.4.'de gösterilmektedir.

**Tablo 4. 4.** Yetkili yönetimi için oluşan sistemsel maliyet.

| İşlem           | Kullanılan Gaz | Gaz Maliyeti (ETH) | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) |
|-----------------|----------------|--------------------|--------------------|-------------------|
| Yönetici ekleme | 58.423         | 0,00008880296      | 0,22               | 8,66              |
| Yönetici silme  | 38.068         | 0,00005786336      | 0,14               | 5,64              |

##### 4.1.2.2. Seçmen Yönetimi

Sistem üzerinde seçmen yönetimi işlemleri gerçekleştirebilmektedir. Bu bölümde yapılan işlemlere ilişkin karşılanması gereken maliyetler Tablo 4.5.'de gösterilmektedir.

**Tablo 4. 5.** Seçmen yönetimi için oluşan sistemsel maliyet.

| İşlem                | Kullanılan Gaz | Gaz Maliyeti (ETH) | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) |
|----------------------|----------------|--------------------|--------------------|-------------------|
| Seçmen ekleme        | 187.190        | 0,00028452880      | 0,71               | 27,74             |
| Seçmen düzenleme     | 59.905         | 0,00009105560      | 0,23               | 8,88              |
| Seçmen pasifleştirme | 38.921         | 0,00005915992      | 0,15               | 5,77              |

|                      |        |               |      |      |
|----------------------|--------|---------------|------|------|
| Seçmen aktifleştirme | 38.921 | 0,00005915992 | 0,15 | 5,77 |
|----------------------|--------|---------------|------|------|

#### 4.1.2.3. Fakülte ve Bölüm Yönetimi

Yetkililer tarafından sistem üzerinde fakülte ve bölüm yönetimi işlemleri gerçekleştirilmektedir. Sistemde, her bölüm bir fakülteye ait olmak zorundadır. Bu nedenle maliyet hesapları birbirinden ayrı yapılamamaktadır. Fakülte ve bölüm yönetimine ilişkin maliyetler Tablo 4.6.'da gösterilmektedir.

**Tablo 4. 6.** Fakülte ve bölüm yönetimi için oluşan sistemsal maliyet.

| İşlem                                | Kullanılan Gaz | Gaz Maliyeti (ETH) | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) |
|--------------------------------------|----------------|--------------------|--------------------|-------------------|
| Fakülte ekleme                       | 134.466        | 0,00020438832      | 0,51               | 19,93             |
| Fakülte düzenleme                    | 42.337         | 0,00006435224      | 0,16               | 6,28              |
| Fakülte pasifleştirme (1 bölümlü)    | 40.950         | 0,00006224400      | 0,15               | 6,07              |
| Fakülte pasifleştirme (5 bölümlü)    | 88.720         | 0,00013485440      | 0,33               | 13,15             |
| Fakültenin aktifleştirme (1 bölümlü) | 36.923         | 0,00005612296      | 0,14               | 5,47              |
| Fakültenin aktifleştirme (5 bölümlü) | 88.720         | 0,00013485440      | 0,33               | 13,15             |
| Bölüm ekleme                         | 171.785        | 0,00026111320      | 0,65               | 25,46             |
| Bölüm düzenleme                      | 45.071         | 0,00006850792      | 0,17               | 6,68              |

#### 4.1.2.4. Seçim Yönetimi

Yetkili işlemleri içerisinde seçim yönetimi işlemlerinin gerçekleştirildiği bir bölüm bulunmaktadır. Bu bölüm içerisinde seçim oluşturma, başlatma, bitirme ve sonuç ilanı işlemleri gerçekleştirilmektedir. Bu işlemlere ilişkin maliyetler Tablo 4.7.'de gösterilmektedir.

**Tablo 4. 7.** Seçim yönetimi için oluşan sistemsal maliyet.

| İşlem                      | Kullanılan Gaz | Gaz Maliyeti (ETH) | Gaz Maliyeti (USD) | Gaz Maliyeti (TL) |
|----------------------------|----------------|--------------------|--------------------|-------------------|
| Seçim oluşturma (2 adaylı) | 617.912        | 0,00093922624      | 2,33               | 91,59             |
| Seçim oluşturma (3 adaylı) | 717.438        | 0,00109050576      | 2,71               | 106,34            |
| Seçim oluşturma (5 adaylı) | 911.120        | 0,00138490240      | 3,44               | 135,04            |
| Seçim başlatma (2 adaylı)  | 169.737        | 0,00025800024      | 0,64               | 25,16             |
| Seçim başlatma (3 adaylı)  | 183.124        | 0,00027834848      | 0,69               | 27,14             |
| Seçim başlatma (5 adaylı)  | 209.886        | 0,00031902672      | 0,79               | 31,11             |

#### 4.1.3. Seçmen Tabanlı İşlem Maliyetleri

Seçmenlerin, oylama sistemi üzerinde gerçekleştirdiği işlemler için gaz harcanmaktadır. İşlemin gerçekleştirildiği ağda bu gaz miktarına karşılık gelen ücretin ödenmesi gerekmektedir. Seçmenlerden ücret alınmaması ve tüm ücretin sistem tarafından karşılanması için geliştirilen relay servisi bu ücretin karşılanmasını sağlamaktadır. Relay servisi tarafından karşılanan

ücretlerin, sistemi kuranlar tarafından ödeneceği bilinmelidir. Bu nedenle, seçmen işlemleri için maliyet analizinin yapılması gerekmektedir. Seçmen işlemleri sırasında oluşan maliyet Tablo 4.8.'de gösterilmektedir.

**Tablo 4. 8.** Seçim işlemleri için oluşan sistemsal maliyet.

| İşlem       | Kullanılan Gaz | Gaz Maliyeti (ETH) | ETH Maliyeti (USD) | ETH Maliyeti (TL) |
|-------------|----------------|--------------------|--------------------|-------------------|
| Oy kullanma | 2.931.414      | 0,00445574928      | 11,06              | 434,49            |

## 4.2. PERFORMANS ANALİZİ

Performans analizi, bir sistemin standartlara veya beklenen hedeflere hangi ölçüde ulaştığını değerlendirmek için kullanılan yöntemdir. Sistemin kullanılabilirliğini değerlendirmek, verimliliğini artırmak, zayıf noktalarını tespit edebilmek ve ölçeklenebilir hale getirebilmek için önemli bir araçtır. Bu bağlamda, sistemin performansının ölçülmesi çalışmanın değerlendirilmesini sağlayacaktır. Bu nedenle, çeşitli test ve analiz senaryoları hazırlanmış ve önemli veriler elde edilmiştir. Elde edilen bulgular, geliştirilen sistemin büyük ölçekli seçim süreçlerinde kullanılabilirliğine dair önemli bir referans kaynağı oluşturmaktadır.

### 4.2.1. Oy Verme Sürecinin Bütünleşik Performans Ölçümü

Oylama sisteminin en önemli işlemini oy kullanma aşaması oluşturmaktadır. Bu aşamanın değerlendirilmesi, sistemin hangi koşullarda nasıl bir performans gösterdiğinin tespit edilmesini sağlamaktadır. Bu kapsamda, sistemin performansını değerlendirebilmek için oy kullanma süreci üzerinde test ve analizler gerçekleştirilmiştir. Bu test ve analizler, JavaScript ve Bash tabanlı geliştirilen bir servis aracılığıyla gerçekleştirilmiştir. Gerçekleştirilen performans değerlendirmesi, bir seçmenin oy verme sürecini baştan sona kapsayan işlemleri içermektedir.

#### 4.2.1.1. Akıllı Sözleşme Tabanlı 32-bit Paillier ile Oy Verme

Geliştirme aşamasında, Paillier şifreleme algoritması için 32-bit anahtar uzunluğu kullanılmıştır. Bu uzunluğun seçilmesindeki neden; akıllı sözleşmeyle hesaplama yaparken gaz kullanım limitine takılmamaktır. Böylece, homomorfik toplamın elde edilmesi ve bu toplam üzerindeki şifreli verilerin çözülerek açık sonuca ulaşılabilmesi için gereken işlemler, akıllı sözleşmeler üzerinde gerçekleştirilmiştir. Her bir seçim senaryosunda, seçmenlerin sıralı şekilde oy kullanmaları sağlanmış; bu süreçte harcanan ortalama gaz miktarı ve geçen toplam süre hesaplanmıştır. Bu doğrultuda, oy kullanım hacmi aşağıdaki formül ile belirlenmiştir.

$$\text{İşlem Hacmi} = \frac{\text{Toplam Oy Sayısı}}{\text{Toplam Süre (s)}}$$

32-bit anahtar uzunluğuyla sıralı oy kullanma performans analizi Tablo 4.9.'da gösterilmektedir. Tablo 4.9.'daki gaz miktarı ve gaz maliyeti, seçmen başına düşen ortalama değerleri göstermektedir. Toplam süre, bir seçim süreci boyunca tüm seçmenlerin bireysel oy kullanma işlemleri için harcadıkları sürelerin toplamını ifade eder. İşlem hacmi ise, sistemin saniye ya da dakika bazında işleyebileceği ortalama oy sayısını ifade etmektedir.

**Tablo 4. 9.** Akıllı sözleşme tabanlı 32-bit Paillier ile oy kullanma performans analizi.

| Seçim | Seçmen Sayısı | Kullanılan Gaz | Gaz Maliyeti (USD) | Toplam Süre (s) | İşlem Hacmi (oy/s) | İşlem Hacmi (oy/dk) |
|-------|---------------|----------------|--------------------|-----------------|--------------------|---------------------|
| 1     | 10            | 589.781        | 1,46               | 3,65            | 2,97               | 178                 |
| 2     | 100           | 587.713        | 1,46               | 31,75           | 3,56               | 213                 |
| 3     | 500           | 586.440        | 1,46               | 149,25          | 3,50               | 210                 |
| 4     | 1.000         | 586.406        | 1,46               | 301,20          | 3,55               | 213                 |

#### 4.2.1.2. JavaScript Tabanlı 2048-bit Paillier ile Oy Verme

İlk çalışmalar sonucunda 32-bit anahtar uzunluğu yetersiz görülmüş ve 2048-bit anahtar uzunluğuna geçiş gerçekleştirilmiştir. Seçmen sayısı, akıllı sözleşmelerin gaz kullanım limiti ve Solidity dilinin değişken boyutları sebebiyle 2048-bit anahtar uzunluğuyla homomorfik toplama işlemi, sunucu tarafında gerçekleştirilememiştir. Şifreleme işlemi sonucu oluşan veriler üzerinde yapılacak matematiksel işlemler, akıllı sözleşmelerin gaz kullanım limitinin aşılmasına neden olmaktadır. Bu nedenle; mimari tasarımda değişiklik yapılarak akıllı sözleşmeler üzerinde yapılan homomorfik toplama ve şifre çözme işlemlerinin ağ dışı (off-chain) olacak şekilde JavaScript ile yapılmasına karar verilmiştir.

Böylece, seçmenin oyu şifrelenmiş halde byte değişkenlerine bölünmüş ve bu şekilde akıllı sözleşmeler tarafından blokzincire kaydedilmiştir. Homomorfik toplamın elde edilmesi ve bu toplam üzerindeki şifreli verilerin çözülerek açık sonuca ulaşılabilmesi için oylar akıllı sözleşmeden ağ dışına aktarılmıştır. Böylece, ilgili işlemler ağ dışında, JavaScript programlama dili kullanılarak gerçekleştirilmiştir. Bu doğrultuda, oy kullanım hacmi aşağıdaki formül ile hesaplanmıştır.

$$\text{İşlem Hacmi} = \frac{\text{Toplam Oy Sayısı}}{\text{Toplam Süre (s)}}$$

Bu işlem neticesinde, Tablo 4.10.'da gösterilen şekilde sistemin pek çok bölümünde performans değişiklikleri olmuştur. Tablo 4.10.'daki gaz miktarı ve gaz maliyeti, seçmen başına düşen ortalama değerleri göstermektedir. Toplam süre, seçmenlerin bireysel olarak oy kullanım sürelerinin toplamını ifade etmektedir. İşlem hacmi ise, sistemin işlemler neticesinde saniye veya dakika bazında işleyebileceği ortalama oy sayısını ifade etmektedir.

**Tablo 4. 10.** JavaScript tabanlı 2048-bit Paillier ile oy kullanma performans analizi.

| Seçim | Seçmen Sayısı | Kullanılan Gaz | Gaz Maliyeti (USD) | Toplam Süre (s) | İşlem Hacmi (oy/s) | İşlem Hacmi (oy/dk) |
|-------|---------------|----------------|--------------------|-----------------|--------------------|---------------------|
| 1     | 10            | 828.847        | 2,06               | 3,37            | 2,74               | 164                 |
| 2     | 100           | 825.591        | 2,05               | 28,09           | 3,15               | 189                 |
| 3     | 500           | 825.098        | 2,05               | 142,86          | 3,35               | 201                 |
| 4     | 1.000         | 825.255        | 2,05               | 281,69          | 3,32               | 199                 |

#### 4.2.1.3. JavaScript Tabanlı Eşzamanlı 2048-bit Paillier ile Oy Verme

Sistemin eşzamanlı oy kullanma performansının değerlendirilmesi gerekmektedir. Bu bağlamda, eşzamanlı olarak sırasıyla 1, 2, 5 ve 10 adet seçmenin oy kullanması sağlanmış ve böylece sistemin göstermiş olduğu performans ölçülmüştür. Eşzamanlı olarak gerçekleştirilen oy kullanma işleminden elde edilen veriler Tablo 4.11.'de gösterilmektedir. Bu işlem neticesinde, sıralı oy kullanımı ve eşzamanlı oy kullanımı arasında performans farklılıkları olduğu gözlemlenmiştir.

Tablo 4.11.'de yer alan işlem hacmi, sistemin eşzamanlı işlemler sonucunda saniye veya dakika bazında işleyebileceği ortalama oy sayısını ifade etmektedir. İşlem hacmi aşağıdaki formül ile hesaplanmıştır.

$$\text{İşlem Hacmi} = \frac{\text{Toplam Oy Sayısı}}{\text{Toplam Süre (s)}}$$

Eşzamanlı değeri, belirli bir anda aynı anda yürütülen işlem veya görev sayısını belirtmektedir. Toplam süre, bir seçim süreci boyunca tüm seçmenlerin oy kullanma işlemleri için harcadığı sürelerin toplamını ifade eder. Eşzamanlı alanında tanımlanan 1 değeri, işlemlerin sıralı bir biçimde gerçekleştirilmesini ifade etmektedir. Bu durum, daha önceki testlerde elde edilen sıralı işlem verileriyle tutarlı sonuçların ortaya çıkmasına yol açmaktadır.

**Tablo 4. 11.** JavaScript tabanlı eşzamanlı 2048-bit Paillier ile oy kullanma performans analizi.

| Seçim | Seçmen Sayısı | Eşzamanlı (x) | İşlem Hacmi (oy/s) | Toplam Süre (s) | İşlem Hacmi (oy/dk) |
|-------|---------------|---------------|--------------------|-----------------|---------------------|
| 1     | 10            | 1             | 2,73               | 3,66            | 164                 |
| 2     | 10            | 2             | 4,04               | 2,48            | 242                 |
| 3     | 10            | 5             | 4,55               | 2,20            | 273                 |
| 4     | 10            | 10            | 5,25               | 1,90            | 315                 |
| 5     | 100           | 1             | 3,24               | 30,87           | 194                 |
| 6     | 100           | 2             | 4,02               | 24,90           | 241                 |
| 7     | 100           | 5             | 4,65               | 21,49           | 279                 |
| 8     | 100           | 10            | 5,05               | 19,81           | 303                 |
| 9     | 500           | 1             | 3,24               | 154,19          | 195                 |
| 10    | 500           | 2             | 3,99               | 125,26          | 240                 |
| 11    | 500           | 5             | 4,70               | 106,36          | 282                 |
| 12    | 500           | 10            | 5,15               | 97,15           | 309                 |
| 13    | 1.000         | 1             | 3,21               | 311,46          | 193                 |
| 14    | 1.000         | 2             | 3,94               | 253,66          | 237                 |
| 15    | 1.000         | 5             | 4,63               | 216,09          | 278                 |
| 16    | 1.000         | 10            | 4,98               | 200,76          | 299                 |

#### 4.2.2. Kriptografik Hash Üretim Performansı

Oylama sisteminde, commitment ve nullifier üretebilmek için hash fonksiyonu kullanılmaktadır. Bu durum, bir seçmenin oy kullanma işlemi sırasında verilerin iki ayrı hash işleminden geçtiğini ortaya koymaktadır. Bu işlemler için Poseidon hash fonksiyonu tercih edilmiştir. Fonksiyonun sıralı ve eşzamanlı yürütme kapasitesinin belirlenmesi, sistemin işlem verimliliği ve ölçeklenebilirliğinin değerlendirilmesine katkı sağlayacaktır. Bu bağlamda, değerlendirme verileri oluşturabilmek için test ve analiz işlemleri gerçekleştirilmiştir. Test ve analiz işlemleri, JavaScript ve Bash tabanlı geliştirilen bir servis aracılığıyla yapılmıştır. Sistemde hem commitment hem de nullifier verileri için ayrı ayrı hash üretim işlemleri gerçekleştirilmektedir. Her biri kendi içinde iki adet hash fonksiyonunu içerse de performans değerlendirmelerinde bu işlemler, bir seçmen için commitment ve nullifier olmak üzere iki ayrı hash işlemi olarak ele alınmıştır.

