İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

BLOK ZİNCİRİ TABANLI ELEKTRONİK SEÇİM SİSTEMİ TASARIMI VE KISMİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Bilal GÜLTEKİN

Bilişim Uygulamaları Anabilim Dalı

Bilişim Uygulamaları Programı

NİSAN 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

BLOK ZİNCİRİ TABANLI ELEKTRONİK SEÇİM SİSTEMİ TASARIMI VE KISMİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Bilal GÜLTEKİN
(708151024)

Bilişim Uygulamaları Anabilim Dalı

Bilişim Uygulamaları Programı

Tez Danışmanı: Prof. Dr. Ertuğrul KARAÇUHA

NİSAN 2019

İTÜ, Bilişim Enstitüsü'nün 708151024 numaralı Yüksek Lisans Öğrencisi Bilal GÜLTEKİN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "BLOK ZİNCİRİ TABANLI ELEKTRONİK SEÇİM SİSTEMİ TASARIMI VE KISMİ UYGULAMASI" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Ertuğrul KARAÇUHA** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. M. Oğuzhan KÜLEKCİ** İstanbul Teknik Üniversitesi	.....
	Prof. Dr. Hülya ŞAHİNTÜRK Yıldız Teknik Üniversitesi	.....

Teslim Tarihi : **11 Nisan 2019**
Savunma Tarihi : **12 Haziran 2019**

Quis custodiet ipsos custodes?

ÖNSÖZ

Yüksek lisans öğrenimim boyunca ve tez sürecinde desteklerini ve yardımlarını eksik etmeyen başta tez danışmanım Prof. Dr. Ertuğrul Karaçuha olmak üzere tüm İstanbul Teknik Üniversite öğretim görevlisi ve çalışanlarına teşekkürlerimi sunarım. Öğrenim hayatım boyunca, tüm öğretmenlerime emekleri, bu süreçte her daim yanımda olan sevgili arkadaşlarıma destekleri ve katkıları için teşekkür ederim. Hayatım boyunca, maddi manevi desteklerini eksik etmeyen başta annem ve babam olmak üzere tüm aile bireylerime teşekkürü bir borç bilirim.

Nisan 2019

Bilal GÜLTEKİN
Kıdemli Yazılım Geliştirici

İÇİNDEKİLER

Sayfa

KISALTMALAR

ABD : Amerika Birleşik Devletleri
DDOS : Distributed DOS (Dağıtık DOS)
DKE : Doğrudan Kayıt Eden
DOS : Denial of Service (Hizmet Engelleme)
GB : Gigabyte
MB : Megabyte
NSA : National Security Agency (Ulusal Güvenlik Ajansı)
OVM : Oy Verme Makinesi
PTB : Physikalisch Technische Bundesanstalt (Ulusal Standart Enstitüsü)
STK : Sivil Toplum Kuruluşu

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

BLOK ZİNCİRİ TABANLI ELEKTRONİK SEÇİM SİSTEMİ TASARIMI VE KISMİ UYGULAMASI

ÖZET

Elektronik seçim sistemleri, elektronik sistemlerdeki gelişmenin ve insan hayatındaki dijitalleşmenin etkisiyle, ülkeler tarafından tercih edilmeye başlanmıştır. Ancak güvenlik ve şeffaflık endişeleri, tartışmalara ve hatta bazı ülkelerde geri adım atılmasına sebep olmuştur.

Bu çalışmada, şeffaf, denetlenebilir ve doğrulanabilir bir seçim sistemi tasarlanmaya çalışılmıştır. Seçim kurumları yerine seçmenlerin merkezi konumda olduğu, seçim kurumlarının siyasi partiler ile birlikte gözlemci ve doğrulayıcı bir rol oynadığı bir süreç oluşturulmuştur.

Tasarımda, finans alanında kendini ispatlamış ve bir çok farklı alandaki uygulamalarda kullanılmaya başlanmış blok zinciri veritabanı tercih edilmiştir. Dağıtık ve şeffaf yapısı sayesinde, tüm bilginin herkes tarafından okunabilmesi ve doğrulanabilmesi sağlanmaya çalışılmıştır. Temel tasarımda, seçmenlerin elektronik kimlik kartlarıyla kimlik doğrulaması yapabildiği ve her seçmenin açık/gizli anahtar çifti olduğu varsayılmıştır. Farklı kimlik doğrulama yöntemleri ve anahtar çiftine sahip olmama durumu için sistem üzerinde yapılabilecek değişiklikler önerilmiştir. Süreç, baskı ve yasadışı teşvik ile oy kullanımını engellemek adına, seçmenlerin oylarını belirli merkezlerde bulunan oy verme makineleri (OVM) üzerinden kullanacakları ve seçmenlere kullandıkları oy ile ilgili herhangi bir belge verilmeyecek şekilde tasarlanmıştır.

Tasarlanan sistem, seçim öncesi bir takım hazırlıklar yapılmasını gerektirmektedir. Hazırlık süresince, seçmen ve seçim listeleri hazırlanır, seçmenlerin açık anahtarları toparlanır. Madencilik faaliyeti yaparak, seçim sürecine ve sistemin güvenilirliğine katkıda bulunmak isteyen kişi ve kurumların bir portal üzerinden açık anahtarlarını kaydetmeleri istenir. Aynı zamanda OVM'lerin açık anahtarları da blok zincirine eklenmek üzere toparlanır.

Yapılan hazırlıklar sonrasında, parti temsilcilerinin ve bilişim uzmanlarının gözlemci ve denetleyici olarak katılım gösterdikleri kök blok üretim süreci başlar. Süreç başlangıcında, seçim için açık/gizli anahtar çifti üretilir. Sonrasında, her seçmenin kullanacağı oy ile ilgili, elektronik oy pusulaları oluşturulur. Elektronik oy pusulası oluşturulduktan sonra özeti alınır ve bu özet elektronik oy pusulası kanıtı olarak saklanır. Aynı pusula, ilgili seçmenin açık anahtarıyla da şifrelenir. Elektronik oy pusulası işlendikten sonra, seçmen ile oyu arasında ilişki kurulmasını önlemek adına, silindiğinden emin olunmalıdır. Bu veriler, hazırlık sürecinde toparlanan açık anahtarlarla birlikte, kök blok içerisine, herkesin erişimine açık bir şekilde kaydedilir. Kök blok seçimin gizli anahtarı ile şifrelendikten sonra, gizli anahtar şifreleme yöntemleriyle parçalanarak siyasi partiler arasında dağıtılır. Açık anahtar ise OVM'lere girilmek ve madenciler tarafından kullanılmak üzere yayınlanır.

Seçim süresince, seçmenler OVM'lere kimliklerini PIN, biyometrik yöntemler vb. ile ispatlayarak, şifreli pusulayı deşifre eder ve oylarını kullanırlar. Verilen oylar, seçim bitmeden önce oy sayımını engellemek için, seçimin açık anahtarıyla şifrelenir. Blok zinciri ağına gönderilen verinin içine, deşifre edilen pusula içeriği de kanıt olarak

eklenir. OVM'ler kullanılan her oyu, şeffaf bir hazne içinde çıktı alarak seçmene onaylatır ve çıktıyı gizli haznede saklar.

Madenciler, blok zinciri ağına gönderilen oyların, geçerli bir OVM tarafından gönderildiğini, oy içerisindeki elektronik oy pusulasının özetinin, kanıt listesinde olduğunu doğrularlar. Önceki blokların da kayıtlı madenciler tarafından oluşturulup oluşturulmadığını kontrol ederler.

Süreç sonunda, siyasi partilerin kendilerinde bulunan parçaları birleştirmeleriyle seçimin gizli anahtarı tekrar oluşturulur. Blok zinciri veritabanında biriken oylar bu anahtarla deşifre edilerek sayılır. Sayım işlemi müstakil olarak yapılabileceği gibi blok zinciri ağında madenciler tarafından da yapılabilir. Sayım işleminden sonra elektronik yapıya müdahale olup olmadığını doğrulamak için, OVM'lerin her oy için aldıkları çıktılar, rastgele OVM belirlenerek blok zincirindeki sonuçlar ile karşılaştırılır.

Sistem tasarımı sunulmadan önce, dünyadaki elektronik seçim uygulamalarından, sistemde kullanılan şifreleme yöntemlerinden ve blok zincirinden bahsedilmiş daha sonra da günümüz şartları çerçevesinde tasarım ilkeleri belirlenmiştir. Tasarlanan sistem, kısmi olarak gerçeklenmiş ve çalışmaya eklenmiştir.

BLOCKCHAIN BASED ELECTRONIC VOTING SYSTEM DESIGN AND PARTIAL APPLICATION

SUMMARY

Electronic voting systems have been started to be tried by countries with the development of electronic systems and digitalization in human life. However, concerns about security and transparency have led to discussions and even step back in some countries.

In this thesis, a transparent, auditable and verifiable electronic voting system is designed and presented. In this system, voters have the responsibilities of observation and validation of the system instead of election offices. People and institutions can contribute to the system and validate votes by mining activity during the election.

In the design, there are preparation, genesis block generation, election, tallying and verifying stages. In the preparation stage, all necessary data is collected. In genesis block generation stage, this data is processed and first, genesis block of the blockchain is generated. Genesis block contains all the information which is needed for the whole process. In the election stage, votes can be cast only through voting machines which are placed in polling stations and verified by the miners. After the election is done, tallying and verifying stage is performed.

Blockchain database, which has proven itself in the field of finance and has been used in many different fields, is preferred in the design. With its transparent and distributed structure, all information of an election can be read and verified by everyone. For an electronic voting system, the blockchain database should be private to let only eligible voters vote. Since elections may determine countries' destiny, it may get the attention of malicious parties which have high computing power. That's why mining activity is kept also private and let only registered miners do. Additionally, the consensus method is derived from proof of stake to consume less energy and avoid centralization in the mining process.

In the primary design, it is assumed that voters can prove their identities with an electronic ID card and have a public/private key pair. Some modifications on primary design are proposed for cases which may have different ID verification methods and lack of key pair. In order to prevent coercion, votes can be cast only through voting machines which are placed in polling stations. Additionally, no receipt is given by the machines to voters for voting.

In the preparation stage, a list of eligible voters is published and checked by political parties. Public keys of eligible voters are collected to limit voting activity on the blockchain network. In order to show candidates and limit voters for voting only they are authorized, candidate lists are prepared as a digital document. People and institutions willing to do mining activity register their public keys through a web portal and these keys are collected. In this way, mining activity is limited among the known participants. Also, public keys of voting machines are collected.

Genesis block generation stage is operated after the preparation stage. In this stage, all collected data is processed, electronic ballots and proofs are generated for every voter. This stage should be observed and audited by political party representatives and IT professionals. Auditing is crucial for providing anonymity of voters.

At the beginning of the genesis block generation stage, a public/private key pair for the election is generated. Electronic ballots are generated which involve random ballot ID, sub election ID for every voter. Hashes of ballots are saved as proof of ballot. Ballots are also encrypted with voter's public key. Both encrypted ballots and hashes of ballots are saved into the genesis block with the collected public keys of voting machines and miners. Generated electronic ballots are deleted after this process.

At the end of the generation stage, the public key of election is published to be entered in voting machines and used by miners. After the private key of election is used for signing the genesis block, it is distributed between political parties by using threshold cryptosystem so that no one even the election office can decrypt used votes and know the intermediate results before the end of the election.

During the election, voters prove their identities to voting machines by using their electronic ID card with some extra steps such as entering a PIN, using biometric data etc. After the authentication process, voting machines decrypt their encrypted ballots and ask for casting it. When voters cast a vote, voting machines print the choice onto a paper inside a closed transparent part and shows to the voter to get approval. After approval, voting machines move printed paper to closed hidden part. Once the voting process is done, voting machines encrypt just choice of voters with the public key of the election and send this information with the content of encrypted ballots to blockchain network after signing. Printed papers help to check if there is any intervention to the election after tallying is done.

Once a vote is cast and sent to blockchain network, it is verified by miners. Miners check if it is signed by authorized voting machine by checking the existence of the public key among the public keys of voting machines and if the hash of the content of decrypted electronic ballot exists among the ballot proofs. Miners apply same verification steps to previous blocks with an extra step to check if blocks are generated and signed by authorized miners.

After the election, the private key of the election is generated by gathering pieces from political parties and is used to decrypt choices in used electronic ballots. Since the tallying process may take a long time for elections have many voters, using the blockchain network is proposed as an option. After tallying, a number of voting machines are selected randomly and their printed votes are compared with their sent votes to the blockchain database in order to check if there is any intervention or failure in the electronic voting system.

It will be appropriate to develop all software and hardware used in the system as open source to provide more transparency. It is known there are some concerns about open source in terms of security and business. However, there are plenty of works show that these concerns are unfounded.

The designed system can be used without voting machines with personal devices by just removing voting machines from the design and miners' checklist. Since proofs are exposed after first voting, repetitive voting should not be let and just first one should be tallied. Additionally, to prevent Sybil attacks, received votes can be broadcasted with timestamp and miners' sign. A transaction ID can be given to voters to let them check if their votes are tallied at the final results. Removing voting machines is not preferred and kept out of scope in the primary design, because this may lead to coercion in some countries in the world.

The size of the blockchain database may cause some problems for mining activity and the application of the design itself. When calculated roughly, the final size is 75 GB for 50,000,000 voters. The size can be decreased by reducing the sizes of keys and output of hashes. Putting sub election IDs inside electronic ballot lets voters cast their votes wherever they want. However, this makes detecting repetitive votes hard at the voting machine level. That's why it should be done at the blockchain network level by miners while producing new blocks or at the tallying stage.

Within the scope of the study, a partial application of the system is developed using Python language. Since it was not possible to create an application for the whole system in this study, blockchain network and vote casting machines are represented by simple classes.

1. GİRİŞ

Demokrasinin temelini oluşturan seçme işlemi, süreç itibariyle demokrasinin uygulanmaya başlandığı tarihten itibaren, gelişen teknoloji ve imkanlar dahilinde evrimleşmiştir. Atina'da taştan oy pusulaları ile, Sparta'da seçmenlerin oylarını bağırmasıyla uygulanan seçim süreci; gelişen teknoloji ile geleneksel seçim sistemi diye adlandırılan oy pusulası, zarf, sandık gibi malzemelerin kullanıldığı bir forma dönmüştür [1]. Bu dönüşümü, sadece teknoloji değil aynı zamanda ihtiyaçlar da tetiklemiştir. İlk dönemlerde şehir devleti ölçeğinde uygulanan demokrasi, günümüzde sivil kurumların, ülkelerin, birliklerin yönetim biçimi olarak uygulanmaktadır.

Son zamanlarda bir çok ülke, seçim sürecini insan hayatında yaşanan dijitalleşmenin de etkisiyle bir adım öteye taşımış ve elektronik seçim sistemlerini kullanmaya başlamıştır [2]. Bu seçim sistemleriyle ilgili en önemli sorun; oy kullanım ve sayım süreciyle ilgili şeffaflık ve güvenlik endişeleridir. Bu çalışmada, blok zinciri temelli bir elektronik seçim sistemi tasarlanarak, bu endişeler şeffaf ve güvenli bir tasarım ile giderilmeye çalışılmıştır. Tasarım belirli ilkeler çerçevesinde şekillendirilmiş, seçmenlerin geleneksel sistemlerde olduğu gibi oy sayımı, oyların güvenliğinin sağlanması gibi konularda cihazlarıyla rol almasının sağlanması amaçlanmıştır.

1.1 Konu ve Önemi

Seçim sistemleri, toplumların hayatlarında önemli bir rol oynamaktadır. Toplumlar; şirket içi kararlar alınması, devlet başkanı seçilmesi, anayasa oylaması vb. bir çok toplu karar verme sürecinde, seçim sistemlerini kullanmak zorunda kalmaktadır. Özellikle, büyük ölçekli seçimlerde, sistemlerin ve sistem kurgularının önemi bir hayli artmaktadır.

Seçim sisteminde, oy verenler, sistemin güvenliğinden ve doğru bir şekilde işlediğinden emin olmaları gerekmektedir. Geleneksel seçim sistemlerinde bu durumu, siyasi parti temsilcileri ve gönüllü seçmenler, sandık başlarında görev alarak sağlamaya çalışmaktadır. Ancak çoğu zaman yeterli olmamakta ve süreç ile ilgili sıkıntılar yaşanmaktadır [3].

İnsan hayatındaki dijitalleşmenin etkisiyle bir çok ülke, elektronik seçim sistemlerine geçmiş, hatta bazı ülkeler internet üzerinden oy kullanılmasına müsaade etmeye başlamıştır. Geçiş yapılan ülkelerde, elektronik seçim sisteminin ne kadar güvenli olduğu ile ilgili tartışmalar sürerken [4], [5], bu tartışmalar Almanya, Belçika, İrlanda gibi ülkelerde geri adım atılmasına sebep olmuştur [6]-[8].

Elektronik seçim sistemleriyle ilgili endişeler çoğunlukla güvenlik temelli endişeler olmaktadır. Seçmenler, elektronik seçim sistemlerini saldırıya, manipülasyona açık olarak görmektedir. Oysa ki, son zamanlarda bankacılık, elektronik devlet sistemleri gibi güvenliğin kritik olduğu sistemlerde kullanılan şifreleme yöntemlerini temel alarak, güvenli bir elektronik seçim sisteminin tasarlanması mümkün gözükmektedir.

Şeffaf, güvenlik sorunları bulunmayan bir elektronik sistemin tasarımı; seçim hilelerinin önüne geçecek, günümüz seçim süreçlerini hızlandıracak, geleneksel seçim süreçlerindeki masrafları azaltacaktır. Gelecekte daha hızlı karar alma mekanizmalarının kurulmasının, katılımcı demokrasinin yaygınlaşması ve etkinleşmesi bakımından her geçen gün önemi artacaktır.

1.2 Çalışmanın Amacı ve İçeriği

Çalışmanın amacı, varolan geleneksel ve elektronik seçim sistemlerinin sorunlarını göz önünde bulundurarak, bunlara çözüm olabilecek bir elektronik seçim sistemi tasarlamaktır.

Günümüz geleneksel ve elektronik seçim sistemlerinde, seçim ile ilgilenen bağımsız devlet kurumları bulunmaktadır. Bu kurumlar, seçim sürecinde merkezi konumda seçmenler gözetiminde; seçimi uygulamakta, oyları saymakta ve aynı zamanda süreci denetlemektedirler. Sistem tasarlanırken, bu kurumlar, merkezi konumdan uzaklaştırılıp daha çok gözlemleyici ve doğrulayıcı bir konuma oturtulmaya çalışılmıştır. Seçmenler ise, cihazlarıyla, seçimin ve oyların güvenliğini sağlayan, seçim bittiğinde oyları sayan, daha merkezi bir konuma getirilmeye çalışılmıştır.

Ayrıca, meclise ve milletvekillerine yönelik ihtiyacın, seçmenlerin oylamalara doğrudan katılımı sağlanarak, azaltılmasına veya kaldırılmasına yönelik araştırmaların önünü açmak amaçlanmıştır.

Bu sistemin tasarlanması için, çağdaş şifreleme yöntemlerinden ve kendini elektronik para alanında ispatlamış bir dağıtık veritabanı olan blok zincirinden faydalanılmıştır. Sistem, günümüzde uygulanmaya çalışılan ilkeler etrafında şekillendirilmeye çalışılmıştır.

2. ELEKTRONİK SEÇİM SİSTEMLERİ

Elektronik seçim, elektronik sistemlerin seçim sistemine, tamamen veya kısmen, müdahil olmasıyla oluşan seçim sistemidir. Elektronik sistem, tüm seçim sistemini devralabileceği gibi sadece sayma veya sadece oylama kısımlarında da etkili olabilmektedir.

Elektronik seçim sistemlerinin, seçim sürecini hızlandırması ve kolaylaştırması, seçim masraflarını düşürmesi gibi faydaları bulunmaktadır. Ancak elektronik seçimin hileye ve tahrife açık olduğu düşüncesi seçmenleri kaygılandırmakta ve elektronik seçim sistemlerinin tercih edilirliğini azaltmaktadır. Bu kaygıları, elektronik sistemlerle ilgili son zamanlarda yaşanan ve geniş yankı uyandıran güvenlik zayıflıkları kuvvetlendirmektedir.

- Ticari elektronik seçim sistemlerinin uygulanması sırasında görülen, sistemlerin şifresiz, varsayılan şifreler veya kırılması kolay şifreler ile kullanımı [9]
- Openssl ve Linux gibi güvenilerek yaygın bir şekilde kullanılan ancak sonradan heartbleed, shellshock gibi önemli zayıflıklar tespit edilen sistemler [10],[11]
- Sistemin tasarım zincirinde, güvenilerek kullanılan yazılımlarda ortaya çıkabilecek zero-day açıklıkları (yeni tespit edilmiş ve henüz düzeltilmemiş açıklıklar)

Elektronik seçim sistemleri, güvenlik endişelerine rağmen, Estonya, ABD, İsviçre ülkelerde aktif olarak kullanılmaktadır. Bu endişeler, Almanya, Belçika, İrlanda gibi bir takım ülkelerde ise geri adım atılmasına ve geleneksel seçim sistemine dönülmesine sebep olmuştur [6]-[8].

2.1 Elektronik Seçim Sistemi Türleri

Günümüzde elektronik seçim sistemleri, sistemin tasarlanış ve uygulanış biçimlerine göre 3 türe ayrılabilmektedir.

2.1.1 Kağıt temelli sistemler

Bu tür sistemlerde, elektronik yapı, geleneksel oylama sistemi işleyişinin çevresine konumlanmıştır. Kağıt pusula, sayısal kağıt (digital paper), optik kağıt gibi pusulalar içeren yapıda, elektronik sistem; oy işaretlemek, oy saymak veya iki görev için de kullanılabilir. Elektronik sistemin müdahil olmadığı süreçlerde; oy elle işaretlenebilir veya elle sayılabilir.

2.1.2 Doğrudan kayıt eden (DKE) sistemler

Doğrudan kayıt eden sistemlerde; oylar düğme, tuş gibi mekanik ara yüzler veya ekran dokunmatik ekran gibi elektro optik ara yüzler yardımıyla elektronik depolama ünitelerine kaydedilir.

2.1.2.1 Yerel DKE sistemler

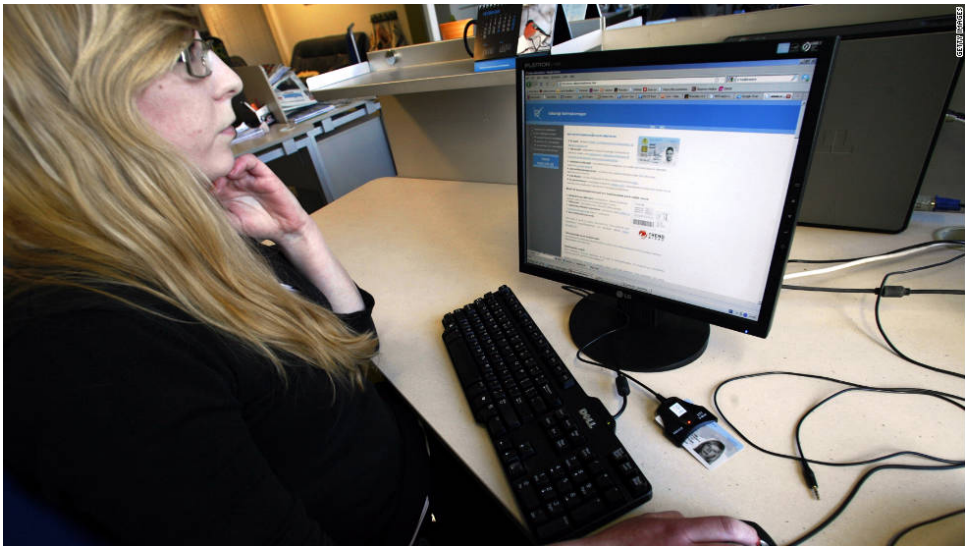
Yerel DKE sistemlerde, seçim sonrası oylar otomatik olarak sayılır ve seçim sonuçları çıkarılabilir depolama ünitesinde ve/veya kağıda basılmış şekilde verilir. Bu sistemde sayım seçim bölgesinde yapılır. Brezilya, Hindistan vb. ülkelerde DKE tipi elektronik seçim sistemleri kullanılmaktadır. Şekil 2.1'de Brezilya'da kullanılan oy verme aygıtı görülmektedir.

Şekil 2.1 : Brezilya'da kullanılan oy verme aygıtı [12].

2.1.2.2 Ağ tabanlı DKE sistemler

Ağ tabanlı DKE sistemlerde, kullanılan oy ağ (internet, telefon) üzerinden merkeze iletilir ve burada depolanır. Oy, kullanıldığı anda iletilebileceği gibi; periyodik olarak veya oylama sonunda toplu olarak da merkeze iletilebilir. Bu sistemde sayım işlemi, seçim bölgesinde veya merkezde yapılabilir.

Aynı zamanda i-seçim (i-voting) diye adlandırılan, internet üzerinden, seçim merkezlerine gidilmeden yapılan seçimler de bu sistem türü içerisinde değerlendirilebilir. Şekil 2.2'de Estonya seçimlerinde bir vatandaşın internet üzerinden oyunu kullandığı görülebilmektedir.



Şekil 2.2 : Estonya'da internet üzerinden oy verilmesi [13].

Ağ tabanlı DKE sistemleri; ABD, İsviçre, Estonya gibi gelişmiş ülkelerde kullanılmaktadır.

2.2 Elektronik Seçim Sistemi Faydaları

2.2.1 Maliyet azaltma

Geleneksel sistem; oy pusulalarının basımı, zarfların alınması, oy gereçlerinin dağıtımı, sürecin gerektirdiği fazla istihdam gibi sebeplerle göz ardı edilemeyecek bir maliyet oluşturmaktadır.

Geleneksel sistemin seçim başı maliyeti, çoğu örnekte elektronik sisteme geçiş maliyetinin altında kalmaktadır. Ancak oy verme makinelerinin tek bir sefer alınması ve uzun süre kullanılabileceği düşünüldüğünde, uzun vadede maliyeti azaltacağı beklenmektedir. Aynı zamanda elektronik seçim sisteminin bir sonraki adımı olarak kabul edebileceğimiz internet üzerinden seçim sistemi; istihdam, oy verme aygıtı gibi bir çok masraf kalemini ortadan kaldıracağı göz önünde bulundurulmalıdır.

