T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

Blok Zincir ile Oy Kullanma Sisteminde Seçmenler için Soğuk Cüzdan

Sefa Nadir YILDIZ

Danışman
Doç. Dr. Mehmet Göktürk

Mayıs, 2019
Gebze, KOCAELİ

Bu çalışma 30/05/2019 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Mehmet Göktürk	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Yakup Genç	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

ÖNSÖZ

Bu bitirme projesinin hazırlanmasında emeği geçenlere, projenin son halini almasında yol gösterici olan Sayın Doç. Dr. Mehmet Göktürk hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Mayıs, 2019 **Sefa Nadir YILDIZ**

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÖZET

Bitcoin ile hayatımıza giren blockchain teknolojisi son dönemde daha sık duyulmaya başlandı. Blockchain teknolojisini şifrelenmiş işlem takibi sağlayan bir dağıtık kayıt defteri olarak tanımlayabiliriz. Adından da anlaşılacağı gibi zincirleme bir modelle inşa edilen, takip edilebilen ama kırılamayan bir yapıya sahip olan blockchain teknolojisi, bir merkeze bağlı olmaksızın işlem yapmaya izin verir. Böylece işlemler aracılara bağlı olmaksızın direkt olarak kişiler arasında güvenli bir şekilde gerçekleştirilebilir.

Blockchain teknolojisi bireysel kullanıcılara dijital kimlik üzerinde bugüne kadar benzeri görülmemiş bir kontrol imkanı sağlamaktadır. Dolayısıyla küresel açık bir hesap defteri olan blockchain sadece kripto paraların üretiminde değil birçok farklı alanda güvenlik, saklama, yönetme ve depolama gibi işlemler için kullanılmaktadır.

Bu çalışma seçim sistemlerinde blockchain teknolojisini kullanarak her seçmene özgü soğuk cüzdan oluşturmaktır. Buradaki amaç seçim sistemlerinde, seçime olan katılımdaki hileyi önlemek, tasarruf sağlamak, katılım sayısını, oy adedini eksiksiz ve doğru bir biçimde belirlemektir. Her seçmen bu sistem de bireysel bir kullanıcı olduğu için kendilerine özgü digital kimlik oluşturmak adına parmak ve avuç izleri şifrelenip referans alınmıştır.

Seçmenlerin parmak ve avuç izleri seçimden önce kayıt edilmiştir. Seçim günü, seçmenin oy kullanma anında elini okutması ile kimlik doğrulaması %100 gerçekleştirilir. Parmak ve avuç izi şifrelenmiş bilgisi ile seçim anında blockchain ağında bireysel soğuk cüzdan oluşturulur ve sistem cüzdana otomatik giriş yapılmasını sağlar. Cüzdan oluşturulduktan sonra seçim ekranı otomatik açılır ve seçmen oyunu kullanır.

SUMMARY

Blockchain technology entering our lives, recently began to be heard more often. We can define Blockchain technology as a distributed registry that provides encrypted transaction tracking. As the name implies, the blockchain technology, built with a chain model, which can be traced but not broken, allows operation without a center. Thus, transactions can be carried out safely between persons, regardless of the intermediary.

Blockchain technology provides individual users with unprecedented control over digital identity. Therefore, the global open account bookkeeping blockchain is used not only in the production of crypto coins but also in many different areas such as security, storage, management and storage.

This study covers the creation of cold wallets for each voter using blockchain technology in the selection systems. The aim is to prevent fraud in election participation, the number of votes in a complete and accurate detected. Every voter is an individual user in this system. The Finger and palm prints are encrypted and referenced to create their own unique digital identity.

The finger and palm prints of the voters were recorded in the database before the election. Authentication is performed 100% when the voter was enter by the system his / her hand in the moment of voting on election day. With the finger and palmprint information, individual cold wallets are created in the blockchain network and the system allows automatic entry into the wallet. After the wallet is created, the selection screen opens automatically and the voters send the vote.