##### 4.2.2.1. Commitment ve Nullifier Hash Üretimi

Bir hash işleminin gerçekleştirilmesi, oylama sürecinin hızını belirleyecek temel öğelerden birisidir. Bu nedenle, değişken aday sayısı ile farklı seçimler oluşturulmuş ve bu seçimlerde her aday için hash üretme işlemi gerçekleştirilmiştir. Bu çalışmada amaçlanan, sıralı biçimde yürütülen hash üretim sürecine ilişkin ortalama hızın tespit edilmesidir. Gerçekleştirilen her işlem, ilgili seçim özelinde ayrı ayrı değerlendirilmektedir. Ortalama hız

belirlendikten sonra, seçmen sayısına bağlı olarak işlemler için gerekli toplam süre hesaplanmaktadır. Bu toplam süre kullanılarak, sistemin saniyede ve dakikada işleyebileceği üretim kapasitesini ifade eden işlem hacmi belirlenebilecektir. Saniyelik üretim kapasitesini temsil eden işlem hacminin hesaplanması için şu formül kullanılmaktadır:

$$\text{İşlem Hacmi} \left( \frac{\text{hash}}{\text{sn}} \right) = \frac{\text{Toplam Hash}(x) * 1000}{\text{Toplam Süre}}$$

Her bir seçmen için iki ayrı hash işlemi gerçekleştirileceğinden, toplam hash üretim sayısı, seçmen sayısının iki katı olarak hesaplanmaktadır. Sıralı oy kullanma işlemleri temel alınarak yapılan ölçümler sonucunda, ortalama hash üretim hızı ve saniyelik üretim kapasitesi Tablo 4.12.'de gösterilmektedir.

**Tablo 4. 12.** Poseidon hash fonksiyonunun oy kullanma performans analizi

| Seçim | Seçmen Sayısı | Toplam Hash (x) | Seçmen Başına (ms) | Toplam Süre (ms) | İşlem Hacmi (hash/s) | İşlem Hacmi (hash/dk) |
|-------|---------------|-----------------|--------------------|------------------|----------------------|-----------------------|
| 1     | 1             | 2               | 0,0795             | 0,0795           | 25.157               | 1.509.434             |
| 2     | 10            | 20              | 0,0786             | 0,7860           | 25.445               | 1.526.718             |
| 3     | 50            | 100             | 0,0788             | 3,9400           | 25.381               | 1.522.843             |
| 4     | 100           | 200             | 0,0811             | 8,1100           | 24.661               | 1.479.655             |
| 5     | 500           | 1.000           | 0,0815             | 40,7500          | 24.540               | 1.472.393             |
| 6     | 1.000         | 2.000           | 0,0830             | 83,0000          | 24.096               | 1.445.783             |

#### 4.2.2.2. Eşzamanlı Commitment ve Nullifier Hash Üretimi

Sistem performansının doğru bir şekilde değerlendirilebilmesi için sistemin eşzamanlı hash üretim hızı ile saniyelik üretim kapasitesinin belirlenmesi gerekmektedir. Bu amaçla, sırasıyla 1, 2, 5 ve 10 adet seçmenin eşzamanlı olarak hash ürettiği varsayılmış ve ilgili test ile analizler gerçekleştirilmiştir. Saniyelik üretim kapasitesini temsil eden işlem hacminin hesaplanması için şu formül kullanılmaktadır:

$$\text{İşlem Hacmi} \left( \frac{\text{hash}}{\text{sn}} \right) = \frac{\text{Toplam Hash} (x) * 1000}{\text{Max Worker Time} (ms)}$$

Max Worker Time, eşzamanlı olarak gerçekleştirilen işlemlerde JavaScript tarafındaki en yavaş eşzamanlı işlemin süresini temsil etmektedir. Böylece hesaplama yaparken en kötü durum (worst case) düşünülerek hesaplama yapılmıştır. Eşzamanlı oy kullanma işlemleri temel alınarak yapılan ölçümler sonucunda, ortalama hash üretim hızı ve saniyelik üretim kapasitesi Tablo 4.13.'de gösterilmektedir.

Tablo 4.13.'deki eşzamanlı ifadesi, belirli bir anda aynı anda yürütülen işlem sayısını ifade etmektedir. Toplam hash, ilgili seçim süresince oluşturulan toplam hash sayısını ifade etmektedir. Seçmen sayısı iki ile çarpılmaktadır; çünkü her seçmen oy kullanırken veriler iki kez hash işlemine tabi tutulmaktadır. Seçmen başına düşen değer, ilgili seçimde bir seçmenin ortalama hash üretim süresini göstermektedir. Seçmen başına düşen değer ile seçmen sayısının çarpılması sonucu, ilgili seçim süresince üretilmesi gereken hash değerlerinin toplam üretim süresi hesaplanabilmektedir. İşlem hacmi ise saniye ve dakika başına üretililecek hash sayısını belirtmektedir.

**Tablo 4. 13.** Poseidon hash fonksiyonu kullanılarak gerçekleştirilen eşzamanlı hash üretimi.

| Seçim | Seçmen Sayısı | Eşzamanlı (x) | Toplam Hash (x) | Seçmen Başına (ms) | Toplam Süre (ms) | Max Worker Time (ms) | İşlem Hacmi (hash/s) | İşlem Hacmi (hash/dk) |
|-------|---------------|---------------|-----------------|--------------------|------------------|----------------------|----------------------|-----------------------|
| 1     | 50            | 2             | 100             | 0,1669             | 8,345            | 13.621               | 13.624               | 817.439               |
| 2     | 50            | 3             | 100             | 0,1819             | 9,095            | 12.020               | 33.003               | 1.980.198             |
| 3     | 50            | 5             | 100             | 0,2084             | 10,42            | 9.986                | 48.077               | 2.884.615             |
| 4     | 50            | 10            | 100             | 0,3448             | 17,24            | 8.258                | 58.140               | 3.488.372             |
| 5     | 100           | 2             | 200             | 0,1553             | 15,53            | 12.997               | 25.773               | 1.546.392             |
| 6     | 100           | 3             | 200             | 0,1678             | 16,78            | 12.176               | 35.778               | 2.146.691             |
| 7     | 100           | 5             | 200             | 0,1781             | 17,81            | 11.507               | 56.180               | 3.370.787             |
| 8     | 100           | 10            | 200             | 0,2977             | 29,77            | 10.040               | 67.114               | 4.026.846             |
| 9     | 500           | 2             | 1.000           | 0,1445             | 72,25            | 14.011               | 27.685               | 1.661.130             |
| 10    | 500           | 3             | 1.000           | 0,1472             | 73,6             | 14.091               | 40.750               | 2.444.988             |
| 11    | 500           | 5             | 1.000           | 0,1542             | 77,1             | 13.255               | 64.851               | 3.891.051             |
| 12    | 500           | 10            | 1.000           | 0,1628             | 81,4             | 12.515               | 122.850              | 7.371.007             |
| 13    | 1.000         | 2             | 2.000           | 0,1433             | 143,3            | 13.979               | 27.913               | 1.674.808             |
| 14    | 1.000         | 3             | 2.000           | 0,1452             | 145,2            | 13.880               | 41.314               | 2.478.827             |
| 15    | 1.000         | 5             | 2.000           | 0,1517             | 151,7            | 13.557               | 65.898               | 3.953.871             |
| 16    | 1.000         | 10            | 2.000           | 0,2236             | 223,6            | 13.076               | 89.445               | 5.366.726             |

#### 4.2.3. Kanıt Üretme Performans Analizi

Sistem, oy kullanma aşamasında sıfır bilgi ispatı kanıtları üretmektedir. Kanıtların üretilme süreci belirli aşamalardan oluşmaktadır. Aşamaların gerçekleştirilmesi belirli bir zaman gerektirmektedir. Bu zamanın, hızlı veya yavaş olması sistemin performansını olumlu veya olumsuz etkilemektedir. Bu nedenle, sistemin kanıt üretme kapasitesinin sıralı ve eşzamanlı olarak belirlenmesi, sistemin işlem verimliliği ve ölçeklenebilirliğinin değerlendirilmesine katkı sağlayacaktır. Bu bağlamda, değerlendirme verileri oluşturabilmek için test ve analiz işlemleri gerçekleştirilmiştir. Bu işlemler, JavaScript ve Bash tabanlı geliştirilen bir servis vasıtasıyla yapılmıştır.

#### 4.2.3.1. JavaScript Tabanlı Kanıt Üretim Performans Analizi

İlk aşamada, değişken aday sayısıyla farklı seçimler oluşturulmuştur. Seçimlerde seçmenlerin sırayla oy kullanması sağlanmıştır. Böylece, sıralı olarak gerçekleştirilen kanıt üretim işlemi için ortalama hız belirlenmiştir. Kanıt üretim süresi tespit edildikten sonra işlem hacmi hesaplaması yapılmıştır. İşlem hacmi hesaplaması için şu formül kullanılmaktadır:

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi}} \times 1000 \times 60$$

Her bir seçim senaryosu kapsamında, seçmenler tarafından üretilen kanıtların süre bazında ortalaması alınarak kanıt üretim süresi belirlenmiştir. Bu ortalama sürelerle dayanarak hesaplanan, dakikada üretililecek toplam kanıt sayısı ise sistemin işlem hacmini ifade etmektedir. Burada gerçekleştirilen tüm hesaplamalar, ilgili seçim özelinde ayrı ayrı değerlendirilmektedir. Hesaplamalar sonucunda Tablo 4.14.'de gösterilen veriler elde edilmektedir.

**Tablo 4. 14.** JavaScript ile kanıt üretiminin sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Kanıt Üretim Süresi (ms) | İşlem Hacmi (kanıt/dk) |
|-------|-------------|---------------|--------------------------|------------------------|
| 1     | 2           | 5             | 292,00                   | 205                    |
| 2     | 2           | 10            | 214,60                   | 280                    |
| 3     | 2           | 50            | 234,20                   | 256                    |
| 4     | 2           | 100           | 221,70                   | 271                    |
| 5     | 2           | 500           | 218,60                   | 274                    |
| 6     | 2           | 1.000         | 216,70                   | 277                    |
| 7     | 3           | 5             | 217,70                   | 276                    |
| 8     | 3           | 10            | 219,80                   | 273                    |
| 9     | 3           | 50            | 218,10                   | 275                    |
| 10    | 3           | 100           | 218,30                   | 275                    |
| 11    | 3           | 500           | 219,00                   | 274                    |
| 12    | 3           | 1.000         | 218,10                   | 275                    |
| 13    | 5           | 5             | 219,20                   | 274                    |
| 14    | 5           | 10            | 223,10                   | 269                    |
| 15    | 5           | 50            | 218,70                   | 274                    |
| 16    | 5           | 100           | 218,50                   | 275                    |
| 17    | 5           | 500           | 218,30                   | 275                    |
| 18    | 5           | 1.000         | 220,40                   | 272                    |
| 19    | 10          | 5             | 220,10                   | 273                    |
| 20    | 10          | 10            | 218,90                   | 274                    |
| 21    | 10          | 50            | 219,00                   | 274                    |
| 22    | 10          | 100           | 219,80                   | 273                    |
| 23    | 10          | 500           | 220,80                   | 272                    |

|    |    |       |        |     |
|----|----|-------|--------|-----|
| 24 | 10 | 1.000 | 218,70 | 274 |
|----|----|-------|--------|-----|

#### 4.2.3.2. JavaScript Tabanlı Eşzamanlı Kanıt Üretim Performans Analizi

Oylama sürecinde performans karşılaştırması yapılabilmesi açısından, sistemdeki eşzamanlı kanıt üretim performansının tespit edilmesi gerekmektedir. Bu bağlamda, 2, 5 ve 10 adet seçmenin eşzamanlı olarak oy kullanacağı ve bunun neticesinde kanıt üreteceği varsayılmıştır. Bu doğrultuda eşzamanlı test işlemleri gerçekleştirilmiştir. Bu işlemler sonucunda eşzamanlı ortalama kanıt üretim süresi tespit edilmiş ve aşağıdaki formülle işlem hacmi hesaplanmıştır.

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi}} \times 1000 \times 60$$

Her seçim senaryosunda, seçmenler tarafından üretilen kanıtların sürelerinin ortalaması, sistemin kanıt üretim süresi olarak kabul edilmektedir. İşlem hacmi ise, dakikada üretilebilecek kanıt sayısını ifade etmektedir. Bu işlemler sonucunda elde edilen veriler Tablo 4.15.'de gösterilmektedir.

**Tablo 4. 15.** JavaScript ile gerçekleştirilen eşzamanlı kanıt üretiminin sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Eşzamanlı (x) | Kanıt Üretim Süresi (ms) | İşlem Hacmi (dk) |
|-------|-------------|---------------|---------------|--------------------------|------------------|
| 1     | 2           | 5             | 2             | 221,70                   | 271              |
| 2     | 2           | 5             | 5             | 217,50                   | 276              |
| 3     | 2           | 5             | 10            | 215,30                   | 279              |
| 4     | 2           | 10            | 2             | 215,50                   | 278              |
| 5     | 2           | 10            | 5             | 223,70                   | 268              |
| 6     | 2           | 10            | 10            | 217,80                   | 275              |
| 7     | 2           | 1.000         | 2             | 214,90                   | 279              |
| 8     | 2           | 1.000         | 5             | 217,90                   | 275              |
| 9     | 2           | 1.000         | 10            | 216,80                   | 277              |
| 10    | 3           | 5             | 2             | 216,70                   | 277              |
| 11    | 3           | 5             | 5             | 235,20                   | 255              |
| 12    | 3           | 5             | 10            | 228,30                   | 263              |
| 13    | 3           | 10            | 2             | 222,20                   | 270              |
| 14    | 3           | 10            | 5             | 217,60                   | 276              |
| 15    | 3           | 10            | 10            | 218,10                   | 275              |
| 16    | 3           | 1.000         | 2             | 216,80                   | 277              |
| 17    | 3           | 1.000         | 5             | 218,40                   | 275              |
| 18    | 3           | 1.000         | 10            | 217,40                   | 276              |
| 19    | 5           | 5             | 2             | 216,20                   | 278              |
| 20    | 5           | 5             | 5             | 218,60                   | 274              |

|    |    |       |    |        |     |
|----|----|-------|----|--------|-----|
| 21 | 5  | 5     | 10 | 216,70 | 277 |
| 22 | 5  | 10    | 2  | 217,10 | 276 |
| 23 | 5  | 10    | 5  | 218,50 | 275 |
| 24 | 5  | 10    | 10 | 217,10 | 276 |
| 25 | 5  | 50    | 2  | 216,00 | 278 |
| 26 | 5  | 50    | 5  | 218,40 | 275 |
| 27 | 5  | 50    | 10 | 216,40 | 277 |
| 28 | 5  | 100   | 2  | 216,50 | 277 |
| 29 | 5  | 100   | 5  | 218,50 | 275 |
| 30 | 5  | 100   | 10 | 215,80 | 278 |
| 31 | 5  | 500   | 2  | 216,50 | 277 |
| 32 | 5  | 500   | 5  | 217,90 | 275 |
| 33 | 5  | 500   | 10 | 219,30 | 274 |
| 34 | 5  | 1.000 | 2  | 217,60 | 276 |
| 35 | 5  | 1.000 | 5  | 219,10 | 274 |
| 36 | 5  | 1.000 | 10 | 218,10 | 275 |
| 37 | 10 | 5     | 2  | 218,70 | 274 |
| 38 | 10 | 5     | 5  | 220,00 | 273 |
| 39 | 10 | 5     | 10 | 219,40 | 273 |
| 40 | 10 | 10    | 2  | 217,20 | 276 |
| 41 | 10 | 10    | 5  | 219,80 | 273 |
| 42 | 10 | 10    | 10 | 216,70 | 277 |
| 43 | 10 | 50    | 2  | 218,60 | 274 |
| 44 | 10 | 50    | 5  | 218,40 | 275 |
| 45 | 10 | 50    | 10 | 218,20 | 275 |
| 46 | 10 | 100   | 2  | 217,30 | 276 |
| 47 | 10 | 100   | 5  | 219,00 | 274 |
| 48 | 10 | 100   | 10 | 217,50 | 276 |
| 49 | 10 | 500   | 2  | 219,20 | 274 |
| 50 | 10 | 500   | 5  | 219,50 | 273 |
| 51 | 10 | 500   | 10 | 220,60 | 272 |
| 52 | 10 | 1.000 | 2  | 217,00 | 276 |
| 53 | 10 | 1.000 | 5  | 223,60 | 268 |
| 54 | 10 | 1.000 | 10 | 218,70 | 274 |

#### 4.2.4. Kanıt Doğrulama Performans Analizi

Üretilen kanıtlar hem JavaScript hem de akıllı sözleşme içerisinde doğrulanmaktadır. Doğrulama işlemi için belirli bir süre harcanması gerekmektedir. Bu süre, oylama işleminin hızını etkileyen unsurlardan birisidir. Bu bağlamda, kanıt doğrulama hızını tespit edebilmek için değişken aday ve seçmen sayısına göre örnek seçim senaryoları oluşturulmuştur. Bu seçim senaryoları kullanılarak kanıt doğrulama hızı analiz edilmiştir. Gerçekleştirilen her işlem, ilgili seçim için ayrı ayrı değerlendirilmektedir. Kanıt doğrulama hızı tespit edilebilmesi için öncelikle kanıtın üretilmesi gerektiğinden, testler kanıt üretme bölümüyle birlikte

tamamlanmıştır. Bu nedenle, kanıt üretim süreleri aynı değerlerle ifade edilmektedir. Tüm test ve analiz işlemleri, JavaScript ve Bash tabanlı geliştirilen bir servis vasıtasıyla yapılmıştır.