Brezilya'da kullanılan oy verme aygıtlarının maliyeti 1.000$ civarındadır. Hindistan'da kullanılan aygıtların maliyeti ise 200$ civarındadır [14]. Bu ülkelerin, elektronik sisteme geçişlerini, elektronik üretimin günümüz kadar ucuz olmadığı bir dönemde gerçekleştirdikleri de göz önünde bulundurulmalıdır.

Sadece sayım sürecinde elektronik aygıt olarak optik okuyucular kullanılan Birleşik Krallık'ta, 47.932.500 kayıtlı seçmenin bulunduğu 2017 Genel Seçimleri'nin maliyeti 140.000.000£ olmuştur [15], [16]. Sadece seçmenlerin ve sandıkların belirlenmesi sürecinde elektronik yazılım kullanan Türkiye'de ise 55.692.841 kayıtlı seçmenin ve 166.657 adet sandığın bulunduğu 2014 Cumhurbaşkanlığı Seçim'in maliyeti 400,000,000₺ olmuştur [17], [18]. Elektronik seçimin süreci hızlandıracağı ve sandık başına oy kullanabilecek seçmen sayısını arttıracağı da göz önünde bulundurulmalıdır.

Internetten seçimi, bir oy verme opsiyonu olarak seçim sürecine dahil etmiş Estonya'da, geleneksel sistem'in oy başı masrafı 2017 Yerel Seçim'leri için 4,37€ ile 20,41€ arasında değişirken, internet üzerinden seçim'in oy başı masrafı 2,32€ olmuştur [19].

2.2.2 Geçersiz oy oranını azaltma

Kağıt oy pusulalarında veya kağıt bazlı elektronik seçimlerde yaşanılan geçersiz oy sorunu, özellikle DKE tipi elektronik seçim sistemlerinde, sistemin bir bütün olması sebebiyle yaşanmamaktadır. Bu tarz sistemlerde, oy gönderilmeden önce seçmenin son kez onayına başvurulmakta ve olası hataların önüne geçilmektedir. Geleneksel seçim sistemlerinde ise mükerrer oy, toplu oy kullanımı gibi seçim hilelerinin kısmen engelleyebilmek adına, herhangi bir hata durumunda seçmene yeni bir oy pusulası verilmemektedir.

Bu konuda MIT Üniversitesi'nden Prof. Charles Stewart, Amerika Birleşik Devletleri'nde, elektronik seçim sistemleri sayesinde 2004 yılında, 2000 yılına nazaran, kağıt temelli sistemlerin gözden kaçırabileceği 1 milyon oyun sayıldığını tahmin ettiğini dile getirmiştir [20].

2.2.3 Seçim sonuçlarının doğruluğunu arttırma

İyi kurgulanmış elektronik seçim sistemleri, seçim sonuçları üzerine duyulması muhtemel şüpheleri ortadan kaldırabilmekte ve geleneksel seçim yöntemlerinde uygulanması muhtemel seçim hilelerinin önüne geçebilmektedir.

Kağıt temelli geleneksel seçim sistemlerinde, hangi oyların geçerli sayılıp, hangilerinin sayılmayacağı ile ilgili sorunlar gündeme gelebilmekte ve bu 2000 yılı Amerika Birleşik Devletleri Başkanlık seçimlerinde olduğu gibi seçim sonuçlarını etkileyebilmektedir [21].

2.2.4 Sonuçların belirlenme hızını arttırma

Elektronik seçim sistemlerinin, geleneksel seçim sistemlerine karşın en büyük faydalarından biri seçim sonuçlarının hızlı ve insan hatasından arınmış bir şekilde belirlenmesini sağlamasıdır.

2.2.5 Seçim hilelerinin önüne geçme

Elektronik seçim sistemleriyle; mükerrer oy, başkasının yerine oy kullanma, toplu oy kullanımı, sandık çalınması gibi seçim hilelerinin önüne kolay bir şekilde geçilebilmektedir. Özellikle gönüllü seçmenler ve diğer parti temsilcileri tarafından iyi gözetlenmeyen sandıklarda; oyların sayımı, kullanımı sırasında ortaya çıkabilecek usülsüzlükler elektronik sistemlerle elimine edilebilir.

2.2.6 Engelli seçmenler için erişilebilirlik

Elektronik seçim sistemleri, yapısı itibariyle engelli insanlar için erişilebilirlik seçenekleri sunmaya daha müsaittir. Engelli seçmenler herhangi bir yardımcıya ihtiyaç duymadan, oylarını ses ve görüntü arayüzleriyle kullanabilirler. Uzaktan elektronik seçim yöntemleri ile seçim merkezlerine gidilmesine bile gerek kalmamaktadır.

Engelli seçmenler için özel seçenekler sunan elektronik seçim sistemi ABD'nin bazı bölgelerinde kullanılmaktadır. Bu sistemler; oy pusulularını sesli olarak okuma, kişisel yardımcı cihazlar için evrensel soket, büyük font gibi seçenekler sunmaktadır [22].

2.3 Bilinen Sorunlar

2.3.1 Teknik hata

Oluşturulan elektronik seçim sisteminin, istem dışı bir şekilde hata vermesi mümkündür. Yazılım ve donanım temelli hatalar neticesinde; oy kullanılamaması, oyların yanlış kaydedilmesi, oyların yanlış sayılması, yetkisiz oy kullanımı gibi sorunlar ortaya çıkabilmektedir.

2.3.2 Hatalı kullanım

Elektronik seçim sisteminin, kuruluşundan ve kullanımından sorumlu görevlilerin süreç ve ekipmanlar hakkında yeterince eğitilmemeleri sebebiyle; oyların karışması, önceki verilerin silinmemesi sebebiyle sayımın fazla çıkması, oylar tutanağa geçirilmeden önce silinmesi gibi sorunlar ortaya çıkabilmektedir.

2.3.3 Sahtekarlık

Elektronik seçim sisteminin geliştirilmesi sırasında veya sonrasında, sonuçları değiştirmeye yönelik sahtekarlık girişimleri olabilir. Sistem geliştirilmesi sırasında, kötü niyetli geliştiriciler tarafından, oyların farklı gösterilip, farklı kaydedilmesini yahut sayımında sorun oluşmasını sağlayacak geliştirmeler yapılabilir. Sonrasında ise sistemin açıklıklarından istifade edilerek, oylamanın sonucunu değiştirmeye yönelik girişimlerde bulunulabilir.

Elektronik seçim sistemlerinde yapılabilecek sahtekarlıklar, birçok elektronik sistemi hedef alan yaygın yöntemler olabileceği gibi seçim özelinde düşünülmüş yöntemler de olabilmektedir.

2.3.3.1 Elektronik bileşenlere kötü amaçlı yazılım yüklenmesi

Sistemlerde kullanılan; mini-pc, mikroişlemci, yazıcı, depolama birimi vb. elektronik bileşenlere, seçim öncesi ve sırasında kötü amaçlı yazılım yüklenmesi ve bu yolla oyların kullanımında, sayımında sahtecilik yapılması mümkün olabilmektedir.

2.3.3.2 Oy sayım sunucularına saldırı

Oyların sayım sunucularına aktarıldığı sistemlerde, sunuculara; mevcut sistemin zayıflıkları, sıfır-gün (zero-day) zayıflıkları vb. kullanılarak erişilebilir ve oyun depolanması, sayılması süreçlerine müdahale edilerek seçime hile karıştırılabilir.

2.3.3.3 Ağın izlenmesi

Özellikle ağ tabanlı seçim sistemlerinde, kullanıcının kullandığı oy merkeze iletilene kadar, gerekli tedbirler alınmadığında izlenebilmekte ve hatta değiştirilebilmektedir.

2.3.3.4 Aygıtların hatalı ayarlanması

Elektronik seçim sistemleri, yapısına ve türüne göre değişmekle birlikte, seçim öncesi farklı ayarlamalara ihtiyaç duyabilmektedir. Bu ayarların, kasıtlı olarak yanlış yapılmasıyla, seçmenler A seçimini yapmak isterken B seçimine yönlendirilebilir veya mükerrer oy kullanımını engellemeyerek, farklı kişilere mükerrer oy kullandırtılabilir.

2.3.3.5 Hizmet reddi saldırıları

Özellikle ağ tabanlı elektronik sistemlerde, sistemin IP'si tespit edilerek, hizmet reddi saldırıları yapılıp, seçim engellenmeye çalışılabilir.

2.3.3.6 Tehdit veya rüşvetle oy satın alma

Sistemin verdiği bir bilgi istismar edilerek, seçmenlerin hangi oyu kullandığı belirlenebilir ve bu kanıt üzerinden seçmenler tehdit edilerek ve seçmenlere rüşvet verilerek oy satın alma işlemleri yapılabilir.

2.3.3.7 Oy pusulası sahtekarlıkları

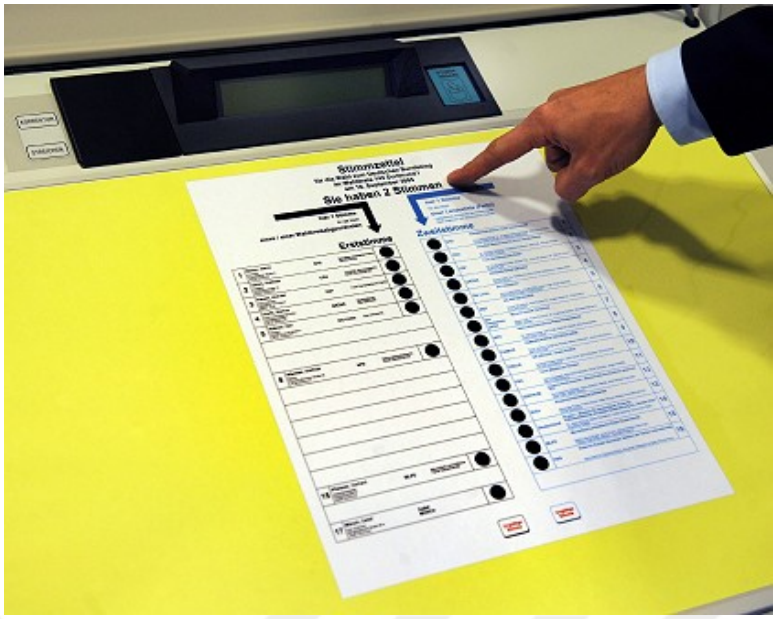
Elektronik sistemin, özellikle sayma sürecine entegre olduğu yapılarda, oy pusulaları tahrif edilerek, çalınarak sahtekarlık yapılabilir. Bunun yanında, kullanılan elektronik oy için kağıt pusula çıktısı veren sistemlerde, pusula oy sandığına atılmayarak, sisteme duyulan güvenin azalması da sağlanmaya çalışılabilir.

2.4 Dünya'daki Uygulamalar

Elektronik seçim sistemine geçmiş veya elektronik sistemlerini seçim sistemlerinin bir bölümünde kullanan birçok ülke bulunmaktadır. Bunların yanında geçiş işlemleri; iyi kurgulanmamış sistemler tercih edilmesi veya seçmenlerin ikna edilememesi gibi sebeplerle başarısız sonuçlanmış Almanya, Belçika, İrlanda gibi ülkeler de bulunmaktadır [6]-[8].

2.4.1 Almanya

Almanya'da, Ulusal Standart Enstitüsü (PTB) tarafından test edilmiş ve onaylanmış 2 adet oy aygıtı bulunmaktadır. Bu aygıtlar Hollanda'lı teknoloji firması NEDAP tarafından geliştirilmiş ESP1 ve ESP2 aygıtlarıdır. Elektronik seçim ilk olarak 1998 yılında Köln şehrinde test edilmiştir. Test sürecinde kullanılan oy verme aygıtları Şekil 2.3'de görülebilmektedir.



Şekil 2.3 : Almanya'da 1998 yılında kullanılan elektronik seçim sistemi [23].

Testler başarılı olarak görülmüş ve 1 yıl sonrasında şehrin Avrupa Parlemontosu seçimlerinde kullanılmıştır. Sonraki süreçte diğer şehirler de akıma uyarak elektronik seçim aygıtlarını kullanmış ve 2005 yılında 2 milyona yakın Alman seçmenin katıldığı Bundestag seçimlerinde kullanılmıştır. Alman seçmeni oy aygıtlarına ve elektronik seçim kavramına oldukça sıcak yaklaşmış ve olumlu karşılamıştır.

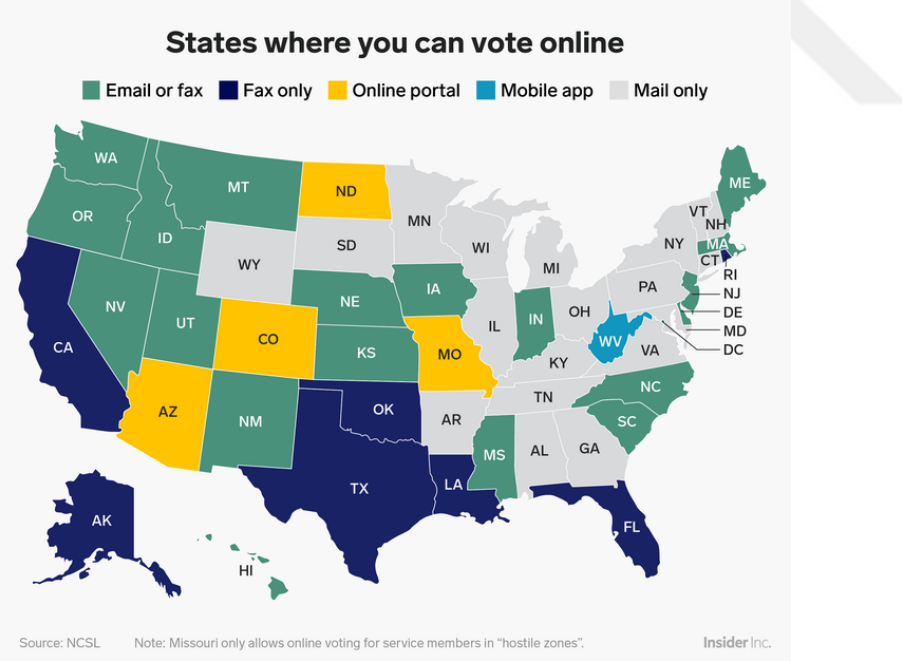
2005 yılındaki seçimden sonra 2 seçmen elektronik seçimin anayasaya aykırı, manipüle edilebilir olduğunu aynı zamanda yapılan seçimin de bu sebeplerle güvenilmez olduğunu iddia ederek Alman Anayasa Mahkemesi'nde dava açmıştır. Dava sonucunda mahkeme elektronik seçime karşı herhangi bir karar almamış ancak şeffaflığa dikkat çekmiştir [23].

2005 yılındaki seçimin devamında, geniş halk desteğini de arkasına alan senato 2008 yılındaki Hamburg eyalet seçimleri için dijital kağıt temelli optik taramalı oy sisteminin kullanılması konusunda çalışmalar başlatmıştır. Ancak 2007'nin son çeyreğinde "Fraktion der Grünen/GAL" ve "the Chaos Computer Club" adlı büyük bilgisayar gruplarından sistemin güvensiz olduğuna dair iddialar ortaya atılmıştır. Bunun üzerine Federal Seçim Ofisi (Bundeswahlamt) sisteme halkın duyduğu güvenle ilgili anket çalışması yapmış ve anket sonucunda ortaya çıkan elektronik sisteme karşı duyulan güvensizlik sebebiyle, elektronik seçim sisteminin kullanım planlarını iptal etmiştir [24]. Böylece Almanya 1998 yılında başlattığı elektronik seçim sistemi çalışmalarını 2009 yılında bitirmiştir.

2.4.2 Amerika Birleşik Devletleri

Amerika Birleşik Devletleri'nde ilk elektronik seçim sistemi, 1964 yılında 7 idari bölgede kullanılmaya başlanmıştır [25]. Özellikle yurt dışında olan vatandaşlar için e-mail, faks, i-seçim ara yüzleri ile kullanılmaya başlanan elektronik seçim; tüm seçmenleri kapsamaya başlamasıyla tartışmalara sebep olmuştur. Arizona'da 2000 yılında, geleneksel seçim yöntemleriyle birlikte kullanılmaya başlanan sistem; güvenlik, bilgisayar okur yazarlığının düşük olduğu kesimlerin oy kullanım yüzdelerinin düşeceği gibi çekincelerle dava edilmiş ancak engellenememiştir [26].

Seçim kurullarının, eyaletlere bağlı olması sebebiyle, federal düzeyde tek bir seçim sistemi bulunmayan ABD'de, kağıt bazlı elektronik seçim sistemleri, DKE elektronik seçim sistemleri, i-seçim sistemleri kullanılmaktadır. Ülkede Delaware, Georgia, Louisiana, New Jersey, South Carolina eyaletlerinde, sürecin herhangi bir yerinde kağıt kullanılmayan seçim sistemi kullanılmaktadır [27]. Elektronik seçim sistemine geçiş, "Help America Vote Act" federal yasası sayesinde hızlanmıştır. Şekil 2.4'teki haritada, eyaletlere göre elektronik seçim kullanımı görülmektedir.



Şekil 2.4 : ABD eyaletlere göre elektronik seçim kullanımı [28].

2016 başkanlık seçimleri sonrasında, Rusya'nın ABD'deki seçimlere müdahale ettiği iddia edilmiş ve elektronik seçim sistemleri eleştirilmiştir [4].

2.4.3 Belçika

Belçika'da elektronik seçim sistemi ilk defa 1991 yılında genel seçimlerde kullanılmaya başlanmıştır. 2 farklı cihazın test edilmesi için, 2 farklı seçim merkezi belirlenmiş ve cihazlar seçim süresince test edilmiştir. Cihazların ikisi de doğrudan kayıt yapmamış ve oy pusulası işaretleme cihazı olarak hizmet vermiştir [29].

Cihazlardan biri dokunmatik ekranlı, diğeri ise manyetik kart, elektronik pusula işaretleme aygıtı ve dokunmatik kalem (light pen, stylus) kullanmaktaydı. Manyetik kartlı aygıt 2004 yılına kadar kullanılmaya devam edildi.

2003 yılında "Ticketing" adlı yeni bir sistemi 2 oy merkezinde denenmiştir. Yeni sistemde, seçmenler oy kullandıktan sonra, kullandıkları oyun kopyası bir yazıcı yardımıyla basılmakta ve seçmen tarafından kontrol edilmekte, daha sonra da bir sandıkta kağıt kopyalar toplanmaktaydı. Sayımda ise kağıt oylar ile elektronik oylar birlikte sayılması, tutarsızlık durumunda kağıt oy sayımının geçerli olacağı bildirilmişti. Bu sayede elektronik oylarda bir değişiklik olup olmadığı kontrol edilebilmesi ve seçimlere şüphe düşmemesi amaçlanmaktaydı. Şekil 2.5'te elektronik seçimde kullanılan aygıtlardan biri görülmektedir.

Şekil 2.5 : Belçika'da kullanılan elektronik seçim aygıtı [30].

Ancak yapılan seçimde, kağıt oylar ve elektronik oyların sayıları hiç bir merkezde eşit çıkmamış, bunun sonucunda Belçika Seçim Kurulu, dönemin yasalarına aykırı olarak, elektronik oyların daha güvenilir olduğunu ve elektronik oyların geçerli olacağını bildirmiştir.

Sonraki seçimlerde “Ticketing” sistemi bir daha kullanılmamış ve manyetik kart sistemine geri dönülmüştür. 1999 yılında bu yana Belçika’da herhangi bir seçim merkezi elektronik seçime dönmemiş ve be sebeple seçmenler arasında elektronik seçim kullanım oranı %44’te kalmıştır.

2.4.4 Brezilya

Brezilya, ilk elektronik seçim denemesini 1996 yılında, Santa Catarina eyaletinde gerçekleştirmiştir. 2000 yılında bu yana da tüm Brezilya’da elektronik seçim sistemi kullanılmakta ve seçim bittikten sonra dakikalar içerisinde sonuçlar ilan edilmektedir [12].

Brezilya’da kullanılan elektronik seçim aygıtları (Şekil 2.6); seçmen kimlik doğrulama, güvenli oylama ve oy sayma işlemlerinin hepsini barındırmaktadır. Siyasi partilerin, seçimi denetleyebilmeleri için, elektronik seçim sistemine erişim hakları bulunmaktadır. Elektronik seçim sistemlerinin kullanılmasından bu yana, seçim sahtekarlığı konusunda herhangi bir dava veya başvuru açılmamış olmasına rağmen, birtakım şikayetler dile getirilmiştir. Sistem, eleştiriler etrafında geliştirilmeye devam edilmiştir. Oyların basılı kopyasının alınmaması eleştirisine karşın, sistem oyların çıktısını alacak şekilde geliştirilmiştir. Aynı zamanda, 2012 yılından bu yana kullanıcıların kimlik doğrulamaları parmak iziyle yapılmaya başlanmış, 2014 yılında bu sistemi 22 milyon seçmen kullanmıştır.

Şekil 2.6 : Brezilya’da kullanılan elektronik seçim aygıtı [31].

Kullanılan sistemin, kaynak kodları ve tasarımları açık olmamakla birlikte, ticari olarak patentlidir. Seçim sürecinde kullanılan, oy kullanma aygıtlarının fiyatı 1000$ civarındadır [14]. Sistemin son halinde, seçmenlerin kullandıkları oyların kağıt çıktısı alınmakta ve cam arkasından el değmeden gösterilmektedir. Seçmen, yanlış oy kullandığını düşünürse oyu iptal edip, farklı bir oy kullanabilmektedir. Onayladığı takdirde, oy el değmeden saklanmaya devam etmektedir.

2.4.5 Estonya

Estonya'da elektronik seçim kullanımı 2005 yılında başlamıştır. Ülke elektronik seçim sistemine hızlı bir şekilde adapte olmuş ve internet tabanlı seçim sistemlerini tercih etmiştir. 2005 yılında uygulanan i-seçim sistemini 9.317 seçmen kullanmış, 2011 yılında ise bu sayı 140.846'yı bulmuştur. 2015 yılında yapılan seçimlerde ise, i-seçim kullanan seçmenlerin sayısı 176.491, oranı %30,5 olmuştur [32].

Ülkede, seçim sistemi, "Estonian ID Card" (çipli kimlik kartı) üzerine kurulmuştur. Akıllı kart sayesinde kimlik doğrulaması yapılmakta ve seçmenin internet üzerinden oy kullanılmasına müsaade edilmektedir. Seçmenin oy kullanabilmesi için kart okuyucu bulundurması gerekmektedir. Sistemin kaynak kodları 2013 yılında yayınlanmıştır [33].

Sistemle ilgili ciddi eleştiriler bulunmaktadır. Birçok bilişim topluluğu sistemi test etmiş ve güvensiz bulmuştur [32]. Michigan Üniversitesi ve Open Rights Topluluğu'ndan siber güvenlik uzmanları sistem üzerinde süreç, kod ve yapı bağlamında yaptıkları incelemelerde bir çok sorunla karşılaşmışlardır [5].

2.4.6 Hindistan

Dünyadaki en yaygın ve sistematik elektronik seçim kullanımı Hindistan'da uygulanmaktadır. Aynı zamanda nüfusu itibariyle, seçmen sayısı bakımından en çok elektronik seçim sistemi kullanılan ülkedir.

Uzun demokrasi geçmişine rağmen, seçim süresince; oy sandıklarının çalınması, oy merkezinin belirli bir partinin taraftarları tarafından ele geçirilip usulsüz oy kullanımı gibi sebepler Hindistan'ı elektronik seçim yönünde çalışmalar yapmaya itmiştir. Bu çalışmaları yaparken, sistemin basit, ucuz ve okuma yazması olmayan seçmenlerin de rahatlıkla oy kullanabilmesine olanak sağlaması gerektiği göz önünde bulundurulmuştur.

Elektronik seçime geçiş çalışmalarına 1982'de başlayan Hindistan'ın, sistemi tüm ülkede yaygınlaştırması 2004 yılını bulmuştur. 2004 yılından bu yana elektronik seçim sistemi tüm ülkede kullanılmaktadır [26].

Hindistan'ın elektronik seçim sisteminde; seçmenlerin, kimlik tespiti için yeni bir yöntem geliştirilmemiş olup, seçim görevlileri tarafından kimlik kontrolü yapılmaktadır. Mükerrer oylar da, parmağa damlatılan mürekkep ile engellenmeye çalışılmaktadır.

Kullanılan elektronik seçim aygıtı; oy verme, kontrol adlarında iki birimden oluşmaktadır. Oy verme birimi, oy pusulası biçiminde tasarlanmıştır. Bu birimde sol kısma, ilgili partilerin adları ve logoları gelmekte, sağ kısımda ise oy vermek için kullanılan düğmeler bulunmaktadır. Oy verme biriminde 16 adaylık yer bulunmaktadır. Bir kontrol birimine 4 farklı oy verme birimi yerleştirilerek, 64 adet adaya kadar çıkarılması mümkündür.

Sistem tasarımı basit ve üretimi maliyeti düşük tutulmaya çalışılmıştır. Güvenlik endişeleri sebebiyle yeniden programlanamayan mikroişlemciler kullanılmıştır. Bu mikroişlemciler fabrikada programlanmakta ve sonrasında değişim mümkün olmamaktadır. Aynı zamanda, sistem kapağı açılmaya çalışılınca kendisini otomatik olarak kapatmaktadır.

Elektronik seçim sürecinde kullanılan, elektronik seçim aygıtlarının birim fiyatı 200$ civarındadır. 390 milyon seçmen bulunan Hindistan'da, yaklaşık 1 milyon OVM bulunmaktadır. OVM'ler seçim sonrası 1214 adet yerel merkezde toplanarak sayım yapılmaktadır. Bu sebeple, yaygın seçim hilelerinin uygulanması pek mümkün değildir. 2004 yılındaki seçimlerde 1 milyon OVM'nin sadece 1800 tanesi arıza yapmıştır [14].

Hindistan kullanılan elektronik seçim sistemi bir çok açıdan sorunsuz çalışmaktadır. Ancak iki firma tarafından üretilen seçim sistemlerinin, yazılımlarının kontrole kapalı olması eleştirilere sebep olmaktadır. Seçmenin kullandığı oylar ile ilgili, Brezilya'daki gibi kağıt bir belge üretilmediği için sistemin doğruluğu denetlenememektedir.