1. GİRİŞ

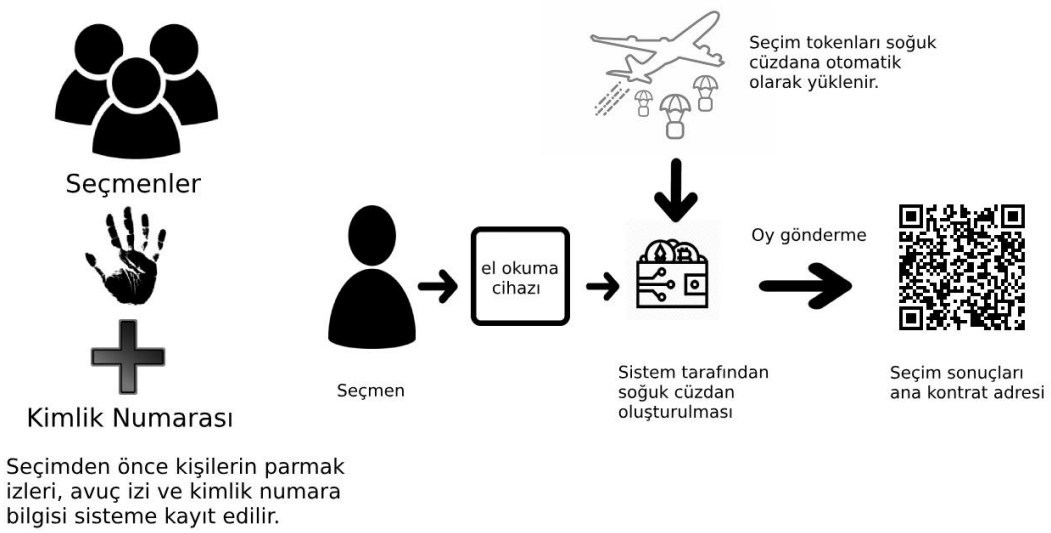
Orijinal adıyla Blockchain, Türkçe karşılığıyla Blok Zinciri teknolojisi aslında tam "adı üstünde" denecek bir tanıma sahiptir. Blok Zinciri, bilgilerin bloklar halinde oluşup merkezi bir veritabanı yerine merkezsiz, herkese açık zincirler halinde saklanması demektir. Bu teknoloji sayesinde internet üzerinden her gelişmesini herkesin takip ettiği, şeffaf, müdahale etmesi neredeyse imkansız umumi veri kayıt sistemleri oluşturulabilir.

Meşhur bir block zinciri örneği düşünecek olursak aklımıza hemen Bitcoin gelir. Bir dijital para birimi olan Bitcoin, üzerinde yapılan tüm işlemlerin açıkça görülebildiği, dışarıdan herhangi bir gücün müdahale edemediği ve yapılan işlemlerin geri alınamadığı bir blok zinciridir. Bizler günlük yaşamda Bitcoin'in daha çok parasal değerini konuşuyor olsak da Bitcoin aslında yeni bir teknoloji çağının başlangıcı, paranın ve ekonominin potansiyel yeni yüzüdür. Günümüz dünyası düşünüldüğünde ekonomiden tıbba, bilimden akademik alanlara, endüstriden ulaşıma kadar tüm sektörler bilgiyle yönetilir. Tüm bu alanlarda bilginin saklanması, işlenmesi, aktarılması kaçınılmazdır. Dolayısıyla Blockchain teknolojisi tüm bu alanlarda kullanılabilir yapıdadır ve görünen o ki bir gün tüm bu alanlarda bir şekilde kullanılacaktır. Bu nedenle blok zinciri teknolojisinin kullanım alanları sınırsınızdır denilebilir

Demokrasi demek özgürlük, şeffaflık ve çoğunluğun kararıyla halkın kendisini yönetmesi demektir. Tüm bu kavramlar doğrultusunda demokrasiyle yönetim hedefine sahip olan herhangi bir ülke, Blok Zinciri teknolojisini kullanarak inanılmaz bir ilerleme kaydedebilir. Herkese açık ve müdahaleye imkan tanımayan Blockchain sistemleriyle yapılacak oylamalar bugüne kadar yapılmış en şeffaf ve en hilesiz oylamalar olacaktır.

Bu proje de oylama sistemlerinde, seçime olan katılımdaki hileyi önlemek, katılım sayısını, oy adedini eksiksiz ve doğru bir biçimde belirlemek ve tasarruf sağlamak için blockchain sistemi üzerinde el okuma cihazı kullanılarak her seçmene özgü bir soğuk cüzdan oluşturmaktır.

Bu projenin işleyiş aşamaları Şekil 1.1'de görülmektedir.

