applied sciences

MDPI

Article

Electronic Voting System Using an Enterprise Blockchain

Camilo Denis González [1], Daniel Frias Mena [1], Alexi Massó Muñoz [1], Omar Rojas [2,3] and Guillermo Sosa-Gómez [2,*]

[1] Institute of Cryptography, University of Havana, Havana 10400, Cuba; camilo.denis@matcom.uh.cu (C.D.G.); daniel.frias@matcom.uh.cu (D.F.M.); alexi.masso@matcom.uh.cu (A.M.M.)
[2] Facultad de Ciencias Económicas y Empresariales, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Jalisco, Mexico; orojas@up.edu.mx
[3] Faculty of Economics and Business, Universitas Airlangga, Surabaya 60286, Indonesia
* Correspondence: gsosag@up.edu.mx; Tel.: +52-33-13682200

Abstract: Conventional electronic voting systems use a centralized scheme. A central administration of these systems manages the entire voting process and has partial or total control over the database and the system itself. This creates some problems, accidental or intentional, such as possible manipulation of the database and double voting. Many of these problems have been solved thanks to permissionless blockchain technologies in new voting systems; however, the classic consensus method of such blockchains requires specific computing power during each voting operation. This has a significant impact on power consumption, compromises the efficiency and increases the system latency. However, using a permissioned blockchain improves efficiency and reduces system energy consumption, mainly due to the elimination of the typical consensus protocols used by public blockchains. The use of smart contracts provides a secure mechanism to guarantee the accuracy of the voting result and make the counting procedure public and protected against fraudulent actions, and contributes to preserving the anonymity of the votes. Its adoption in electronic voting systems can help mitigate part of these problems. Therefore, this paper proposes a system that ensures high reliability by applying enterprise blockchain technology to electronic voting, securing the secret ballot. In addition, a flexible network configuration is presented, discussing how the solution addresses some of the security and reliability issues commonly faced by electronic voting system solutions.

Keywords: Hyperledger Fabric; blockchain; election; permissioned; permissionless; non-fungible token (NFT); Hardware Security Module (HSM); SoftHSM; enterprise blockchain

Citation: Denis González, C.; Frias Mena, D.; Massó Muñoz, A.; Rojas, O.; Sosa-Gómez, G. Electronic Voting System Using an Enterprise Blockchain. *Appl. Sci.* **2022**, *12*, 531. https://doi.org/10.3390/app12020531

Academic Editors: Zheng Chang, Jun Wu and Shancang li

Received: 27 November 2021
Accepted: 27 December 2021
Published: 6 January 2022

1. Introduction

A voting system implements a mechanism to choose between various options based on the decision of the voters; voting can be used in political elections, in companies, associations, even in organizational events. The adoption of a voting system depends on the needs and resources of the implementer. When considering a voting system, some critical factors must be taken into account [1]; otherwise, the voting process may be compromised. Among such factors one finds: voting untraceability, which refers to a vote not being traced back to the actual identity of the voter, ensuring that they are secure and confidential [2]; precision, or a system's ability to count votes accurately; unchangeability, which ensures that the system cannot be manipulated to favor one party over another; verifiability—since the entire electoral process must be verifiable, the system should have the capability to check a vote and determine if a vote has been altered; receipt-freeness, since it might be desirable that a voter does not create a receipt that shows how they voted [3–5]; dispute-freeness, so anyone can verify that the protocol runs correctly and that each voter acted according to the rules of the protocol; accessibility, to ensure that the voters cast their vote easily and accessibly; and decentralization, so the system does not allow a single entity to control the counting of votes and determine the outcome of an election. There are some electoral processes where the voting untraceability property is not necessary; for

example, in scenarios when the information of "who" voted "for whom" or "why" is in the public domain.

Voting security is a matter of interest to any company, institution, or country considering its use, especially when it comes to voting for a political candidate, a law or regulation. Thus, should one continue to rely on centralized voting systems? To replace a conventional system with a new voting system, it is essential to limit fraud and make the voting process attributable and verifiable [1,6]. What would happen if, instead of allocating long hours and many people for manual vote counting, there was a blockchain-based application, recording the vote of each person, ensuring that double voting is not possible, and guaranteeing its unalterability?

With the rise of Bitcoin and the increased popularity of other crypto systems like Ethereum, many institutions and people, in general, have been gaining knowledge about the key benefits that come with blockchain technology. Among such benefits, there is the possibility of developing decentralized apps that include cryptocurrencies and create the capability of managing industries' supply chains, financial systems, games, and other industries and scenarios. Electronic voting is one of the scenarios the scientific community tries to improve to increase integrity, anonymity, and non-repudiation. All those are fundamental requirements for a voting system, and blockchain may help achieve them that [7–16].

The main contribution of this paper is to present an alternative for participatory management processes, such as electronic voting, focusing on the following core values: trust, transparency, and immutability. Using Hyperledger Fabric as a framework, the paper describes the design and the proof of concept to a system that can be adapted to more than one electing scenario, integrated with a mechanism to protect cryptographic artifacts from users and network nodes.

This article is structured as follows: Section 2 presents simple definitions for blockchain and Hyperledger Fabric and exposes some of its main features and components. Section 3 explains some of the challenges related to the implementation of a voting system using blockchain and why Hyperledger Fabric was chosen as a framework to develop the proposed solution. Section 4 describes the proposed electronic voting solution, its design and network structure, along with other parts of the proposal like digital assets, protocol stages, identity management, and access control. Some notes related to implementation challenges and future research work are discussed in Section 5. Finally, Section 6 summarizes the relevance and implications of the proposed system.