#### 4.2.4.1. JavaScript Tabanlı Kanıt Doğrulama Performans Analizi

JavaScript tabanlı doğrulama süreci kapsamında, dakika başına işlem hacmi; bir dakikalık süre zarfında doğrulanabilecek toplam kanıt sayısını göstermektedir. Bu değer hesaplanmasında aşağıda verilen formül kullanılmıştır:

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi} + \text{Kanıt Doğrulama Süreci}} \times 1000 \times 60$$

Sıralı kanıt üretim sürelerinin belirlendiği test senaryolarına paralel olarak, JavaScript tabanlı kanıt doğrulama işlemleri seçim bazında yürütülmüştür. Her seçimde seçmen başına gerçekleştirilen doğrulama işlemlerinin süreleri alınarak, ortalama doğrulama süresi hesaplanmış ve sistemin doğrulama performansı bu değerler üzerinden analiz edilmiştir. Bu hesaplama sonucunda, işlem hacmi olarak ifade edilen dakika başına doğrulanabilen toplam kanıt sayısı belirlenmiştir ve Tablo 4.16’da gösterilmektedir.

Tablo 4.16’daki kanıt üretim süresi, seçim bazında her seçmen için gereken ortalama kanıt üretim süresini göstermektedir; bu değer, tüm seçmenler için bireysel olarak gereken sürenin ortalamasını ifade etmektedir. Benzer şekilde, doğrulama süresi de her seçmen için gereken ortalama kanıt doğrulama süresini ifade etmekte olup, tüm seçmenler için bireysel olarak gereken sürenin ortalamasıdır. Toplam bölümü ise, bir seçmenin bireysel kanıt üretim süresi ile doğrulama süresinin toplamını ifade etmektedir. Bu değer, bir seçmenin kanıt doğrulama sürecini bütüncül olarak değerlendirmek için kullanılmaktadır.

**Tablo 4. 16.** JavaScript ile kanıt doğrulamanın sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Kanıt Üretim Süresi (ms) | Doğrulama (ms) | Toplam (ms) | İşlem Hacmi (dk) |
|-------|-------------|---------------|--------------------------|----------------|-------------|------------------|
| 1     | 2           | 5             | 292,00                   | 5,90           | 297,90      | 201              |
| 2     | 2           | 10            | 214,60                   | 5,60           | 220,20      | 272              |
| 3     | 2           | 50            | 234,20                   | 5,80           | 240,00      | 250              |
| 4     | 2           | 100           | 221,70                   | 6,20           | 227,90      | 263              |
| 5     | 2           | 500           | 218,60                   | 5,60           | 224,20      | 268              |
| 6     | 2           | 1.000         | 216,70                   | 5,50           | 222,20      | 270              |
| 7     | 3           | 5             | 217,70                   | 5,60           | 223,30      | 269              |
| 8     | 3           | 10            | 219,80                   | 5,60           | 225,40      | 266              |
| 9     | 3           | 50            | 218,10                   | 5,60           | 223,70      | 268              |

|    |    |       |        |      |        |     |
|----|----|-------|--------|------|--------|-----|
| 10 | 3  | 100   | 218,30 | 5,60 | 223,90 | 268 |
| 11 | 3  | 500   | 219,00 | 5,60 | 224,60 | 267 |
| 12 | 3  | 1.000 | 218,10 | 5,70 | 223,80 | 268 |
| 13 | 5  | 5     | 219,20 | 5,70 | 224,90 | 267 |
| 14 | 5  | 10    | 223,10 | 5,70 | 228,80 | 262 |
| 15 | 5  | 50    | 218,70 | 5,80 | 224,50 | 267 |
| 16 | 5  | 100   | 218,50 | 5,60 | 224,10 | 268 |
| 17 | 5  | 500   | 218,30 | 5,90 | 224,20 | 268 |
| 18 | 5  | 1.000 | 220,40 | 5,60 | 226,00 | 265 |
| 19 | 10 | 5     | 220,10 | 5,70 | 225,80 | 266 |
| 20 | 10 | 10    | 218,90 | 5,60 | 224,50 | 267 |
| 21 | 10 | 50    | 219,00 | 5,60 | 224,60 | 267 |
| 22 | 10 | 100   | 219,80 | 5,60 | 225,40 | 266 |
| 23 | 10 | 500   | 220,80 | 5,70 | 226,50 | 265 |
| 24 | 10 | 1.000 | 218,70 | 5,70 | 224,40 | 267 |

Doğrulama sürecine ilişkin performans ölçümlerinin doğru yapılabilmesi ve sistemin eşzamanlı yük altında nasıl davrandığının anlaşılabilmesi için JavaScript ile kanıt doğrulama sürecinin eşzamanlı olarak değerlendirilmesi gerekmektedir. JavaScript ile dakika başına eşzamanlı doğrulama sürecine ilişkin işlem hacmi, dakika başına doğrulanabilecek eşzamanlı kanıt sayısını göstermektedir. Bu değer hesaplanmasında aşağıdaki formül kullanılmıştır.

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi} + \text{Kanıt Doğrulama Süreci}} \times 1000 \times 60$$

Eşzamanlı kanıt üretim sürelerinin belirlendiği test senaryolarına ek olarak, JavaScript tabanlı eşzamanlı kanıt doğrulama işlemleri de seçim bazında yürütülmüştür. Her seçimde, seçmen başına gerçekleştirilen doğrulama işlemlerinin süreleri üzerinden ortalama değerler elde edilmiş; bu ortalama süreler doğrultusunda sistemin doğrulama performansı analiz edilmiştir. Eşzamanlı testlerden elde edilen verilerle, kanıt üretim ve doğrulama süreleri belirlenmiştir. Bu sürelerin toplamı, sistemdeki eşzamanlı kanıt doğrulama süresini bütüncül olarak göstermektedir. Hesaplanan toplam süreye dayanarak, dakika başına gerçekleştirilebilecek doğrulama sayısını gösteren işlem hacmi elde edilmiş ve ilgili hesaplamalar Tablo 4.17.'de verilmiştir.

**Tablo 4. 17.** JavaScript ile eşzamanlı kanıt doğrulamanın sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Eşzamanlı (x) | Kanıt Üretim Süresi (ms) | Doğrulama (ms) | Toplam (ms) | İşlem Hacmi (dk) |
|-------|-------------|---------------|---------------|--------------------------|----------------|-------------|------------------|
| 1     | 2           | 5             | 2             | 221,70                   | 5,90           | 228,70      | 262              |
| 2     | 2           | 5             | 5             | 217,50                   | 5,70           | 227,50      | 264              |

|    |    |       |    |        |      |         |     |
|----|----|-------|----|--------|------|---------|-----|
| 3  | 2  | 5     | 10 | 215,30 | 5,90 | 230,30  | 261 |
| 4  | 2  | 10    | 2  | 215,50 | 5,80 | 227,50  | 264 |
| 5  | 2  | 10    | 5  | 223,70 | 5,90 | 238,70  | 251 |
| 6  | 2  | 10    | 10 | 217,80 | 5,90 | 237,80  | 252 |
| 7  | 2  | 1.000 | 2  | 214,90 | 5,60 | 1216,90 | 49  |
| 8  | 2  | 1.000 | 5  | 217,90 | 5,70 | 1222,90 | 49  |
| 9  | 2  | 1.000 | 10 | 216,80 | 5,60 | 1226,80 | 49  |
| 10 | 3  | 5     | 2  | 216,70 | 5,50 | 223,70  | 268 |
| 11 | 3  | 5     | 5  | 235,20 | 5,70 | 245,20  | 245 |
| 12 | 3  | 5     | 10 | 228,30 | 5,80 | 243,30  | 247 |
| 13 | 3  | 10    | 2  | 222,20 | 5,80 | 234,20  | 256 |
| 14 | 3  | 10    | 5  | 217,60 | 6,00 | 232,60  | 258 |
| 15 | 3  | 10    | 10 | 218,10 | 5,80 | 238,10  | 252 |
| 16 | 3  | 1.000 | 2  | 216,80 | 5,70 | 1218,80 | 49  |
| 17 | 3  | 1.000 | 5  | 218,40 | 5,60 | 1223,40 | 49  |
| 18 | 3  | 1.000 | 10 | 217,40 | 5,70 | 1227,40 | 49  |
| 19 | 5  | 5     | 2  | 216,20 | 5,70 | 223,20  | 269 |
| 20 | 5  | 5     | 5  | 218,60 | 5,70 | 228,60  | 262 |
| 21 | 5  | 5     | 10 | 216,70 | 5,70 | 231,70  | 259 |
| 22 | 5  | 10    | 2  | 217,10 | 5,80 | 229,10  | 262 |
| 23 | 5  | 10    | 5  | 218,50 | 5,70 | 233,50  | 257 |
| 24 | 5  | 10    | 10 | 217,10 | 5,60 | 237,10  | 253 |
| 25 | 5  | 50    | 2  | 216,00 | 5,60 | 268,00  | 224 |
| 26 | 5  | 50    | 5  | 218,40 | 5,60 | 273,40  | 219 |
| 27 | 5  | 50    | 10 | 216,40 | 5,70 | 276,40  | 217 |
| 28 | 5  | 100   | 2  | 216,50 | 5,70 | 318,50  | 188 |
| 29 | 5  | 100   | 5  | 218,50 | 5,80 | 323,50  | 185 |
| 30 | 5  | 100   | 10 | 215,80 | 5,60 | 325,80  | 184 |
| 31 | 5  | 500   | 2  | 216,50 | 5,80 | 718,50  | 84  |
| 32 | 5  | 500   | 5  | 217,90 | 5,60 | 722,90  | 83  |
| 33 | 5  | 500   | 10 | 219,30 | 5,80 | 729,30  | 82  |
| 34 | 5  | 1.000 | 2  | 217,60 | 5,80 | 1219,60 | 49  |
| 35 | 5  | 1.000 | 5  | 219,10 | 5,70 | 1224,10 | 49  |
| 36 | 5  | 1.000 | 10 | 218,10 | 5,80 | 1228,10 | 49  |
| 37 | 10 | 5     | 2  | 218,70 | 5,80 | 225,70  | 266 |
| 38 | 10 | 5     | 5  | 220,00 | 5,70 | 230,00  | 261 |
| 39 | 10 | 5     | 10 | 219,40 | 5,70 | 234,40  | 256 |
| 40 | 10 | 10    | 2  | 217,20 | 5,70 | 229,20  | 262 |
| 41 | 10 | 10    | 5  | 219,80 | 5,70 | 234,80  | 256 |
| 42 | 10 | 10    | 10 | 216,70 | 5,80 | 236,70  | 253 |
| 43 | 10 | 50    | 2  | 218,60 | 5,70 | 270,60  | 222 |
| 44 | 10 | 50    | 5  | 218,40 | 5,60 | 273,40  | 219 |
| 45 | 10 | 50    | 10 | 218,20 | 5,80 | 278,20  | 216 |
| 46 | 10 | 100   | 2  | 217,30 | 5,80 | 319,30  | 188 |
| 47 | 10 | 100   | 5  | 219,00 | 5,80 | 324,00  | 185 |
| 48 | 10 | 100   | 10 | 217,50 | 5,70 | 327,50  | 183 |
| 49 | 10 | 500   | 2  | 219,20 | 5,70 | 721,20  | 83  |
| 50 | 10 | 500   | 5  | 219,50 | 5,70 | 724,50  | 83  |

|    |    |       |    |        |       |         |    |
|----|----|-------|----|--------|-------|---------|----|
| 51 | 10 | 500   | 10 | 220,60 | 5,80  | 730,60  | 82 |
| 52 | 10 | 1.000 | 2  | 217,00 | 5,60  | 1219,00 | 49 |
| 53 | 10 | 1.000 | 5  | 223,60 | 12,00 | 1228,60 | 49 |
| 54 | 10 | 1.000 | 10 | 218,70 | 5,60  | 1228,70 | 49 |

#### 4.2.4.2. Akıllı Sözleşme Tabanlı Kanıt Doğrulama

Akıllı sözleşme aracılığıyla gerçekleştirilen dakika başına kanıt doğrulama süreci, bir dakikalık süre içinde doğrulanabilen toplam kanıt sayısını ifade etmektedir. Bu değerin hesaplanmasında aşağıdaki formül kullanılmıştır.

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi} + \text{Kanıt Doğrulama Süreci}} \times 1000 \times 60$$

Kanıt üretme performans analizinde yer alan sıralı kanıt üretim testleri kapsamında, akıllı sözleşme tabanlı doğrulama işlemleri seçim bazında gerçekleştirilmiş ve seçmen bazında ortalama doğrulama süresi hesaplanmıştır. Bu hesaplama sonucunda, işlem hacmi olarak ifade edilen dakika başına doğrulanabilen toplam kanıt sayısı belirlenmiştir ve Tablo 4.18.'de gösterilmektedir.

Tablo 4.18.'deki yer alan kanıt üretim süresi, seçim bazında her bir seçmen için gereken ortalama kanıt üretim süresini göstermektedir. Bu değer, tüm seçmenler için bireysel olarak hesaplanan kanıt üretim sürelerinin ortalamasını ifade etmektedir. Benzer şekilde, doğrulama süresi de her bir seçmen için gereken ortalama kanıt doğrulama süresini göstermekte olup, tüm seçmenler özelinde hesaplanan bireysel doğrulama sürelerinin ortalamasını yansıtmaktadır. Toplam bölümü ise, bir seçmenin bireysel kanıt üretim süresi ile doğrulama süresinin toplamını ifade etmektedir. Bu değer, bir seçmenin kanıt doğrulama sürecini bütüncül olarak değerlendirmek için kullanılmaktadır.

**Tablo 4. 18.** Akıllı sözleşmeyle kanıt doğrulamanın sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Kanıt Üretim Süresi (ms) | Doğrulama (ms) | Toplam (ms) | İşlem Hacmi (dk) |
|-------|-------------|---------------|--------------------------|----------------|-------------|------------------|
| 1     | 2           | 5             | 292,00                   | 17,90          | 309,90      | 194              |
| 2     | 2           | 10            | 214,60                   | 13,60          | 228,20      | 263              |
| 3     | 2           | 50            | 234,20                   | 15,30          | 249,50      | 240              |
| 4     | 2           | 100           | 221,70                   | 14,80          | 236,50      | 254              |
| 5     | 2           | 500           | 218,60                   | 14,00          | 232,60      | 258              |
| 6     | 2           | 1.000         | 216,70                   | 13,80          | 230,50      | 260              |
| 12    | 3           | 1.000         | 218,10                   | 14,40          | 232,50      | 258              |
| 16    | 5           | 100           | 218,50                   | 14,00          | 232,50      | 258              |

|    |    |       |        |       |        |     |
|----|----|-------|--------|-------|--------|-----|
| 17 | 5  | 500   | 218,30 | 14,00 | 232,30 | 258 |
| 18 | 5  | 1.000 | 220,40 | 13,60 | 234,00 | 256 |
| 19 | 10 | 5     | 220,10 | 14,10 | 234,20 | 256 |
| 20 | 10 | 10    | 218,90 | 14,00 | 232,90 | 258 |
| 21 | 10 | 50    | 219,00 | 13,70 | 232,70 | 258 |
| 22 | 10 | 100   | 219,80 | 13,70 | 233,50 | 257 |
| 23 | 10 | 500   | 220,80 | 14,30 | 235,10 | 255 |
| 24 | 10 | 1.000 | 218,70 | 13,30 | 232,00 | 259 |

Akıllı sözleşme tabanlı kanıt doğrulama sürecinin, yalnızca sıralı işlemler altında değil, eşzamanlı işlem yükü gibi gerçekçi senaryolar altında da değerlendirilmesi gerekmektedir. Bu doğrultuda ilgili test ve analizler gerçekleştirilmiştir. Bu bağlamda, seçim bazında doğrulama işlemi için gereken toplam süre tespit edilmiştir. Toplam süre kullanılarak işlem hacmini oluşturan dakika başına doğrulanabilecek kanıt sayısı hesaplanmıştır. İşlem hacmi hesaplanmasında aşağıdaki formül kullanılmaktadır.

$$\text{İşlem Hacmi} = \frac{1}{\text{Kanıt Üretim Süresi} + \text{Kanıt Doğrulama Süreci}} \times 1000 \times 60$$

Seçim bazında elde edilen ortalama eşzamanlı kanıt üretim süresi, akıllı sözleşme tabanlı eşzamanlı ortalama kanıt doğrulama süresi ve buna karşılık gelen işlem hacmi Tablo 4.19.'da gösterilmektedir.