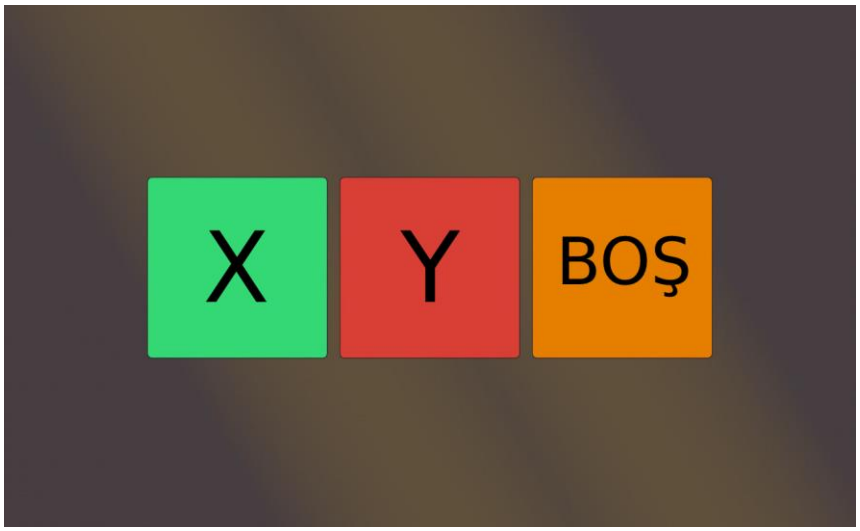


Şekil 1.1 Proje İşleyiş Aşamaları

Seçimden önce, şeçim yönetimi tarafından seçmenlerin kimlik numaraları, parmak izleri ve avuç izi sisteme kayıt edilmelidir. Parmak izleri ve avuç izi şifrelenerek, dijital bir imza haline getirilerek kayıt edilir. Bu aşama da blok zinciri sisteminde güvenlik nedeniyle soğuk cüzdan oluşturulmamaktadır. Seçim günü, seçmenler oy kullanmak için, el okuma cihazına elini okutur. Parmak ve avuç izi bilgisi daha önce sisteme kayıt edildiği için sisteme giriş yaparken doğrulama işlemi kayıtlar üzerinden yapılır. Eşleşme başarılı olursa blok zinciri üzerinde yeni bir soğuk cüzdan oluşur. Oluşan bu soğuk cüzdan farklı akıllı sözleşmeleri desteklemektedir. Yani seçime katılan partilerin akıllı sözleşmeleri birbirinden farklı olsa da bu cüzdan içinde bir arada bulunabilirler. Seçimden önce blok zincirinde seçime katılan parti sayısı kadar akıllı sözleşme yani token oluşturulmuştur. Her parti tokenin adedi seçime katılan seçmen sayısı kadardır.

Cüzdan oluşumu tamamlanınca blok zinciri tarafından her partinin 1 adet tokeni cüzdana aktarılır. Bu yönteme airdrop yöntemi denmektedir. Seçmenin soğuk cüzdanına açık adres ve özel adres olmak üzere 2 anahtar atanır. Özel adres seçmenin kimlik bilgisi, parmak ve avuç izi bilgilerinin şifrelenmesi ile oluşturulmuştur. Cüzdanın kilidini açmaya yarar. Yani kişinin kendisinden başka hiç kimse bu cüzdana erişip oy kullanamaz. Açık adres ise kişinin cüzdanının dışarıdan kontrol edilmesini sağlar ama kim olduğu bilinemez. Çünkü dışarıdan gözüken sadece dijital sayısal bir veridir. Böylelikle anonimlik sağlanmış olur.

Parti tokenları geldikten sonra oy kullanma ara yüzü kiosk makinesi üzerinde aktif hale gelmiş olur. Seçimde kaç parti varsa ekran o kadar buton gözükmektedir. Ekstra olarak boş oy atmak içinde bir buton bulunmaktadır. Atılacak bu boş oy da blok zincirinde bir akıllı sözleşme olarak bulunmaktadır.

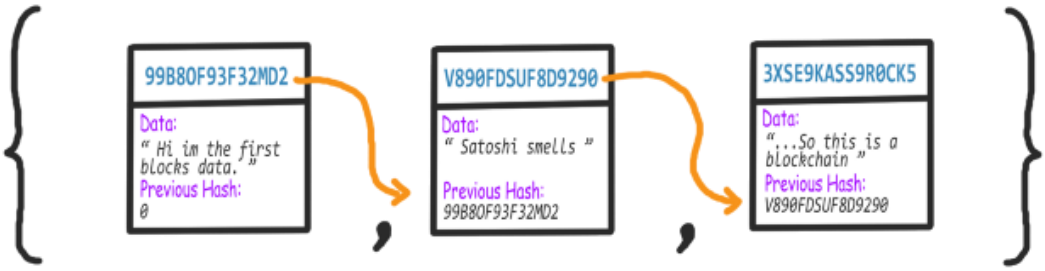


Şekil 1.2 Örnek Oy Kullanma Ekranı

Seçmen kiosk makinesinde oy kullanmak istediği butona tıklayarak oyunu kullanır. Burada butona bastıktan sonra gerçekleşen işlem seçim ana kontrat adresine yani seçim sandığına bir token göndermektir. Bu işlem yapılırken gerçekleşen ifade dijital olarak imzalanır ve blok zincirinde gönderilmek üzere iletim aşamasına(transcation) gönderilir.