2.4.7 Kanada

Kanada'da federal ve temsilci seçimleri için kağıt oy pusulaları kullanılmasına rağmen, belediye seçimleri için 1990 yılından bu yana elektronik seçim sistemleri tercih edilmektedir. Bazı bölgelerde ise i-seçim sistemi uygulanmakta ve federal komisyon tarafından, Kanada genelinde i-seçim sistemine geçilmesi tavsiye edilmektedir. Ülkede elektronik seçim için herhangi bir standart bulunmamaktadır [34].

2.4.8 Venezuela

İlk elektronik seçim denemesini 1998 yılında, başkanlık seçimlerinde yapan ülke, 2004 yılında sistemini, oyun kağıt çıktısını verecek şekilde güncellemiştir [26]. Elektronik seçim sistemini, "Smartmatic" firmasından temin eden Venezuela [35], 40.000'den fazla elektronik oy aygıtı, seçim sırasında kullanmaktadır.

Seçim sisteminde, mükerrer oyu önlemek ve kimlik kontrolü yapmak amacıyla, her seçmenin parmak izi alınarak, merkezi kimlik sunucularından kontrol edilir. Daha sonra kullanıcının oyunu kullandığı bilgisi de bir sunucuda tutularak, mükerrer oyların önüne geçilmeye çalışılır.

Kimlik onaylandıktan sonra, seçmen oyunu kullanmak üzere kapalı bölüme geçer. Dokunmatik oy pusulasından oy vermek istediği kişi veya partiyi seçerek, kontrol birimi ekranından onaylar. Seçmen oyunu onayladıktan sonra, oyu kaydedilir ve kullandığı oy kağıt bir çıktı olarak verilir. Seçmen kullandığı oyu kağıt üzerinde de kontrol ettikten sonra, kağıdı ilgili sandığa atar. Böylece, elektronik oyların sağlamasını yapmak üzere, kağıt bir delil oluşturulmuş olur. Seçim süresi bittikten sonra, oylar merkezi sunucuya aktarılır ve sayım yapılır.

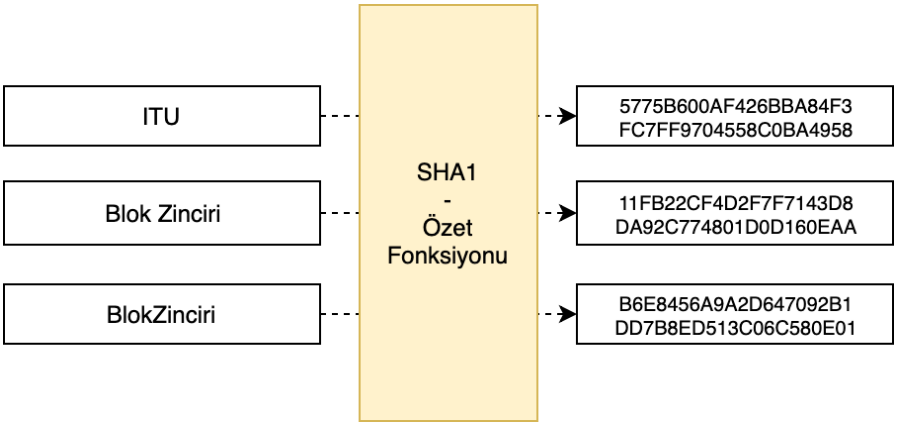
Özellikle Hugo Chavez'in görevden alınması ile ilgili 2004 yılında yapılan seçimler sonrası, sistemde hile olduğu iddiası ortaya çıktı. Bunun üzerine Seçim Kurulu, muhalefetin iddiasına cevap olarak, oyların %1'inin tekrar sayılması için rastgele seçim aygıtları belirledi. Ancak sonraki süreçte, muhalefet aygıtların rastgele seçilmediği, seçim sonrası anketler ile seçim sonuçlarının uyuşmadığı konusunda görüşlerini sürdürdü. Oy aygıtlarıyla, merkez arasındaki iletişimin iki yönlü olmasını da kuşkulu bulduklarını ileten muhalefet, aynı zamanda elektronik seçim sisteminde hile yapmanın çok kolay tespit etmenin ise çok zor olduğu yönünde görüş bildirdi [14].

3. ŞİFRELEME TEMELLERİ

Günümüzde, hemen hemen her elektronik sistemde, güvenlik için şifreleme (kriptografik) yöntemleri kullanılmaktadır. Özellikle sistemin dağınık olduğu durumlarda, bu yöntemler daha fazla önem kazanmaktadır. Bu bölümde, tasarımda kullanılan şifreleme temellerinden özetle bahsedilmiştir.

3.1 Özet Fonksiyonu

Özet fonksiyonu, değişken uzunluklu verileri, belirli uzunlukta özet verisine haritalayan algoritmadır. İdeal bir özet fonksiyonunda; özet hesaplaması kolay olmalı, belirli bir özete ait mesajı oluşturmak zor olmalıdır. Özet fonksiyonları, sayısal imza, dosya ve mesaj bütünlüğü doğrulama, anahtar üretimi vb. alanlarda kullanılmaktadır. Kullanılan algoritmaya göre özet uzunluğu değişkenlik göstermektedir. MD5, SHA-1, SHA-256/224, SHA-512/384 özet fonsiyon algoritmalarına örnek olarak verilebilir. Şekil 3.1'de SHA-1 algoritmasıyla hesaplanmış özetler gözükmektedir.



Şekil 3.1 : SHA-1 özet fonksiyonu çıktıları.

3.2 Sıfır Bilgiyle İspat

Sıfır bilgiyle ispat, bir tarafın bildiği bir bilgiyi, diğer tarafa bu bilgiyi ifşa etmeden ispat etmesi yöntemidir. Bu yöntemde, iddia veya bilgi sahibi, bilgiyi bildiğini ispat ederken, bilgiyle ilgili herhangi ipucu vermez.

Bu yöntem etkileşimli ve etkileşimsiz olarak uygulanabilir. Etkileşimsiz türde, sadece bir taraf aktif olarak kanıtı doğrularken; etkileşimli türde iki taraf da daha önceden belirlenmiş bir yol ile haberleşirler.

Sıfır bilgiyle ispat; bütünlük, sağlamlık ve sıfır bilgi özelliklerini sağlamalıdır:

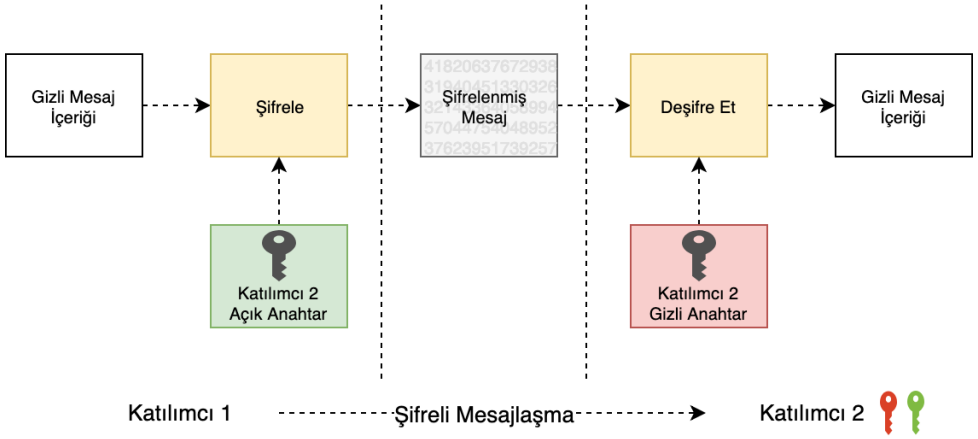
- Bütünlük: eğer iddia doğru ise, dürüst doğrulayıcı, dürüst bir iddia sahibi tarafından buna ikna edilir.
- Sağlamlık: eğer iddia yanlış ise, kötü niyetli bir iddia sahibi, doğrulayıcıyı iddiasının doğru olduğu konusunda ikna edemez.
- Sıfır Bilgi: eğer iddia doğru ise, doğrulayıcı bu iddianın doğru olduğu dışında herhangi bir şey öğrenemez.

Bu yöntemde, doğrulayıcı küçük bir olasılıkla, kötü niyetli bir iddia sahibi tarafından ikna edilebilir. Bu ihtimali minimuma indirgemek için süreç tekrar uygulanabilir. Sıfır bilgiyle ispat yöntemi, kimlik doğrulama sistemlerinde, blok zinciri yapılarında vb. kullanılmaktadır.

3.3 Açık Anahtar Şifrelemesi

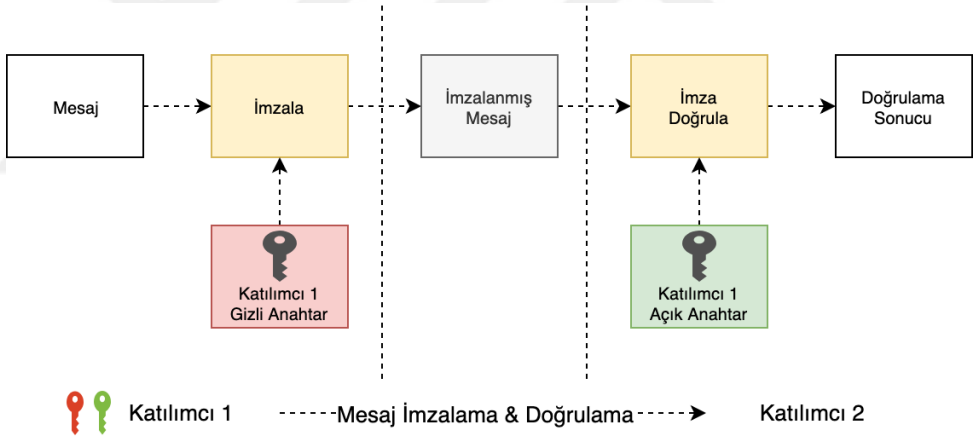
Açık anahtar şifrelemesi (asimetrik şifreleme), açık ve gizli anahtar çiftlerini kullanarak, şifreleme, elektronik imza vb. işlemlerin yapılmasını sağlayan, yaygın olarak kullanılan bir şifreleme sistemidir. Anahtar çifti oluşturulduktan sonra, gizli anahtarın saklanması, açık anahtarın ise iletişim kurulacak taraflarla paylaşılması gerekmektedir.

Şifreleme işlemi, mesaj gönderilecek katılımcının açık anahtarı kullanılarak yapılır. Deşifre etmek için ilgili katılımcının gizli anahtarının kullanılması gerekir. Açık anahtar ile şifreleme işleminin akışı Şekil 3.2'de görülebilmektedir.



Şekil 3.2 : Açık anahtar şifrelemesi akış özeti.

Elektronik imza ise, imzalayacak katılımcının gizli anahtarı kullanılarak gerçekleştirilir. İmza, sürecin diğer katılımcıları tarafından, imzalayan katılımcının açık anahtarı kullanarak kontrol edilir. İmzalama sürecinin akış özeti Şekil 3.3'de görülebilir. RSA, Diffie-Hellman, ElGamal bu sistemi kullanan algoritmalara örnek olarak verilebilir.



Şekil 3.3 : Açık anahtar elektronik imza akış özeti.

Sistemin yaygın kullanılıyor olması, şifrelemeyi aşmaya çalışan kişi ve kurumların açık hedefi haline getirmektedir. Devletlerin veya güçlü kurumların elinde yüksek işlem gücü bulunabileceği ve bunun anahtar çiftlerini elde etmek için kullanılabileceği göz önünde bulundurulmalıdır. Bu sebeple anahtar çifti uzunlukları belirlenirken dikkatli olunmalıdır. Ayrıca istihbarat kurumlarının da sistemi manipüle etmek adına çalışmalar yapabileceği göz ardı edilmemelidir. Örneğin NSA'in, RSA algoritmasının geliştiricilerine, sisteme kusurlu bir rastgele sayı üreticisinin dahil edilmesi için, 10 milyon dolar rüşvet verdiği iddia edilmektedir [36].

Açık anahtar şifrelemesi, bir çok sistemde olduğu gibi, elektronik seçim sisteminde de yaygın olarak kullanılmaktadır. Haberleşmenin şifrelenmesi, kimlik doğrulama, elektronik imzalama vb. işlemlerde tercih edilmektedir.

3.4 Eşik Şifreleme Sistemi

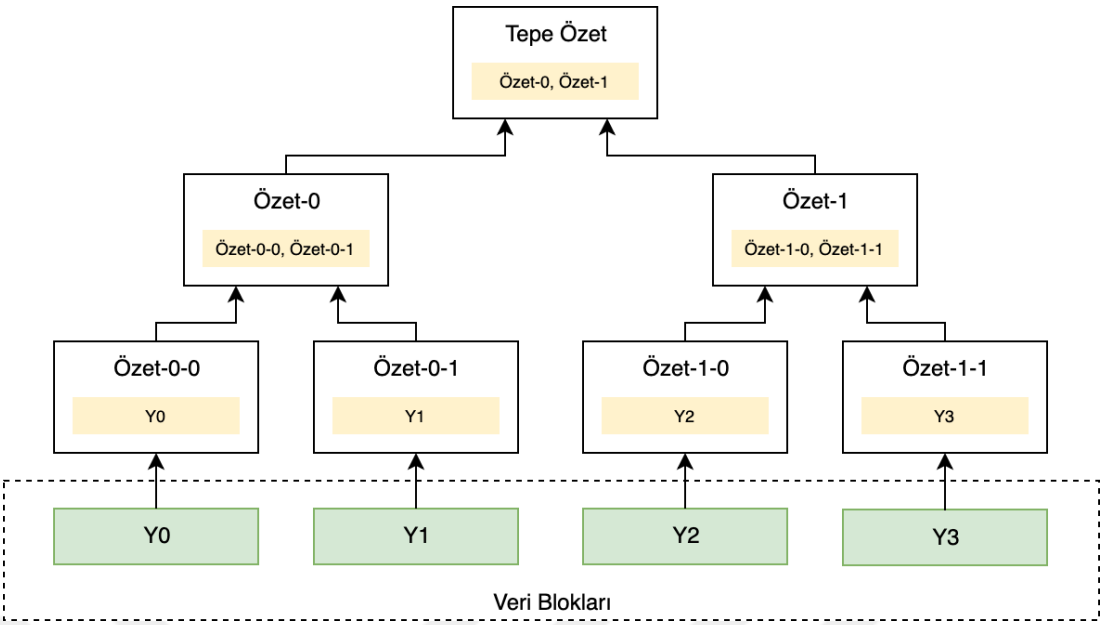
Eşik şifreleme sistemi, deşifre etme veya elektronik imzalama işlemlerinin, yalnızca belirli bir eşik sayı üzerinde katılımcının iş birliği yaparak gerçekleşmesini sağlayan şifreleme sistemidir. Sistem RSA, ElGamal, DSA algoritmalarıyla çalışabilmektedir.

Şifreleme ve deşifre etme işlemlerinde, mesaj açık anahtar ile şifrelenir, gizli anahtar ise katılımcı taraflar arasında dağıtılır. Başlangıçta belirlenen sayı kadar katılımcı, işbirliği yapar ve kendilerine verilen parçaları kullanırsa, mesaj deşifre edilebilir.

Sistem, elektronik seçim sisteminde, kullanılan önemli anahtarların siyasi partiler arasında dağıtılması için kullanılabilir. Böylece seçim sisteminin merkeziyetçilikten uzaklaşması ve riskin dağıtılması sağlanabilir.

3.5 Merkle Ağacı

Merkle ağacı (Merkle tree, Hash tree), veri özetlerini birbirine bağlayarak, büyük veri yapılarını verimli ve güvenli bir biçimde doğrulanmasını sağlar. Bu yöntemde, Şekil 3.4’te görülebilecği üzere, her yaprak veri bloğunun, her düğüm ise alt düğümlerin özet değerlerini içerir. Merkle ağacı; dosya sistemlerinde, dağıtık versiyon kontrol sistemlerinde, P2P ağlarında ve blok zinciri gibi dağıtık veritabanlarında kullanılmaktadır.



Şekil 3.4 : Merkle ağacı yapısı.

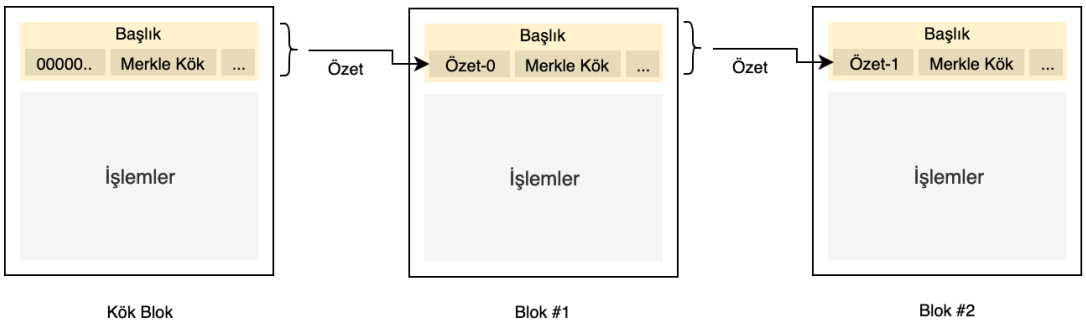
4. BLOK ZİNCİRİ

Blok zinciri (Blockchain), bilginin tüm katılımcılarla ağ üzerinden paylaşılmasını sağlayan, dağıtık bir veritabanıdır. Bilgi bloklar halinde, önceki bloğun özeti içerisinde bulundurulacak şekilde depolanır. Bloklar madenci adı verilen, katılımcılar tarafından oluşturulur ve diğer katılımcılar tarafından, önceden tanımlanmış kurallara uygunluğu kontrol edilir.

2008 yılında Satoshi Nakamoto tarafından bir makalede Bitcoin adlı sanal, dağıtık para sisteminin altyapısını oluşturacak şekilde açıklanmış [37] ve 2009 yılında hayata geçirilmiştir [38]. Blok zinciri altyapısı, günümüze kadar farklı projeler için, farklı şekillerde geliştirilmeye devam etmiştir.

4.1 Çalışma Yapısı

Blok zincirinin temel çalışma yapısında; işlemler (alt veriler), belirli limitler içerisinde bloklara depolanmakta ve bloklar önceki bloğun başlık bilgisinin özet değerini içererek verinin bütünlüğü sağlanmaktadır. Veritabanı, tüm katılımcıların erişimine açık, dağıtık bir yapıdadır. Blok zinciri sisteminde; madenci (miner), oluşturucu, doğrulayıcı adları verilen özel katılımcılar, ilgili veritabanının güvenliğine ve sağlamlığına, blok oluşturarak ve blokları doğrulayarak katkıda bulunurlar. Blok zincirinin temel yapısı Şekil 4.1'de görülebilmektedir.



Şekil 4.1 : Blok zinciri temel yapısı.

Bu temel yapının dışında, tüm kurallar, sistemin oluşturucusu veya oluşturucuları tarafından belirlenir. Bu kurallar uzlaşı kuralları veya doğrulama kuralları adını alır (consensus rules, validation rules). Zaman içerisinde, blok zinciri yapısını geliştirmek, yeni özellikler getirmek adına, fikir birliğiyle uzlaşı kurallarında değişiklikler yapılabilmektedir. Uzlaşı kuralları aşağıdaki konuları kapsayabilmektedir:

- Blok zinciri tipi
- Blok zinciri yapısı
- Blok yapısı
- İşlemlerin (alt veriler) özellikleri
- Doğrulama kuralları
- Fikir birliği yöntemi
- Kullanılacak algoritmalar

Uzlaşı kuralları çerçevisinde belirlenen bir düzende seçilen madenciler, blok zinciri için blok oluştururlar ve yine bu kurallara göre blokları doğrularlar. Birçok sistemde, madenci, blok ekleme işleminden kazanç sağlar.

Oluşturulan sisteme göre değişmekle birlikte, her katılımcının madenci olarak, tüm veritabanını indirmesine gerek olmamaktadır. Merkle ağacı yapısının yardımıyla, blokların sadece başlık kısımları indirilerek, doğrulama ve kontrol yapılabilmektedir.

4.1.1 Blok zinciri tipi

Günümüzde yaygın olarak kullanılan; açık (public), özel (private) ve kurul (consortium) adlarında, 3 temel blok zinciri tipi bulunmaktadır. Blok zinciri tipi, veriye erişim düzeyini ve kullanıcıların sisteme katılım düzeyini belirlemektedir. Belirtilen tipler dışında, özel olarak tanımlanan yapılar da bulunmaktadır.

4.1.1.1 Açık blok zinciri

Açık blok zincirinde, kullanıcıların veriye erişimi ve sisteme katılımı bakımından herhangi bir kısıtlama bulunmamaktadır. İnternet bağlantısına bağlı herhangi bir kullanıcı, veri gönderebilir, madenci olabilir. Açık blok zincirine örnek olarak, Bitcoin, Ethereum, Litecoin verilebilir.

4.1.1.2 Özel blok zinciri

Özel blok zincirinde, verilere, sistem yöneticisinin müsaadesi olmadan erişilemez. Aynı zamanda sistemde katılımcı olmak için de özel izin gerekir. Bu tip daha çok, şirketler tarafından tercih edilmektedir.

4.1.1.3 Kurul blok zinciri

Kurul blok zinciri, verilere erişim ve sisteme katılım bakımından özel blok zinciri tipine benzemektedir. Tek farkı, erişim ve katılım kararlarının, bir kurul tarafından verilmesidir.

4.1.2 Doğrulama kuralları

Blok zinciri, dağıtık bir sistem olması nedeniyle, sistemin herkes tarafından aynı şekilde doğrulanması çok önemlidir. Bu da önceden tanımlanan doğrulama kuralları tarafından sağlanmaktadır. Tüm katılımcılar, sistem ile etkileşirken bu kurallar çerçevesinde etkileşir ve madenciler de doğrulamayı bu kurallar çerçevesinde yaparlar.

Doğrulama kuralları, oluşturulan blokların limitleri, yapılan işlemlerin yetkili kullanıcılar tarafından yapılıp yapılmadığı, istenen verilerin blok içerisinde bulunup bulunmadığı gibi bir çok adımı doğrular ve verinin bütünlüğünü sağlar.

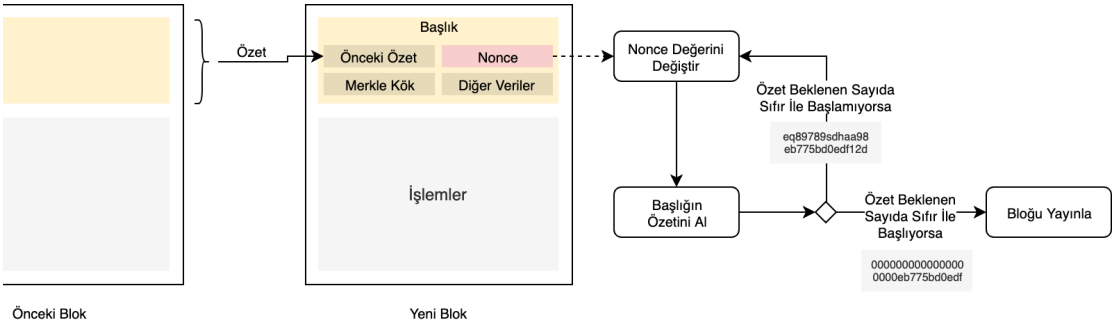
4.1.3 Fikir birliği yöntemi

Blok zinciri ağının kararlı ve dağıtık bir şekilde çalışabilmesi için, katılımcılar arasında fikir birliğine varılabilmesi ve olası kaba kuvvet saldırılarının önünce geçilmesi için bir yöntemin belirlenmesi gerekmektedir. Bu yöntem, blokları nasıl bir yol izlenerek oluşturulacağını, bunları oluşturacak madencilerin nasıl seçileceğini, kötü niyetli katılımcıların nasıl engelleneceğini belirler. Halihazırda yaygın olarak kullanılmakta olan iki yöntem bulunmakta ve sistemine göre özel yöntemler geliştirilebilmektedir.

4.1.3.1 Emek ispatı

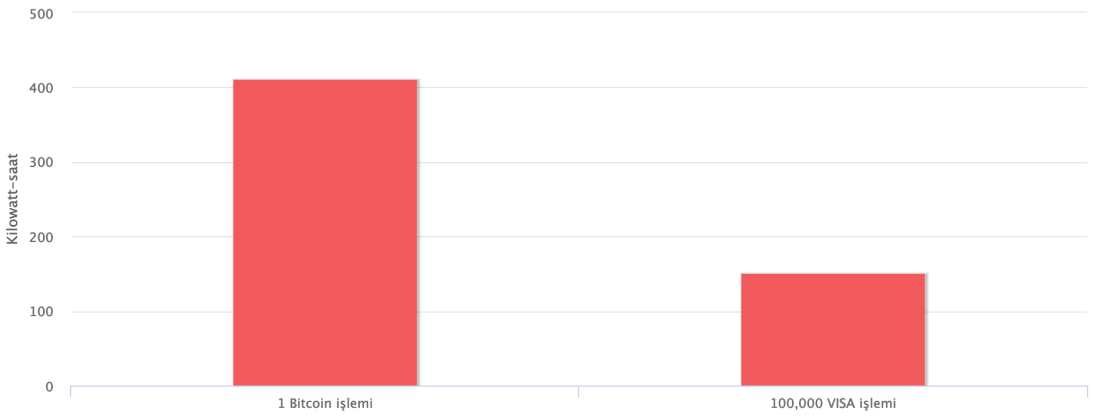
Emek ispatı (proof of work) yönteminde, madenciler blokları hazırladıktan sonra, her blok için çözülmesi zor ancak doğrulaması kolay bir problemin çözümünü de bloğa ekleyerek, ağa emek harcadıklarını ispat ederler. Bu problemin çözümüyle ağdaki diğer katılımcılar, bloğu geçerli bir blok olarak değerlendirirler.

Bu problem Bitcoin'de, özetinin önceden belirlenmiş kadar sıfır ile başlayacak şekilde bir blok oluşturulmasıdır. Çözüm, bloğun içerisindeki "nonce" değerinin farklı sayılarla denenmesiyle bulunmaktadır. İlgili blok bulunduktan sonra ağda yayınlanmakta ve bir sonraki blok için yeni bir süreç başlamaktadır. Hedeflenen periyotta blok üretimini yakalayabilmek adına, problemin çözümü zorlaştırılıp, kolaylaştırılabilmektedir. Bloğu oluşturan madenci, işlem ücretleriyle birlikte ve her blokta üretilen Bitcoin'i kazanmış olur. Sürecin özeti şekil 4.2'de görülebilmektedir.



Şekil 4.2 : Emek ispatı süreç diagramı.