**Tablo 4. 19.** Akıllı sözleşmeyle eşzamanlı kanıt doğrulamanın sistem üzerindeki analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Eşzamanlı (x) | Kanıt Üretim Süresi (ms) | Kanıt Doğrulama Süresi (ms) | Toplam (ms) | İşlem Hacmi (dk) |
|-------|-------------|---------------|---------------|--------------------------|-----------------------------|-------------|------------------|
| 1     | 2           | 5             | 2             | 221,70                   | 14,60                       | 228,70      | 262              |
| 2     | 2           | 5             | 5             | 217,50                   | 14,10                       | 227,50      | 264              |
| 3     | 2           | 5             | 10            | 215,30                   | 14,70                       | 230,30      | 261              |
| 4     | 2           | 10            | 2             | 215,50                   | 14,50                       | 227,50      | 264              |
| 5     | 2           | 10            | 5             | 223,70                   | 14,70                       | 238,70      | 251              |
| 6     | 2           | 10            | 10            | 217,80                   | 14,70                       | 237,80      | 252              |
| 16    | 2           | 1.000         | 2             | 214,90                   | 13,50                       | 1216,90     | 49               |
| 17    | 2           | 1.000         | 5             | 217,90                   | 14,70                       | 1222,90     | 49               |
| 18    | 2           | 1.000         | 10            | 216,80                   | 14,20                       | 1226,80     | 49               |
| 19    | 3           | 5             | 2             | 216,70                   | 13,80                       | 223,70      | 268              |
| 20    | 3           | 5             | 5             | 235,20                   | 14,60                       | 245,20      | 245              |
| 21    | 3           | 5             | 10            | 228,30                   | 14,30                       | 243,30      | 247              |
| 22    | 3           | 10            | 2             | 222,20                   | 14,80                       | 234,20      | 256              |
| 23    | 3           | 10            | 5             | 217,60                   | 14,10                       | 232,60      | 258              |
| 24    | 3           | 10            | 10            | 218,10                   | 14,60                       | 238,10      | 252              |
| 34    | 3           | 1.000         | 2             | 216,80                   | 14,40                       | 1218,80     | 49               |
| 35    | 3           | 1.000         | 5             | 218,40                   | 14,60                       | 1223,40     | 49               |

|    |    |       |    |        |       |         |     |
|----|----|-------|----|--------|-------|---------|-----|
| 36 | 3  | 1.000 | 10 | 217,40 | 14,80 | 1227,40 | 49  |
| 37 | 5  | 5     | 2  | 216,20 | 14,50 | 223,20  | 269 |
| 38 | 5  | 5     | 5  | 218,60 | 14,30 | 228,60  | 262 |
| 39 | 5  | 5     | 10 | 216,70 | 13,60 | 231,70  | 259 |
| 40 | 5  | 10    | 2  | 217,10 | 14,60 | 229,10  | 262 |
| 41 | 5  | 10    | 5  | 218,50 | 13,60 | 233,50  | 257 |
| 42 | 5  | 10    | 10 | 217,10 | 14,20 | 237,10  | 253 |
| 52 | 5  | 1.000 | 2  | 217,60 | 14,10 | 1219,60 | 49  |
| 53 | 5  | 1.000 | 5  | 219,10 | 14,00 | 1224,10 | 49  |
| 54 | 5  | 1.000 | 10 | 218,10 | 14,60 | 1228,10 | 49  |
| 55 | 10 | 5     | 2  | 218,70 | 14,10 | 225,70  | 266 |
| 56 | 10 | 5     | 5  | 220,00 | 14,10 | 230,00  | 261 |
| 57 | 10 | 5     | 10 | 219,40 | 13,90 | 234,40  | 256 |
| 58 | 10 | 10    | 2  | 217,20 | 14,10 | 229,20  | 262 |
| 59 | 10 | 10    | 5  | 219,80 | 14,10 | 234,80  | 256 |
| 60 | 10 | 10    | 10 | 216,70 | 15,30 | 236,70  | 253 |
| 61 | 10 | 50    | 2  | 218,60 | 14,20 | 270,60  | 222 |
| 62 | 10 | 50    | 5  | 218,40 | 14,00 | 273,40  | 219 |
| 63 | 10 | 50    | 10 | 218,20 | 13,70 | 278,20  | 216 |
| 64 | 10 | 100   | 2  | 217,30 | 14,80 | 319,30  | 188 |
| 65 | 10 | 100   | 5  | 219,00 | 14,20 | 324,00  | 185 |
| 66 | 10 | 100   | 10 | 217,50 | 14,10 | 327,50  | 183 |
| 67 | 10 | 500   | 2  | 219,20 | 14,40 | 721,20  | 83  |
| 68 | 10 | 500   | 5  | 219,50 | 13,70 | 724,50  | 83  |
| 69 | 10 | 500   | 10 | 220,60 | 14,30 | 730,60  | 82  |
| 70 | 10 | 1.000 | 2  | 217,00 | 14,00 | 1219,00 | 49  |
| 71 | 10 | 1.000 | 5  | 223,60 | 15,20 | 1228,60 | 49  |
| 72 | 10 | 1.000 | 10 | 218,70 | 14,10 | 1228,70 | 49  |

#### 4.2.4.3. JavaScript ve Akıllı Sözleşme Tabanlı Kanıt Doğrulama Performans Karşılaştırması

Kanıt doğrulama süreci hem JavaScript ortamında hem de akıllı sözleşme üzerinde yürütülebilmektedir. Seçmen oy kullandığında, öncelikle JavaScript tabanlı doğrulama işlemi gerçekleştirilir. Takip ededen aşamada, oy verisi akıllı sözleşmeye iletilerek hem kaydedilir hem de kanıt doğrulama işlemi uygulanır. Bu nedenle, her iki yöntemle ilişkin analizler gerçekleştirilmiş ve karşılaştırmalı değerlendirmeler hazırlanmıştır. Sunulan veriler, önceki aşamalarda elde edilen sonuçların karşılaştırılmasını içermektedir. Gerçekleştirilen testler sonucunda ortalama kanıt doğrulama süresi hesaplanmış ve JavaScript ile akıllı sözleşme tabanlı doğrulama sürelerinin karşılaştırılmasında aşağıdaki formül temel alınmıştır:

$$\text{Hız Oranı} = \frac{\text{Kanıt Doğrulama Süresi (Akıllı Sözleşme)}}{\text{Kanıt Doğrulama Süresi (JavaScript)}}$$

JavaScript ve akıllı sözleşmeyle gerçekleştirilen kanıt üretim ve doğrulama sürecini içeren veriler Tablo 4.20.'de gösterilmektedir.

**Tablo 4. 20.** JavaScript ve akıllı sözleşmenin kanıt doğrulamasının karşılaştırmalı analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Kanıt Üretim Süresi (ms) | JavaScript (ms) | Akıllı Sözleşme (ms) | Hız Oranı |
|-------|-------------|---------------|--------------------------|-----------------|----------------------|-----------|
| 1     | 2           | 5             | 292,00                   | 5,90            | 17,90                | 3,03      |
| 2     | 2           | 10            | 214,60                   | 5,60            | 13,60                | 2,43      |
| 3     | 2           | 50            | 234,20                   | 5,80            | 15,30                | 2,64      |
| 4     | 2           | 100           | 221,70                   | 6,20            | 14,80                | 2,39      |
| 5     | 2           | 500           | 218,60                   | 5,60            | 14,00                | 2,50      |
| 6     | 2           | 1.000         | 216,70                   | 5,50            | 13,80                | 2,51      |
| 7     | 3           | 5             | 217,70                   | 5,60            | 14,50                | 2,59      |
| 8     | 3           | 10            | 219,80                   | 5,60            | 14,10                | 2,52      |
| 9     | 3           | 50            | 218,10                   | 5,60            | 15,00                | 2,68      |
| 10    | 3           | 100           | 218,30                   | 5,60            | 14,50                | 2,59      |
| 11    | 3           | 500           | 219,00                   | 5,60            | 14,20                | 2,54      |
| 12    | 3           | 1.000         | 218,10                   | 5,70            | 14,40                | 2,53      |
| 13    | 5           | 5             | 219,20                   | 5,70            | 13,60                | 2,39      |
| 14    | 5           | 10            | 223,10                   | 5,70            | 14,10                | 2,47      |
| 15    | 5           | 50            | 218,70                   | 5,80            | 13,90                | 2,40      |
| 16    | 5           | 100           | 218,50                   | 5,60            | 14,00                | 2,50      |
| 17    | 5           | 500           | 218,30                   | 5,90            | 14,00                | 2,37      |
| 18    | 5           | 1.000         | 220,40                   | 5,60            | 13,60                | 2,43      |
| 19    | 10          | 5             | 220,10                   | 5,70            | 14,10                | 2,47      |
| 20    | 10          | 10            | 218,90                   | 5,60            | 14,00                | 2,50      |
| 21    | 10          | 50            | 219,00                   | 5,60            | 13,70                | 2,45      |
| 22    | 10          | 100           | 219,80                   | 5,60            | 13,70                | 2,45      |
| 23    | 10          | 500           | 220,80                   | 5,70            | 14,30                | 2,51      |
| 24    | 10          | 1.000         | 218,70                   | 5,70            | 13,30                | 2,33      |

Sistemde oy kullanma, kanıt üretme ve kanıt doğrulama sürecinin eşzamanlı olarak değerlendirilmesi gerekmektedir. Bu nedenle, daha önceki çalışmalarda sırasıyla 2, 5 ve 10 seçmenin bu işlemleri eşzamanlı olarak gerçekleştirmesi sağlanmıştır. Gerçekleştirilen işlemler sonucunda, JavaScript ve akıllı sözleşme ortamlarında eşzamanlı kanıt doğrulama süresinin ortalaması tespit edilmiştir. JavaScript ile akıllı sözleşme tabanlı doğrulama sürelerinin karşılaştırılması için aşağıdaki formül kullanılmıştır:

$$\text{Hız Oranı} = \frac{\text{Kanıt Doğrulama Süresi (Akıllı Sözleşme)}}{\text{Kanıt Doğrulama Süresi (JavaScript)}}$$

JavaScript ve akıllı sözleşme ortamlarında eşzamanlı kanıt doğrulama süresini içeren veriler Tablo 4.21.'de gösterilmektedir. Bu veriler doğrultusunda JavaScript ve akıllı sözleşme

ortamlarında gerçekleştirilen eşzamanlı kanıt doğrulama işlemlerinde hız farkları olduğu tespit edilmiştir. Bu hız farklılıklarının tespiti hangi servisinin daha fazla işlem gücü harcayacağını belirlenmesini sağlamaktadır.

**Tablo 4. 21.** JavaScript ve akıllı sözleşmenin aracılığıyla gerçekleştirilen eşzamanlı kanıt doğrulama süreçlerinin karşılaştırmalı analizi.

| Seçim | Aday Sayısı | Seçmen Sayısı | Eşzamanlı (x) | Kanıt Üretim Süresi (ms) | JavaScript (ms) | Akıllı Sözleşme (ms) | Hız Oranı |
|-------|-------------|---------------|---------------|--------------------------|-----------------|----------------------|-----------|
| 1     | 2           | 5             | 2             | 221,70                   | 5,90            | 14,60                | 2,47      |
| 2     | 2           | 5             | 5             | 217,50                   | 5,70            | 14,10                | 2,47      |
| 3     | 2           | 5             | 10            | 215,30                   | 5,90            | 14,70                | 2,49      |
| 4     | 2           | 10            | 2             | 215,50                   | 5,80            | 14,50                | 2,50      |
| 5     | 2           | 10            | 5             | 223,70                   | 5,90            | 14,70                | 2,49      |
| 6     | 2           | 10            | 10            | 217,80                   | 5,90            | 14,70                | 2,49      |
| 7     | 2           | 1.000         | 2             | 214,90                   | 5,60            | 13,50                | 2,41      |
| 8     | 2           | 1.000         | 5             | 217,90                   | 5,70            | 14,70                | 2,58      |
| 9     | 2           | 1.000         | 10            | 216,80                   | 5,60            | 14,20                | 2,54      |
| 10    | 3           | 5             | 2             | 216,70                   | 5,50            | 13,80                | 2,51      |
| 11    | 3           | 5             | 5             | 235,20                   | 5,70            | 14,60                | 2,56      |
| 12    | 3           | 5             | 10            | 228,30                   | 5,80            | 14,30                | 2,47      |
| 13    | 3           | 10            | 2             | 222,20                   | 5,80            | 14,80                | 2,55      |
| 14    | 3           | 10            | 5             | 217,60                   | 6,00            | 14,10                | 2,35      |
| 15    | 3           | 10            | 10            | 218,10                   | 5,80            | 14,60                | 2,52      |
| 16    | 3           | 1.000         | 2             | 216,80                   | 5,70            | 14,40                | 2,53      |
| 17    | 3           | 1.000         | 5             | 218,40                   | 5,60            | 14,60                | 2,61      |
| 18    | 3           | 1.000         | 10            | 217,40                   | 5,70            | 14,80                | 2,60      |
| 19    | 5           | 5             | 2             | 216,20                   | 5,70            | 14,50                | 2,54      |
| 20    | 5           | 5             | 5             | 218,60                   | 5,70            | 14,30                | 2,51      |
| 21    | 5           | 5             | 10            | 216,70                   | 5,70            | 13,60                | 2,39      |
| 22    | 5           | 10            | 2             | 217,10                   | 5,80            | 14,60                | 2,52      |
| 23    | 5           | 10            | 5             | 218,50                   | 5,70            | 13,60                | 2,39      |
| 24    | 5           | 10            | 10            | 217,10                   | 5,60            | 14,20                | 2,54      |
| 25    | 5           | 1.000         | 2             | 217,60                   | 5,80            | 14,10                | 2,43      |
| 26    | 5           | 1.000         | 5             | 219,10                   | 5,70            | 14,00                | 2,46      |
| 27    | 5           | 1.000         | 10            | 218,10                   | 5,80            | 14,60                | 2,52      |
| 28    | 10          | 5             | 2             | 218,70                   | 5,80            | 14,10                | 2,43      |
| 29    | 10          | 5             | 5             | 220,00                   | 5,70            | 14,10                | 2,47      |
| 30    | 10          | 5             | 10            | 219,40                   | 5,70            | 13,90                | 2,44      |
| 31    | 10          | 10            | 2             | 217,20                   | 5,70            | 14,10                | 2,47      |
| 32    | 10          | 10            | 5             | 219,80                   | 5,70            | 14,10                | 2,47      |
| 33    | 10          | 10            | 10            | 216,70                   | 5,80            | 15,30                | 2,64      |
| 34    | 10          | 50            | 2             | 218,60                   | 5,70            | 14,20                | 2,49      |
| 35    | 10          | 50            | 5             | 218,40                   | 5,60            | 14,00                | 2,50      |
| 36    | 10          | 50            | 10            | 218,20                   | 5,80            | 13,70                | 2,36      |
| 37    | 10          | 100           | 2             | 217,30                   | 5,80            | 14,80                | 2,55      |

|    |    |       |    |        |       |       |      |
|----|----|-------|----|--------|-------|-------|------|
| 38 | 10 | 100   | 5  | 219,00 | 5,80  | 14,20 | 2,45 |
| 39 | 10 | 100   | 10 | 217,50 | 5,70  | 14,10 | 2,47 |
| 40 | 10 | 500   | 2  | 219,20 | 5,70  | 14,40 | 2,53 |
| 41 | 10 | 500   | 5  | 219,50 | 5,70  | 13,70 | 2,40 |
| 42 | 10 | 500   | 10 | 220,60 | 5,80  | 14,30 | 2,47 |
| 43 | 10 | 1.000 | 2  | 217,00 | 5,60  | 14,00 | 2,50 |
| 44 | 10 | 1.000 | 5  | 223,60 | 12,00 | 15,20 | 1,27 |
| 45 | 10 | 1.000 | 10 | 218,70 | 5,60  | 14,10 | 2,52 |

#### 4.2.5. Paillier Tabanlı Oy Verisi Şifreleme Performans Analizi

Sistemdeki oylar, 2048-bit anahtar uzunluğu kullanılarak Paillier şifreleme algoritmasıyla şifrelenmekte ve akıllı sözleşme üzerinde saklanmaktadır. Seçim sonuçları açıklanırken, oylar homomorfik olarak toplanmakta ve bu toplam üzerindeki şifreli verilerin çözülerek açık sonuca ulaşılabilmesi için işlemler gerçekleştirilmektedir. Bu nedenle, oylama işlemi sırasında şifreleme algoritmasının nasıl bir performans gösterdiğinin tespit edilmesi gerekmektedir.