2. YÖNTEM

Blok zincir, bloklar listesi olduğu için ve seçmenler oy gönderdiği zaman gerçekleşen işlem blok zincir ağına bir blok olarak kayıt edileceği için öncelikli olarak bir block sınıfı oluşturdum.



Şekil 2.1 Blokların Birbiri ile Olan Durumunun Gösterimi

Her blok yalnızca kendinden önceki var olan bloktaki dijital imza değerini içermez. Ancak kendi dijital imzaası bir önceki bloğun dijital imza değerinden hesaplanır. Önceki bloğun verileri değiştirilirse, önceki bloğun dijital imzası, bundan sonraki blokların tüm değerlerini etkilemek için sırayla değişecektir. Dijital imza değerlerinin hesaplanması ve karşılaştırılması bir blok zincirin bize geçersiz olup olmadığını görmemizi sağlayacaktır.

```
import java.util.Date;

public class Block {

    public String hash;
    public String previousHash;
    private String data;     //our data will be a simple message.
    private long timeStamp; //as number of milliseconds since 1/1/1970.

    //Block Constructor.
    public Block(String data,String previousHash ) {
        this.data = data;
        this.previousHash = previousHash;
        this.timeStamp = new Date().getTime();
        this.hash = calculateHash();
    }
}
```

Şekil 2.2 Blok Sınıfı

Görüldüğü gibi Şekil 2.2'de temel blok sınıfı dijital imzayı saklayacak bir string hash değişkeni içeriyor. previousHash değişkeni ise blok verilerimizi tutmak için önceki bloğun hash ve verilerini saklar.

Bloklardan gerçekleşen verileri şifrelemek için seçebilecek birçok şifreleme algoritması var. Ancak ben Bitcoin'in de kullandığı SHA256 şifreleme algoritmasını tercih ettim. Bu şifreleme algoritmasını oluşturmak için StringUtil sınıfını yazdım ve bu sınıfın içinde applySha256 methodunu oluşturdum.

```
public class StringUtil {

    public static String applySha256(String input){
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");

            byte[] hash = digest.digest(input.getBytes("UTF-8"));
            StringBuffer hexString = new StringBuffer();
            for (int i = 0; i < hash.length; i++) {
                String hex = Integer.toHexString(0xff & hash[i]);
                if(hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        }
        catch(Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Şekil 2.3 SHA256 Şifreleme Algoritması

Şekil 2.3'te applySha256 methodu veriyi bir string değişkeni olarak alır ve imzalayarak dijital bir veri olarak çıktı verir.

Hash bilgisi yani dijital imza oluşturmak için applySha256 methodunu içinde kullandığım calculateHash methodunu oluşturdum.

calculateHash methodunda StringUtil sınıfı üzerinde applySha256 methodu çağrılır. applySha256 gerçekleşen işlemi bir string ifadesi olarak alacağı için bir önceki bloğun dijital imzası(previousHash), işlemin gerçekleşme zamanı(timeStamp) ve gerçekleşen olaya verilen sabit veri yapısı(örneğin A kişi C partisine oy gönderdi) bir araya getirilerek şifreleme algoritmasına gönderilir.

```
public String calculateHash() {
    String calculatedhash = StringUtil.applySha256(
            previousHash +
            Long.toString(timeStamp) +
            data
            );
    return calculatedhash;
}
```

Şekil 2.4 calculateHash Methodu

Hash değişkeninin hesaplanan dijital imzaya eşit olduğunu ve önceki bloğun hash değerinin pervioushHash değişkenine eşit olduğunu kontrol etmek için isChainValid methodunu yazdım. Bu method Boolean tipinde olup bloğun geçerli olup olmamasına göre true veya false değeri döndürnektedir.

```
public static Boolean isChainValid() {
    Block currentBlock;
    Block previousBlock;
    for(int i=1; i < blockchain.size(); i++) {
        currentBlock = blockchain.get(i);
        previousBlock = blockchain.get(i-1);
        //compare registered hash and calculated hash:
        if(!currentBlock.hash.equals(curre ntBlock.calculateHash()) ){
            System.out.println("Current Hashes not equal");
            return false;
        }
        //compare previous hash and registered previous hash
        if(!previousBlock.hash.equals(currentBlock.previousHash) ) {
            System.out.println("Previous Hashes not equal");
            return false;
        }
    }
    return true;
}
```

Şekil 2.5 Blok Doğrulama Kontrolü

Blok sınıfıyla ilgili işlemler tamamlandıktan sonra her seçmene özgü soğuk cüzdan oluşturmak için Wallet isminde bir sınıf oluşturdum.