Emek ispatı, çözülmesi gereken problemin çok fazla deneme yanılma yapılmasını gerektirmesi ve her bir madencinin her blok için ayrı ayrı problemi çözmeye uğraşması sebebiyle, fazla güç tüketimine sebep olmaktadır. Şekil 4.3'te, 1 Bitcoin işlemi ve 100.000 Visa işleminin güç tüketimi olarak karşılaştırıldığı grafik görülmektedir. Grafik sadece kullanılan cihazların işlem gücü baz alınarak oluşturulmuş, ofis ve çalışanların sebep olduğu güç tüketimi hesaba katılmamıştır.



Şekil 4.3 : Bitcoin, VISA güç tüketim grafiği [39].

Birden fazla madencinin, kazançlarını arttırma amacıyla bir araya gelmeleri ve madencilik havuzu oluşturmaları, blok zinciri ağlarını tehdit etmektedir. Madencilik havuzlarının, yüzde 51 saldırılarına alt yapı oluşturabilme tehlikesi bulunmakta ve dağıtık olarak kurgulanan yapıyı merkezileştirmektedir. Aynı zamanda, emek ispatı kullanılan ağlarda, ağ üzerindeki yoğunluğa göre ölçeklenebilirlik problemleri oluşmaktadır.

4.1.3.2 Hisse ispatı

Hisse ispatı (proof of stake) yönteminde, madenciler emek ispatı yönteminde olduğu gibi problem çözmezler. Bu yöntemde, sonraki bloğu işleyecek madenci, belirli parametrelere bakılarak rastgele seçilir. Seçilen madenci, bloğu oluşturur ve ağda yayınlar. Seçimi belirleyen parametreler, sistemden sisteme değişmekle birlikte, madencinin sistemdeki maddi varlığı, uzun süredir harcanmayan parası veya rehin bırakılan parası olabilmektedir. Bazı sistemlerde, seçilmek isteyen madenci bir kontrat veya yazılımsal arayüz yardımıyla parasını seçim yapan sisteme rehin bırakır.

Blok yayınlandıktan sonra ağın bloğu doğrulaması beklenir. Paranın rehin bırakıldığı sistemlerde, doğrulamadan sonra madenci parasını geri alabilir ve blokta bulunan işlem ücretleri de hesabına geçer. Blokta hata olması durumunda, madenci ceza olarak rehin bıraktığı parasının bir kısmını kaybeder.

Hisse ispatı yöntemi, blok oluşturma yöntemini madenciler arasında rastgele dağıtması ve madencilerin emek ispatında olduğu gibi problem çözmesi gerekmemesi sebebiyle daha az güç tüketir. Aynı zamanda, emek ispatı yöntemine nazaran merkezileşme ihtimali daha düşüktür.

4.2 Blok Zincirinin Avantajları

Çalışma yapısı itibariyle blok zincirinin bir çok avantajı bulunmaktadır. Tamamen dağıtık olması, yüksek erişilebilirlik imkanı sunmakta ve DDOS saldırılarına karşı sistemi dayanıklı kılmaktadır. Dağıtık ve şeffaf yapı, aynı zamanda, merkezi kontrol yapıları bulunmayan sistemler kurulmasına olanak sağlamakta, bu da sistemlerde kişi ve kurumlara güvenme gereğini ortadan kaldırmaktadır.

Zincir yapısı sayesinde ise verinin bütünlüğü sağlanmaktadır. Eski bir veriyi değiştirmek, tüm zincirin yeniden oluşturulmasını gerektireceğinden, oldukça zordur. Verinin şeffaf ve herkes tarafından erişilebilir olması da, tüm veriyi kontrol edilebilir ve doğrulanabilir kılmaktadır.

4.3 Blok Zinciri Dezavantajları

Her altyapı ve sistemde olduğu gibi, blok zincirinin de bir takım dezavantajları bulunmaktadır. Dağıtık olması sebebiyle, her bir madenci doğrulama işlemlerini yapması gerekmekte ve bu da aynı işlemin farklı bilgisayarlarda tekrarı manasına gelmektedir. Bu tekrar, güç tüketimini arttırmaktadır. Farklı fikir birliği yöntemleri ile güç tüketimi azaltılsa da, doğrulamanın tekrar tekrar yapılmasının önüne geçecek bir yapı kurulamamıştır.

Dağıtık olması aynı zamanda sistem oluşturulmasını ve kurulumunu, merkezi yapılara göre daha zor kılmaktadır. Bu durum, sistemde yapılacak güncellemeleri de zorlaştırmakta ve verinin çatallanmasına (hard fork, soft fork) sebep olabilmektedir.

4.4 Elektronik Seçimde Kullanım

Dağıtık ve şeffaf altyapı, blok zincirini, elektronik seçim sistemleri için oldukça uygun bir alt yapı haline getirmektedir. Sadece, blok zincirinin özelliklerinin, elektronik seçim özelinde, dikkatli bir şekilde belirlenmesi ve uygulanması gerekmektedir.

Seçimlerde, seçmenlerin kimlik ve sayıları belirli olduğu için, blok zinciri tipi olarak, herkesin katılımcı olabildiği açık blok zinciri tipi kullanılmaması gerekmektedir. Verilerin herkes tarafından erişilebilir olması ve bu sayede seçimde şeffaflığı sağlamak adına da, özel blok zinciri tipinin kullanılmaması gerekmektedir. Seçim sistemi için, katılımın izne tabi olduğu verilerin ise açık olarak herkes tarafından erişilebileceği özel bir sistem kurmak yerinde bir tercih olacaktır.

Yapılan seçimler, ülkelerin kaderlerini belirleyeceğinden, siteme müdahalede bulunma fikri, elinde yüksek işlem gücü bulunduran kurum ve devletlerin ilgisini çekebilir. Bu sebeple ağ üzerinde madencilik faaliyeti yapacak kişi ve kurumlar da izne tabi olmalıdır. Böylece blok zincirini hedef alabilecek, %51 çoğunluk saldırısı gibi fikir birliği saldırı yöntemlerinden veritabanı korunmuş olur.

Olası şüpheleri ortadan kaldırmak adına, seçim ile ilgili tüm verilerin de açık ve kontrol edilebilir olması gerekmektedir. Bu sebeple, kullanılacak tüm oyların, elektronik pusulalar ile önceden belirlenmesi, madencilik faaliyeti yapacak kişi ve kurumların da açık bir şekilde ilan edilmesi yerinde olacaktır. Bunu sağlamak adına, bu tarz veriler önceden hazırlanıp, blok zincirinin kök bloğuna (genesis block) konabilir.

Sistem seçim süresince çalışacak olsa bile, sistemin fazla güç tüketmemesi faydalı olacaktır. Aynı zamanda doğrulama işleminin de yüksek işlem gücüne bağlı olmaması, aksine madencilik faaliyeti için kaydolan herkese eşit fırsat verilmesi yerinde bir tercih olacaktır. Böylece yüksek işlem gücü bulunduran madencilerin, ağ üzerinde hakimiyet kurmasının önüne geçilir. Ayrıca merkeziyetçilikten de kaçınılmış olunur. Bu nedenlerden yola çıkarak, hisse ispatı yöntemine yakın bir yöntem belirlenebilir. Bu yöntemde, sonraki bloğu doğrulayacak madenci rastgele belirlenebilir.

5. TASARIM İLKELERİ

Tasarımın belirli bir çizgide ilerleyebilmesi ve başarılı olup olmadığının belirlenebilemesi için ilkeler belirlenmesi gerekmektedir. Bu ilkeler belirlenirken, geleneksel seçim sistemleri, günümüz demokrasi kültürü, günümüz şartları ve tasarlanacak sistemin elektronik olması göz önünde bulundurulmuştur.

Belirlenen ilkelerin, uygulandığında çıktı olarak, güvenilirlik ve tutarlılık sağlaması, aynı zamanda zorla oy kullandırma, oy satın alma, insanların siyasi tercihleri sebebiyle fişlenmesi gibi istenmeyen olayların önüne geçmesi amaçlanmıştır.

5.1 Bütünlük

Seçim sistemi, sistem üzerindeki önemli verilerin ve özellikle oyların hiçbir şekilde değiştirilemeyeceğini garanti etmelidir.

Geleneksel seçim sisteminde bu durum, kullanılan oyların mühürlü sandıklara konulmasıyla ve sandıkların gönüllü seçmenler veya taraflar tarafından gözlenmesiyle sağlanmaktadır.

5.2 Gizlilik

Seçim sistemi, seçmen ile oyu arasında herhangi bir ilişki kurulmasına müsaade etmemelidir. İlişki kurulmasını engelleyerek; fişleme, tehditle oy talep etme, oy satın alma gibi istenmeyen durumların oluşmasının önüne geçilmiş olur.

Geleneksel seçim sisteminde bu durum, oyu gizli kullandırtarak ve sandığa kapalı bir zarf içinde atılması istenerek sağlanmaktadır.

5.3 Tekillik

Sistem, tek bir seçmenin tek oy kullanabilmesini ve bir oyun bir defa sayılmasını sağlamalıdır.

Geleneksel seçim sisteminde, tüm süreç seçmenlerin ve tarafların temsilcilerinin gözü önünde gerçekleştiği varsayılarak bunun sağlandığı düşünülmektedir. Ancak toplu oy kullanımı, mükerrer oy kullanımı, başkası yerine oy kullanımı gibi vakalara rastlanabilmektedir [3], [40].

5.4 Doğruluk

Elektronik seçim sistemi, sonucu doğru olarak belirlemeli ve sonucun doğru olduğunu seçmenlere garanti edecek şekilde tasarlanmalıdır. Aynı zamanda gerektiğinde bunu ispatlayabilecek kayıtları tutmalı ve şüpheye yer bırakmayacak şekilde sunabilmelidir.

5.5 Kararlılık

Sistem çalışması esnasında ortaya çıkabilecek sorunlar önceden hesaplanarak tasarlanmalı ve çalışması herhangi bir şekilde bölünmemelidir. Seçim sistemine yönelik saldırılar ile seçim baltalanmaya veya seçime şaibe düşürülmek istenebilir.

5.6 Güvenlik

Sistem tasarlanırken, seçime hile karıştırmak isteyen kişilerin varlığının farkında olunmalı ve bununla ilgili önlemler alınmalıdır.

5.7 Kimlik

Seçmen ve yetkili kişilerin sisteme erişimi, kimlik kontrolü sağlanarak yapılmalıdır. Yetkisiz kişilerin, başkalarının adına oy kullanmalarının veya işlem yapmalarının önüne geçilmedir.

5.8 Şeffaflık

Sistem ve seçim süreci olabildiğince şeffaf olmalı ve seçmen güveni sağlanması adına seçmenler bu konu hakkında azami derecede bilgilendirilmelidir.

5.9 Denetlenebilirlik

Sistem üzerinde yapılan işlemler ile ilgili seçmen ve verilen oy ifşa edilmeyecek şekilde kayıt tutulmalı ve bu kayıtlara gerektiğinde yetkili kişiler ulaşabilmelidir.

6. SİSTEM TASARIMI

Elektronik seçim sistemi tasarlamak, tasarım gereklilikleri sebebiyle, oldukça karmaşık bir süreçtir. Tüm tasarım blokları kendini ispatlamış teknolojilerle oluşturulmalı ve şeffaf olmalıdır.

Sistem tasarlanırken; geleneksel seçim sistemleri ve günümüz şartları göz önünde bulundurularak, sistemin uyması gereken bazı ilkeler belirlenmiştir. Sistem bu ilkeler çerçevesinde; bankacılık, istihbarat vb. alanlarda güvenilerek kullanılan şifreleme yöntemleri ve algoritmalar ile tasarlanmaya çalışılmıştır. Sistem tasarımı, seçimin toplum içerisindeki her kesimden bireyi ilgilendirdiği göz önünde tutularak, basit tutulmaya çalışılmıştır.

Elektronik seçim sistemi tasarımı, sadece elektronik bir sistem tasarımı değil, aynı zamanda bir süreç tasarımı gerektirmektedir. Bu sebeple tasarım açıklanırken, öncelikle süreçten bahsedilmiş, daha sonra alt süreçler ve elektronik tasarımlar detaylı şekilde açıklanmıştır.

6.1 Tasarıma Genel Bakış

Seçim süreci, zorla oy kullandırma, oy satın alma vb. istenmeyen olayların önüne geçebilmek için, oy verme işlemi OVM'ler üzerinden gerçekleşecek şekilde kurgulanmıştır. Ancak kurgulanan sistem, birkaç basit düzenlemeyle cep telefonlarından, kişisel bilgisayarlardan kullanılabilir hale gelmektedir.

Tasarlanan süreçte, seçim ile ilgili tüm bilgiler, herkese açık ve herkes tarafından doğrulanabilen blok zinciri tabanlı bir veritabanına kaydedilir. Bu veritabanı, gönüllü kişi veya kurumların, cihazlarına seçim vaktinden önce başlayacak şekilde açık kaynaklı bir madencilik yazılımını kurup çalıştırmalarıyla faal olur ve ağ üzerinde dağıtılır. Madencilik adı verilen bu işlemler, verilerin kurallara uygunluğunun kontrol edilmesini ve yine kurallar içerisinde ağdan gelen verilerin yeni bloklar olarak işlenmesini içerir. Süreç içerisinde blok zinciri temelli veritabanına yapılabilecek saldırıları önlemek amacıyla kişi veya kurumların bir devlet portalına kendi açık anahtarlarını kaydetmeleri gerekmektedir.

Seçim başlangıcında OVM'ler ağa bağlanarak, önceden girilen seçim açık anahtarı yardımıyla, dağıtık veritabanını kaydeder ve doğrular, bilgileri okuyup kurulum yapar. Aday listeleri, ilgili resimler dahil bu ağdan alınır, seçimin açık anahtarı hariç bu cihazlara herhangi bir veri girilmesi gerekmez. Bu sayede insan kaynaklı, kurulum hatalarının önüne azami ölçüde geçilmiş olunur.

Geleneksel ve varolan elektronik seçim süreçlerinden farklı olarak, tasarlanan sistem, seçimin güvenliğini sağlamak adına, parti ve ilgili STK temsilcilerinin de katılarak gözlemlediği ve kontrol ettiği bir kök blok üretim süreci içermektedir. Bu süreçte, seçmen listesi ve seçmenlerin kayıtlı açık anahtarlarıyla, her seçmene özel pusulalar, kanıtlar basılır. Süreci gözlemleyen ilgili parti ve kişilerin tek dikkat etmesi gereken husus bu sürecin sonunda, üretilen özel pusulalar ile kanıtlar arasında kayıtlı herhangi bir ilişki kalmaması ve şifrelenen pusulaların içeriklerinin silinmesidir.

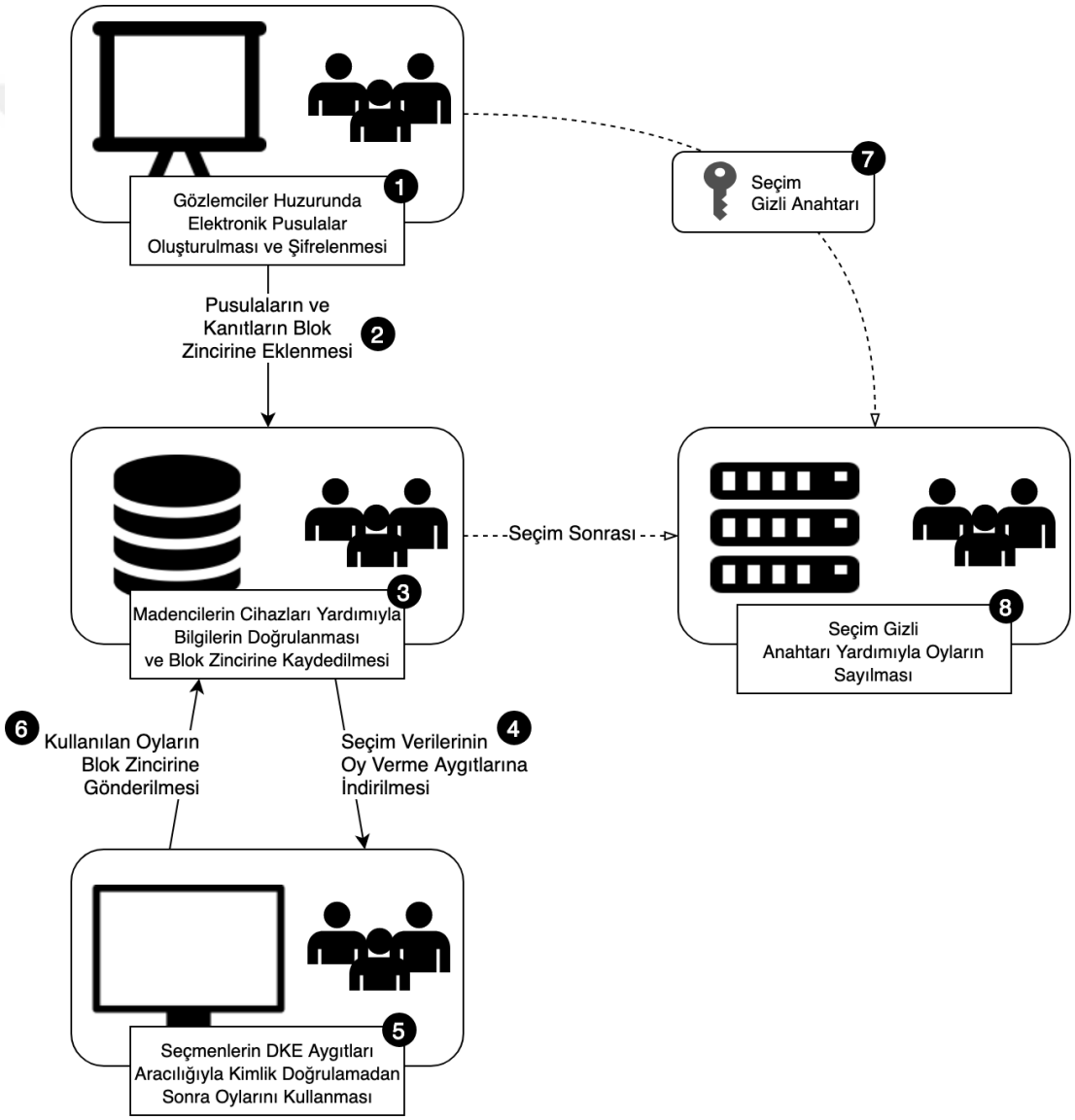
Pusulalar üretilirken, pusulanın içerisine oyun hangi bölge, seçim vb. ile ilgili olduğu bilgisi de girilir. Bu bilgi sayesinde seçmenler istedikleri aygıt üzerinden, sadece bağlı oldukları seçmeni oldukları bölge veya seçim için oylarını kullanabilir.

Kök blok üretim sürecinde, seçim açık ve gizli anahtarları da üretilir. Açık anahtar yayınlanır, gizli anahtar ise blok zinciri veritabanı kök bloğu oluşturulduktan sonra, kriptografik yöntemlerle parçalanıp seçim kurulu ve siyasi partiler arasında dağıtılır.

Üretilen elektronik pusulalar hem sadece ilgili seçmenin pusulayı işleyebileceği hem de madencilerin ilgili pusulanın doğru kişi tarafından kullanılıp, kullanılmadığını kontrol edeceği şekilde blok zinciri veritabanına eklenir. Bu süreçte seçmen ile kimliği arasında herhangi bir ilişki kurulamaması sağlanır.

Verilen oylar seçimin açık anahtarıyla şifrelenir ve böylece seçim bitmeden oyların tespitinin önüne geçilir. Seçim sonunda gizli şifre yayınlanarak herkesin oyları sayıp sonucu doğrulayabilmesi sağlanır.

Herkese açık sistem, veritabanı ve doğrulama yapısıyla; şeffaflığın en üst düzeyde sağlanmaya çalışıldığı bu tasarımda, ilgili tüm yazılım ve donanımların da açık kaynaklı olarak geliştirilmesinin daha uygun olacağı düşünülmektedir. Bu konuda daha sonra ayrıntılı irdelenmiştir (Bkz.: 6.12 Geliştirme Yöntemi). Şekil 6.1'de kurgunun özeti görülebilmektedir. Sistemin ve kurgulanan süreçlerin geniş özeti EK A.1'da görülebilir.



Şekil 6.1 : Tasarıma genel bakış.

6.2 Kayıt ve Kimlik Doğrulama

Dünya üzerinde, bir çok farklı kimlik doğrulama, seçmen kaydı süreci işletilmektedir. Geleneksel seçim yönteminde, genellikle bu işlem farklı partilerden temsilcilerin oluşturduğu, sandık başında bekleyen kurul tarafından yapılmaktadır. Seçim öncesi seçmen kaydı ise bazı ülkelerde bir zorunluluk iken, bazı ülkede kayıt otomatik olarak yapılmakta ve herhangi bir ek işlem gerektirmemektedir.

Dijital sistemlerin yayınlaşmasıyla, bazı ülkeler şekil 6.2'de görülen elektronik kimlik kartları kullanmaya başlamıştır. Bu kimlik kartlarının içerisine biyokimlik doğrulama, PIN doğrulama gibi işlevler ekleyerek daha güvenilir hale getirilmiştir.

Şekil 6.2 : Türkiye elektronik kimlik kartı.

Aynı işlevlerin bulunduğu farklı ülkelerde bile sistemler farklı çalışabilmektedir. Örneğin, bazı ülkelerde biyokimlik doğrulama herhangi bir merkezi sunucu doğrulamasına ihtiyaç duymadan çalışabilirken, bazı ülkeler sadece doğrulama sonrası merkezi sunucuya bildirim yapılmasını şart koşmakta ve bazı ülkelerde ise bu işlem merkezi sunucuya ihtiyaç duymaktadır.

Kimlik doğrulama sistemlerinin farklı çalışması, elektronik seçim sistemini de farklılaştırmaktadır. Tasarlanan sistemde, tüm süreç ve alt süreçlerde ekleme veya çıkarma yapılmasına ihtiyaç duyulabilmektedir.

Bu çalışma temelde, seçmenlerin açık ve gizli anahtarları oldukları, elektronik kimlik kartlarında gizli anahtarlarının bulunduğu ve kimlik doğrulamanın herhangi bir merkezi sunucu ihtiyacı duymadan yapılabildiği varsayılmıştır. Kart üzerinde bu anahtar ile şifre çözme işlemi yapılabilmesi bir artı olmakla birlikte, zorunluluk olarak görülmemiştir. Aynı zamanda kayıt konusunda, kayıt işleminin seçime katılım oranını düşürdüğü de varsayılarak, temel tasarım kayda gerek duyulmayacak şekilde yapılmıştır.

Kimlik doğrulama sürecinin elektronik olarak işlemesi, olası insan hatalarını ve kötü niyetli yaklaşımları ortadan kaldırmaktadır. Seçmenlerin elektronik kartı olmadığı, açık gizli anahtarlarının olmadığı veya kimlik doğrulama için merkezi sunucuya ihtiyaç duyulduğu diğer kimlik doğrulama yöntemleri ve seçmen kaydı gerektirebilecek bazı durumlar için de tasarım üzerinde değişiklik önerilerinde bulunulmuştur (Bkz.: 6.10 Farklı Kimlik Doğrulama Yöntemleri).

6.3 Oy Verme Makineleri

Seçim sisteminde oy kullanımı, baskı ile oy kullanımını engellemek adına, belirli merkezlerde bulunan OVM (Oy Verme Makinesi) yardımıyla yapılacak şekilde tasarlanmıştır. Bu sayede rüşvet, zorlama, tehdit ile oy kullanımının önüne geçilmek istenmektedir. Aynı zamanda seçmenlerin kendi bilgisayarları üzerinden oy kullandıklarında, cihazlarındaki güvenlik problemleri sebebiyle yaşayabilecekleri sorunların da önüne geçilmesi amaçlanmaktadır.

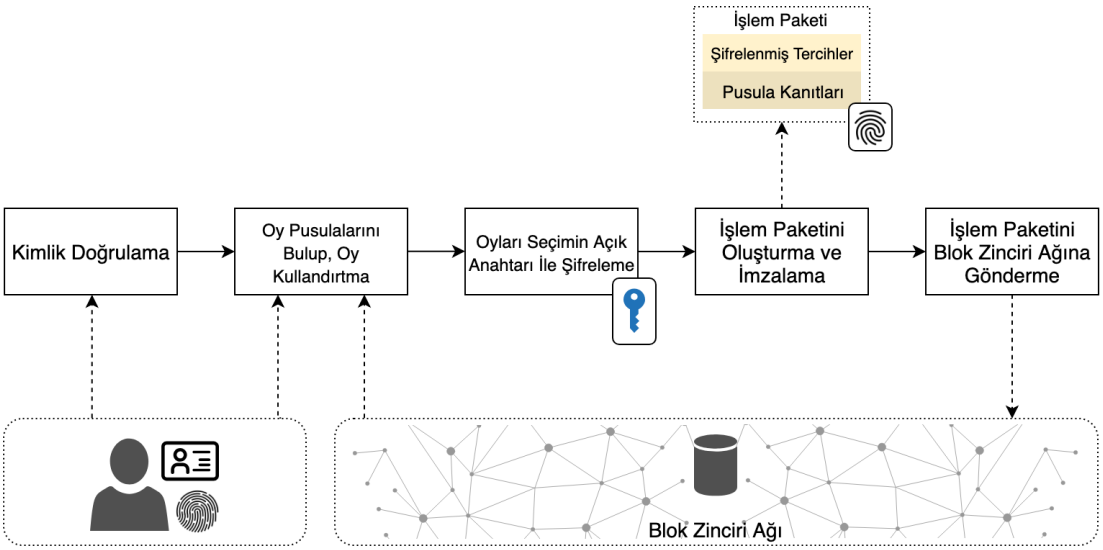
OVM'ler; ekran, bilgi girişi sağlayacak arayüz (butonlar, klavye veya dokunmatik özelliği vb.) ve yazıcı'dan oluşmaktadır. Yazıcı dışarıya herhangi bir çıktı vermez, şeffaf bir hazne içerisinde kullanılan oyları basar ve kullanıcı onayına sunar.

6.3.1 Çalışma mantığı

Seçmen oy kullanırken, OVM öncelikle seçmenin kimliğini elektronik kimlik kartı ve biyometrik veriler ile doğrular. Sonrasında OVM, seçmen adına üretilmiş pusulaları blok zinciri veritabanında bulup, seçmene sırayla gösterip, seçim yapmasını bekler.