##### 4.2.5.1. Paillier Şifreleme Analizi

İlk aşamada birbirinden bağımsız seçimler oluşturulmuştur. Bu seçimlerde, seçmenlerin sırayla oy kullanması sağlanmış; oylar şifrelenerek kaydedilmiştir. Seçim tamamlandığında ise, şifreli oylar homomorfik olarak toplanmış ve şifre çözme işlemine tabi tutulmuştur. Bu süreçler sonucunda, ortalama şifreleme, toplama ve deşifreleme süresinin belirlenmesi ve toplam sürenin hesaplanması amaçlanmıştır. İşlem hacmi, sistemin belirli bir zaman biriminde gerçekleştirebildiği toplam kriptografik işlemler bütünüdür. Bu bağlamda işlem hacmi; oy verisinin şifrelenmesi, homomorfik toplama işlemleri ve deşifreleme süreçlerini kapsamaktadır. Testler sonucunda, dakikada gerçekleştirilebilen bu işlemlerin sayısı hesaplanarak sistemin zaman-temelli işlem kapasitesi belirlenmiştir. Bu doğrultuda, aşağıda gösterilen işlem hacmi formülü kullanılmıştır.

$$\text{İşlem Hacmi} = \frac{\text{Oy Sayısı}}{\text{Toplam}} \times 1000 \times 60$$

Bu hesaplamalar doğrultusunda sıralı gerçekleştirilen işlemler için Paillier şifreleme algoritmasının performans değerleri Tablo 4.22.'de gösterilmektedir. Şifreleme, toplama ve deşifreleme süreleri, ilgili seçimde bir oyun şifrelenmesine ilişkin ortalama süreyi göstermektedir. Bu ortalama, seçim sürecine katılan tüm seçmenlerin bireysel olarak gerçekleştirdiği işlemlerin sürelerinin ortalamasına dayanır. Toplam ise, bu bireysel sürelerin toplamını ifade etmektedir. Böylece, dakika başına gerçekleştirilebilecek işlem sayısını

gösteren işlem hacmi hesaplanabilmektedir. Bu testler gerçekleştirirken farklı seçim senaryolarında farklı Paillier anahtar boyutları seçilerek, anahtar boyutunun sistem performansı üzerindeki etkilerinin ortaya çıkartılması amaçlanmıştır.

**Tablo 4. 22.** Paillier şifreleme algoritmasının sistem üzerindeki analizi.

| Seçim | Anahtar Boyutu (Bit) | Aday Sayısı | Oy Sayısı | Şifreleme (ms) | Toplama (ms) | Deşifreleme (ms) | Toplam (ms) | İşlem Hacmi (oy/dk) |
|-------|----------------------|-------------|-----------|----------------|--------------|------------------|-------------|---------------------|
| 1     | 32                   | 10          | 500       | 16,60          | 0,2          | 0,01             | 16,81       | 1.784.652           |
| 2     | 32                   | 10          | 1.000     | 33,00          | 0,4          | 0,2              | 33,60       | 1.785.714           |
| 3     | 1024                 | 2           | 50        | 418,00         | 0,4          | 9,6              | 428,00      | 7.009               |
| 4     | 1024                 | 2           | 500       | 4.139,20       | 2,4          | 8,2              | 4.149,80    | 7.229               |
| 5     | 1024                 | 2           | 1.000     | 8.179,60       | 5,2          | 8                | 8.192,80    | 7.324               |
| 6     | 1024                 | 3           | 50        | 612,00         | 0,2          | 12,4             | 624,60      | 4.803               |
| 7     | 1024                 | 3           | 500       | 6.133,00       | 4            | 12               | 6.149,00    | 4.879               |
| 8     | 1024                 | 3           | 1.000     | 12.312,80      | 8            | 12               | 12.332,80   | 4.865               |
| 9     | 1024                 | 5           | 50        | 1.043,40       | 0,8          | 20,2             | 1.064,40    | 2.818               |
| 10    | 1024                 | 5           | 500       | 10.335,80      | 6,2          | 20,4             | 10.362,40   | 2.895               |
| 11    | 1024                 | 5           | 1.000     | 20.313,80      | 13           | 20,2             | 20.347,00   | 2.949               |
| 12    | 1024                 | 10          | 50        | 2.022,80       | 1,2          | 40,2             | 2.064,20    | 1.453               |
| 13    | 1024                 | 10          | 500       | 20.396,80      | 13,2         | 39,8             | 20.449,80   | 1.467               |
| 14    | 1024                 | 10          | 1.000     | 41.154,80      | 27           | 40,6             | 41.222,40   | 1.456               |
| 15    | 2048                 | 2           | 50        | 8.920,80       | 2,8          | 188,8            | 9.112,40    | 329                 |
| 16    | 2048                 | 2           | 500       | 27.800,40      | 14,4         | 89,6             | 27.904,40   | 1.075               |
| 17    | 2048                 | 2           | 1.000     | 55.381,60      | 18           | 54,8             | 55.454,40   | 1.082               |
| 18    | 2048                 | 3           | 50        | 3.939,80       | 1            | 78,6             | 4.019,40    | 746                 |
| 19    | 2048                 | 3           | 500       | 38.507,80      | 12,8         | 76,6             | 38.597,20   | 777                 |
| 20    | 2048                 | 3           | 1000      | 75.602,20      | 24,8         | 75,6             | 75.702,60   | 793                 |
| 21    | 2048                 | 5           | 50        | 6.277,00       | 2            | 124,8            | 6.403,80    | 468                 |
| 22    | 2048                 | 5           | 500       | 63.033,00      | 20,8         | 126              | 63.179,80   | 475                 |
| 23    | 2048                 | 5           | 1.000     | 125.112,60     | 41,4         | 126              | 125.280,00  | 479                 |
| 24    | 2048                 | 10          | 50        | 13.529,00      | 4,6          | 266              | 13.799,60   | 217                 |
| 25    | 2048                 | 10          | 500       | 127.627,60     | 41,2         | 253              | 127.921,80  | 235                 |
| 26    | 2048                 | 10          | 1.000     | 243.275,60     | 78,8         | 237,4            | 243.591,80  | 246                 |

#### 4.2.5.2. Eşzamanlı Paillier Şifreleme Analizi

Sistemin işlem kapasitesini doğru bir şekilde değerlendirebilmek için, şifreleme algoritmasının eşzamanlı biçimde çalıştırılması ve dakikada gerçekleştirilebilecek işlem sayısının hesaplanması gerekmektedir. Bu hesaplama, sistemin yüksek işlem yoğunluğuna sahip senaryolarda ne ölçüde ölçeklenebilir olduğunu ortaya koymak açısından kritik bir öneme sahiptir. Bu kapsamda, sistemin farklı yoğunluk koşullarındaki performansını değerlendirebilmek amacıyla, 2, 3, 5 ve 10 adet seçmenin eşzamanlı şekilde oy kullanması sağlanmıştır. Böylece oy kullanma işlemi sırasında gerçekleşen şifreleme süreçlerinin süreleri

ölçülmüş; her bir oy için ortalama şifreleme, homomorfik toplama ve şifre çözme süreleri ayrı ayrı belirlenmiştir. Elde edilen bu süreler üzerinden sistemin işlem hacmi, bir diğer ifadeyle dakikada gerçekleştirebileceği toplam işlem sayısı, aşağıdaki formül yardımıyla hesaplanmıştır.

$$\text{İşlem Hacmi} = \frac{\text{Oy Sayısı}}{\text{Toplam}} \times 1000 \times 60$$

Tüm hesaplamalar neticesinde Paillier şifreleme algoritmasının sistem üzerindeki performansı Tablo 4.23.'de gösterilmiştir. Bu süreler, seçimdeki tüm seçmenlerin bireysel olarak gerçekleştirdiği işlemlerin ortalamasından alınmıştır. Toplam bölüm ise, bu bireysel sürelerin toplamını ifade etmektedir. Eşzamanlı bölüm, belirli bir anda aynı anda işlem gerçekleştiren seçmen sayısını belirtmektedir. Böylece, eşzamanlı işlemler neticesinde dakika başına gerçekleştirilebilecek işlem sayısını gösteren işlem hacmi hesaplanabilmektedir. Ayrıca, Paillier şifreleme algoritmasında farklı anahtar boyutları seçilerek testler çeşitlendirilmiştir.

**Tablo 4. 23.** Paillier şifreleme algoritmasının eşzamanlı sistem üzerindeki analizi.

| Seçim | Anahtar Boyutu (Bit) | Aday Sayısı | Oy Sayısı | Eşzamanlı (x) | Şifreleme (ms) | Toplama (ms) | Deşifreleme (ms) | Toplam (ms) | İşlem Hacmi (oy/dk) |
|-------|----------------------|-------------|-----------|---------------|----------------|--------------|------------------|-------------|---------------------|
| 1     | 32                   | 10          | 500       | 2x            | 16,80          | 0,01         | 0,01             | 16,82       | 1.783.590,96        |
| 2     | 32                   | 10          | 500       | 3x            | 16,50          | 0,01         | 0,01             | 16,52       | 1.815.980,63        |
| 3     | 32                   | 10          | 500       | 5x            | 16,60          | 0,03         | 0,01             | 16,64       | 1.802.884,62        |
| 4     | 32                   | 10          | 500       | 10x           | 16,50          | 0,03         | 0,01             | 16,54       | 1.813.784,76        |
| 5     | 1024                 | 2           | 50        | 2x            | 409,80         | 0,01         | 8,6              | 418,41      | 7.170,00            |
| 6     | 1024                 | 2           | 50        | 3x            | 409,80         | 0,03         | 8                | 417,83      | 7.179,95            |
| 7     | 1024                 | 2           | 50        | 5x            | 410,10         | 0,02         | 8,1              | 418,22      | 7.173,26            |
| 8     | 1024                 | 2           | 50        | 10x           | 409,90         | 0,03         | 8,1              | 418,03      | 7.176,52            |
| 9     | 1024                 | 2           | 500       | 2x            | 4.089,80       | 2,6          | 8                | 4.100,40    | 7.316,36            |
| 10    | 1024                 | 2           | 500       | 3x            | 4.090,30       | 2,7          | 8,2              | 4.101,20    | 7.314,93            |
| 11    | 1024                 | 2           | 500       | 5x            | 4.137,30       | 2,6          | 8                | 4.147,90    | 7.232,58            |
| 12    | 1024                 | 2           | 500       | 10x           | 4.137,10       | 2,6          | 8,6              | 4.148,30    | 7.231,88            |
| 13    | 1024                 | 2           | 1.000     | 2x            | 8.186,40       | 5            | 8,2              | 8.199,60    | 7.317,43            |
| 14    | 1024                 | 2           | 1.000     | 3x            | 8.205,30       | 5            | 8                | 8.218,30    | 7.300,78            |
| 15    | 1024                 | 2           | 1.000     | 5x            | 8.120,90       | 5,1          | 8,1              | 8.134,10    | 7.376,35            |
| 16    | 1024                 | 2           | 1.000     | 10x           | 8.216,60       | 5,5          | 8,2              | 8.230,30    | 7.290,14            |
| 17    | 1024                 | 3           | 50        | 2x            | 613,00         | 0,4          | 12,2             | 625,60      | 4.795,40            |
| 18    | 1024                 | 3           | 50        | 3x            | 612,70         | 0,3          | 12               | 625,00      | 4.800,00            |
| 19    | 1024                 | 3           | 50        | 5x            | 613,60         | 0,5          | 12,2             | 626,30      | 4.790,04            |
| 20    | 1024                 | 3           | 50        | 10x           | 616,80         | 0,4          | 12,1             | 629,30      | 4.767,20            |
| 21    | 1024                 | 3           | 500       | 2x            | 6.137,80       | 3,8          | 12               | 6.153,60    | 4.875,20            |
| 22    | 1024                 | 3           | 500       | 3x            | 6.187,80       | 4            | 12,2             | 6.204,00    | 4.835,59            |
| 23    | 1024                 | 3           | 500       | 5x            | 6.187,60       | 3,9          | 12,3             | 6.203,80    | 4.835,75            |
| 24    | 1024                 | 3           | 500       | 10x           | 6.160,40       | 3,8          | 12,3             | 6.176,50    | 4.857,12            |

|    |      |    |       |     |           |      |      |           |          |
|----|------|----|-------|-----|-----------|------|------|-----------|----------|
| 25 | 1024 | 3  | 1.000 | 2x  | 12.274,20 | 7,8  | 12   | 12.294,00 | 4.880,43 |
| 26 | 1024 | 3  | 1.000 | 3x  | 12.307,20 | 8,8  | 14   | 12.330,00 | 4.866,18 |
| 27 | 1024 | 3  | 1.000 | 5x  | 12.324,50 | 7,9  | 12,4 | 12.344,80 | 4.860,35 |
| 28 | 1024 | 3  | 1.000 | 10x | 12.361,40 | 7,8  | 12,2 | 12.381,40 | 4.845,98 |
| 29 | 1024 | 5  | 50    | 2x  | 1.057,60  | 0,8  | 20,2 | 1.078,60  | 2.781,38 |
| 30 | 1024 | 5  | 50    | 3x  | 1.042,20  | 0,8  | 22,7 | 1.065,70  | 2.815,05 |
| 31 | 1024 | 5  | 50    | 5x  | 1.037,70  | 0,6  | 20,4 | 1.058,70  | 2.833,66 |
| 32 | 1024 | 5  | 50    | 10x | 1.042,20  | 0,6  | 20,5 | 1.063,30  | 2.821,41 |
| 33 | 1024 | 5  | 500   | 2x  | 10.205,00 | 6,6  | 20   | 10.231,60 | 2.932,09 |
| 34 | 1024 | 5  | 500   | 3x  | 10.191,00 | 6,7  | 20   | 10.217,70 | 2.936,08 |
| 35 | 1024 | 5  | 500   | 5x  | 10.171,80 | 6,6  | 22   | 10.200,40 | 2.941,06 |
| 36 | 1024 | 5  | 500   | 10x | 10.157,50 | 6,5  | 20,2 | 10.184,20 | 2.945,74 |
| 37 | 1024 | 5  | 1.000 | 2x  | 20.286,60 | 14   | 20,2 | 20.320,80 | 2.952,64 |
| 38 | 1024 | 5  | 1.000 | 3x  | 20.269,00 | 13,2 | 20   | 20.302,20 | 2.955,34 |
| 39 | 1024 | 5  | 1.000 | 5x  | 20.268,30 | 12,9 | 19,9 | 20.301,10 | 2.955,50 |
| 40 | 1024 | 5  | 1.000 | 10x | 20.250,10 | 13,2 | 19,9 | 20.283,20 | 2.958,11 |
| 41 | 1024 | 10 | 50    | 2x  | 2.032,60  | 1    | 40,2 | 2.073,80  | 1.446,62 |
| 42 | 1024 | 10 | 50    | 3x  | 2.037,70  | 1,2  | 40,7 | 2.079,60  | 1.442,59 |
| 43 | 1024 | 10 | 50    | 5x  | 2.027,70  | 1,4  | 39,9 | 2.069,00  | 1.449,98 |
| 44 | 1024 | 10 | 50    | 10x | 2.032,30  | 1,2  | 40,2 | 2.073,70  | 1.446,69 |
| 45 | 1024 | 10 | 500   | 2x  | 20.211,60 | 13,2 | 39,8 | 20.264,60 | 1.480,41 |
| 46 | 1024 | 10 | 500   | 3x  | 20.153,80 | 13   | 39,8 | 20.206,60 | 1.484,66 |
| 47 | 1024 | 10 | 500   | 5x  | 20.137,50 | 12,9 | 39,7 | 20.190,10 | 1.485,88 |
| 48 | 1024 | 10 | 500   | 10x | 21.392,50 | 13,2 | 40   | 21.445,70 | 1.398,88 |
| 49 | 1024 | 10 | 1.000 | 2x  | 41.075,20 | 27   | 40,6 | 41.142,80 | 1.458,34 |
| 50 | 1024 | 10 | 1.000 | 3x  | 40.372,00 | 26,3 | 40   | 40.438,30 | 1.483,74 |
| 51 | 1024 | 10 | 1.000 | 5x  | 44.108,40 | 26,1 | 39,8 | 44.174,30 | 1.358,26 |
| 52 | 1024 | 10 | 1.000 | 10x | 55.020,20 | 44,7 | 68   | 55.132,90 | 1.088,28 |
| 53 | 2048 | 2  | 50    | 2x  | 2.257,40  | 0,8  | 45,6 | 2.303,80  | 1.302,20 |
| 54 | 2048 | 2  | 50    | 3x  | 2.264,80  | 0,3  | 45,8 | 2.310,90  | 1.298,20 |
| 55 | 2048 | 2  | 50    | 5x  | 2.271,00  | 0,9  | 45,6 | 2.317,50  | 1.294,50 |
| 56 | 2048 | 2  | 50    | 10x | 4.229,60  | 1,6  | 88,5 | 4.319,70  | 694,49   |
| 57 | 2048 | 2  | 500   | 2x  | 29.412,00 | 13,8 | 88,8 | 29.514,60 | 1.016,45 |
| 58 | 2048 | 2  | 500   | 3x  | 35.198,50 | 7,3  | 45   | 35.250,80 | 851,04   |
| 59 | 2048 | 2  | 500   | 5x  | 31.285,30 | 11,4 | 70   | 31.366,70 | 956,43   |
| 60 | 2048 | 2  | 500   | 10x | 28.060,90 | 8,9  | 55,2 | 28.125,00 | 1.066,67 |
| 61 | 2048 | 2  | 1.000 | 2x  | 52.972,00 | 17   | 52,4 | 53.041,40 | 1.131,19 |
| 62 | 2048 | 2  | 1.000 | 3x  | 52.972,00 | 16,5 | 51,3 | 53.039,80 | 1.131,23 |
| 63 | 2048 | 2  | 1.000 | 5x  | 51.165,30 | 16,6 | 51,3 | 51.233,20 | 1.171,12 |
| 64 | 2048 | 2  | 1.000 | 10x | 52.545,80 | 16,9 | 52,6 | 52.615,30 | 1.140,35 |
| 65 | 2048 | 3  | 50    | 2x  | 3.994,40  | 1,2  | 80,2 | 4.075,80  | 736,05   |
| 66 | 2048 | 3  | 50    | 3x  | 4.119,80  | 1,2  | 81,5 | 4.202,50  | 713,86   |
| 67 | 2048 | 3  | 50    | 5x  | 3.913,50  | 1,2  | 78,5 | 3.993,20  | 751,28   |
| 68 | 2048 | 3  | 50    | 10x | 3.966,40  | 1,3  | 79,6 | 4.047,30  | 741,23   |
| 69 | 2048 | 3  | 500   | 2x  | 38.244,40 | 12,4 | 77,4 | 38.334,20 | 782,59   |
| 70 | 2048 | 3  | 500   | 3x  | 37.989,70 | 12,3 | 76,8 | 38.078,80 | 787,84   |
| 71 | 2048 | 3  | 500   | 5x  | 37.916,80 | 12,3 | 76   | 38.005,10 | 789,37   |
| 72 | 2048 | 3  | 500   | 10x | 37.793,80 | 12,4 | 76,3 | 37.882,50 | 791,92   |