```
public class Wallet {
    public PrivateKey privateKey;
    public PublicKey publicKey;
    public Wallet(){
        generateKeyPair();
    }
    public void generateKeyPair() {
        try {
            KeyPairGenerator keyGen = KeyPairGenerator.getInstance("ECDSA","BC");
            SecureRandom random = SecureRandom.getInstance("SHA1PRNG");
            ECGenParameterSpec ecSpec = new ECGenParameterSpec("prime192v1");
            // Initialize the key generator and generate a KeyPair
            keyGen.initialize(ecSpec, random);   //256 bytes provides an acceptable security level
                KeyPair keyPair = keyGen.generateKeyPair();
                // Set the public and private keys from the keyPair
                privateKey = keyPair.getPrivate();
                publicKey = keyPair.getPublic();
        }catch(Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Şekil 2.6 Seçmene Özgü Soğuk Cüzdan Sınıfı

Bu aşamada cüzdanın özel anahtarı(private key) sınıf içinde oluşturulmaktadır. El okuma cihazı system ile entegre edildiği zaman bu aşama da cüzdan için sadece açık anahtar(public key) oluşturulacaktır. Bu anahtarlar oluşturulurken java da bulunan KeyPairGenerator, SecureRandom ve ECGenParameterSpec sınıfları kullanılmıştır. Oluşan keypair değişkeninde getPrivate ve getPublic methodları ile soğuk cüzdanın özel ve açık anahtarları belirlenir.

Cüzdan sınıfı oluşturulduktan sonra seçmenler oy kullandığı zaman bu işlemin şifrelenmesi, seçmenin oy ekranından sadece 1 kere oy kullanabilmesi ve seçim kontrat adresine başarıyla yola çıkması yani iletim(transaction) aşamasına %100 doğrulukta geçmesi için var olan StringUtil sınıfına applyECDSASig methodunu ekledim.

İmza oluşturma method applicationECDSASig, seçmenlerin özel anahtarını ve gerçekleştirdikleri olayın dize girişini alır. Bu bilgileri kullanarak imzalar ve bir bayt dizisi döndürür.

Bu oluşan imzanın doğruluğunun kontrol edilmesi gerekir. Bu kontrolü de verifyECDSASig methodu üstlenir. Bu method imza, ortak anahtar ve dize verilerini alır. verifyECDSASig method imzanın geçerlilik durumuna göre doğru ya da yanlış sonucunu verir.

```
//Applies ECDSA Signature and returns the result ( as bytes ).
public static byte[] applyECDSASig(PrivateKey privateKey, String input) {
    Signature dsa;
    byte[] output = new byte[0];
    try {
        dsa = Signature.getInstance("ECDSA", "BC");
        dsa.initSign(privateKey);
        byte[] strByte = input.getBytes();
        dsa.update(strByte);
        byte[] realSig = dsa.sign();
        output = realSig;
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    return output;
}
//Verifies a String signature
public static boolean verifyECDSASig(PublicKey publicKey, String data, byte[] signature) {
    try {
        Signature ecdsaVerify = Signature.getInstance("ECDSA", "BC");
        ecdsaVerify.initVerify(publicKey);
        ecdsaVerify.update(data.getBytes());
        return ecdsaVerify.verify(signature);
    }catch(Exception e) {
        throw new RuntimeException(e);
    }
}
public static String getStringFromKey(Key key) {
    return Base64.getEncoder().encodeToString(key.getEncoded());
}
```

Şekil 2.7 İmza Oluşturma

İmzalama işleminden sonra halledilmesi gereken durum gerçekleşen işlemin iletim(transcation) aşamasına %100 doğrulukta bağlanabilmesidir. Bu durumun doğruluğunun sağlanması için generateSignature ve verifiySignature methodlarını ekledim.

generateSignature methodu seçmenin cüzdan adresinin açık anahtarını, seçim sandığının kontrat adresini(oyun ulaşması gereken akıllı kontrat) ve A seçmeni B partisine oy attı verisi birleştirerek bir veri oluşturur. Bu method ise seçmenin cüzdanının özel anahtarı ve oluşan veriyi kullanarak oy kullanılmasını imzalayarak dijital bir imza döndürür.