Her kullanılan pusula sonrası sonrası, OVM seçmenin seçimini şeffaf hazne içindeki yazıcı ile kağıda basıp seçmene gösterir ve onayını talep eder. Seçmen onayladıktan sonra, kağıt şeffaf hazne içerisinden gizli hazneye doğru kayar. Yeniden oy vermeyi talep ederse kağıda oyun yeniden kullanıldığı bilgisi ile birlikte yeni oy girilir ve tekrar onaya sunulur.

OVM, yapılan seçimleri, seçimin açık anahtarı ile şifreler ve pusula kanıtı ile birlikte paketler. Bu paketi, kendi gizli anahtarı ile imzalar ve blok zinciri veritabanına gönderir. İşlem kimlik numarasını, gönderilen paketleri seçmen kimliğini ifşa etmeyecek şekilde kendi üzerine kaydeder. Sürecin özeti şekil 6.3'te görülmektedir.



Şekil 6.3 : OVM çalışma mantığı.

6.3.2 Güvenlik önlemleri

OVM'ler, fiziksel ve yazılımsal müdahalelere karşı olabildiğince güvenli olarak tasarlanmalıdır. Tüm süreçte olduğu gibi, OVM içerisindeki yazılımın da açık kaynaklı olarak geliştirilmesi gerekmektedir (Bkz.: 6.12 Geliştirme Yöntemi).

OVM'ler olası yazılım müdahalelerine karşı, güvenli önyükleme (secure boot) özelliği içermelidirler. Bu özellik sayesinde OVM, sadece güvenli bir kaynak tarafından imzalanmış yazılımı ön yükleme yaparak çalıştıracaktır. Böylece yazılıma herhangi bir şekilde dışarıdan müdahale olduğunda, OVM yazılımı yüklemeyecek ve çalışmayacaktır.

OVM'ler üzerinde açık gizli anahtar çifti üretimi, erişimi ve yönetimi olabildiğince güvenli olmalıdır. Bu anahtarların üretimi sadece cihaz içerisinde yapılmalı, herhangi bir şekilde güncelleme işlemi içermemeli sadece yeniden üretilebilmelidir. Gizli anahtara da hiçbir şekilde erişilememeli ve şifreleme, şifre çözme, imzalama işlemleri işletim sistemi seviyesinde değil, donanım seviyesinde yapılmalıdır.

Cihazlara herhangi bir şekilde fiziksel müdahale gerçekleştiğinde bu algılanmalı ve içerisindeki gizli anahtar silinmelidir. Bu tarz kasalar hali hazırda üretilmekte ve kullanılmaktadır.

Olası zafiyetlerin tespiti ve verebileceği zararları minimuma indirgemek için makineler Mercuri yöntemi kullanacak şekilde tasarlanmıştır. Mercuri yönteminde, kullanılan oylar şeffaf bir bölmede bir kağıda basılarak yazılmakta ve seçmenin onayına sunulmaktadır [41]. Bu sayede, seçmen tarafından doğrulanmış kağıt delil üretilmektedir. Bu deliller daha sonra, seçime müdahale olup olmadığını doğrulamak adına rastgele seçilip, blok zinciri içerisindeki oylarla karşılaştırılabilir.

Seçmenler oy kullanırken, verdikleri oyların etraftakiler yada sırada bekleyenler tarafından görülmemesi için, özel ekranlar kullanılabilir veya cihazın, ekranın etrafı perdeyle çevrelenebilir.

6.3.3 Makine özellikleri

OVM'ler olası elektrik kesintilerine karşı, batarya ile desteklenmelidirler. Aynı zamanda internet kesintileri ve internet olmayan bölgelerde çalışması gerekebileceği göz önünde bulundurularak, OVM'nin en azından kök blok cihaza indirildikten sonra, internetsiz çalışabilmesi gerekmektedir. OVM bunu, kullanılan oyları kendi üzerine kaydederek, internet sağlandığı zaman blok zincirine göndererek sağlayabilir.

6.3.4 Kullanıcı deneyimi

Seçmenler, farklı eğitim seviyesi ve mesleklerde olabileceğinden, arayüz olabildiğince basit tutulmalıdır. Seçmenlerin arayüzü hızlıca algılayabilmesi ve kullanabilmesi için, çoğu insanın kullandığı ATM, akıllı telefonlar vb. üzerinde bulunan kullanıcı deneyimi taklit edilebilir.

Pusulanın gösterildiği ve seçimin yapıldığı ekranın geleneksel pusulalara benzemesi, özellikle yaşlı seçmenlerin yaşayabileceği olası karışıklıkları önleyebilir.

6.3.5 Erişilebilirlik

Teknolojinin sağladığı imkanlar dahilinde; engelli seçmenlerin oylarını, yardıma ihtiyaç duymayacakları şekilde kullanabilmeleri sağlanabilir. Gizlilik ilkesinin her seçmen için geçerli olmasına azami derecede önem gösterilmelidir. Aşağıdaki erişilebilirlik seçenekleri uygulanabilir gözükmektedir:

- Ayarlanabilir ekran yüksekliği
- Yüksek konstrat modu

- Büyük yazı boyutu modu
- Kulaklık yardımıyla seçim yapılma imkanı

6.4 Hazırlıklar

Seçim hazırlıkları yapılırken, temel hedef seçim sürecinde kullanılabilmesi için gerekli olan veriyi hazırlamaktır. Bu süreçte, seçim kurulu, partiler ve seçmenlere yükümlülükler düşmektedir.

Öncelikle, geleneksel seçim sürecinde olduğu gibi, seçmen listesinin seçim kurulu tarafından oluşturulup, partiler tarafından da denetlenmesi gerekmektedir. Seçmen listesindeki seçmenlere ait açık anahtarlar da alınmalı ve kök blok üretim sürecinde kullanılmak üzere hazır bulundurulmalıdır. Seçim kurulu, yapılacak seçimleri, adayları ve ilgili resimleri önceden belirlenmiş bir formatta hazırlamalı ve partilere onaylatmalıdır.

Seçim kurulu, madencilik faaliyeti yaparak, oyları doğrulayıp sistemin tutarlılığına yardımcı olmak isteyen kişi ve kurumları, açık gizli anahtar çifti oluşturmaları için bilgilendirmeli ve daha sonra açık anahtarı, kimlik doğrulaması yaparak kayıt edecekleri bir portal sunmalıdır. Madencilik için kayıt edilen açık anahtarlar, blok zinciri kök bloğuna eklenecek şekilde hazır bulundurulmalıdır.

OVM'lerin kontrolleri ve bakımları yapılmalı, açık gizli anahtar çifti yenilenmeli ve kök blok üretimi sürecinde kullanılmak üzere, açık anahtarlar toparlanmalıdır.

6.5 Kök Blok Üretimi

Bu süreçte, seçimin açık gizli anahtar ikilisi, elektronik oy pusulaları ve pusula kanıtları üretilir. Üretilen veriler, blok zincirinin temelini oluşturmak üzere, kök blok içerisine eklenir. Seçim gizli anahtarının veya seçmenler ile pusula kanıtları arasında ilişkilerin kaydedilmesi başta belirtilen tasarım ilkelerine zarar vereceğinden, süreç gözetim içerisinde, önceden yayınlanmış yazılımlar ile yürütülmelidir. Üretim süreci sırasında, seçim kurulunun yönetici konumda, siyasi parti ve ilgili STK'ların temsilcilerinin de gözlemci ve denetleyici konumda olması gerekmektedir.

Gözlemcilerin ekseriyetle bilişim uzmanlarından oluşması yararlı olacaktır. Gözlemcilerin süreç içerisinde gözetmesi ve kontrol etmesi gerekenler şöyle listenebilir:

- Kullanılan donanımların güvenilir olduğundan emin olmak
- Kullanılan donanımların herhangi bir şekilde ağ veya herhangi bir başka bilgisayarlarla iletişiminin olmadığını belirlemek
- Donanıma yapılan tüm yazılım kurulumlarının açık kaynak, değiştirilmemiş yazılımlar olduğunu tespit etmek
- Kök bloğun doğruluğunu kontrol etmek

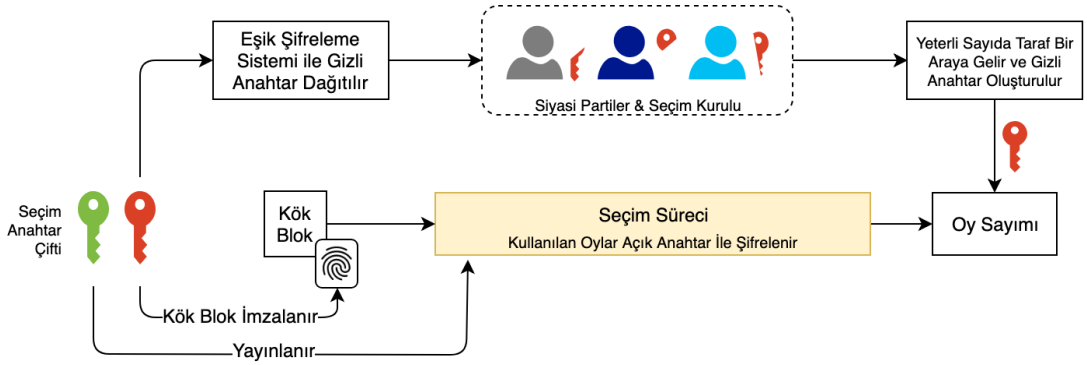
Süreç sonunda, kök blok içeriğinde; seçim listesi, pusula kanıtları, şifrelenmiş pusulalar, madencilerin açık anahtarları ve OVM'lerin açık anahtarları bulunmaktadır. Oluşturulan kök blok, blok zinciri veritabanının ilk bloğunu oluşturacak şekilde ağ üzerinden yayınlanmaya başlanır.

6.5.1 Seçim açık gizli anahtar çifti

Seçim sürecinde kullanılmak için bir açık gizli anahtar çifti gerekmektedir. Açık anahtar seçim öncesinde, gizli anahtar ise seçim sonrasında yayınlanır. Bu anahtar çifti ile kök blok doğrulaması, verilen oyların gizlenmesi ve seçim sonuçlarının vaktinden önce bilinmesinin engellenmesi sağlanır.

Anahtarlar, kök blok üretimi başında üretilerek, üretilen kök bloğun da gizli anahtar ile imzalanması sağlanır. Aynı zamanda üretilen açık anahtar, OVM'lere girilmek üzere kaydedilir ve aygıtların ağa göndermeden önce, yapılan tercihi bu anahtarla şifrelemesi sağlanır. Böylece sadece gizli anahtar yayınlandığı zaman oylar sayılabilecektir.

Gizli anahtarın sadece seçim kurulunda bulunması, kök bloğun sonradan değiştirilmesi, oyların önceden sayılması gibi riskler oluşturmaktadır. Kök blok üretimi sürecinin sonunda, gizli anahtar, eşik şifreleme sistemi (Bkz.: 3.4 Eşik Şifreleme Sistemi) kullanılarak parçalanması bu riskleri en aza indirebilir. Anahtar parçalandıktan sonra, parçalar seçim kurulu ve siyasi parti temsilcilerine dağıtılır, seçim kurulu veya siyasi partiler, parçaların belirlenen adetini bir araya getirmedikçe, gizli anahtarı tekrar oluşturamayacak ve süreç bitmeden sonuçları bilemeyeceklerdir. Anahtar çiftinin, sistemde kullanım diyagramı şekil 6.4'te görülebilir.



Şekil 6.4 : Seçim açık gizli anahtar kullanım diyagramı.

Partilerin veya seçim kurulunun seçimi bloke etmemeleri için gizli anahtar parçalarının dağıtımında, seçimin koşulları göz önünde bulunarak düzenlemeler yapılabilir. Örneğin, 3 partinin bulunduğu bir seçimde, gizli anahtarın tekrar elde edilmesi için en az 3 parçanın bir araya gelmesi şartıyla, anahtar 5'e bölünebilir. Bu durumda gizli anahtarın 2 parçası seçim kuruluna verilerek, seçim kurulunun tek bir partiyle iş birliğine giderek, gizli anahtarı tekrar elde etmesi sağlanabilir. Böyle bir senaryoda, sadece partiler de bir araya gelerek, seçim kurulundaki parçalara ihtiyaç duymadan, gizli anahtarı tekrar üretebilir.

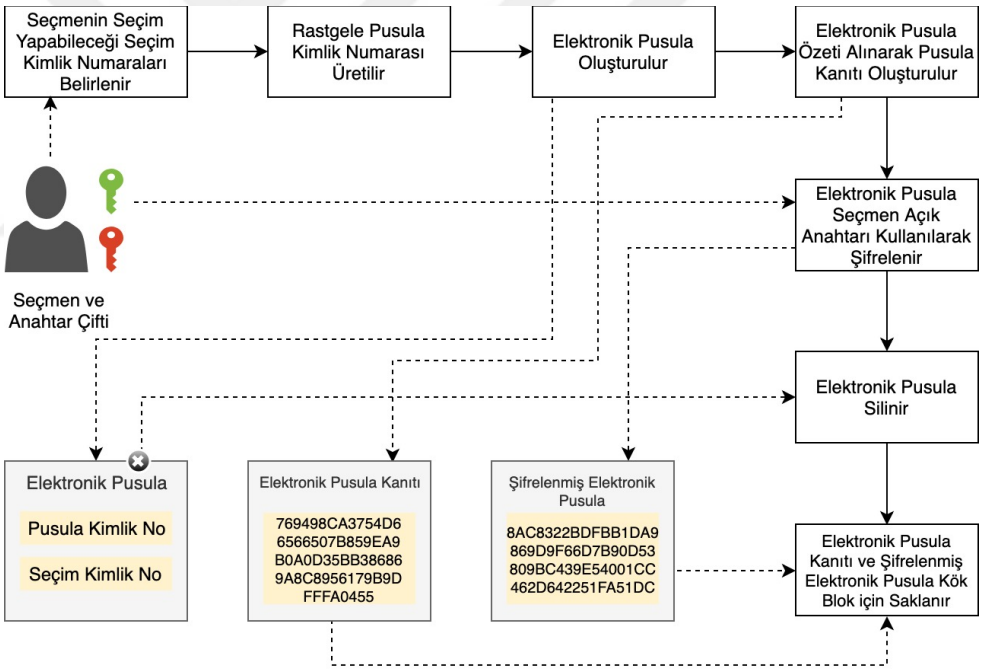
6.5.2 Elektronik oy pusulaları ve pusula kanıtları

Kök blok üretimi sürecinde, her bir seçmen için oy kullanabileceği her seçime birer elektronik pusula, her bir pusula için de madencilerin doğrulama yapmasını sağlayacak birer kanıt üretilir ve bu veriler kök bloğa eklenir.

Üretilen pusula içerisinde; pusula kimliği ve seçim kimlik numarası bulunmalıdır. Pusula kimliği, rastgele üretilmiş karakter dizisi olarak oluşturulur. Bu sayede pusulanın tahmin edilmesi güç verisi olur ve dolayısıyla taklit edilmesinin önüne geçilir. Seçim kimlik numarası ise pusulanın hangi seçim için üretildiğini gösterir, bu sayede OVM'ler seçmene ilgili adayları sunar.

Her seçmen için üretilen pusula, aynı seçmenin açık anahtarı ile şifrelenir. Bu sayede bu pusulayı sadece seçmen çözüp, bilgileri kanıt olarak sunup oyunu kullanır. Burada kullanılan asimetrik algoritmanın, aynı açık anahtar ve aynı metinde aynı çıktıyı vermemesi, rastgelelik içermesi gerekmektedir. Bu sayede sürecin sonunda, seçim kurulunun yada seçmenlerin açık anahtarlarını bulunduran ilgili devlet kurumlarının, tüm ihtimalleri deneyerek, hangi seçmenin hangi adaya oy verdiğini öğrenmesinin önüne geçilir.

Pusula kimliği ve seçim kimlik numarasını barındıran, pusula içeriklerinin özeti (hash) alınır ve bu özet de pusula kanıtı olarak kök bloğa eklenir. Kök bloğa eklenen bu kanıt ile madenciler doğrulama yapabilir. Bu sayede madencilerin pusula ve seçmen hakkında herhangi bir bilgi sahibi olmadan, açık bir şekilde doğrulama yapabilmesi sağlanır. Tek bir seçmen için, elektronik oy pusulası üretim süreci şekil 6.5'te görülebilmektedir.



Şekil 6.5 : Elektronik oy pusulası üretimi.

6.5.3 OVM'lerin açık anahtarları

Blok zincirine, oy gönderiminin sadece OVM'ler tarafından yapıldığının garanti edilmesi için, OVM'lerde açık gizli anahtar çifti üretilir ve OVM'ler tarafından gönderilen oylar, OVM'lerin gizli anahtarları ile imzalanır.

Madencilerin imzanın, geçerli bir OVM'ye ait olup olmadığını belirleyebilmesi için, makinelerin açık anahtarları kök blok içerisine eklenir. Böylece madenci, gelen oyun geçerli bir OVM üzerinden gönderildiğini doğrulayabilir.

6.6 Seçim Süreci

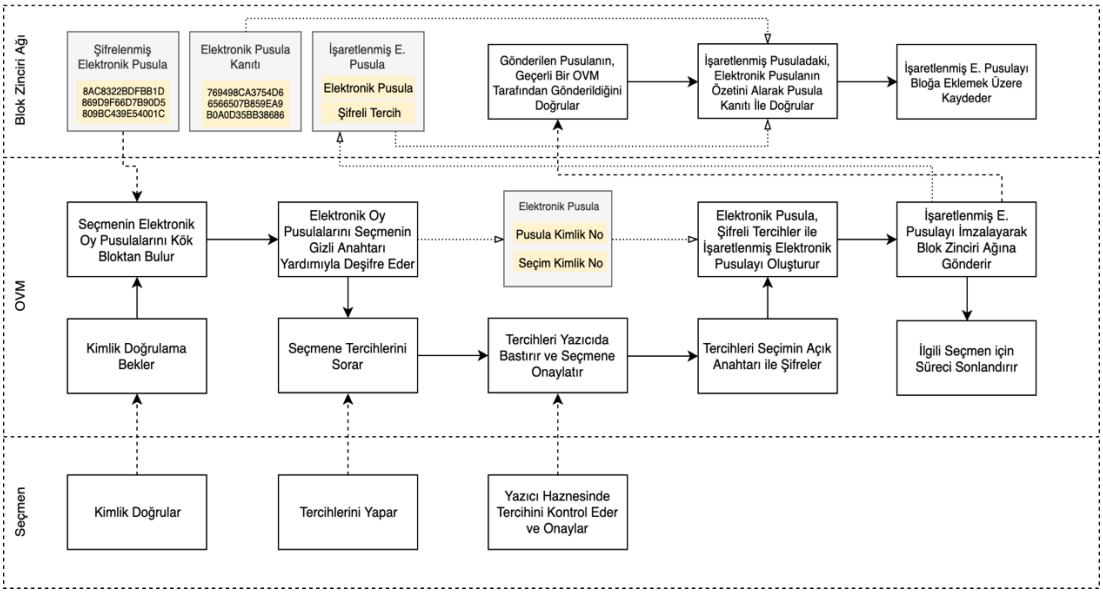
Seçim sürecinden önce, OVM'lere ilgili seçimin açık anahtarı girilmeli ve sadece kök bloğu içeren blok zinciri veritabanını ağdan indirmesi sağlanmalıdır. Bu işlemler tamamlandıktan sonra, OVM'ler oy kullanımına hazır hale gelmiş olur. İlgili merkezlere yerleştirilen OVM'ler üzerinden, seçmenler oy kullanırlar.

Seçmenler, seçim süreci için belirlenen zaman aralıklarında, OVM'lerin bulundukları merkezlere giderler. Elektronik kimlik kartlarını okutup, önceden belirlenen ekstra kimlik doğrulama metoduyla (biyokimlik, PIN, yüz tarama vb.) kimliklerini doğrularlar.

Kimlik doğrulamasından sonra, OVM blok zincirinden ilgili seçmenin tercihini belirtmesi gereken şifreli pusulaları bulur. Seçmenin gizli anahtarıyla, şifreli pusulayı çözer. Pusula içerisindeki seçim kimlik numarasından, ilgili aday listesini getirir ve seçmene, seçim yapması için sunar.

Seçmen aday listesinden seçimini yaptıktan sonra OVM, daha önce de belirtildiği gibi, seçmenin tercihini şeffaf hazne içerisindeki yazıcıdan basar ve ekranda gösterdiği uyarıyla seçmene kağıt üzerindeki yazıyı kontrol ettirir. Seçmen oyunu kontrol ettikten sonra, ekran üzerinden onaylar. Tüm elektronik pusulalar için bu süreç tekrarlandıktan sonra, seçmen için oy verme işlemi tamamlanmış olur.

OVM bu süreçte, "6.3.1 Çalışma mantığı" bölümünde ayrıntılı bahsedildiği gibi, oyu seçimin açık anahtarıyla şifreler ve pusula kanıtıyla bir paket oluşturur. Bu paketi de kendi gizli anahtarıyla imzalayıp blok zinciri veritabanına gönderir. Gönderdiği tüm paketleri, seçmenle ilişkili herhangi bir veri tutmadan kaydeder. Sürecin özeti şekil 6.6'da gözükmektedir.



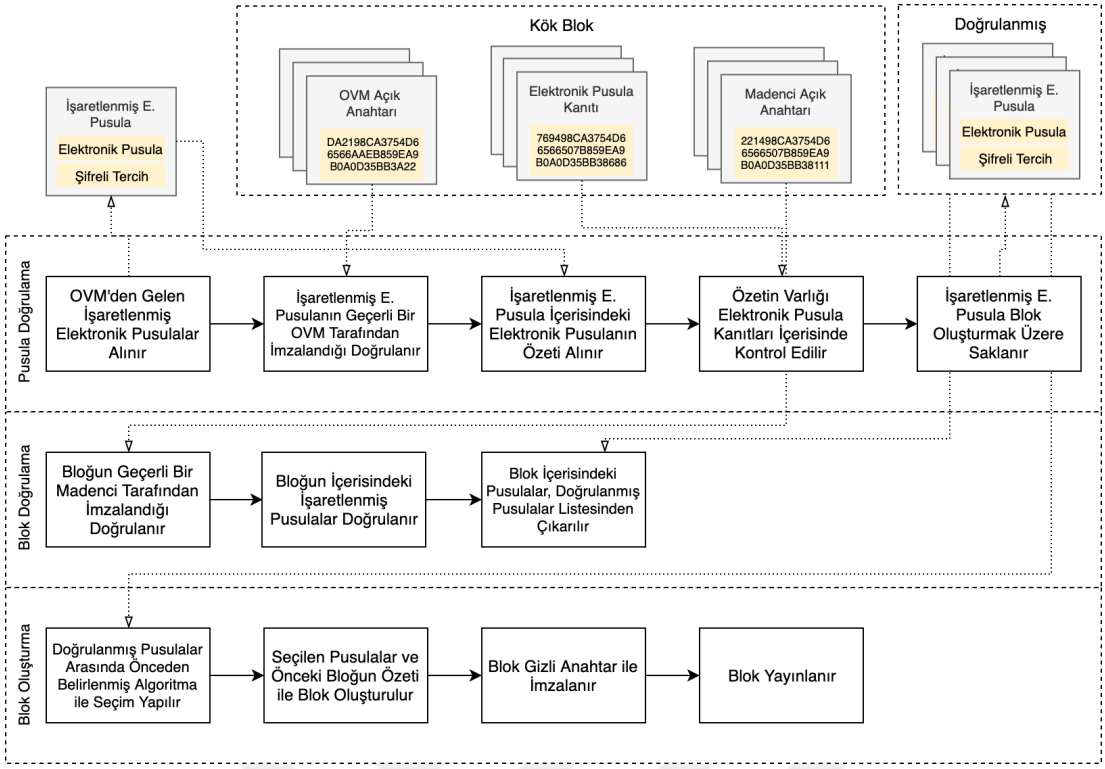
Şekil 6.6 : Seçim süreci özeti.

6.7 Madencilik Faaliyeti

Gönüllü kişi ve kurumlar, seçim süreci içerisinde, aygıtlarına indirdikleri açık kaynak bir yazılımla madencilik faaliyeti yürütür ve seçim sürecinin doğru bir şekilde işlemesine bu şekilde katkıda bulunurlar. Bu yazılımlara, daha önceden oluşturup açık anahtarı kayıt yaptırdıkları anahtar çiftinin, gizli anahtarını ve seçimin açık anahtarını girerek, madencilik faaliyetlerine başlarlar. Madenciler isteğe bağlı olarak kendi oluşturdukları yazılımlar ile de madencilik faaliyeti yürütebilirler.

Madenci, blok zinciri ağını dinler, gönderilen işaretlenmiş pusulaları toplar. Aynı zamanda toparladığı işaretlenmiş pusulaları doğrulayarak ve devamlı olarak yeni oluşturulan bloklarda kontrol ederek, işaretlenmiş pusula listesinin hiçbir bloğa dahil edilmeyen paketlerden oluşmasını sağlar.

Madenci, blok oluşturma yetkisi kendisine verildiğinde, işaretlenmiş pusula listesinden rastgele seçim yapar, belirlenen niteliklerde bir blok oluşturup, madencinin açık anahtarıyla imzalar. Yeni bloğu, geçerli zincire ek olarak oluşturduğundan emin olur. Genel madencilik faaliyeti özeti şekil 6.7'de görülebilmektedir.



Şekil 6.7 : Madencilik faaliyeti özeti.

Madencilik faaliyeti, daha önce belirtildiği gibi, kimlik bilgileri ve açık anahtar girilerek yapılan bir kayıt işlemi gerektirir. Bu sebeple, madencilerin yaptıkları tüm faaliyetler, kimlikleriyle ilişkilendirilebilir olur. Kimlik bilgisi, kötü niyetli madenciler için caydırıcı bir kayıttır. Kötü niyetli madencilerin bulunması sistemde gecikmelere neden olabilse de, veritabanında hatalı verilere ve seçimde hile yapılabilmesine olanak vermez.

6.7.1 İşaretlenmiş pusulaların doğrulanması

Madenciler kendilerine ulaşan her işaretlenmiş pusulayı doğrularlar. Öncelikle işaretlenmiş pusulanın, yetkili bir OVM tarafından imzalandığını, kök blok içerisindeki listeden kontrol ederler. Sonrasında işaretlenmiş pusulanın içerisinde yer alan kanıt verisinin özetinin, kök blokta yer aldığını doğrularlar.