|     |      |    |       |     |            |      |       |            |        |
|-----|------|----|-------|-----|------------|------|-------|------------|--------|
| 73  | 2048 | 3  | 1.000 | 2x  | 75.813,20  | 24,6 | 76,2  | 75.914,00  | 790,37 |
| 74  | 2048 | 3  | 1.000 | 3x  | 75.865,20  | 25   | 75,3  | 75.965,50  | 789,83 |
| 75  | 2048 | 3  | 1.000 | 5x  | 76.439,50  | 24,6 | 76    | 76.540,10  | 783,90 |
| 76  | 2048 | 3  | 1.000 | 10x | 75.985,70  | 24,7 | 75,8  | 76.086,20  | 788,58 |
| 77  | 2048 | 5  | 50    | 2x  | 6.320,00   | 2    | 127,4 | 6.449,40   | 465,16 |
| 78  | 2048 | 5  | 50    | 3x  | 6.298,80   | 2    | 127,2 | 6.428,00   | 466,71 |
| 79  | 2048 | 5  | 50    | 5x  | 6.336,30   | 2    | 126,4 | 6.464,70   | 464,06 |
| 80  | 2048 | 5  | 50    | 10x | 6.308,10   | 2    | 126,4 | 6.436,50   | 466,09 |
| 81  | 2048 | 5  | 500   | 2x  | 63.062,40  | 21   | 127   | 63.210,40  | 474,61 |
| 82  | 2048 | 5  | 500   | 3x  | 62.967,00  | 20,5 | 132,5 | 63.120,00  | 475,29 |
| 83  | 2048 | 5  | 500   | 5x  | 62.984,50  | 21   | 130,8 | 63.136,30  | 475,16 |
| 84  | 2048 | 5  | 500   | 10x | 62.861,30  | 20,7 | 126   | 63.008,00  | 476,13 |
| 85  | 2048 | 5  | 1.000 | 2x  | 125.031,60 | 40,8 | 124,6 | 125.197,00 | 479,24 |
| 86  | 2048 | 5  | 1.000 | 3x  | 126.018,00 | 40,2 | 124,3 | 126.182,50 | 475,50 |
| 87  | 2048 | 5  | 1.000 | 5x  | 122.829,10 | 40,4 | 124,5 | 122.994,00 | 487,83 |
| 88  | 2048 | 5  | 1.000 | 10x | 127.444,10 | 41,6 | 126,8 | 127.612,50 | 470,17 |
| 89  | 2048 | 10 | 50    | 2x  | 13.593,60  | 4,4  | 265,6 | 13.863,60  | 216,39 |
| 90  | 2048 | 10 | 50    | 3x  | 13.318,50  | 4,2  | 268,5 | 13.591,20  | 220,73 |
| 91  | 2048 | 10 | 50    | 5x  | 13.102,60  | 4,4  | 262,2 | 13.369,20  | 224,40 |
| 92  | 2048 | 10 | 50    | 10x | 12.971,00  | 4,2  | 259,1 | 13.234,30  | 226,68 |
| 93  | 2048 | 10 | 500   | 2x  | 126.174,00 | 41,2 | 253,4 | 126.468,60 | 237,21 |
| 94  | 2048 | 10 | 500   | 3x  | 126.235,50 | 41,8 | 260,8 | 126.538,10 | 237,08 |
| 95  | 2048 | 10 | 500   | 5x  | 127.169,60 | 41,5 | 255,6 | 127.466,70 | 235,36 |
| 96  | 2048 | 10 | 500   | 10x | 126.895,90 | 41,5 | 257,5 | 127.194,90 | 235,86 |
| 97  | 2048 | 10 | 1.000 | 2x  | 229.234,40 | 75,6 | 229,6 | 229.539,60 | 261,39 |
| 98  | 2048 | 10 | 1.000 | 3x  | 229.282,30 | 75,8 | 229,3 | 229.587,40 | 261,34 |
| 99  | 2048 | 10 | 1.000 | 5x  | 230.590,90 | 80   | 230,2 | 230.901,10 | 259,85 |
| 100 | 2048 | 10 | 1.000 | 10x | 230.163,00 | 76,3 | 230,3 | 230.469,60 | 260,34 |

#### 4.2.6. Paillier Şifreleme Algoritması Sistemdeki Zaman Karmaşıklığı

Paillier şifreleme algoritması, homomorfik özellikleri nedeniyle sistem içerisinde tercih edilmiştir. Paillier şifreleme algoritmasında anahtar uzunluğu 2048-bit olduğu için algoritmanın zaman karmaşıklığının hesaplanması gerekmektedir.

Paillier şifrelemesinde güvenli bir yapı oluşturabilmek seçilen asal sayıların boyutuyla doğrudan ilişkilidir. Bu nedenle,  $p$  ve  $q$  asal sayıları 1024 bit uzunluğunda seçildiği için modül  $n=p \times q$  yaklaşık 2048 bit uzunluğunda olmaktadır. İşlemlerin gerçekleştiği  $n^2$  ise yaklaşık 4096 bit uzunluğundadır. Bu bit uzunlukları, hesaplamaların karmaşıklığını belirleyen temel parametrelerdir.

Paillier sisteminde üç temel işlem mevcuttur: şifreleme, homomorfik toplama ve deşifreleme. Zaman karmaşıklıkları anahtar boyutuyla orantılıdır ve bu işlemlerin zaman karmaşıklıkları,  $k$  (bit uzunluğu) cinsinden aşağıdaki gibi ifade edilmektedir:

- **Şifreleme:** İki adet modüler üs alma işlemi içerdiğinden karmaşıklığı  $O(k^3)$  olarak hesaplanır. Fakat her seçmen için yapılacağından toplam karmaşıklık  $O(n.k^3)$  olacaktır.
- **Homomorfik Toplama:** Sadece modüler çarpma işlemi olduğundan, karmaşıklığı  $O(k^2)$  seviyesindedir.
- **Deşifreleme:** Tek modüler üs alma ve modüler çarpma işlemi içerir. Karmaşıklığı  $O(k^3)$  olarak değerlendirilir.



## 5. TARTIŞMA

Oylama sistemleri, kurum ve kuruluşların karar alma süreçlerinde kullanılan yöntemlerden birisidir. Dijitalleşmenin olmadığı dönemlerde bu süreçler için geleneksel oylama sistemleri kullanılırken, günümüzde dijitalleşmesinin etkisiyle elektronik oylama sistemleri alternatif olarak tercih edilmektedir. Elektronik oylama sistemlerinin, geleneksel oylama yöntemlerinin yerini alabilmesi için öncelikle yüksek düzeyde güvenilirlik sağlaması gerekmektedir. Bu güvenilirliğin temin edilebilmesi için ise sistemlerin doğru şekilde tasarlanması ve titizlikle geliştirilmesi zorunludur. Maliyet, geleneksel sistemlerden elektronik sistemlere geçiş sürecini etkileyecek kriterlerden bir diğeridir. Sistemlerin geliştirilmesi ve bakımının yapılması için ortaya çıkacak maliyetin kabul edilebilir düzeyde olması tercih edilmelerine neden olmaktadır. Ayrıca, geliştirilen sistemlerin beklenen performans kriterlerini karşılaması önemli bir unsurdur. Sistemlerin kesintisiz hizmet sunması ve artan seçmen ile seçim yetkilisi sayısına rağmen kabul edilebilir performans seviyelerinde çalışmayı sürdürebilmesi kritik bir gerekliliktir. Bu nedenle; güvenlik, maliyet ve performans dengesinin doğru oluşturulması kritik önem taşımaktadır. Bu çalışma kapsamında geliştirilen blokzinciri tabanlı merkezi olmayan üniversite ölçekli oylama sistemi, seçim süreçlerinde güvenilirlik, şeffaflık, doğrulanabilirlik ve gizlilik kriterlerini sağlaması amacıyla tasarlanmış ve geliştirilmiştir. Bu dört kriter, geliştirilen sistemin dayandığı temelleri oluşturmaktadır. Bu temel yapıların doğru maliyet ve performansla çalıştığını göstermek amacıyla test ve analizler gerçekleştirilmiş, elde edilen veriler doğrultusunda bulgular ortaya konmuştur. Bulgular, sistemin teknik olarak başarılı sonuçlar verdiğini göstermektedir.

Sistemi değerlendirmek için öncelikle maliyet analizi yapılmıştır. Oylama sisteminin uygulanabilirliğini tespit edebilmek ve kaynak kullanımını değerlendirmek için gerekli bir yöntemdir. Maliyet analizine planlama ve bakım ücretleri dahil edilmemiştir. Çünkü bu işlemler oylama sisteminin kullanılacağı üniversite bünyesinde yapılması gereken işlemlerdir. Maliyet analizinde kullanılan kur bilgileri, çalışmanın yapıldığı tarih baz alınarak değerlendirilmelidir. ETH, USD ve TL para birimleri arasındaki kur dönüşümleri tarihsel süreçte değişkenlik gösterebilmektedir. Bu nedenle, çalışmanın tekrar değerlendirileceği zaman bu birimsel parametrelerin zamana uygun hale getirilmesi gerekmektedir. Ayrıca, Ethereum ağlarının gaz kullanım ücretleri birbirinden farklıdır. Seçilecek ağa göre maliyet tablosunda artış veya azalış

olabilmektedir. Maliyet analizi; sistemin kurulması, yetkili işlemleri ve seçmen işlemleri olarak üç alt başlığa bölünmüştür.

Sistemin kurulmasında oluşan maliyetler, akıllı sözleşmelerin dağıtılması aşamasını içeren ücretleri içermektedir. Yapılan analizler sonucunda, sistemin kurulum maliyetindeki artışın, varsayılan olarak eklenen verilerden kaynaklandığı tespit edilmiştir. Özellikle sözleşmelerin dağıtımı sırasında, varsayılan olarak sisteme dahil edilen yüksek seçmen sayısı, toplam maliyetin artmasına yol açmaktadır. Bu veriler doğrultusunda varsayılan verilerle, maliyet arasında doğru orantı olduğu tespit edilmiştir.

Yetkili işlemleri kapsamındaki maliyetler, seçim yetkililerinin sistemde gerçekleştirdiği operasyonlar sonucu oluşan ücretleri kapsamaktadır. Yetkili, seçmen, seçim, fakülte ve bölüm yönetimindeki işlemler için oluşan maliyetler, yetkili işlemleri kapsamında değerlendirilmektedir. Yetkili yönetimi, aktif olan yetkililerin diğer yetkilileri yönetebilmesini sağlayan bölümdür. Yetkililerin, seçmenler üzerinde gerçekleştirebildikleri işlemler ise seçmen yönetimini oluşturmaktadır. Yetkili ve seçmen yönetimi kapsamında gerçekleştirilen işlemler için standart gaz kullanımı gerektiğinden, sabit ücretlerin karşılanması zorunlu hale gelmektedir. Seçmen yönetiminin sözleşme dağıtılırken değil de bu aşamada gerçekleştirilmesi, maliyetin artışına neden olan ana unsurdur. Bu yüzden, seçmen ekleme işlemlerinin, sözleşmeler dağıtılırken kullanılan JSON ayar dosyalarıyla yapılması daha doğru bir yaklaşım olacaktır. Fakülte ve bölüm yönetimi, üniversite bünyesinde bulunan fakülte ve bölümlerin eklenmesi, düzenlenmesi, aktif ve pasif hale getirilmesi işlemlerini kapsamaktadır. Bu bölüm içerisinde fakülte ve bölüm ekleme ücretleri diğer işlemlere göre yüksektir. Bir fakültenin aktif veya pasif hale getirilmesindeki oluşan ücret, fakültenin bünyesinde barındırdığı bölümlere göre artış göstermektedir. Bu nedenle, çok fazla bölüme sahip bir fakültenin aktif ve pasif hale getirilmesi diğer tüm işlemlerden daha fazla ücret ödenmesini gerektirecektir. Seçim yönetimi, yetkililerin seçim süreçleri için gerçekleştirdikleri seçim oluşturma, başlatma ve bitirme işlemlerini kapsamaktadır. Prototip ilk geliştirilirken, seçim oluşturma işlemi sırasında seçime katılacak tüm seçmenler seçim bilgileriyle birlikte kaydedilmekteydi. Bu yaklaşımın, çok fazla seçmenin olduğu bir seçimde yüksek gaz kullanımına ve bunun sonucunda da gaz kullanım limitinin aşılmasına neden olduğu tespit edilmiştir. Bu nedenle seçim oluşturma maliyetini en çok etkileyen ana etken seçmen sayısı olmuştur. Yüksek gaz kullanımı, maliyet artışına ve sistemin çalışmasına engel oluşturduğundan dolayı mimari değişiklikler yapılmak zorunda kalınmıştır. Bunun için seçim oluşturulurken kullanılmak üzere seçmen snapshot sistemi

geliştirilmiştir. Böylece, seçim oluşturulurken seçmen uygunluk kriteri ve zaman damgası blokzinciri üzerine kalıcı olarak kaydedilmiştir. Seçim bilgileri içerisinde bütün seçmenlerin bilgilerinin tutulmasına gerek kalmadığından yüksek gaz kullanım sorunu çözülmüştür. Bu bağlamda seçim yönetimi değerlendirildiğinde, en son gelinen nokta itibarıyla seçim maliyetini en çok artıran işlemin aday sayısı olduğu görülebilmektedir.

Seçmen işlemleri kapsamındaki maliyetler, seçmenlerin oy kullanma aşamasında gerçekleştirdikleri ücretleri kapsamaktadır. Oy kullanma sürecine ilişkin maliyetler; sıfır bilgi ispatı (ZKP) kanıtının oluşturulması, 2048-bit anahtarla şifrelenmiş oyun blokzincire iletilmesi, kanıtın akıllı sözleşme aracılığıyla doğrulanması ve şifreli verinin saklanması işlemlerinden kaynaklanmaktadır. Anahtar uzunluğu burada kullanılan gazın artmasına neden olan en önemli etmendir. Tüm bu işlemlerde veri boyutuyla maliyet arasında doğrusal bir ilişki olduğu gözlemlenmiştir.

Elektronik üniversite ölçekli bir blokzinciri oylama sistemi geliştirilirken, işlem ve hesaplama maliyetleri önemli bir unsur olarak öne çıkmaktadır. Bu çalışmada önerilen sistemin, özellikle akıllı sözleşme işlemlerinde kullanılan gaz ücretleri açısından, başka çalışmada sunulan modele kıyasla daha maliyetli olduğu görülmektedir [76]. Daha sade bir yapı benimsenmişken, bizim sistemimizde kimlik doğrulama, oy şifreleme ve çok aşamalı veri doğrulama gibi ek güvenlik ve ölçeklenebilirlik önlemleri yer almaktadır. Bu gelişmiş yapı, işlem başına daha fazla gaz tüketimine neden olmakta ve toplam maliyeti artırmaktadır. Ancak, bu ek maliyetin, sistemin güvenliği ve güvenilirliği açısından kabul edilebilir bir düzeyde olduğu değerlendirilmektedir.