verifiySignature methodu ise seçmenin cüzdanın açık anahtarı, generateSignature methıdundan gelen veri ve signature bilgisi alarak bu imzanın doğruluğunu kontrol eder. Eğer imza başarılı ise true sonucu döner ve oy gönderilmek üzere iletim(transaction) aşamasına bağlanır. Sonuç false ise işlem gerçekleşmez.

```
//Signs all the data we dont wish to be tampered with.
public void generateSignature(PrivateKey privateKey) {
    String data = StringUtil.getStringFromKey(sender) + StringUtil.getStringFromKey(reciepient) + Float.toString(value) ;
    signature = StringUtil.applyECDSASig(privateKey,data);
}
//Verifies the data we signed hasnt been tampered with
public boolean verifiySignature() {
    String data = StringUtil.getStringFromKey(sender) + StringUtil.getStringFromKey(reciepient) + Float.toString(value) ;
    return StringUtil.verifyECDSASig(sender, data, signature);
```

Şekil 2.8 İmza Doğrulama

İmza doğruluğu başarılı ise seçmen oy ekranında gösterilen butonların aktifliği kaldırılır ve işlevsiz hale getirilir. Böylece ikinci ve daha fazla oy kullanma durumu engellenmiş olur. Oy kullanma işlemi tamamlandıktan sonra sıradaki seçmen oy kullanacağı için cüzdandan otomatik çıkış yapılır.

Seçmen oy kullanmak için elini okuttuğu sırada WalletConnection sınıfı devreye girerek getConnection method ile veri tabanı ile iletişime geçilir.

```
public class WalletConnection {

    // create a function to connect with mysql database
    public static Connection getConnection(){

        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost/java_login_register", "root", "");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
        return con;
    }
}
```

Şekil 2.9 Seçmen Soğuk Cüzdan Girişi Veri Tabanı Bağlantısı

Veri tabanı ile iletişim sağlandıktan sonra şifrelenen el izi verisi cüzdan girişini sağladığı için checkPrivateKey methodu ile kontrol edilerek cüzdanın açılıp açılamadığını kontrol eder. Özel anahtar sistemde mevcut ise doğru değeri vererek cüzdanın açılmasını sağlar ve oy kullanma ekranı gelir.

```
public boolean checkPrivatekey(String prvKey)
{
    PreparedStatement ps;
    ResultSet rs;
    boolean enterWallet = false;
    String query = "SELECT * FROM `the_app_users` WHERE `private_key` =?";
    try {
        ps = MyConnection.getConnection().prepareStatement(query);
        ps.setString(1, prvKey);
        rs = ps.executeQuery();
        if(rs.next())
        {
            enterWallet = true;
        }
    } catch (SQLException ex) {
        Logger.getLogger(RegisterForm.class.getName()).log(Level.SEVERE, null, ex);
    }
     return enterWallet;
}
```

Şekil 2.10 El İzi Şifrelenmiş Anahtar ile Cüzdanı Aktif Etme

3. BULGULAR

Bu proje de oy kullanıldığı zaman blok zincir de blokların meydana gelip gelmediğini kontrol etmek için isChainValid methodu ile birlikte kullanarak bir test hazırlanmıştır.

```
import java.util.ArrayList;
import com.google.gson.GsonBuilder;

public class TestBlock {

    public static ArrayList<Block> blockchain = new ArrayList<Block>();
    public static int difficulty = 5;

    public static void main(String[] args) {

        blockchain.add(new Block("Hi im the first block", "0"));
        System.out.println("Trying to Mine block 1... ");
        blockchain.get(0).mineBlock(difficulty);

        blockchain.add(new Block("Yo im the second block",blockchain.get(blockchain.size()-1).hash));
        System.out.println("Trying to Mine block 2... ");
        blockchain.get(1).mineBlock(difficulty);

        blockchain.add(new Block("Hey im the third block",blockchain.get(blockchain.size()-1).hash));
        System.out.println("Trying to Mine block 3... ");
        blockchain.get(2).mineBlock(difficulty);

        System.out.println("\nBlockchain is Valid: " + isChainValid());

        String blockchainJson = new GsonBuilder().setPrettyPrinting().create().toJson(blockchain);
        System.out.println("\nThe block chain: ");
        System.out.println(blockchainJson);
    }
}
```