6.7.2 Blokların doğrulanması

Önceki blokların doğrulanması, geçerli bir blok zinciri oluşturmak adına gereklidir. Burada yapılması gereken doğrulanan blok içindeki pusulaların doğrulanması ve bloğun da geçerli bir madenci tarafından imzalandığının belirlenmesidir. Bunun için madencinin açık anahtarının, kök blokta olup olmadığı kontrol edilir.

6.7.3 Blok oluşturulması

"4. Blok Zinciri" başlığında değinildiği gibi, blok zincirinde fikir birliği, hisse ispatı (Bkz.: 4.1.3.2 Hisse İspatı) benzeri bir yöntemle yapılacaktır. Bu yönteme göre bloğun işlenmesi için belirli bir madenciye yetki verilecek ve yetkilendirilen madenci blok oluşturup imzalayarak blok zincirine ekleyecektir.

Madenci bloğu oluştururken, elindeki işaretlenmiş pusula listesinden rastgele pusulalar seçecek ve bunlarla, önceden belirlenmiş limitlere uygun olarak, blok oluşturacaktır. Bu bloğun içerisine aynı zamanda önceki blok başlığının özeti, imzalayanın OVM'nin bilgisi bulunacaktır.

Madenci blok içeriğini oluşturduktan sonra, gizli anahtarıyla imzalayıp, ağda yayınlayarak diğer madenciler ve dinleyicilere haber verecektir.

6.7.4 Madenciliğe teşvik

Madencilik, tasarlanan seçim sürecinde oldukça önemli bir faaliyettir. Tasarlanan sistem için başta değinilen, seçim kurullarını merkezi konumdan, gözlemci konuma kaydırmayı sağlayan en önemli parçalardan biridir.

Sistemin dağınık ve şeffaf olarak yürütülmesine yardımcı olmakta aynı zamanda vatandaşların, aygıtlarına yükledikleri bir yazılım ile bile olsa, süreç içerisinde aktif rol oynamalarını sağlamaktadır. Bu yönüyle, geleneksel seçim sistemlerindeki sandık başında gözlemci olma, oyların sayılması sürecine katılma vb. faaliyetlere benzemektedir.

Madencilik faaliyetinin, önemi gereği ve özellikle yeterli madenci kayıt olmadığı durumlarda, teşvik edilmesine ihtiyaç duyulabilir. Özellikle özel kurum ve kuruluşların, madencilik faaliyetini reklam kanallarında ve sosyal medya faaliyetlerinde, kendilerinin de parçası olduklarını belirterek bildirimde bulunması, bu yönde bilinç oluşmasına katkı sağlayacaktır. Seçim kurulu, madencilik faaliyetinde bulunan ve belirli sayıda blok işlemiş madencilere ödül vadetmesi de madencilik faaliyetine duyulan ilgiliyi arttıracaktır.

6.8 Sayım Süreci

Oy verme süreci sona erdikten sonra, seçim kurulu ve partiler ellerindeki seçimin gizli anahtarı parçalarını yayınlarlar. Bu parçalarla, eşik şifreleme sistemi (Bkz.: 3.4 Eşik Şifreleme Sistemi) kullanılarak, gizli anahtar yeniden oluşturulur ve yayınlanır. Elde edilen gizli anahtar ile şifrelenmiş oylar çözülüp, sayım işlemi yapılabilir.

Sayım süreci için iki farklı yaklaşım kullanılabilir. Birinci yaklaşımda; herkes isteyen kendi imkanları dahilinde oyları çözer ve sayar. İkinci yaklaşımda ise, sayım işlemi blok zinciri ağı içerisinde yapılır.

6.8.1 Müstakil sayım

Bu yaklaşımda, fazla oyun kullanıldığı durumlar için oy sayımı oldukça uzun sürebilir. şekil 6.8'de günümüz için iyi bir işlemci sayılabilecek, Intel i5 2.7 GHz ile yapılan RSA hız testi süreleri gözükmektedir. Bu veriler baz alınıp, 1024 bit'lik bir anahtar çifti kullanıldığı varsayıldığında, 10 milyon oyluk bir seçimde, oyların tek bir aygıt ile deşifre edilmesi yaklaşık 15 saat sürmektedir.

```
→  ~ openssl speed rsa -decrypt
Doing 512 bit private rsa's for 10s: 11287 512 bit private RSA's in 9.89s
Doing 512 bit public rsa's for 10s: 104571 512 bit public RSA's in 9.85s
Doing 1024 bit private rsa's for 10s: 1852 1024 bit private RSA's in 9.73s
Doing 1024 bit public rsa's for 10s: 21876 1024 bit public RSA's in 9.75s
Doing 2048 bit private rsa's for 10s: 283 2048 bit private RSA's in 9.87s
Doing 2048 bit public rsa's for 10s: 4994 2048 bit public RSA's in 8.63s
Doing 4096 bit private rsa's for 10s: 38 4096 bit private RSA's in 9.22s
Doing 4096 bit public rsa's for 10s: 1635 4096 bit public RSA's in 9.89s
LibreSSL 2.6.5
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
                  sign    verify    sign/s verify/s
rsa  512 bits 0.000876s 0.000094s   1141.3  10616.3
rsa 1024 bits 0.005254s 0.000446s    190.3   2243.7
rsa 2048 bits 0.034876s 0.001728s     28.7    578.7
rsa 4096 bits 0.242632s 0.006049s      4.1    165.3
```

Şekil 6.8 : RSA hız testi sonuçları.

Sayımın makul bir sürede bitirilebilmesi için, yüksek işlem gücü veya paralel çalışma gerekebilir. Bu da sayımın herkes tarafından yapılabilir ve doğrulanabilir olmasının önüne geçer. Bu yaklaşımda, partiler, STK'lar, seçim kurulu sayımı makul sürede bitirmeye muktedirlerdir. Ancak bunun, özellikle yüksek oy sayısının bulunduğu seçimlerde, seçmen seviyesinde yapılması pek mümkün gözükmemektedir.

6.8.2 Madenciler ile sayım

Sayım işlemi, oyların toparlanması ve doğrulanması sürecine destek veren madenciler ile de yapılabilir. Seçim kurulu, elde ettiği seçim gizli anahtarını, yine seçim gizli anahtarıyla imzalayarak blok zincirine ekler. Burada olası karışıkları önlemek adına parti temsilcilerinin, seçim gizli anahtarı parçalarını kamuoyu ile değil, direk seçim kuruluyla paylaşması gerekmektedir. Bunu ağı izleyen madenciler, bir işaret olarak kabul edip sayım işlemini başlatırlar.

Bu yaklaşımda, üretilen blokların tüm madenciler arasında aynı anda paylaştırılmasının ve bir bloğun birkaç farklı madenci tarafından deşifre edilip sayılmasının sağlanması gerekmektedir.

Her madenci, bloklar arasından belirli bir algoritmaya göre seçim yaparak blokları deşifre eder. Bu sırada deşifre ettiği blokların sonuçlarını ağda yayınlar. Gelen yayınları da dinleyerek kaydeder ve gelen aynı blok sonuçlarını da karşılaştırarak doğrular. Tüm blokların sonuçlarına, her blok için önceden belirlenmiş doğrulama adetine ulaşan madenci, tüm sonuçları toparlar ve seçim sonuçlarını bir blok olarak, gizli anahtar bloğunun hemen ardına imzalayıp ekler.

Seçim sonuçları blok olarak eklendikten sonra, diğer madenciler de kendilerinde bulunan bilgileri toparlayarak son bloğu doğrularlar ve aynı blok başlığının özetini imzalayarak son bloğa eklerler. Eğer ulaştıkları sonuçlar, imzalanan blok ile örtüşmüyorsa, yeni bir blok oluşturup, imzalayarak; gizli anahtarın yayınlandığı bloktan dallanma yaparlar. Bu durumda, diğer madenciler kendi hesaplamalarıyla uyuşan bloğu imzalalarlar ve o bloğun ardına eklerler.

En çok madenci tarafından onay alan blok, sonuç olarak kabul edilebilir. Sayım sonuçları aynı zamanda, yüksek işlemci gücüne sahip kurumlar tarafından, müstakil olarak da kontrol edilebilir.

6.9 Müdahele Kontrolü

Elektronik sistemler her ne kadar tüm endüstrilerde güvenilip kullanılsa da her gün kullanılan sistemlerin bir şekilde açıkları bulunmakta ve sistemler zafiyete uğramaktadır. Bu sebeple, önem derecesi yüksek sistemlerde, sistem harici kontrol yapıları oluşturulmaktadır.

Tasarlanan elektronik seçim sisteminde de bu durum gözetilmiş ve Mercuri yönteminin kullanılması ve verilen oyların, kullanıldıkları cihazlarda çıktı alınarak saklanması düşünülmüştür. Olası bir müdahalede, çıktı alınarak kaydedilen oylar ile OVM'den blok zincirine gönderilen oylar bir birini tutmayacağı için, müdahale anlaşılabilmektedir.

Bu süreçte seçim kurulu tarafından rastgele OVM'ler seçilir ve/veya siyasi partilerden de kontrol edilmesi için OVM'ler seçmeleri istenir. Seçilen OVM'lerde çıktı alınan oylar sayıldıktan sonra, blok zincirine gönderilen oylar ile karşılaştırılıp, sürecin sağlıklı bir şekilde ilerleyip ilerlemediği belirlenir.

6.10 Farklı Kimlik Doğrulama Yöntemleri

Kimlik doğrulama süreçleri için uygulanan uluslararası bir standart bulunmamakta, ülkeler, farklı kimlik doğrulama alt yapılarını kullanabilmektedir. Bu sebeple tasarlanan sistemin, farklı kimlik yapılarına uyarlanabilmesi önemlidir. Tasarımda varsayılan olarak tercih edilen kimlik doğrulama yöntemi, sunucu doğrulaması gerektirmeyen, elektronik kimlik kartı ile gerçekleştirilen kimlik doğrulama yöntemidir. Farklı kimlik yöntemlerinin kullanılması, süreç ve tasarım üzerinde bazı güncellemelerle mümkün olabilmektedir.

6.10.1 Elektronik kimlik kartı ile çevrimiçi doğrulama

Elektronik kimlik kartının çevrimiçi doğrulama yapması, doğrulama yapan kurumun, verilen oy ile yapılan doğrulama vaktini karşılaştırarak, seçmenin gizliliğini ihlal edilmesine sebep olabilir. Bu bilgiye sahip olan kurumun, madenci olması ve gelen oyların zamanlarını kayıt altına alması da söz konusu olabilir.

Bu durumun önüne, oyların blok zincirine gönderilmeden önce OVM üzerinde bekletilip, karıştırılmasıyla geçilebilir. Böylece kötü niyetli madenciler gelen oyların zaman bilgilerini kaydetseler bile, oyun gerçekten kullanıldığı vakti bilemeyeceklerdir.

6.10.2 Elektronik olmayan kimlik kartı

Elektronik olmayan kimlik kartlarının süreç içerisinde kullanılması, tasarlanan sistemde kimlik doğrulaması ve her seçmen için pusula basılması sürecini sorunlu bir hale getirmektedir. Bu tarz kimlik doğrulama yönteminde, elektronik pusula üretim sürecinde her seçmen için gizli anahtar da üretilmesi gerekmektedir.

Burada dikkat edilmesi gereken husus, toplu ve yetkisiz oy kullanımlarını önlemek adına, üretilen gizli anahtarların, ilgili seçmen haricinde kişiler tarafından kullanılabilir olmamasını sağlamaktır. Bu gizli anahtarların seçmene direk olarak ulaştırılması ile sağlanabilir. Ancak bu sürece ek yük getirecek ve gizli anahtarların ulaşmaması durumunda, bu anahtarların yeniden üretilmesi, geçersiz kılınması gibi işlemlerin sürece dahil edilmesi ihtiyacını doğuracaktır.

Bu sebeple, seçmenin gizli anahtarının, seçmenin kendisinin erişebileceği bilgiler ile (kart üzerindeki numara bilgileri, anne kızlık soyadı vb.) şifrelenip blok zincirine eklenmesi en makul çözüm olmaktadır. Böylece seçmen OVM başında kendisinden istenen bilgileri cihaza girerek, gizli anahtarına erişebilir. Buna ek olarak, OVM başında bir kurulun yada resmi yetkilinin kimlik kontrolü yapması, kimlik doğrulama sürecine katkı sağlayacaktır.

6.11 OVM'siz Kullanım

Tasarlanan sistem, elektronik kimlik kartının başarılı bir şekilde kullanıldığı kimlik doğrulama yöntemleriyle, OVM'lere ihtiyaç duymadan kullanılabilir. Ancak bunun baskı ile oy kullanımını arttırabileceği göz önünde bulundurulmalıdır.

Teknik olarak yapılması gereken, OVM'lerin açık gizli anahtarlarının süreç içerisinden ve madencilerin doğrulama sürecinden çıkarılmasıdır. Bunların dışında, bir seçmen kanıtını ağda yayınladığında, kötü niyetli kişilerin aynı kanıtla farklı oylar kullanmasının önüne geçilmedilir. Bu durum, mükerrer oya müsaade edilmeyip sadece ilk oy sayılarak ve kullanılan oyun madenciler tarafından zaman damgası ile imzalanarak ağda yayılmaya devam etmesi ile sağlanabilir. Diğer bir alternatif de sistemin, elektronik pusula içerisine, gizli anahtar yerleştirilecek ve açık anahtarın blok zincirine eklenecek şekilde düzenlenmesi olabilir. Böylece seçmen pusulasını deşifre edip, kullandığı oyu gizli anahtar ile imzalayabilir. Madenciler de açık anahtar ile bunu kontrol edebilir.

Bunlar sağlandıktan sonra seçmenler, elektronik kimlik kartı okuma aygıtlarını kullanarak, açık kaynak oy kullanma yazılımları aracılığıyla, oylarını kendi bilgisayarları üzerinden kullanabilirler.

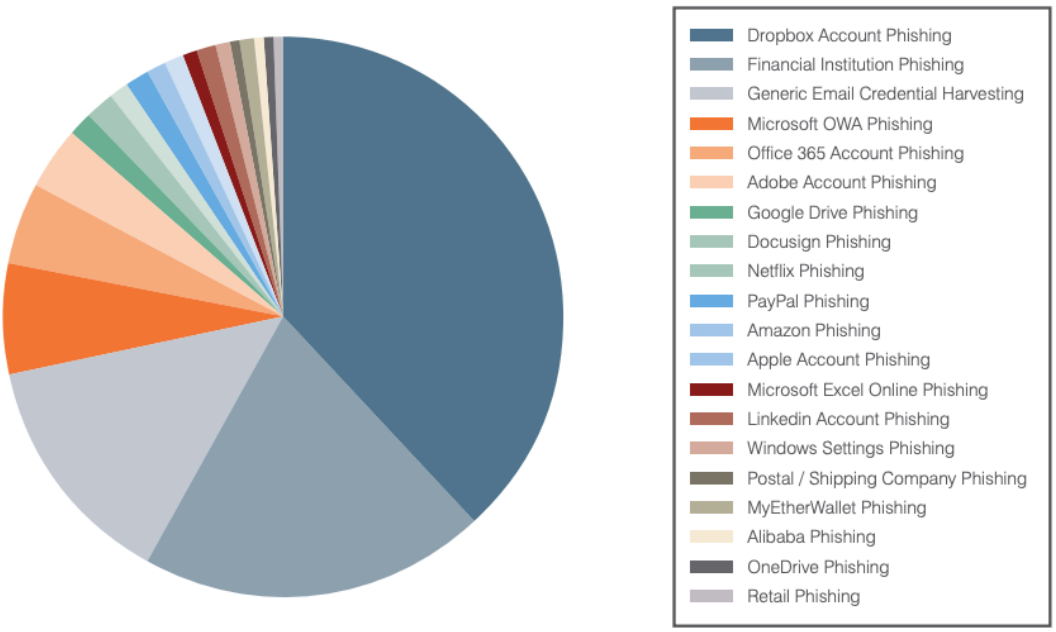
6.12 Geliştirme Yöntemi

Sistemin temel hedefi, seçim sürecini şeffaf bir şekilde otomatize etmektir. Bu amaç etrafında, kullanılan tüm yazılım ve donanımların da açık kaynak olarak geliştirilmesi ve kamuoyu ile paylaşılması yerinde olacaktır.

Açık kaynağın, kodun ve yapısının incelenmesine müsaade etmesi sebebiyle, kullanıcılarına güven vermektedir. Her daldan bilişim uzmanının kontrolüne açık olması, olası kötü niyet şüphelerini ortadan kaldırır.

Açık kaynak geliştirme ile ilgili bazı endişeler de bulunmaktadır. Bunların başında, kaynağın ve yapının açık olmasının bazı güvenlik sorunlarına yol açacağı düşüncesi gelmektedir. Kaynağı kapatmanın, sistemde bulunan açıkların ortaya çıkmasını engelleyeceği ve saldırıları önleyeceği düşüncesi çoğu zaman geçerli değildir. Guido 2009 yılında açık ve kapalı kaynaklı yazılımlarda güvenliği incelendiği araştırmasında, 6 incelemenin 3'ünde açık kaynak daha güvenli bulunmuş, diğer 3 durumda sonuç iki geliştirme yöntemi için aynı çıkmıştır [42].

Bir diğer endişe, açık kaynağın, oluşturulan yapının kolayca taklit edilerek, oltalama saldırılarına alet edilme ihtimalidir. Tasarlanan seçim sisteminin, sadece birkaç arayüzden oluşmadığı, OVM'lerle birlikte oltalama saldırısı yapılması oldukça zor olan bir durumda olduğu göz önünde bulundurulmalıdır. Şekil 6.9'da bulunan 2017 yılı oltalama saldırıları oranları incelendiğinde, kapalı kaynak kullanımının, oltalama saldırılarını zorlaştırmadığı ve önüne geçmediği görülebilmektedir. Oltalama saldırılarına hedef olmada en büyük paya sahip Dropbox, kapalı kaynak geliştirme yapmakta, finans kurumları ve email sistemlerinin de kaynaklarının çoğu kapalı tutulmaktadır. Kapalı kaynak kullanımının oltalama saldırılarının önüne geçmediği, arayüzlerin günümüz şartlarında kolay ve hızlı bir şekilde taklit edilebildiği bilinmektedir.



Şekil 6.9 : 2017 yılı oltalama saldırıları tuzak türleri. [43]

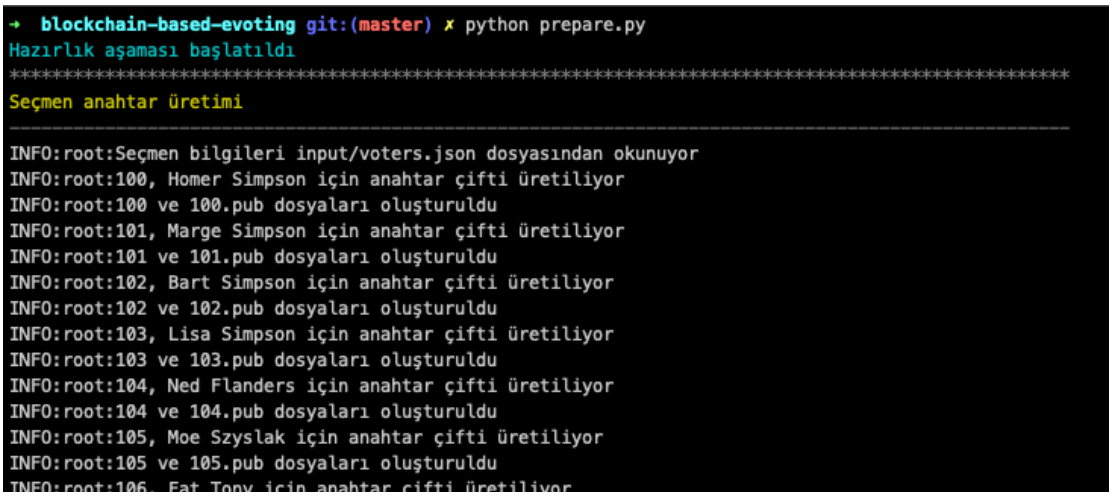
Açık kaynak ile ilgili bir diğer endişe de lisans ve patent sorunlarına dayalı ticari endişelerdir. Kaynağını açık tutarak, ciddi değerlenme ve gelire ulaşmış birçok şirket bulunmaktadır. Bu şirket ve ürünlere örnek olarak MySQL, PostgreSQL, Docker, Nginx, MongoDB vb. gösterilebilir. Microsoft, Google, Facebook, Netflix vb. bir çok şirketin de kodlarının bir bölümünü açık kaynak olarak yayınladığı göz ardı edilmemelidir.

7. UYGULAMA

Çalışma kapsamında, sistemin kısmi bir uygulaması, Python dili kullanılarak geliştirilmiştir. Sistemin tamamen uygulamaya geçirilmesinin bu çalışma dahilinde mümkün olmaması nedeniyle, blok zinciri ağı ve OVM'ler uygulama içerisine dahil edilmemiş, birer sınıf ile temsil edilerek, sorumlulukları kod içerisinde uygulanmıştır. Kodun tamamı EK A.2'da görülebilir.

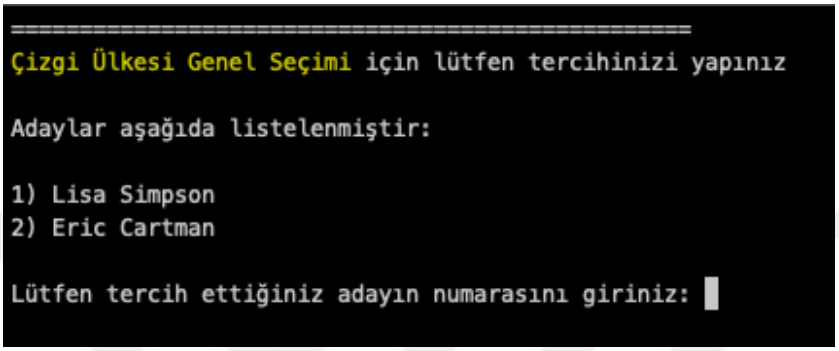
Uygulama "input" klasöründe bulunan verilere göre sistemi hazırlamakta, oy kullanılmasına müsaade etmekte daha sonra da verilen oyları sayıp sonuçları "output" klasörüne kaydetmektedir. Uygulama ile ilgili temel ayarlamalar "app/config.py" dosyasındaki sabitler ile yapılabilmektedir. "Input" klasöründe "election.json" dosyası seçim bilgilerini, "miners.json" madenci bilgilerini, "vcms.json" dosyası OVM bilgilerini, "voters.json" dosyası ise seçmen bilgilerini içermektedir.

Uygulama içerisinde ana dizinde bulunan "prepare.py" dosyası çalıştırılarak, daha önce belirtilen hazırlık aşamaları gerçekleştirilmektedir. Bu dosya ile "input" klasöründe bilgileri verilen seçime, seçmenlere, madencilere ve OVM'lere ait anahtar çifti üretimi yapılmakta, sonrasında ise kök blok üretilmektedir. Şekil 7.1'de dosyanın çalıştırıldıktan sonra oluşturduğu çıktı görülebilmektedir. Oluşturulan tüm anahtarlar ve kök blok, "data" klasörü içerisine kaydedilmektedir.



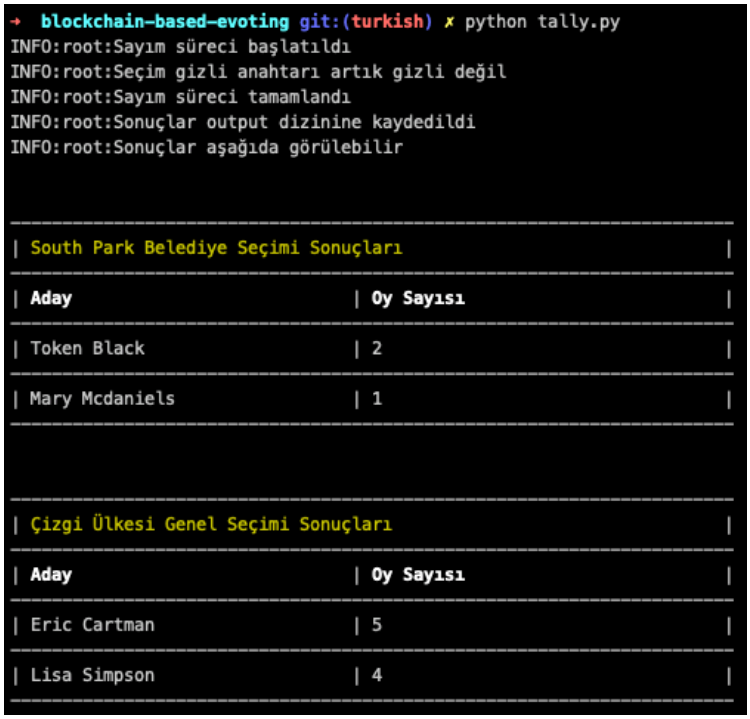
Şekil 7.1 : Uygulama hazırlık aşaması çıktısı.

Ana dizinde bulunan “vote.py” ise, çalıştırıldıktan sonra seçmen kimliği istemekte, verilen numaraya göre seçmen adına oluşturulmuş elektronik seçim pusulalarını, kök blok içerisinden bulup, seçmenin gizli anahtarıyla deşifre etmektedir. Deşifre işleminden sonra, seçmene seçimlerini sormakta ve her seçimi pusula kanıtıyla birlikte imzalayarak, madenci sınıfına göndermektedir. Madenci sınıfı ise her oyu kontrol edip, belirli sayıda oy gelince, blok oluşturup, imzalayıp blok zincirine dahil etmektedir. Şekil 7.2’de “vote.py” dosyasının çıktısı görülebilmektedir.



Şekil 7.2 : Uygulama oy verme aşaması.

Sayım sırasında, seçim anahtarının yayınlandığı varsayılır. Ana dizinde bulunan “tally.py” dosyası ile verilen oylar seçim gizli anahtarı ile deşifre edilip, sayılır. Sayım sonucu “output/results.json” dosyasına kaydedilir ve aynı zamanda ekrana tablo olarak yazdırılır. Şekil 7.3’te sayım sonuç tablosu görülebilmektedir.



Şekil 7.3 : Uygulama sayım çıktısı.

8. SONUÇ VE ÖNERİLER

Çalışma kapmasında, blok zinciri tabanlı elektronik seçim sistemi, belirlenen ilkeler doğrultusunda, mimari olarak kurgulanmış; algoritma seçimi, veri yapıları gibi detaylara girilmemiştir. Sistem tasarlanırken; finans, askeri, istihbarat vb. kritik alanlarda güvenilerek kullanılan şifreleme algoritmaları ve yöntemleri tercih edilmiştir.