Sistemin doğru ve verimli bir şekilde çalıştığını değerlendirebilmek için performans analizi yapılmıştır. Bu analiz, oylama sisteminin ölçeklenebilir ve güncellenebilir bir yapıya sahip olduğunun görülebilmesi için kullanılabilecek etkili bir yöntemdir. Analiz kapsamında, sistemin işlem kapasitesi ve kaynak kullanımı detaylı olarak incelenmiştir. Performans analizi, kişisel bilgisayar üzerinde gerçekleştirilen verilerden ortalama değerler hesaplanarak hazırlanmıştır. Bu bağlamda üniversitelerin, bu uygulamayı bir bulut ortamında çalıştırılacağı varsayılmalı ve başka parametrelerin de devreye gireceği unutulmamalıdır.

Performans değerlendirmesi için ilk aşamada oy kullanma işlemi için çalışmalar yapılmıştır. Bu kapsamda, saniye başına kullanılabilecek oy sayısı hesaplanabilmiştir. Fakat oylama sistemi geliştirilirken ilk aşamada, Paillier şifreleme algoritması için 32-bit anahtar

uzunluğu seçilmiştir. Şifrelenmiş oylar üzerinde gerçekleştirilen homomorfik toplamın oluşturulması ve bu toplam üzerindeki şifreli verilerin çözülerek açık sonuca ulaşılması, akıllı sözleşmeler üzerinde gerçekleştirilmiştir. Yapılan işlemler sonucunda 32-bit yetersiz görülmüş ve 2048-bit anahtar uzunluğunun kullanılmasına karar verilmiştir. Fakat 2048-bit anahtar boyutuyla şifrelenmiş çok sayıda oy, homomorfik olarak akıllı sözleşmeler üzerinde toplanamamıştır. Çünkü, bu boyuttaki işlemler, akıllı sözleşmelerin gaz kullanım limitine takılmaktadır. Bu nedenle, şifrelenmiş oylar parçalanarak byte olarak sözleşmeler tarafından kaydedilmiş, homomorfik toplama işlemi ve bu toplam üzerindeki şifreli verilerin çözülmesi, tarayıcı üzerinde kullanılan JavaScript diliyle yapılmıştır. Bunun sonucunda oylama başına kullanılan gaz miktarında artış olmuş ve saniye başına oy kullanım oranında düşüş gözlemlenmiştir. Bunun yanı sıra, oy kullanma işleminin eşzamanlı gerçekleşmesi için çalışmalar gerçekleştirilmiş ve sistemin göstermiş olduğu performans değerleri incelenmiştir. Bu kapsamda, performans değerlendirmesi amacıyla 2, 5 ve 10 adet seçmenin eşzamanlı olarak oy kullanması sağlanmıştır. Yerel bilgisayar üzerinde eşzamanlı seçmen sayısındaki artış, sistemin işlem hacminde belirgin bir artışına neden olmuştur.

Sistemde, güvenliğin ve bütünlüğün sağlanması amacıyla hash fonksiyonu kullanılmaktadır. Hash fonksiyonu olarak Circom devreleri için optimize edilen Poseidon tercih edilmiştir. Sistemin hash üretim hızı, doğrudan oy kullanma performansını etkileyeceğinden dolayı değerlendirilmesi gereken bir parametredir. Farklı seçmen sayılarının ve seçim senaryolarının yer aldığı durumlarda, sıralı ve eşzamanlı oy kullanma işlemleri sırasında hash üretim hızı değerlendirilmiş; sistemin hash üretim kapasitesinin yeterli düzeyde olduğu tespit edilmiştir. Hash üretim hızlarında işlem hacmi bazında yüksek rakamlara ulaşılmaktadır. Fakat, sistemin değerlendirmesi sistem içerisindeki en yavaş bileşene göre yapıldığından, oylama performansını etkileyecek en önemli unsurun sıfır bilgi ispatı kanıtı ve Paillier şifreleme işlemi olduğu gözlemlenmektedir.

Oy kullanma aşamasında, oyun geçerliliğinin doğrulanması için sıfır bilgi ispatı kanıtları üretilmektedir. Kanıt üretim performansı, oy kullanma hızını doğrudan etkilediğinden, bu bölümde yapılacak analizler sistemin genel performans değerlendirmesine katkılar sağlayacaktır. Bu nedenle, sıfır bilgi ispatı kanıtlarının üretim performansını değerlendirmek amacıyla analizler yapılmıştır. Bu bağlamda sıfır bilgi ispatı devresinde constraints ile işlem hacmi arasında ter orantılı bir ilişki gözlemlenmiştir. Constraint sayısındaki artış, kanıt üretme süresinde bir artışa neden olmakta ve böylece birim zamandaki üretilen kanıt sayısı

düşmektedir. Fakat gerçekleştirilen analizler, dakika başına üretilen kanıt sayısının kabul edilebilir aralıkta olduğunu göstermektedir. Eşzamanlı bir şekilde kanıt üretilmesinin performansa olan etkisini değerlendirmek amacıyla eşzamanlı oylama işlemleri gerçekleştirilmiştir. Bu aşamada, 2, 5 ve 10 adet seçmenin eşzamanlı oy kullanımı sırasında kanıt üretim hızları değerlendirilmiştir. Bu bağlamda, sıralı ve eşzamanlı gerçekleştirilen testlerde, işlem hacmi yönünden olumlu bir gelişme gözlemlenmemiştir. Sistemin, eşzamanlı işlem hacmi performansının artırılması için bu bölümde geliştirmelerin yapılması önerilmektedir.

Üretilen kanıtların, tarayıcı ve akıllı sözleşme üzerinde doğrulanması gerekmektedir. Bu nedenle, kanıt doğrulama sürecinin hızı, sistem performansını doğrudan etkileyecek unsurlardan birisidir. Bu kapsamda, tarayıcı ve akıllı sözleşmeler üzerinde kanıtların doğrulanması ayrı ayrı değerlendirilmiştir. Bu değerlendirmeler, kanıt doğrulama sürecinin hem sıralı hem de eşzamanlı gerçekleşmesi durumlarını da içermektedir. Her durumda, tarayıcı üzerinde JavaScript ile gerçekleştirilen doğrulama işleminin, akıllı sözleşmeden daha hızlı olduğu tespit edilmiştir. Öte yandan, oylama sisteminde sıfır bilgi ispatı işlemleri için toplam 4.0MB dosya indirilmesi gerekmektedir. Bu dosyalar, oy kullanma aşamasında seçmenin tarayıcısı üzerinde otomatik olarak yüklenmekte ve tarayıcı önbelleği (browser cache) üzerinde tutulmadığı durumda, her oy kullanma işlemi için yeniden indirilmektedir. Ayrıca, bu dosyalar içerisindeki 2.3MB boyutlu WASM circuit dosyası, hızın düşük olduğu durumlarda bottleneck oluşmasına neden olmaktadır.

Oylama işlemi sırasında, oyların şifrelenmesi işlemi için Paillier şifreleme algoritması kullanılmaktadır. Bu nedenle, Paillier şifreleme algoritmasının sistem içerisinde değerlendirilmesi için analizler yapılmıştır. Paillier şifreleme algoritması; anahtar üretimi, şifreleme, homomorfik toplama ve şifre çözme olmak üzere dört aşamada kullanılmaktadır. Sistemin sıralı ve eşzamanlı oy kullanımı senaryolarında, Paillier şifreleme algoritmasının performans değerlendirmesi yapılmıştır. Farklı senaryolarda, Paillier şifreleme algoritmasının performansını etkileyen en önemli unsurun anahtar uzunluğu olduğu tespit edilmiştir. Anahtar boyutundaki artış şifreleme, toplama ve deşifreleme süresini artırmaktadır. Böylece, dakika başına düşen işlem sayısında düşüş gözlemlenmektedir. Ayrıca, eşzamanlı gerçekleştirilen işlemlerin belirli ölçüde sıralı işlemlerden hızlı olduğu verilerle gösterilmiştir.

Paillier şifreleme algoritmasının sistemdeki işlemlere ilişkin zaman karmaşıklığı analizi, kullanılan anahtar uzunluğunun işlem performansı üzerindeki etkisini ortaya koymaktadır. Bu analiz, 2048-bit anahtar uzunluğuyla şifreleme, homomorfik toplama ve deşifreleme işlemlerini değerlendirmektedir. 2048-bit anahtar uzunluğunun güncel güvenlik standartlarını sağladığı bilinmektedir. Ancak bu karmaşıklık, büyük ölçekli seçimler açısından dikkate alınmalıdır.

Bu çalışma, üniversite ölçeğinde gerçekleştirilen bir uygulama olması nedeniyle, benzer alandaki çalışmalarla karşılaştırılabilecek nitelikte önemli bir örnektir. Örneğin, [73] numaralı çalışmada şifreleme işlemi için AES veya DES algoritmaları tercih edilirken, bütünlük kontrolü amacıyla SHA-256 hash fonksiyonu kullanılmıştır. Söz konusu çalışmada, oylama işleminin tamamlanabilmesi için oyların şifresinin çözülmesi gerekmektedir. Bu durum, bireysel oyların açığa çıkma ihtimalini doğurduğundan, güvenlik kriterinin sağlanmasına engel olmaktadır. Ayrıca, oyların doğrulanabilmesi için herhangi bir mekanizma bulunmamaktadır. Bu çalışmada temel güvenlik mekanizmalarıyla yetinilmişken, önerdiğimiz sistemde seçim bütünlüğü, oy gizliliği ve seçmen doğrulaması gibi kritik unsurlar için daha kapsamlı ve saldırılara karşı dayanıklı yöntemler uygulanmıştır. Sunulan elektronik oylama sistemi, önerilen diğer modele kıyasla daha gelişmiş güvenlik önlemleri içermesi bakımından öne çıkmaktadır. Bu durum, sistemin özellikle üniversite gibi güvenlik hassasiyeti yüksek ortamlarda tercih edilebilirliğini artırmaktadır. Bu alanda yapılan bir diğer çalışma, blokzinciri tabanlı bir oylama sistemi olup, kapsamlı değerlendirmelere yer vermektedir [74]. Söz konusu çalışmada, sistemin yerel ve bulut ortamlarında gösterdiği performans sonuçları analiz edilmiştir. Ancak, güvenlik önlemlerinin yetersiz olduğu ve performans sonuçlarının mevcut sistemimize kıyasla daha düşük seviyede kaldığı tespit edilmiştir. Çalışmada blokzinciri üzerindeki işlem trafiği sınırlı tutulurken, bizim modelimiz optimize edilmiş akıllı sözleşme yapıları ve paralel işlem mekanizmaları sayesinde daha fazla seçmenin aynı anda oy kullanabilmesine olanak tanımaktadır. Bu iyileştirmeler sayesinde sistem, büyük kullanıcı gruplarında dahi performans kaybı yaşamadan etkin biçimde çalışabilmektedir. Blokzinciri tabanlı oylama sistemlerine yönelik bir diğer örnek ise [75] numaralı çalışmadır. Bu çalışmada sözleşme sayısının azlığı ve sistemin iş akışı açısından daha basit bir yapıya sahip olması nedeniyle maliyetlerin, bizim çalışmamıza kıyasla daha düşük olduğu görülmektedir. Bizim sistemimizde seçmen sayısının fazla olması maliyeti artıran bir unsur iken, karşılaştırılan sistemde de seçmen sayısı, seçmenlere ait e-posta ve kimlik (ID) bilgileri ile seçim ayarları gibi unsurlar maliyet üzerinde

etkili olmuştur. Söz konusu çalışmada oyların şifrenmesi için Paillier şifreleme algoritması kullanılmış, ancak oyların doğrulanmasına yönelik herhangi bir mekanizma önerilmemiştir.

Geliştirdiğimiz sistem, oyların gizliliğini sağlamak amacıyla homomorfik şifreleme algoritması, oy doğrulamasını gerçekleştirmek için sıfır bilgi ispatı (Zero-Knowledge Proof) ve sistem altyapısı olarak blokzinciri teknolojisini kullanmaktadır. Bu bileşenlerin birlikte ve optimize biçimde uygulanması sonucunda, sistemimiz benzer çalışmalara kıyasla yüksek düzeyde verimlilik ve güvenlik sunmaktadır.



## 6. SONUÇ VE ÖNERİLER

Dijitalleşmenin getirdiği yenilikler, kriptografik tekniklerin gücü ve blokzinciri altyapısıyla, geleneksel oylama sistemlerinin elektronik oylama sistemlerine dönüşümünü modern bir yaklaşımla gerçekleştirebilmek amaçlanmaktadır. Geliştirilen sistemin; güvenilirlik, şeffaflık, doğrulanabilirlik ve gizlilik kriterlerini sağlaması hedefler arasındadır. Bu dört önemli kriter tüm çalışmanın temel yapı taşını oluşturmaktadır.

Sıfır bilgi ispatı, oylama sisteminin hedeflerini sağlaması için kullanılan yöntemlerden bir tanesidir. Oy kullanıldığında oluşturulan sıfır bilgi ispatı kanıtı ve açık parametreler, blokzincir üzerine yazılmaktadır ve böylece ağa erişim izni olan herkes tarafından görülebilmektedir. Sıfır bilgi ispatı kanıtı, kullanılan oyun içeriğinin görüntülenmesini engellemektedir. Bu nedenle, ağdaki kişiler tarafından görülmesi herhangi bir olumsuz duruma neden olmamaktadır. Bunun yanı sıra, erişilen kanıt ve açık parametrelerle oyun geçerli olduğu doğrulanabilmektedir. Böylece oy gizli kalırken, doğrulama yapılabilmesi sağlanmaktadır. Gerçekleştirilen testler ve performans analizi çalışmalarında sıfır bilgi ispatının doğru bir şekilde çalıştığına dair veriler elde edilmiştir.

Blokzincir, işlemlerin güvenli ve şeffaf bir şekilde kayıt altında tutmak için kullanılan bir teknolojidir. Bu çalışmada, akıllı sözleşme desteği sebebiyle Ethereum platformu tercih edilmiştir. Sistemde kullanılan akıllı sözleşmelerin dağıtılması için gereken maliyetlerin belirlenmesi için maliyet analizi yapılmıştır. Akıllı sözleşmelerle gerçekleştirilen işlemlerin değerlendirilmesi amacıyla da performans analizi oluşturulmuş ve analizler sonucunda kritik veriler elde edilmiştir. Sistemin kurulması aşamasında seçmen veya yetkilisi sayısı, maliyet arasında doğrusal bir ilişki olduğu tespit edilmiştir. Seçmen sayısının artışı veri boyutunu artırmakta ve bu durum da sistemde harcanan gaz miktarının yükselmesine neden olmaktadır. Geliştirilen oy sisteminin maliyetini artıran nedenlerden bir tanesi de ağ seçimidir. Güvenli ve düşük gaz maliyetine sahip bir ağın tercih edilmesi, işlem sürecinde oluşan ücretlerin azaltılmasına katkı sağlamaktadır.

Sistemin ilk aşamalarında seçim oluşturma işlemlerinde seçim bilgileri içerisinde seçime katılacak olan tüm seçmenlerin bir listesi eklenmiştir. Fakat bu durumun maliyet ve performans değerlendirmelerinde iyi sonuçlar vermediği gözlemlenmiştir. Yüksek seçmen sayısı gaz kullanım limitinin aşılmasına ve seçim başına düşen maliyetin artmasına neden olmuştur. Bu

nedenle, seçim oluşturulurken sistemde bulunan ve seçime katılabilecek seçmenlerin anlık görüntüsünü (snapshot) alacak bir yapı tasarlanmıştır. Böylece aktif olan seçim için seçmen manipülasyonu yapılması engellenmiş ve yüksek boyutlu veri tekrarının önüne geçilmiştir. Bu yapı, gaz kullanım maliyetini düşürmüş ve sistemin performansında artış olmasını sağlamıştır.

Sistemde kullanılan oyların gizli bir şekilde saklanması için Paillier şifreleme algoritması kullanılmıştır. Seçmenin oyu 2048-bit anahtar uzunluğuyla şifrelediği için homomorfik toplama işlemi yapılırken, akıllı sözleşmelerin gaz kullanım limiti aşılmaktadır. Bu durumun önüne geçmek için oylar tarayıcı üzerine çekilerek JavaScript diliyle homomorfik toplanır ve şifre çözme işlemi bu şekilde gerçekleştirilir. Bu işlem blokzinciri ağı dışında (off-chain) yapılmaktadır. Bu bölümde yapılacak yeni çalışmalar, burada oluşabilecek güvenlik sorunlarının önüne geçilmesini sağlayacaktır. Ayrıca yüksek boyutlu anahtar uzunluğu oy başına ödenmesi gereken ücretin artışına neden olmaktadır. Burada uygulanabilecek optimizasyon yöntemleri maliyetlerin azalmasını sağlayacaktır. Oyların gizli kalarak doğrulanabilmesini sağlamak için sıfır bilgi ispatından yararlanılmıştır. Yapılan çalışmalar neticesinde, kanıt üretme ve doğrulama sürelerinde başarılı sonuçlar elde edilmiştir. Oyların doğrulanması için kanıtlar ve açık parametreler blokzincir üzerinde görülebilmektedir. Bu yöntemin tercih edilmesindeki amaç sisteme olan güvenin sağlanmasını sağlamaktır. Sıfır bilgi ispatı ve şifreleme algoritmaları sistemi oluşturan en kritik bileşenlerdir. Kriptografik algoritmalar kapsamında hash fonksiyonundan yararlanılmıştır. Burada devre içerisinde kullanım kolaylığı ve gösterdiği performans değerlerinden dolayı Poseidon algoritması tercih edilmiştir.