Şekil 3.1 Blok Oluşumu Testi

Bir blok zincirinde ilk bloğa başlangıç(genesis) blok denir. Bu nedenle Şekil 3.1'de öncelikli olarak başlangıç bloğu oluşturulmuş olup, diğer bloklar oluşturulurken de bu başlangıç bloğunun dijital imzasından faydanılmıştır. Şekil 3.2'de blokların işlenmesi sonucu oluşan dijital imzaları ve başlangıç bloğunun içerii gözükmektedir.

```
Trying to Mine block 1...
Block Mined!!! : 00000731a61c365f093fcbab8741d2ea29191c1da408dfcdd9332b568fe38cce
Trying to Mine block 2...
Block Mined!!! : 000003dae8a626c87dbadb34325a8b272a52e3ce45e4da2c2959eb81a0c8cb9a
Trying to Mine block 3...
Block Mined!!! : 000001c813a29475aed4d4e9dd1af2f7530c177f1e3f0bcf1d944cb2ec3d96e8

Blockchain is Valid: true

The block chain:
[
  {
    "hash": "00000731a61c365f093fcbab8741d2ea29191c1da408dfcdd9332b568fe38cce",
    "previousHash": "0",
    "data": "Hi im the first block",
    "timeStamp": 1513418517793,
    "nonce": 1453771
  },
  {
    "hash": "000003dae8a626c87dbadb34325a8b272a52e3ce45e4da2c2959eb81a0c8cb9a",
    "previousHash": "00000731a61c365f093fcbab8741d2ea29191c1da408dfcdd9332b568fe38cce"
```

Şekil 3.2 Oluşan Blokların İşlenmesi ve Digital İmzalarının Sonuçları

Bir sonraki aşamada seçmenler için soğuk cüzdan alt yapısı ve iletim aşamasına(transaction) iletilme metotları tamamlandığı için bunların testi yapılmıştır. Şekil 3.3'te iki farklı soğuk cüzdan oluşturulup, cüzdanlarda token miktarı belirtilerek birbirleri arasında işlem yapmaları test edilmiştir.

```
//add our blocks to the blockchain ArrayList:
Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());

//Create wallets:
walletA = new Wallet();
walletB = new Wallet();
Wallet coinbase = new Wallet();

//create genesis transaction, which sends 100 NoobCoin to walletA:
genesisTransaction = new Transaction(coinbase.publicKey, walletA.publicKey, 100f, null);
genesisTransaction.generateSignature(coinbase.privateKey);   //manually sign the genesis transaction
genesisTransaction.transactionId = "0"; //manually set the transaction id
genesisTransaction.outputs.add(new TransactionOutput(genesisTransaction.reciepient,
genesisTransaction.value, genesisTransaction.transactionId));
UTXOs.put(genesisTransaction.outputs.get(0).id, genesisTransaction.outputs.get(0));

System.out.println("Creating and Mining Genesis block... ");
Block genesis = new Block("0");
genesis.addTransaction(genesisTransaction);
addBlock(genesis);

//testing
Block block1 = new Block(genesis.hash);
System.out.println("\nWalletA's balance is: " + walletA.getBalance());
System.out.println("\nWalletA is Attempting to send funds (40) to WalletB...");
block1.addTransaction(walletA.sendFunds(walletB.publicKey, 40f));
addBlock(block1);
System.out.println("\nWalletA's balance is: " + walletA.getBalance());
System.out.println("WalletB's balance is: " + walletB.getBalance());

Block block2 = new Block(block1.hash);
System.out.println("\nWalletA Attempting to send more funds (1000) than it has...");
block2.addTransaction(walletA.sendFunds(walletB.publicKey, 1000f));
addBlock(block2);
System.out.println("\nWalletA's balance is: " + walletA.getBalance());
System.out.println("WalletB's balance is: " + walletB.getBalance());

Block block3 = new Block(block2.hash);
System.out.println("\nWalletB is Attempting to send funds (20) to WalletA...");
block3.addTransaction(walletB.sendFunds( walletA.publicKey, 20));
System.out.println("\nWalletA's balance is: " + walletA.getBalance());
System.out.println("WalletB's balance is: " + walletB.getBalance());

isChainValid();
```

Şekil 3.3 Akıllı Kontratlar Cüzdanlar Arasında Token İletimi

Şekil 3.4'te cüzdanlar yani akıllı sözleşmeler arasında iletim yapılırken oluşan blokların işlenme sonucu oluşan dijital imza değerleri ve aktarım işlemlerinin doğruluğunun kontrolü için cüzdan için token miktarları gösterilmiştir.

```
Creating and Mining Genesis block...
Transaction Successfully added to Block
Block Mined!!! : 000107236b351965f2734a3fb20053902ff4389d8a978bdc9e935d55e5966b36

WalletA's balance is: 100.0

WalletA is Attempting to send funds (40) to WalletB...
Transaction Successfully added to Block
Block Mined!!! : 00043602c474fa655ee99b4126c7828742cc2f3e66fa8b90184d1bc002f43456

WalletA's balance is: 60.0
WalletB's balance is: 40.0

WalletA Attempting to send more funds (1000) than it has...
#Not Enough funds to send transaction. Transaction Discarded.
Block Mined!!! : 000cb374154e23cf00f899818c37efae5592a761331be002b41c2b0c64f21b4a

WalletA's balance is: 60.0
WalletB's balance is: 40.0

WalletB is Attempting to send funds (20) to WalletA...
Transaction Successfully added to Block

WalletA's balance is: 80.0
WalletB's balance is: 20.0
Blockchain is valid
```