Sistem, çalışması ve sürdürülmesi herhangi bir merkezi yapıya ihtiyaç duymayacak şekilde planlanmıştır. Seçmenler, kendi cihazlarıyla, yüksek işlem gücü ve enerji harcamadan seçim sürecini kontrol etmekte ve doğrulamaktadır. Yine bu amaç etrafında, geleneksel seçim yöntemlerinden farklı şeffaf süreçler de tanımlanmış ve açıklanmıştır. Günümüz geleneksel ve elektronik seçim sistemleri için ciddi bir sorun olan denetlenebilirlik, açık ve şeffaf bir yaklaşımla sağlanmaya çalışılmış, bu sebeple blok zinciri tercih edilmiştir.

Seçim sistemleri için diğer bir ciddi sorun olan, baskı ve yasadışı teşvik ile oy kullanımının önüne geçmek adına; sistem ve süreç, oyların oy verme merkezlerinde bulunan OVM'ler aracılığıyla kullanılacak şekilde tasarlanmıştır. Buna rağmen kurgulanan sistem internet üzerinden oy kullanımına, basit düzenlemelerle adapte olabilmektedir. Baskı ve yasadışı teşvik ile oy kullanımının düşük olduğu ülkelerde, maliyetleri azaltmak ve kolaylık sağlamak adına bu yöntem tercih edilebilir. Farklı kimlik doğrulama yöntemleri de göz önünde bulundurulmuş ve ufak düzenlemelerle sistemin bu tarz yöntemlere uyarlanabilmesi sağlanmaya çalışılmıştır. Sistem, pilot bölgeler seçilerek test edilebilir.

8.1 Değerlendirme

Sistemin OVM'ler üzerinden çalışıyor olması, her ne kadar uzun süreli kullanımda maliyeti düşerecek olsada, kurulum ve hazırlık aşamasında maliyet sorunlarını gündeme getirebilir. Kullanılan OVM sayısını azaltmak için, seçim süreleri uzatılabilir.

Tasarlanan sistem için, soruna yol açabilecek en önemli kısım, seçim sonucu oluşacak veritabanının büyüklüğü olabilir. Veritabanı boyutunun büyük olması, madencilik faaliyetini zorlaştırabilir ve seçmenlerin madencilik faaliyetine ilgi göstermemesine sebep olabilir. Çizelge 8.1'de, seçmen sayısına göre, kök blok ve nihai blok zinciri veritabanının boyutları yaklaşık olarak hesaplanmıştır. Seçim listeleri, verileri düzenleyecek ayraçlar, blok başlıkları vb. doğrudan hesaba katılmamış toplam veriye eklenen %30 sapma içerisinde değerlendirilmiştir. Özet fonksiyonu çıktısı 512 bit, açık anahtarlar 2048 bit, şifrelenmiş oy pusulaları ve kullanılan oylar 512 byte olarak hesaplanmıştır. Her seçmen için bir adet elektronik oy pusulası olacağı varsayılmıştır. OVM sayısı, her 500 seçmene 1 adet OVM düşecek şekilde; madenci sayısı da seçmen sayısının %1'i olacak şekilde alınmıştır.

Çizelge 8.1 : Blok zinciri veritabanı seçmen sayısına göre yaklaşık boyut hesaplaması.

Seçmen Sayısı	Kök Blok				Oylar	Toplam (+%30 Sapma)
	Şifrelenmiş Elektronik Pusulalar	Elektronik Pusula Kanıtları	OVM (%2) Açık Anahtarları	Madenci (%1) Açık Anahtarları		
1 milyon	512 MB	64 MB	6 MB	3 MB	512 MB	~1.5 GB
10 milyon	5120 MB	640 MB	60 MB	30 MB	5120 MB	~15 GB
50 milyon	25600 MB	3200 MB	300 MB	150 MB	25600 MB	~75 GB
100 milyon	51200 MB	6400 MB	600 MB	300 MB	51200 MB	~150 GB

Çizelgede görüldüğü üzere, seçmen sayısı arttıkça, veritabanı boyutu da ciddi oranlarda artmaktadır. Seçimin birkaç gün sürmesi sebebiyle, anahtar çiftleri veya özet bilgileri üzerine yapılması olası kaba kuvvet saldırısının kısa sürede tamamlanması gerekecektir. Bu durum göz önünde bulundurulduğunda, geçici olarak üretilen anahtar ve özet çıktılarının boyutları düşürülebilir. Bu işlem, veritabanı boyutunu azaltır. Alternatif olarak internet bağlantısının istikrarlı olduğu yerlerde, madenciler dağıtık bir servis gibi kullanılabilir ve OVM'lerin madencilerden veri sorgulaması sağlanabilir.

Değinilmesi gereken diğer bir konu da, mükerrer oy kullanımıdır. Oyun tekrar kullanımının tespiti, her seçmene istediği OVM üzerinden oy kullanım hakkı tanınacağından, ancak madenciler tarafından yapılabilir. Oy kullanıldığı zaman, elektronik pusula kanıtı blok zinciri ağında yayınlanmaktadır. Bu sebeple sadece ilk kullanılan oyun geçerli sayılması, güvenlik bakımından daha yerinde olacaktır. OVM'lerin kullanıldığı düzende, OVM'nin kimlik kontrolünü ve imzasını aşmak zor olsa da, farklı kimlik kontrol yöntemlerinin kullanıldığı düzende ilk oyun geçerli olması gerekmektedir.

Sistem, baskı ve yasadışı teşvik ile oy kullanımına olanak sağlamamak için, seçmenlere herhangi bir işlem kimlik numarası bilgisi vermemektedir. Baskı ve yasadışı teşvik ile oy kullanımının sorun olarak görülmediği ülkelerde, Mercuri yöntemi yerine, seçmenlere kullandıkları oylarla ilgili işlem kimlik numaraları verilebilir. Seçmenlerin oylarının nihai veritabanında bulunup bulunmadığını kendileri kontrol etmeleri istenebilir.

KAYNAKLAR

[1] **Athenian democracy**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 20.12.2018, adres: https://en.wikipedia.org/wiki/Athenian_democracy

[2] **M. Hapsara, A. Imran ve T. Turner**. (2017, Ocak). E-Voting in Developing Countries, *Lecture Notes in Computer Science*.

[3] **Teyit.org**. (2018, Temmuz 12). Seçim2018: Seçim günü teyit.org'a gönderilen usulsüzlük ihbarları, Alındığı tarih: 21.12.2018, adres: https://teyit.org/secim-2018-secim-gunu-teyit-orga-gonderilen-secim-usulsuzlugu-ihbarlari/

[4] **P. Musgrave**. (2016, Kasım 28). If you're even asking if Russia hacked the election, Russia got what it wanted. Alındığı tarih: 24.12.2018, adres: https://www.washingtonpost.com/posteverything/wp/2016/11/28/whether-or-not-russians-hacked-the-election-they-messed-with-our-democracy/?utm_term=.6bd7d821a8e3.

[5] **D. Springall, T. Finkenauer ve Z. Durumeric**. (2014). Security Analysis of the Estonian Internet Voting System, *21st ACM Conference on Computer and Communications Security*, *Scottsdale, Arizona, USA*.

[6] **Institute, The National Democratic**. (t.y.). The National Democratic Institute. Alındığı tarih: 24.12.2018, adres: https://www.ndi.org/e-voting-guide/examples/constitutionality-of-electronic-voting-germany

[7] **European Commission**. (2004, Mayıs 31). Electronic voting in Belgium. Alındığı tarih: 24.12.2018, adres: https://joinup.ec.europa.eu/community/epractice/case/electronic-voting-belgium

[8] **European Digital Rights**. (2004, Mayıs 5). Ireland cancels e-voting. Alındığı tarih: 25.12.2018, adres: https://edri.org/edrigramnumber2-9evote/

[9] **The Slate Group LLC**. (2015, Nisan 16). The Worst Voting Machine in America. Alındığı tarih: 26.12.2018, adres: https://slate.com/technology/2015/04/avs-winvote-virginia-voting-machines-password-was-admin.html

[10] **Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey ve J. A. Halderman**. (2014). The Matter of Heartbleed. *Internet Measurement Conference*, *Vancouver, BC, Canada, 2014.*

[11] **Shellshock (software bug)**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 26.12.2018, adres: https://en.wikipedia.org/wiki/Shellshock_(software_bug)

[12] **Electronic voting in Brazil**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 31.12.2018, adres: https://en.wikipedia.org/wiki/Electronic_voting_in_Brazil

[13] **Turner Broadcasting System, Inc**. (2011, Kasım 8). Why can't Americans vote online. Alındığı tarih: 31.12.2018, adres:

https://edition.cnn.com/2011/11/08/tech/web/online-voting/index.html

[14] **Y. D. D. M. AKIN**. (2006). Elektronik Oy Verme Sistemlerinde Güvenlik: Deneyimler ve Türkiye İçin Öneriler. *İstanbul Üniversitesi İktisat Fakültesi Ekonometri ve İstatistik Dergisi, cilt 3, 2006.*

[15] **Office for National Statistics**. (2018, Mart 22). Electoral statistics, UK: 2017. Alındığı tarih: 27.12.2018, adres: https://www.ons.gov.uk/peoplepopulationandcommunity/elections/electoralregistration/bulletins/electoralstatisticsforuk/2017

[16] **BBC**. (2017, Eylül 13). Snap general election cost over £140m. Alındığı tarih: 27.12.2018, adres: https://www.bbc.com/news/uk-politics-41258026

[17] **2014 Türkiye cumhurbaşkanlığı seçimi**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 27.12.2018, adres: https://tr.wikipedia.org/wiki/2014_T%C3%BCrkiye_cumhurba%C5%9Fkanl%C4%B1%C4%9F%C4%B1_se%C3%A7imi

[18] **Gazete Vatan**. (2014, Haziran 16). Seçimin maliyeti 400 milyon TL. Alındığı tarih: 27.12.2018, adres: http://www.gazetevatan.com/secimin-maliyeti-400-milyon-tl-648834-gundem/

[19] **R. Krimmer, D. Duenas-Cid, I. Krivonosova, P. Vinkel ve A. Koitmae**. (2018). How Much Does an e-Vote Cost? Cost Comparison per Vote in Multichannel Elections in Estonia. *International Joint Conference on Electronic Voting, Bregenz, Austria, 2018.*

[20] **Electronic Voting**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 30.12.2018, adres: https://en.wikipedia.org/wiki/Electronic_voting

[21] **2000 United States presidential election recount in Florida**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 30.12.2018, adres: https://en.wikipedia.org/wiki/2000_United_States_presidential_election_recount_in_Florida

[22] **The County of Santa Cruz**. (2019, Ocak 1). Voters with Disabilities. Alındığı tarih: 01.01.2019, adres: http://www.votescount.com/Home/VoterswithDisabilities.aspx

[23] **Spiegel Gruppe**. (2008, Ekim 28). Court to Examine Security of Electronic Voting. Alındığı tarih: 31.12.2018, adres: http://www.spiegel.de/international/germany/election-worries-court-to-examine-security-of-electronic-voting-a-587001.html

[24] **The National Democratic Institute**. (t.y.). The Constitutionality of Electronic Voting in Germany. Alındığı tarih: 31.12.2018, adres: https://www.ndi.org/e-voting-guide/examples/constitutionality-of-electronic-voting-germany

[25] **Brave New Ballot**. (t.y.). Electronic Voting in USA. Alındığı tarih: 01.01.2019, adres: http://www.bravenewballot.org/electronic-voting-in-usa/

[26] **Electronic voting by country**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 01.01.2019, adres: https://en.wikipedia.org/wiki/Electronic_voting_by_country

[27] **I. Akwei**. (2016, Aralık 7). U.S. Presidential Election 2016: How Americans are voting. Alındığı tarih: 05.01.2019, adres:

http://www.africanews.com/2016/11/07/us-presidential-election-2016-how-americans-are-voting/

[28] **S. Gal ve G. Panetta**. (2018, Aralık 2). Midterm elections 2018: 25 states that allow electronic voting. Alındığı tarih: 01.01.2019, adres: https://www.businessinsider.com/22-states-that-allow-you-to-vote-online-2016-9

[29] **Electronic voting in Belgium**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 08.01.2019, adres: https://en.wikipedia.org/wiki/Electronic_voting_in_Belgium

[30] **The Wire**. (t.y.). A Vote For The Future: Electronic Voting... Is It time. Alındığı tarih: 01.01.2019, adres: http://thewire.org.au/story/vote-future-electronic-voting-time/

[31] **Merco Press**. (2014, Ekim 4). Investors on edge waiting for the results of Sunday's voting in Brazil. Alındığı tarih: 10.01.2019, adres: http://en.mercopress.com/2014/10/04/investors-on-edge-waiting-for-the-results-of-sunday-s-voting-in-brazil

[32] **Electronic voting in Estonia**. (t.y.) *Wikimedia Foundation, Inc.* Alındığı tarih: 01.01.2019, adres: https://en.wikipedia.org/wiki/Electronic_voting_in_Estonia

[33] **Err Arhiv**. (2013, Temmuz 12). E-Voting Source Code Made Public. Alındığı tarih: 10.01.2019, adres: https://news.err.ee/107779/e-voting-source-code-made-public

[34] **Electronic voting in Canada**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 01.01.2019, adres: https://en.wikipedia.org/wiki/Electronic_voting_in_Canada

[35] **Smartmatic**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 01.01.2019, adres: https://en.wikipedia.org/wiki/Smartmatic#Venezuela

[36] **J. Menn, Thomson Reuters**. (2014, Mart 31). Alındığı tarih: 06.02.2014, adres: https://www.reuters.com/article/us-usa-security-nsa-rsa/exclusive-nsa-infiltrated-rsa-security-more-deeply-than-thought-study-idUSBREA2U0TY20140331

[37] **S. Nakamoto**. (2008). Bitcoin. Alındığı tarih: 09.02.2019, adres: https://bitcoin.org/bitcoin.pdf

[38] **Bitcoin**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 09.01.2019, adres: https://en.wikipedia.org/wiki/Bitcoin

[39] **Digiconomist**. (t.y.). Bitcoin Energy Consumption Index. Alındığı tarih: 11.02.2019, adres: https://digiconomist.net/bitcoin-energy-consumption

[40] **The Guardian**. (2018, Mart 19). Vladimir Putin secures record win in Russian presidential election. Alındığı tarih: 25.12.2018, adres: https://www.theguardian.com/world/2018/mar/19/vladimir-putin-secures-record-win-in-russian-presidential-election

[41] **Mercuri method**. (t.y.). *Wikimedia Foundation, Inc.* Alındığı tarih: 27.01.2019, adres: https://en.wikipedia.org/wiki/Mercuri_method

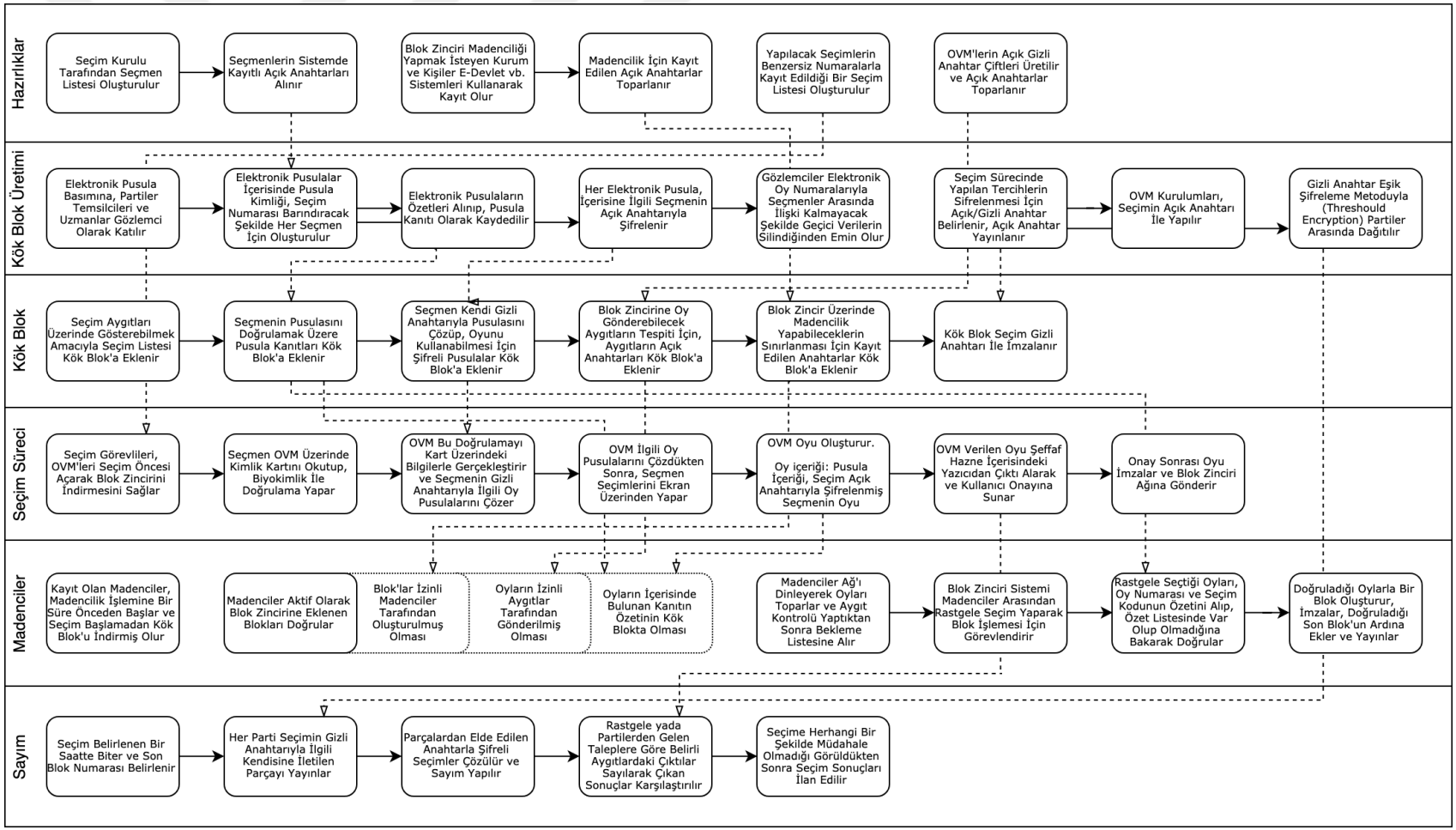
[42] **G. Schryen**. (2009). Security of Open Source and Closed Source Software: An Empirical Comparison of Published Vulnerabilities. *15th Americas Conference on Information Systems, San Francisco, 2009.*

[43] **Proofpoint, Inc**. (2018). The Human Factor 2018.

EKLER

EK A.1 : Blok zinciri temelli elektronik seçim süreci geniş özeti
EK A.2 : Uygulama kaynak kodu

EK A.1



Şekil A.1 : Blok zinciri temelli elektronik seçim süreci geniş özeti.

EK A.2

Kaynak kod, https://github.com/bgultekin/blockchain-based-e-voting adresinden indirilebilir.

Dosya: prepare.py

```
1. # -*- coding: utf-8 -*-
2. from colors import *
3. from app.prepare import *
4.
5. """
6. Preparation step for the election.
7. """
8.
9. print(color("Hazırlık aşaması başlatıldı", fg="cyan"))
10.  print(color("*", fg="gray") * 100)
11.
12.  print(color("Seçmen anahtar üretimi", fg="yellow"))
13.  print(color("-", fg="gray") * 100)
14.
15.  voter_keys.generate()
16.
17.  print(color("*", fg="gray") * 100)
18.  print(color("Madenci anahatar üretimi", fg="yellow"))
19.  print(color("-", fg="gray") * 100)
20.
21.  miner_keys.generate()
22.
23.  print(color("*", fg="gray") * 100)
24.  print(color("OVM (Oy Verme Makinesi) anahtar üretimi",
    fg="yellow"))
25.  print(color("-", fg="gray") * 100)
26.
27.  vcm_keys.generate()
28.
29.  print(color("*", fg="gray") * 100)
30.  print(color("Kök blok üretimi", fg="yellow"))
31.  print(color("-", fg="gray") * 100)
32.
33.  genesis_block.generate()
34.
35.  print(color("*", fg="gray") * 100)
36.  print(color("Hazırlık aşaması tamamlandı", fg="cyan"))
```

Dosya: vote.py

```
1. # -*- coding: utf-8 -*-
2. from app.vcm import VCM
3.
4. # we are voting in Springfield
5. vcm = VCM("100")
6. vcm.start()
```

Dosya: tally.py

```
1. # -*- coding: utf-8 -*-
2. import json, logging, os
```

```
3. from colors import *
4. from app import config, crypto
5. from app.blockchain import Blockchain
6.
7. logging.info("Sayım süreci başlatıldı")
8.
9. bc = Blockchain()
10.  blocks = bc.get_blocks()
11.
12.  genesis_block = blocks[0]
13.  blocks = blocks[1:]
14.
15.  logging.info("Seçim gizli anahtarı artık gizli değil")
16.  election_private_key =
   crypto.load_private_key_file(config.ELECTION_KEYS_FOLDER,
   "election")
17.
18.  # read election data and count
19.  with open(config.ELECTION_FILE) as json_file:
20.      election = json.load(json_file)
21.
22.  for sub_election_id, sub_election in election.iteritems():
23.      sub_election["results"] = {key: 0 for key in
   sub_election["candidates"]}
24.
25.  for block in blocks:
26.      for ballot in block["content"]:
27.          election_id =
   ballot["content"]["proof"]["election_id"]
28.          vote = crypto.decrypt(ballot["content"]["vote"],
   election_private_key)
29.
30.          election[str(election_id)]["results"][vote] += 1
31.
32.  logging.info("Sayım süreci tamamlandı")
33.
34.  # dump results
35.  with open(os.path.join(config.OUTPUT_FOLDER,
   "results.json"), "w") as json_file:
36.      json.dump(election, json_file)
37.
38.  logging.info(u"Sonuçlar %s dizinine kaydedildi" %
   config.OUTPUT_FOLDER)
39.  logging.info("Sonuçlar aşağıda görülebilir\n\n")
40.
41.  # print it beautifully
42.  for sub_election_id, sub_election in election.iteritems():
43.      print("-" * 70)
44.      print("| " + color(sub_election["name"] + u" Sonuçları",
   fg="yellow").ljust(75) + " |")
45.      print("-" * 70)
46.      print("| " + color("Aday", style="bold").ljust(38) + " |
   " + color("Oy Sayısı", style="bold").ljust(43) + " |")
47.      print("-" * 70)
48.
49.      for candidate, result in
   sub_election["results"].iteritems():
50.          print("| %s | %s |" % (candidate.ljust(30),
   str(result).ljust(33)))
51.          print("-" * 70)
52.
```

```
53.       print("\n")
```

Dosya: requirements.txt

```
1. cryptography==2.3.1
2. ansicolors==1.1.7
```

Dosya: input/election.json

```
1. {
2.     "100": {
3.         "name": "Çizgi Ülkesi Genel Seçimi",
4.         "candidates": ["Lisa Simpson", "Eric Cartman"]
5.     },
6.     "101": {
7.         "name": "Springfield Belediye Seçimi",
8.         "candidates": ["Bart Simpson", "Joe Quimby", "Fat
   Tony"]
9.     },
10.      "102": {
11.          "name": "South Park Belediye Seçimi",
12.          "candidates": ["Token Black", "Mary Mcdaniels"]
13.      }
14.  }
```

Dosya: input/miner.json

```
1. [{
2.     "id": "100",
3.     "citizen_id": "108"
4. }, {
5.     "id": "101",
6.     "citizen_id": "207"
7. }]
```

Dosya: input/vcms.json

```
1. [{
2.   "id": "100",
3.     "location": "Springfield"
4. }, {
5.   "id": "200",
6.     "location": "South Park"
7. }]
```

Dosya: input/voters.json

```
1. [{
2.   "id": "100",
3.   "name": "Homer",
4.     "surname": "Simpson",
5.     "eligible_for_voting": [100, 101]
6. }, {
7.   "id": "101",
8.   "name": "Marge",
```

```
    "surname": "Simpson",
     "eligible_for_voting": [100, 101]
}, {
        "id": "102",
        "name": "Bart",
    "surname": "Simpson",
    "eligible_for_voting": [100, 101]
}, {
        "id": "103",
        "name": "Lisa",
    "surname": "Simpson",
    "eligible_for_voting": [100, 101]
}, {
        "id": "104",
        "name": "Ned",
    "surname": "Flanders",
    "eligible_for_voting": [100, 101]
}, {
        "id": "105",
        "name": "Moe",
    "surname": "Szyslak",
    "eligible_for_voting": [100, 101]
}, {
        "id": "106",
        "name": "Fat",
    "surname": "Tony",
    "eligible_for_voting": [100, 101]
}, {
        "id": "107",
        "name": "Apu",
    "surname": "Nahasapeemapetilon",
    "eligible_for_voting": [100, 101]
}, {
        "id": "108",
        "name": "Jeff",
    "surname": "Albertson",
    "eligible_for_voting": [100, 101]
}, {
        "id": "200",
        "name": "Kyle",
    "surname": "Broflovski",
    "eligible_for_voting": [100, 102]
}, {
        "id": "201",
        "name": "Stan",
    "surname": "Marsh",
    "eligible_for_voting": [100, 102]
}, {
        "id": "202",
        "name": "Eric",
    "surname": "Cartman",
    "eligible_for_voting": [100, 102]
}, {
        "id": "203",
        "name": "Butters",
    "surname": "Stotch",
    "eligible_for_voting": [100, 102]
}, {
        "id": "204",
        "name": "Wendy",
    "surname": "Testaburger",
```

```
    "eligible_for_voting": [100, 102]
}, {
        "id": "205",
        "name": "Token",
    "surname": "Black",
    "eligible_for_voting": [100, 102]
}, {
        "id": "206",
        "name": "Mary",
    "surname": "Mcdaniels",
    "eligible_for_voting": [100, 102]
}, {
        "id": "207",
        "name": "Jenkins",
    "surname": "South",
    "eligible_for_voting": [100, 102]
}]
```