Sistem üzerinde gerçekleştirilen test ve analiz işlemleri sonucunda elde edilen bulgular; blokzinciri tabanlı oylama sisteminin, üniversite çapında düzenlenen oylama süreçlerinde önemli bir potansiyele sahip olduğunu göstermektedir. Elde edilen veriler sistemin yeterli performans gösterdiğini, kullanılan kriptografik tekniklerin günümüz şartlarında yeterli olduğunu göstermektedir. Prototip çalışması kapsamında geliştirilen arayüzün kullanıcı dostu olduğu ve entegrasyonun kolay bir şekilde yapılabilirdiği görülebilmektedir.

## KAYNAKLAR

- [1] Dinçkol, B., 2017, Atina Demokrasisinden Roma Cumhuriyetine-" Demos" tan" Populus Romanus" a. Marmara Üniversitesi Hukuk Fakültesi Hukuk Araştırmaları Dergisi, 23(3), 751-765.
- [2] Ekinci, E., 2016, Devrimden günümüze fransız siyasi sisteminin evrimi. Kırıkkale Üniversitesi Sosyal Bilimler Dergisi, 6(1), 149-172.
- [3] Vladucu, M. V., Dong, Z., Medina, J., & Rojas-Cessa, R., 2023, E-voting meets blockchain: A survey. IEEE Access, 11, 23293-23308.
- [4] Killer, C., Rodrigues, B., Scheid, E. J., Franco, M. F., & Stiller, B., 2022, Blockchain-based voting considered harmful?. IEEE Transactions on Network and Service Management, 19(3), 3603-3618.
- [5] Taş, R., & Tanrıöver, Ö. Ö., 2020, A systematic review of challenges and opportunities of blockchain for E-voting. Symmetry, 12(8), 1328.
- [6] Nikhare, R. V., & Chandavarkar, B. R., 2023, A comparative analysis of traditional versus blockchain-based voting systems. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.
- [7] Jafar, U., Aziz, M. J. A., & Shukur, Z., 2021, Blockchain for electronic voting system—review and open research challenges. Sensors, 21(17), 5874.
- [8] Hanifatunnisa, R., & Rahardjo, B., 2017, Blockchain based e-voting recording system design. In 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA) (pp. 1-6). IEEE.
- [9] Çabuk, U. C., Adiguzel, E., & Karaarslan, E., 2020, A survey on feasibility and suitability of blockchain techniques for the e-voting systems. arXiv preprint arXiv:2002.07175.
- [10] Aberer, K., & Hauswirth, M., 2002, An Overview of Peer-to-Peer Information Systems. In WDAS (Vol. 14, No. 2002, pp. 171-188).
- [11] Chang, V., Wills, G., & De Roure, D., 2010, A review of cloud business models and sustainability. In 2010 IEEE 3rd International Conference on Cloud Computing (pp. 43-50). IEEE.
- [12] Bhargava, A., Rana, A., & Bhargava, D., 2022, Blockchain based digital voting system: A secure and decentralized electoral process. In 2022 5th International Conference on Contemporary Computing and Informatics (IC3I) (pp. 479-485). IEEE.
- [13] Čapko, D., Vukmirović, S., & Nedić, N., 2022, State of the art of zero-knowledge proofs in blockchain. In 2022 30th Telecommunications Forum (TELFOR) (pp. 1-4). IEEE.
- [14] Shahzad, B., & Crowcroft, J., 2019, Trustworthy electronic voting using adjusted blockchain technology. IEEE Access, 7, 24477-24488.

- [15] Vokerla, R. R., Shanmugam, B., Azam, S., Karim, A., De Boer, F., Jonkman, M., & Faisal, F., 2019, An overview of blockchain applications and attacks. In 2019 international conference on vision towards emerging trends in communication and networking (ViTECoN) (pp. 1-6). IEEE.
- [16] Haber, S., & Stornetta, W. S., 1991, How to time-stamp a digital document (pp. 437-455). Springer Berlin Heidelberg.
- [17] Kumar, N., Kumar, K., Aeron, A., & Verre, F., 2025, Blockchain technology in supply chain management: Innovations, applications, and challenges. Telematics and Informatics Reports, 100204.
- [18] Hellani, H., Samhat, A. E., Chamoun, M., El Ghor, H., & Serhrouchni, A., 2018, On blockchain technology: Overview of bitcoin and future insights. In 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET) (pp. 1-8). IEEE.
- [19] Georgiadis, E., 2019, How many transactions per second can bitcoin really handle? Theoretically. Cryptology ePrint Archive.
- [20] Monrat, A. A., Schelén, O., & Andersson, K., 2019, A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access, 7, 117134-117151.
- [21] Islam, M. R., Rahman, M. M., Mahmud, M., Rahman, M. A., Mohamad, M. H. S., & Embong, A. H., 2021, A review on blockchain security issues and challenges. In 2021 IEEE 12th control and system graduate research colloquium (ICSGRC) (pp. 227-232). IEEE.
- [22] Lin, I. C., & Liao, T. C., 2017, A survey of blockchain security issues and challenges. Int. J. Netw. Secur., 19(5), 653-659.
- [23] Krishnan, R., Smith, M. D., & Telang, R., 2003, The economics of peer-to-peer networks. Journal of Information Technology Theory and Application (JITTA), 5(3), 6.
- [24] Zhang, C., Sun, J., Zhu, X., & Fang, Y., 2010, Privacy and security for online social networks: challenges and opportunities. IEEE network, 24(4), 13-18.
- [25] Nakamoto, S., & Bitcoin, A., 2008, A peer-to-peer electronic cash system. Bitcoin.– URL: <https://bitcoin.org/bitcoin.pdf>, 4(2), 15.
- [26] Mourtzis, D., Angelopoulos, J., & Panopoulos, N., 2023, Blockchain integration in the era of industrial metaverse. Applied Sciences, 13(3), 1353.
- [27] Colomo-Palacios, R., Sánchez-Gordón, M., & Arias-Aranda, D., 2020, A critical review on blockchain assessment initiatives: A technology evolution viewpoint. Journal of Software: evolution and process, 32(11), e2272.
- [28] Angelis, J., & Da Silva, E. R., 2019, Blockchain adoption: A value driver perspective. Business horizons, 62(3), 307-314.

- [29] Goldwasser, S., Micali, S., & Rackoff, C., 2019, The knowledge complexity of interactive proof-systems. In *Providing sound foundations for cryptography: On the work of shafi goldwasser and silvio micali* (pp. 203-225).
- [30] Chahoki, A. Z., Herlihy, M., & Roveri, M., 2025, SoK: Concurrency in Blockchain-A Systematic Literature Review and the Unveiling of a Misconception. *arXiv preprint arXiv:2506.01885*.
- [31] Lawlor, B., Chalk, S., Frey, J., Hayashi, K., Kochalko, D., Shute, R., & Sopek, M., 2025, Blockchain technology: driving change in the scientific research workflow. *Pure and Applied Chemistry*, 97(4), 279-330.
- [32] Paul, A., & Das, B., 2025, A Review on the Minimization of Energy Consumption in Blockchain Technology. In *2025 8th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* (pp. 1-5). IEEE.
- [33] Usta, A., & Doğantekin, S., 2018, Blockchain 101 v2, Bankalararası Kart Merkezi.
- [34] Küfeoğlu, S., & Özkuran, M., 2019, Bitcoin mining: A global review of energy and power demand. *Energy Research & Social Science*, 58, 101273.
- [35] Ouyang, Z., Shao, J., & Zeng, Y., 2021, PoW and PoS and related applications. In *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)* (pp. 59-62). IEEE.
- [36] Chaganti, R., Bhushan, B., & Ravi, V., 2023, A survey on Blockchain solutions in DDoS attacks mitigation: Techniques, open challenges and future directions. *Computer Communications*, 197, 96-112.
- [37] Ethereum Community, 2025, Ethereum Whitepaper, <https://ethereum.org/en/whitepaper/>, [Ziyaret Tarihi: 01 Mart 2025]
- [38] Kapengut, E., & Mizrach, B., 2023, An event study of the ethereum transition to proof-of-stake. *Commodities*, 2 (2), 96-110.
- [39] Ahn, J., Yi, E., & Kim, M., 2024, Ethereum 2.0 Hard Fork: Consensus Change and Market Efficiency. *The Journal of The British Blockchain Association*.
- [40] What is Ethereum's block structure? <https://www.educative.io/answers/what-is-ethereums-block-structure> [Ziyaret Tarihi: 01 Mart 2025]
- [41] Zhou, H., Milani Fard, A., & Makanju, A., 2022, The state of ethereum smart contracts security: Vulnerabilities, countermeasures, and tool support. *Journal of Cybersecurity and Privacy*, 2(2), 358-378.
- [42] Kezadri Hamiaz, M., & Driss, M., 2025, Ethereum Smart Contracts Under Scrutiny: A Survey of Security Verification Tools, Techniques, and Challenges. *Computers*, 14(6), 226.

- [43] Liu, H., Luo, X., Liu, H., & Xia, X., 2021, Merkle tree: A fundamental component of blockchains. In 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS) (pp. 556-561). IEEE.
- [44] Kabla, A. H. H., Anbar, M., Manickam, S., Al-Amiedy, T. A., Cruspe, P. B., Al-Ani, A. K., & Karuppayah, S., 2022, Applicability of intrusion detection system on Ethereum attacks: a comprehensive review. IEEE Access, 10, 71632-71655.
- [45] Jani, S., 2017, An overview of ethereum & its comparison with bitcoin. Int. J. Sci. Eng. Res, 10(8), 1-6.
- [46] Feng, H., Hu, Y., Kou, Y., Li, R., Zhu, J., Wu, L., & Zhou, Y., 2024, {SlimArchive}: A Lightweight Architecture for Ethereum Archive Nodes. In 2024 USENIX Annual Technical Conference (USENIX ATC 24) (pp. 1257-1272).
- [47] Teng, H., Tian, W., Wang, H., & Yang, Z., 2022, Applications of the decentralized finance (defi) on the ethereum. In 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) (pp. 573-578). IEEE.
- [48] Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q., 2020, Symmetric encryption algorithms: Review and evaluation study. International Journal of Communication Networks and Information Security, 12(2), 256-272.
- [49] Zhang, Q., 2021, An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption. In 2021 2nd international conference on computing and data science (CDS) (pp. 616-622). IEEE.
- [50] Kumar, M. R., Jeya, I. J. S., & Athira, S., 2025, Analysis of Encryption Techniques for Secure Data in Cloud Environment. In 2025 International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-4). IEEE.
- [51] Denton, B., & Adhami, R., 2011, Modern hash function construction. In Proceedings of the International Conference on Security and Management (SAM) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- [52] Günsay, E., A, 2021, Zero-Knowledge range proofs and applications on decentralized contructions, Thesis (Master), Middle East Technical University,
- [53] Arora, S., & Barak, B. 2009, Computational complexity: a modern approach. Cambridge University Press.
- [54] Nithya, B., & Sripriya, P., 2016, A review of cryptographic algorithms in network security. International Journal of Engineering and Technology (IJET), 8(1), 324-331.
- [55] Zhou, L., Diro, A., Saini, A., Kaisar, S., & Hiep, P. C., 2024, Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities. Journal of Information Security and Applications, 80, 103678.

- [56] Saad, S. M. S., & Radzi, R. Z. R. M., 2020, Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *International Journal of Innovative Computing*, 10(2).
- [57] Kaur, S., Chaturvedi, S., Sharma, A., & Kar, J., 2021, A research survey on applications of consensus protocols in blockchain. *Security and Communication Networks*, 2021(1), 6693731.
- [58] de Ocariz Borde, H. S., 2022, An overview of trees in blockchain technology: merkle trees and merkle patricia tries. University of Cambridge: Cambridge, UK.
- [59] Jezek, K., 2021, Ethereum data structures. arXiv preprint arXiv:2108.05513.
- [60] Zhai, S., Yang, Y., Li, J., Qiu, C., & Zhao, J., 2019, Research on the Application of Cryptography on the Blockchain. In *Journal of Physics: Conference Series* (Vol. 1168, p. 032077). IOP Publishing.
- [61] Gupta, S., Goyal, N., & Aggarwal, K., 2014, A review of comparative study of md5 and ssh security algorithm. *International Journal of Computer Applications*, 104(14).
- [62] Aydın, N., 2022, Klasik Oylama Yönteminin Elektronik Oylama Yöntemlerine Evrilmesi. *Akdeniz Üniversitesi Hukuk Fakültesi Dergisi*, 12(2), 711-744.
- [63] Aktan, C. C., & Yay, S., 2021, Kleroterion: Antik Yunan'da Kayırmacılık ve Kollamacılığı Önlemeye Yönelik Bir Sortisyon-Rotasyon Uygulaması. *Organizasyon ve Yönetim Bilimleri Dergisi*, 13(1), 1-19.
- [64] Hallene, A., Allen J. M., 2024, The Evolution of Voting Technology, [https://www.americanbar.org/groups/senior\\_lawyers/resources/voice-of-experience/2024-september/the-evolution-of-voting-technology/](https://www.americanbar.org/groups/senior_lawyers/resources/voice-of-experience/2024-september/the-evolution-of-voting-technology/), [Ziyaret Tarihi: 18 Haziran 2025]
- [65] Zukerman, T. D., 1925, The Voting Machine: Report on the History, Use, and Advantages of Mechanical Means for Casting and Counting Ballots.
- [66] Risnanto, S., Abd Rahim, Y. B., & Herman, N. S., 2020, Success implementation of e-voting technology in various countries: a review. *FoITIC*, 150-155.
- [67] Newkirk, G. M., 2008, Trends in American trust in voting technology. *InfoSENTRY Services*, (March 17, 2008), white paper.
- [68] Everett, S. P., Greene, K. K., Byrne, M. D., Wallach, D. S., Derr, K., Sandler, D., & Torous, T., 2008, Electronic voting machines versus traditional methods: Improved preference, similar performance. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 883-892).
- [69] Krimmer, R., Duenas-Cid, D., & Krivonosova, I., 2021, New methodology for calculating cost-efficiency of different ways of voting: is internet voting cheaper?. *Public money & management*, 41(1), 17-26.

- [70] Hajian Berenjestanaki, M., Barzegar, H. R., El Ioini, N., & Pahl, C., 2023, Blockchain-based e-voting systems: a technology review. *Electronics*, 13(1), 17.
- [71] Alamleh, H., & AlQahtani, A. A. S., 2021, Analysis of the design requirements for remote internet-based E-voting systems. In 2021 IEEE World AI IoT Congress (AIIoT) (pp. 0386-0390). IEEE.
- [72] Onur, C., 2023, ElectAnon: A Blockchain-based, Anonymous, Robust, and Scalable Ranked-choice Voting Protocol, Thesis (Master), Boğaziçi University
- [73] Singh, A., & Chatterjee, K., 2018, SecEVS: secure electronic voting system using blockchain technology. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 863-867). IEEE.
- [74] Cadiz, J. V., Mariscal, N. A. M., & Ceniza-Canillo, A. M., 2021, An empirical analysis of using blockchain technology in e-voting systems. In 2021 1st International Conference in Information and Computing Research (iCORE) (pp. 78-83). IEEE.
- [75] Dagher, G. G., Marella, P. B., Milojkovic, M., & Mohler, J., 2018, Broncovote: Secure voting system using ethereum's blockchain.
- [76] Lyu, J., Jiang, Z. L., Wang, X., Nong, Z., Au, M. H., & Fang, J., 2019, A secure decentralized trustless E-voting system based on smart contract. In 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 570-577). IEEE.

# İNTİHAL RAPORU İLK SAYFASI



Sayfa 2 of 173 - Bütünlük Genel Bakış

Gönderi Kimliği trn:oid::1:3280330266

## 3% Genel Benzerlik

Her veri tabanı için çıkarılan kaynaklar da dâhil tüm eşleşmelerin kombine toplamı.

### Rapordan Filtrelenen

- Bibliyografya

### Ön Sıradaki Kaynaklar

- 2% İnternet kaynakları
- 2% Yayınlar
- 1% Gönderilen çalışmalar (Öğrenci Makaleleri)

### Bütünlük Bayrakları

#### İnceleme için 0 Bütünlük Bayrağı

Herhangi bir şüpheli metin manipölasyonu belirlenmedi.

Sistemimizin algoritmaları bir belgede, onu normal bir gönderiden ayırabilecek her türlü tutarsızlığı derinlemesine inceler. Tuhaf bir şey fark edersek incelemeniz için bayrak ekleriz.

Bir Bayrak mutlaka bir sorun olduğunu göstermez. Ancak daha fazla inceleme için dikkatinizi vermenizi öneririz.

## KURUM İZNİ YAZILARI

**Uyarı:** Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

- ☐ Kurum izni gerekmektedir.
- ☒ Kurum izni gerekmemektedir.

Ahmet Can Çiçek  
(İmza)