Şekil 3.4 Akıllı Kontratlar Arası Token İletimi Test Sonuçları

4. TARTIŞMA VE SONUÇ

Son dönemlerde üzerine en fazla konuşulan konuların başında blok zincir teknolojisi ve ona bağlı olarak gelişen kripto para birimleri geliyor. Geliştirilen bu karmaşık şifreleme sistemi sayesinde, veri saklama ve aktarımının güvenli bir mekanizma ile gerçekleştirilmesinin finans piyasalarında oluşturduğu etki kayda değer bir noktaya ulaştı. Bu teknolojinin, ekonomik sahada meydana getirdiği değişim, ekonomi dışı sahalar için de konuşulmaya başlamış durumda. Akla ilk gelen alanların başında, korunaklı bir veri seyahati ile gerçekleşecek oy verme mekanizması geliyor.

Dünya genelinde vatandaşların seçimlere gösterdiği ilgi her geçen yıl azalıyor. Sandıkta kullanacağı bir oyun, seçim sonuçlarına etki etmeyeceğini düşünen bir çok seçmen, sandığa gidip oy vermenin maliyetinin, oy kullanma sonucu oluşacak siyasi atmosferden alacağı kazanca göre daha yüksek görmektedir. Bu noktada, sandığa gidip oy kullanma sürecini kısaltan bir oy verme mekanizmasının, oy kullanma davranışını etkileyeceği düşünülebilir.

Blok zincir teknolojisi değiştirilemez, saldırıya uğrayamaz bir şekilde verileri güvenli sakladığı için bireyin verdiği oyu güvenli bir şekilde kaydedip, seçimde meydana gelebilecek hile, sahte oy ve eksik oy gibi durumların önüne geçebilir.

Bu projede seçimlerdeki güvenirliliğin arttırılması için her seçmen için bir soğuk cüzdan oluşturulması hedeflenmiş olup seçimdeki her seçenek soğuk cüzdan da akıllı bir sözleşme yani token olarak konumlandırılmıştır. Her soğuk cüzdan kişiye özel olup eşsizdir. Kişinin parmak ve avuç izini okutarak cüzdana erişim sağlaması ile son derece güvenli olduğu görülmüştür. Yapılan testler sonucunda oy kullanma yani token gönderimleri başarıyla test edilmiştir.

Seçmenin verdiği oyu, güvenilir bir algoritma ile sisteme aktarabilecek bir yapı aynı zamanda temsili demokrasi ile doğrudan demokrasi arasındaki aşılmaz olarak görülen uçurumu da bir ölçüde kapatmaya yardımcı olabilir. Böylelikle önümüzdeki yıllarda, kripto paralar ile hayatımıza giren blok zincir teknolojisinin seçim sistemi başta olmak üzere hayatın başka alanlarına nüfuz etmesine şahitlik edeceğiz.

KAYNAKLAR

BlockChain Asset

[1] https://www.nasdaq.com/article/blockchain-is-about-to-change-how-asset-ownership-works-cm1149350

Crowd Funding Blockchain

[2]https://www.nasdaq.com/article/a-new-era-of-crowdfunding-blockchain-cm1138237

How Blockchains Could Change The World

[3]https://www.mckinsey.com/industries/high-tech/our-insights/how-blockchains-could-change-the-world

Wallets Addresses

[4]https://support.blockchain.com/hc/en-us/articles/207746403-Wallets-Addresses

Blockchain Wallet

[5]https://blog.softwaremill.com/what-is-a-blockchain-wallet-bbb30fbf97f8

Cold Wallet For Storing

[6]https://www.investinblockchain.com/top-cold-wallets-for-storing-cryptocurrencies/

Cold Wallet-Hot Wallet

[7]https://blog.liquid.com/hot-wallet-vs-cold-wallet-how-should-you-store-crypto

Blockchain and Cyber

[8]https://www2.deloitte.com/tr/en/pages/technology-media-and-telecommunications/articles/blockchain-and-cyber.html

Blockchain Transaction

[9]https://coincentral.com/what-is-a-blockchain-transaction-anyway/

Blockchain Transaction Processing

[10]https://www.researchgate.net/publication/325116198_Blockchain_Transaction_Processing

Blockchain Confirmations

[11]https://coincentral.com/blockchain-confirmations/