Dosya: app/blockchain.py

```
# -*- coding: utf-8 -*-
import os, glob, json
import config, crypto

class Blockchain:
    """
    This class is simple abstract class
    just for showing responsibility of blockchain
    in the code and content of blocks.

     It just simulates blockchain behavior
     and saves given data to files.
     """

     def __init__(self):
         self._blocks = []
         block_files = self._list_block_files()

         for block_file in block_files:
             with open(block_file) as file:
                 self._blocks.append(json.load(file))

     """
     Add a block to chain and create a file.

     :param self: self class
     :param block_content: block
     """
     def add_block(self, block):
         block_count = len(self._blocks)

         # if this is not the genesis_block
         # then we should validate it
         if (block_count > 0):
             self._validate_block(block)

         self._blocks.append(block)

         block_file_name = self._name_block(block_count)

```

```
41.             with open(os.path.join(config.BLOCKCHAIN_FOLDER,
   block_file_name), "w") as block_file:
42.                 block_file.write(json.dumps(block,
   sort_keys=True))
43.
44.         """
45.         Get a block content.
46.
47.         :param self: self class
48.         :param index: index of the block
49.
50.         :return: block as a dictionary
51.         """
52.         def get_block(self, index):
53.             return self._blocks[index]
54.
55.         """
56.         Get all blocks.
57.
58.         :param self: self class
59.
60.         :return: blocks as a list
61.         """
62.         def get_blocks(self):
63.             return self._blocks
64.
65.         """
66.         Get the last block content.
67.
68.         :param self: self class
69.
70.         :return: the last block as a dictionary
71.         """
72.         def get_last_block(self):
73.             last_index = len(self._blocks) - 1
74.
75.             return self._blocks[last_index]
76.
77.         """
78.         Clear database and delete all files.
79.
80.         :param self: self class
81.         """
82.         def purge(self):
83.             block_files = self._list_block_files()
84.             self._blocks = []
85.
86.             for block_file in block_files:
87.                 os.remove(block_file)
88.
89.         """
90.         List block files.
91.
92.         :param self: self class
93.
94.         :return: sorted files as a list
95.         """
96.         def _list_block_files(self):
97.             block_files =
   glob.glob(os.path.join(config.BLOCKCHAIN_FOLDER, "*.json"))
98.
```

```
99.             return sorted(block_files)
100.
101.        """
102.        Name a block.
103.
104.        :param self: self class
105.        :param index: index of the block
106.        """
107.        def _name_block(self, index):
108.            return str(index) + '.json'
109.
110.        """
111.        Validate a block.
112.        This is normally responsibility of miners.
113.
114.        :param self: self class
115.        :param index: block
116.        """
117.        def _validate_block(self, block):
118.            miner_id = block["header"]["miner_id"]
119.            miner_public_key =
    self.get_block(0)["content"]["miners"][str(miner_id)]["public_
    key"].encode('ascii')
120.
121.            crypto.verify(block["header"]["signature"],
    json.dumps(block["content"], sort_keys=True),
    crypto.load_public_key(miner_public_key))
```

Dosya: app/config.py

```
1. ELECTION_FILE = 'input/election.json'
2. ELECTION_KEYS_FOLDER = 'data/election_keys'
3.
4. VOTERS_FILE = 'input/voters.json'
5. VOTER_KEYS_FOLDER = 'data/voter_keys'
6.
7. VCM_FILE = 'input/vcms.json'
8. VCM_KEYS_FOLDER = 'data/vcm_keys'
9.
10.  MINERS_FILE = 'input/miners.json'
11.  MINER_KEYS_FOLDER = 'data/miner_keys'
12.
13.  BLOCKCHAIN_FOLDER = 'data/blockchain'
14.  BLOCKCHAIN_BLOCK_VOTE_LIMIT = 2
15.
16.  OUTPUT_FOLDER = 'output'
```

Dosya: app/crypto.py

```
1. # -*- coding: utf-8 -*-
2. import os, logging, base64
3. from cryptography.hazmat.backends import default_backend
4. from cryptography.hazmat.primitives.asymmetric import rsa
5. from cryptography.hazmat.primitives import hashes
6. from cryptography.hazmat.primitives.asymmetric import padding
7. from cryptography.hazmat.primitives import serialization
8. import config
9.
10.  logging.basicConfig(level=logging.INFO)
```

```
11.
12.  """
13.  Generate key pair with RSA algorithm.
14.
15.  return private_key and public_key objects
16.  """
17.  def generate_key_pair():
18.      private_key = rsa.generate_private_key(
19.          public_exponent=65537,
20.          key_size=2048,
21.          backend=default_backend()
22.      )
23.
24.      public_key = private_key.public_key()
25.
26.      return private_key, public_key
27.
28.  """
29.  Serialize private key.
30.
31.  :param private_key: private key to serialize
32.  """
33.  def serialize_private_key(private_key):
34.      return private_key.private_bytes(
35.          encoding=serialization.Encoding.PEM,
36.
   format=serialization.PrivateFormat.TraditionalOpenSSL,
37.          encryption_algorithm=serialization.NoEncryption()
38.      )
39.
40.  """
41.  Serialize public key.
42.
43.  :param public_key: public key to serialize
44.  """
45.  def serialize_public_key(public_key):
46.      return public_key.public_bytes(
47.          encoding=serialization.Encoding.PEM,
48.
   format=serialization.PublicFormat.SubjectPublicKeyInfo
49.      )
50.
51.  """
52.  Create a file with given content
53.
54.  :param folder: folder to write keys
55.  :param name: name of output files
56.  :param private_key: serialized private key
57.  :param public_key: serialized public key
58.  """
59.  def write_key_pair(folder, name, private_key, public_key):
60.      with open(os.path.join(folder, name), "w") as file:
61.          file.write(private_key)
62.
63.      with open(os.path.join(folder, name + ".pub"), "w") as
   file:
64.          file.write(public_key)
65.
66.  """
67.  Load private key
68.
```

```
69.  :param key: key
70.  """
71.  def load_private_key(key):
72.      return serialization.load_pem_private_key(
73.          key,
74.          password=None,
75.          backend=default_backend()
76.      )
77.
78.  """
79.  Load public key from
80.
81.  :param key: key
82.  """
83.  def load_public_key(key):
84.      return serialization.load_pem_public_key(
85.          key,
86.          backend=default_backend()
87.      )
88.
89.  """
90.  Load private key from a file
91.
92.  :param folder: folder to key file
93.  """
94.  def load_private_key_file(folder, name):
95.      with open(os.path.join(folder, name), "rb") as file:
96.          return load_private_key(file.read())
97.
98.  """
99.  Load public key from a file
100.
101. :param folder: folder to key file
102. """
103. def load_public_key_file(folder, name):
104.     with open(os.path.join(folder, name + ".pub"), "rb") as
   file:
105.         return load_public_key(file.read())
106.
107. """
108. Return random string base64 encoded
109.
110. :param byte_length: byte length of random output
111. """
112. def random_string(byte_length = 64):
113.     random_bytes = os.urandom(byte_length)
114.     return base64.b64encode(random_bytes)
115.
116. """
117. Return sha256 hash of the message base64 encoded
118.
119. :param message: message
120. """
121. def sha256(message):
122.     digest = hashes.Hash(hashes.SHA256(),
   backend=default_backend())
123.     digest.update(message)
124.     return base64.b64encode(digest.finalize())
125.
126. """
```

```
127. Encrypt given message with given public key and return
   base64 encoded
128.
129. :param message: message
130. :param public_key: public_key
131. """
132. def encrypt(message, public_key):
133.     encrypted_message = public_key.encrypt(
134.         message,
135.         # make RSA probabilistic with OAEP, random output is
   needed
136.         padding.OAEP(
137.             mgf=padding.MGF1(algorithm=hashes.SHA256()),
138.             algorithm=hashes.SHA256(),
139.             label=None
140.         )
141.     )
142.
143.     return base64.b64encode(encrypted_message)
144.
145. """
146. Decrypt given base64 encoded ciphertext with given private
   key
147.
148. :param message: message
149. :param public_key: public_key
150. """
151. def decrypt(ciphertext, private_key):
152.     ciphertext_decoded = base64.b64decode(ciphertext)
153.
154.     return private_key.decrypt(
155.         ciphertext_decoded,
156.         padding.OAEP(
157.             mgf=padding.MGF1(algorithm=hashes.SHA256()),
158.             algorithm=hashes.SHA256(),
159.             label=None
160.         )
161.     )
162.
163. """
164. Sign given message with given private key and return base64
   encoded
165.
166. :param message: message
167. :param private_key: private_key
168. """
169. def sign(message, private_key):
170.     signature = private_key.sign(
171.         message,
172.         padding.PSS(
173.             mgf=padding.MGF1(hashes.SHA256()),
174.             salt_length=padding.PSS.MAX_LENGTH
175.         ),
176.         hashes.SHA256()
177.     )
178.
179.     return base64.b64encode(signature)
180.
181. """
182. Verify given base64 encoded signature with given private key
   and message.
```

```
183. If not verified, throw
   cryptography.exceptions.InvalidSignature
184.
185. :param signature: signature
186. :param message: message
187. :param public_key: public_key
188. """
189. def verify(signature, message, public_key):
190.     return public_key.verify(
191.         base64.b64decode(signature),
192.         message,
193.         padding.PSS(
194.             mgf=padding.MGF1(hashes.SHA256()),
195.             salt_length=padding.PSS.MAX_LENGTH
196.         ),
197.         hashes.SHA256()
198.     )
```

Dosya: app/miner.py

```
1. # -*- coding: utf-8 -*-
2. import os, glob, json
3. import config, crypto, blockchain
4.
5. class Miner:
6.     """
7.     This class is simple abstract class
8.     just for showing responsibility of miner
9.     in the blockchain network.
10.
11.      It just simulates miners' behavior,
12.      validates votes and saves.
13.      """
14.
15.      blockchain = blockchain.Blockchain()
16.
17.      def __init__(self, id):
18.          self._votes = []
19.          self.id = id
20.          self.private_key =
   crypto.load_private_key_file(config.MINER_KEYS_FOLDER,
   str(self.id))
21.
22.      """
23.      Add a vote to waiting list.
24.
25.      :param self: self class
26.      :param vote: vote data
27.      """
28.      def add_vote(self, vote):
29.          vote["header"]["hash_of_proof"] = crypto.sha256(
30.              json.dumps(vote["content"]["proof"],
   sort_keys=True)
31.          )
32.
33.          self.validate_vote(vote)
34.          self.check_used_vote(vote)
35.
36.          self._votes.append(vote)
37.
```

```
38.          if len(self._votes) >=
   config.BLOCKCHAIN_BLOCK_VOTE_LIMIT:
39.              self.blockchain.add_block({
40.                  "content": self._votes,
41.                  "header": {
42.                      "signature":
   crypto.sign(json.dumps(self._votes, sort_keys=True),
   self.private_key),
43.                      "miner_id": self.id
44.                  }
45.              })
46.
47.              self._votes = []
48.
49.      """
50.      Validate a vote.
51.
52.      :param self: self class
53.      :param vote: vote dictionary
54.      """
55.      def validate_vote(self, vote):
56.          genesis_block = self.blockchain.get_block(0)
57.
58.          # check proof
59.          proofs =
   genesis_block["content"]["electronic_ballot_proofs"]
60.
61.          if not vote["header"]["hash_of_proof"] in proofs:
62.              raise Exception("Invalid proof")
63.
64.          # check VCM
65.          vcm_id = vote["header"]["vcm_id"]
66.          vcm_public_key =
   genesis_block["content"]["vcms"][vcm_id]["public_key"].encode(
   'ascii')
67.
68.          crypto.verify(vote["header"]["signature"],
   json.dumps(vote["content"], sort_keys=True),
   crypto.load_public_key(vcm_public_key))
69.
70.      """
71.      Check if a vote is already used.
72.      This can/should be done with different and efficient
   structures in a real instance.
73.      This can be done on tallying process, if it won't be
   done in blockchain network.
74.
75.      :param self: self class
76.      :param vote: vote dictionary
77.      """
78.      def check_used_vote(self, vote):
79.          blocks = self.blockchain.get_blocks()
80.          blocks = blocks[1:]
81.          blocks.append({
82.              "content": self._votes
83.          })
84.
85.          for block in blocks:
86.              for vote_in_block in block["content"]:
87.                  if vote_in_block["header"]["hash_of_proof"]
   == vote["header"]["hash_of_proof"]:
```

```
88.                     raise Exception("This vote is already
   used")
89.
```

Dosya: app/vcm.py

```
1. # -*- coding: utf-8 -*-
2. import os, glob, json, time
3. from colors import *
4. import config, crypto, blockchain, miner
5.
6. class VCM:
7.     """
8.     This class is simple abstract class
9.     just for showing responsibility of
10.      VCM (Vote Casting Machine).
11.
12.      It just simulates VCMs' behavior.
13.      """
14.
15.      blockchain = blockchain.Blockchain()
16.      miner = miner.Miner(100)
17.      election_public_key =
   crypto.load_public_key_file(config.ELECTION_KEYS_FOLDER,
   "election")
18.
19.      def __init__(self, id):
20.          self.id = id
21.          self.private_key =
   crypto.load_private_key_file(config.VCM_KEYS_FOLDER,
   str(self.id))
22.          self.genesis_block = self.blockchain.get_block(0)
23.
24.      """
25.      Start the machine.
26.
27.      :param self: self class
28.      """
29.      def start(self):
30.          while True:
31.              os.system("clear")
32.
33.              print("Oy Verme Arayüzüne Hoşgeldiniz.")
34.
35.              voter_id = raw_input("Lütfen Kimlik Numaranızı
   Giriniz: ")
36.
37.              print("Seçim anahtarı doğrulanıyor...")
38.
39.              self._verify_genesis_block()
40.
41.              print("Seçim anahtarı doğrulama başarılı")
42.              print("Elektronik pusulalarınız getiriliyor...")
43.
44.              voter_decrypted_ballots =
   self._get_voter_ballots(voter_id)
45.
46.              print("Oy verme zamanı...")
47.
48.              # let user read output
```

```
49.             self._wait_and_clear(2)
50.
51.             for ballot in voter_decrypted_ballots:
52.                 election_id_to_vote = ballot["election_id"]
53.                 election_to_vote =
   self.genesis_block["content"]["election"][str(election_id_to_v
   ote)]
54.
55.                 print("=" * 50)
56.                 print(color(election_to_vote["name"],
   fg="yellow") + u" için lütfen tercihinizi yapınız \n")
57.                 print("Adaylar aşağıda listelenmiştir: \n")
58.
59.                 for index, candidate in
   enumerate(election_to_vote["candidates"], start=1):
60.                     print("%d) %s" % (index, candidate))
61.
62.                 while True:
63.                     candidate_index = raw_input("\nLütfen
   tercih ettiğiniz adayın numarasını giriniz: ")
64.                     candidate_index = int(candidate_index)
65.
66.                     if candidate_index > 0 and
   candidate_index <= len(election_to_vote["candidates"]):
67.                         chosen_candidate =
   election_to_vote["candidates"][candidate_index - 1]
68.
69.                         self.miner.add_vote(
70.
   self._cast_vote(chosen_candidate, ballot)
71.                         )
72.
73.                         print("\n")
74.                         print(color(chosen_candidate,
   fg="yellow") + u" adayına oy verdiniz")
75.                         print("=" * 50)
76.
77.                         self._wait_and_clear(1)
78.
79.                         break
80.                     else:
81.                         print(u"Lütfen 1 ile %d arasında bir
   sayı giriniz\n" % (len(election_to_vote["candidates"])))
82.
83.             print("Oy verdiğiniz için teşekkür ederiz!")
84.             print("Yenide başlatılıyor...")
85.
86.             self._wait_and_clear(3)
87.
88.     """
89.     Verify genesis block.
90.
91.     :param self: self class
92.     """
93.     def _verify_genesis_block(self):
94.
   crypto.verify(self.genesis_block["header"]["signature"],
   json.dumps(self.genesis_block["content"], sort_keys=True),
   self.election_public_key)
95.
96.     """
```

```
97.      Get given voter's ballots.
98.
99.      :param self: self class
100.     :param voter_id: voter_id
101.     """
102.     def _get_voter_ballots(self, voter_id):
103.         voter_decrypted_ballots = []
104.         voter_private_key =
   crypto.load_private_key_file(config.VOTER_KEYS_FOLDER,
   voter_id)
105.         voter_ballots =
   self.genesis_block["content"]["encrypted_ballots"][voter_id]
106.
107.         for ballot in voter_ballots:
108.             decrypted_ballot = crypto.decrypt(ballot,
   voter_private_key)
109.
   voter_decrypted_ballots.append(json.loads(decrypted_ballot))
110.
111.         return voter_decrypted_ballots
112.
113.     """
114.     Cast a vote with given parameters and sign it.
115.
116.     :param self: self class
117.     :param chosen_candidate: chosen candidate
118.     :param proof: ballot content
119.     """
120.     def _cast_vote(self, chosen_candidate, proof):
121.         content = {
122.             "proof": proof,
123.             "vote": crypto.encrypt(str(chosen_candidate),
   self.election_public_key)
124.         }
125.
126.         return {
127.             "content": content,
128.             "header": {
129.                 "signature": crypto.sign(json.dumps(content,
   sort_keys=True), self.private_key),
130.                 "vcm_id": self.id
131.             }
132.         }
133.
134.     """
135.     Wait given seconds and clear terminal.
136.
137.     :param self: self class
138.     :param wait: wait
139.     """
140.     def _wait_and_clear(self, wait):
141.         time.sleep(wait)
142.         os.system("clear")
```

Dosya: app/__init__.py

```
1. # empty init file
```

Dosya: app/prepare/__init__.py

```
1. __all__ = ["genesis_block", "miner_keys", "voter_keys",
   "vcm_keys"]
```

Dosya: app/prepare/genesis_block.py

```
1. # -*- coding: utf-8 -*-
2. import json, logging, os
3. from app import config, crypto
4. from app.blockchain import Blockchain
5.
6. """
7. Create genesis block and include all needed
8. information into it.
9. """
10.  def generate():
11.      logging.info("Kök blok üretimi başlatıldı")
12.
13.      # election keys
14.      election_private_key, election_public_key =
   crypto.generate_key_pair()
15.      election_private_key_serialized =
   crypto.serialize_private_key(election_private_key)
16.      election_public_key_serialized =
   crypto.serialize_public_key(election_public_key)
17.
18.      crypto.write_key_pair(config.ELECTION_KEYS_FOLDER,
   "election", election_private_key_serialized,
   election_public_key_serialized)
19.
20.      logging.info("Seçim anahtar çifti üretildi")
21.      logging.info("Seçmenler seçmen listesinden okunuyor
   (voters.json)")
22.
23.      electronic_ballot_proofs = []
24.      encrypted_ballots = {}
25.
26.      # electronic ballots
27.      with open(config.VOTERS_FILE) as json_file:
28.          voters = json.load(json_file)
29.
30.          for voter in voters:
31.              logging.info(u"%s, %s %s için elektronik
   pusulalar üretiliyor" %(voter["id"], voter["name"],
   voter["surname"]))
32.              voter_public_key =
   crypto.load_public_key_file(config.VOTER_KEYS_FOLDER,
   voter["id"])
33.              encrypted_ballots[voter["id"]] = []
34.
35.              for sub_election in
   voter["eligible_for_voting"]:
36.                  ballot_id = crypto.random_string()
37.                  electronic_ballot = {
38.                      "ballot_id": ballot_id,
39.                      "election_id": sub_election
40.                  }
41.
42.                  electronic_ballot_json =
   json.dumps(electronic_ballot)
```

```
43.                     electronic_ballot_proof =
   crypto.sha256(electronic_ballot_json)
44.                     electronic_ballot_encrypted =
   crypto.encrypt(electronic_ballot_json, voter_public_key)
45.
46.
   electronic_ballot_proofs.append(electronic_ballot_proof)
47.
   encrypted_ballots[voter["id"]].append(electronic_ballot_encryp
   ted)
48.
49.
50.       logging.info("Seçim bilgisi okunuyor")
51.
52.       # election data
53.       with open(config.ELECTION_FILE) as json_file:
54.           election = json.load(json_file)
55.
56.       logging.info("Madencilerin bilgileri ve açık anahtarları
   okunuyor")
57.
58.       # miners data and public keys
59.       with open(config.MINERS_FILE) as json_file:
60.           miners_list = json.load(json_file)
61.           miners = {}
62.
63.           for miner in miners_list:
64.               with open(os.path.join(config.MINER_KEYS_FOLDER,
   miner["id"] + ".pub"), "rb") as file:
65.                   miners[miner["id"]] = {
66.                       "public_key": file.read(),
67.                       "citizen_id": miner["citizen_id"]
68.                   }
69.
70.       logging.info("OVM'lerin (Oy Verme Makinesi) bilgileri ve
   açık anahtarları okunuyor")
71.
72.       # vcm data and public keys
73.       with open(config.VCM_FILE) as json_file:
74.           vcms_list = json.load(json_file)
75.           vcms = {}
76.
77.           for vcm in vcms_list:
78.               with open(os.path.join(config.VCM_KEYS_FOLDER,
   vcm["id"] + ".pub"), "rb") as file:
79.                   vcms[vcm["id"]] = {
80.                       "public_key": file.read(),
81.                       "location": vcm["location"]
82.                   }
83.
84.       # creating the genesis block
85.       logging.info("Kök blok üretiliyor")
86.
87.       genesis_block_content = {
88.           "encrypted_ballots": encrypted_ballots,
89.           "electronic_ballot_proofs":
   electronic_ballot_proofs,
90.           "election": election,
91.           "miners": miners,
92.           "vcms": vcms
93.       }
```

```
94.
95.      logging.info("Kök blok imzalanıyor")
96.
97.      genesis_block = {
98.          "header": {
99.              "signature":
   crypto.sign(json.dumps(genesis_block_content, sort_keys=True),
   election_private_key)
100.         },
101.         "content": genesis_block_content
102.     }
103.
104.     logging.info("Kök blok kaydediliyor")
105.
106.     bc = Blockchain()
107.     bc.purge()
108.     bc.add_block(genesis_block)
109.
110.     logging.info("Kök blok üretimi tamamlandı")
```

Dosya: app/prepare/miner_keys.py

```
1. # -*- coding: utf-8 -*-
2. import json, logging
3. from app import config, crypto
4.
5. """
6. Create miner keys.
7. """
8. def generate():
9.     logging.info("Madenci bilgileri %s dosyasından okunuyor" %
   config.MINERS_FILE)
10.
11.      with open(config.MINERS_FILE) as json_file:
12.          miners = json.load(json_file)
13.
14.          for miner in miners:
15.              logging.info(u"#%s (%s) Madencisi için anahtar
   çifti üretiliyor" %(miner["id"], miner["citizen_id"]))
16.
17.              private_key, public_key =
   crypto.generate_key_pair()
18.
19.              private_key_serialized =
   crypto.serialize_private_key(private_key)
20.              public_key_serialized =
   crypto.serialize_public_key(public_key)
21.
22.              crypto.write_key_pair(config.MINER_KEYS_FOLDER,
   miner["id"], private_key_serialized, public_key_serialized)
23.
24.              logging.info(u"%s ve %s.pub dosyaları
   oluşturuldu" %(miner["id"], miner["id"]))
```

Dosya: app/prepare/vcm_keys.py

```
1. # -*- coding: utf-8 -*-
2. import json, logging
3. from app import config, crypto
```

```
4.
5. """
6. Create VCM (Vote Casting Machine) keys.
7. """
8. def generate():
9.     logging.info("OVM'ler %s dosyasından okunuyor" %
   config.VCM_FILE)
10.
11.     with open(config.VCM_FILE) as json_file:
12.         vcms = json.load(json_file)
13.
14.         for vcm in vcms:
15.             logging.info(u"#%s OVM (%s) için anahtar çifti
   üretiliyor" %(vcm["id"], vcm["location"]))
16.
17.             private_key, public_key =
   crypto.generate_key_pair()
18.
19.             private_key_serialized =
   crypto.serialize_private_key(private_key)
20.             public_key_serialized =
   crypto.serialize_public_key(public_key)
21.
22.             crypto.write_key_pair(config.VCM_KEYS_FOLDER,
   vcm["id"], private_key_serialized, public_key_serialized)
23.
24.             logging.info(u"%s ve %s.pub dosyaları
   oluşturuldu" %(vcm["id"], vcm["id"]))
```

Dosya: app/prepare/voter_keys.py

```
1. # -*- coding: utf-8 -*-
2. import json, logging
3. from app import config, crypto
4.
5. """
6. Create voter keys.
7. """
8. def generate():
9.     logging.info("Seçmen bilgileri %s dosyasından okunuyor" %
   config.VOTERS_FILE)
10.
11.     with open(config.VOTERS_FILE) as json_file:
12.         voters = json.load(json_file)
13.
14.         for voter in voters:
15.             logging.info(u"%s, %s %s için anahtar çifti
   üretiliyor" %(voter["id"], voter["name"], voter["surname"]))
16.
17.             private_key, public_key =
   crypto.generate_key_pair()
18.
19.             private_key_serialized =
   crypto.serialize_private_key(private_key)
20.             public_key_serialized =
   crypto.serialize_public_key(public_key)
21.
22.             crypto.write_key_pair(config.VOTER_KEYS_FOLDER,
   voter["id"], private_key_serialized, public_key_serialized)
23.
```

```
24.                 logging.info(u"%s ve %s.pub dosyaları
    oluşturuldu" %(voter["id"], voter["id"]))
```

ÖZGEÇMİŞ

Ad Soyad: Bilal Gültekin

Doğum Yeri ve Tarihi: Muş / 12.05.1991

Adres: Avcılar / İSTANBUL

E-Posta: bilal@gultekin.me

Lisans: İstanbul Teknik Üniversitesi, Elektronik Mühendisliği, 2009 - 2014

Mesleki Deneyim ve Ödüller: Kıdemli Yazılım Geliştirici - SOCi, Inc. (2018)

Kurucu Ortak / Yazılım Geliştirici - Alfaron (2017 - 2018)

Takım Lideri Geliştirici - Space Tech (2016 - 2017)

Kurucu Ortak / Takım Lideri Geliştirici - Shift Teknoloji (2015 - 2016)

Yazılım Geliştirici - Bilişim Teknolojileri Ajansı (2014 - 2015)

Serbest Yazılım Geliştirici (2009 - 2014)

Yazılım Geliştirici Stajyeri - Turkcell (2013)

Gömülü Yazılım Stajyeri - İTÜ GSTL (2012)

Yazılım Geliştirici Stajyeri - Gerger (2011)

IT Stajyeri - Aykanlar Group (2010)