2. Blockchain

From a broad perspective, enterprise blockchain technology is an information storage system also defined as a registry distributed among different members, such as companies, institutions, partnerships, corporations, among others, cryptographically protected, and organized in blocks of related transactions. Expressed from a systemic point of view, a blockchain is a decentralized universal registry, anonymous if desired, immutable, and free of falsifications, where a set of transactions is stored in organized blocks that are chained together. Each block is arranged chronologically and contains a block number, an alphanumeric code known as HASH digitally signed using public-key cryptography, the transactions carried out and the HASH of previous block, see Figure 1.

Among the main characteristics of the blockchain are: decentralized architecture thanks to a network of distributed peer nodes that communicate which each other (see Figure 2) and the use of consensus algorithms; each node of the same network has an exact copy of the stored data; transactions (data) are stored in blocks linked to each other using cryptographic techniques (hash functions, mainly), which facilitate the monitoring of any transaction; transactions are stored in blocks that remain unchanged over time; the blocks are stored in a database commonly called ledger, which incorporates advanced cryptography; there is distributed control over who can add new transactions in the ledger; for a new block to become a permanent part of the blockchain, it must have the consensus of all (or a part defined according to the protocol applied) of the nodes in the network.

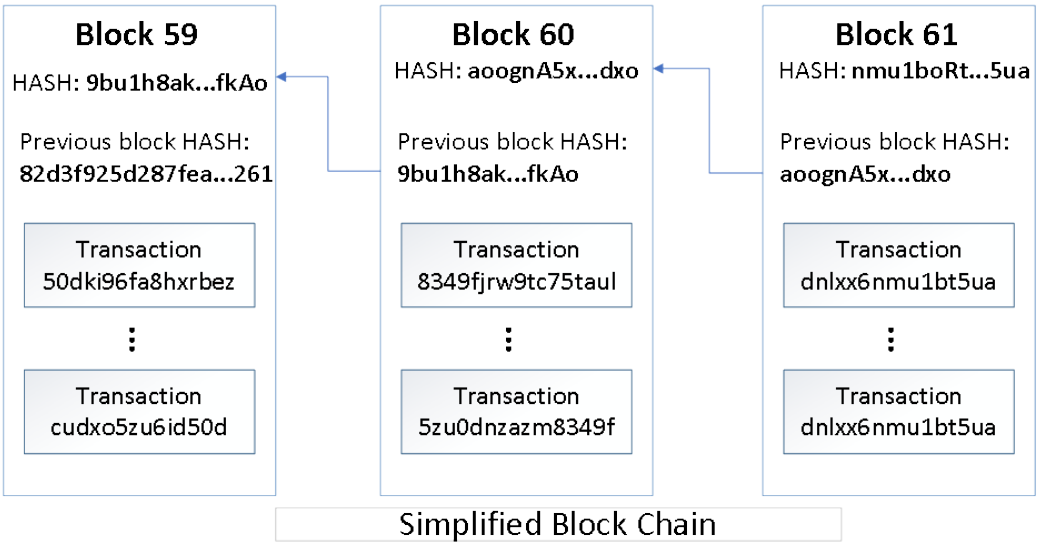

Figure 1. Blocks committed in a ledger.

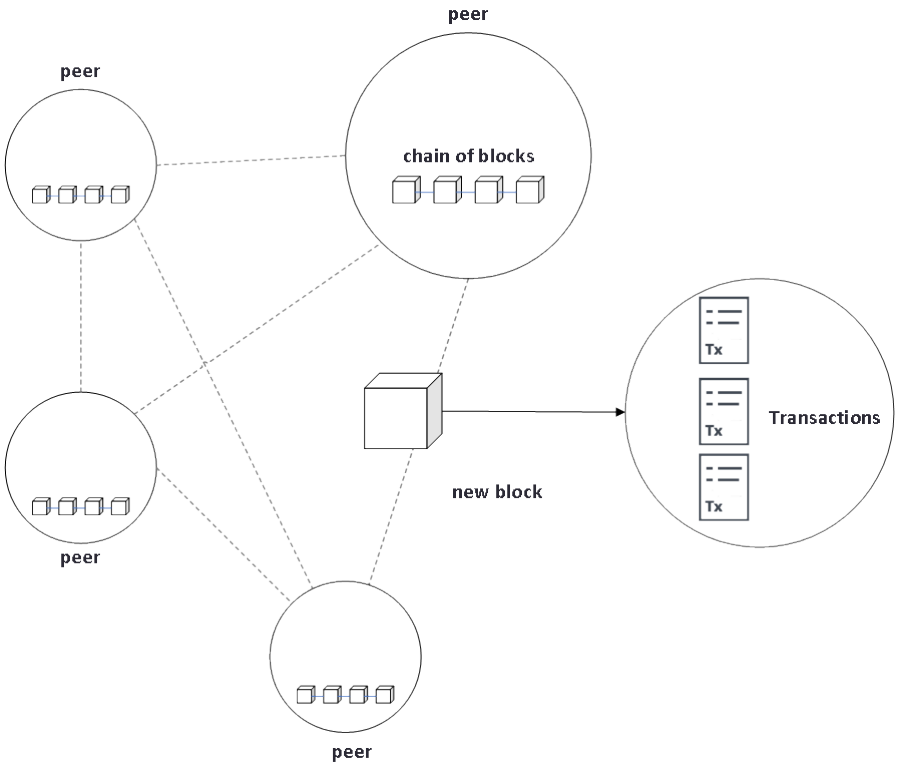

Figure 2. Distributed network between peers.

Currently, there are several types of blockchains, each with unique capabilities and characteristics that adapt to different needs. The most widely used terms to classify blockchains in the literature are: permissioned, permissionless, public, private, and hybrid [13].

2.1. Hyperledger Fabric (HF)

Hyperledger Fabric 2.2 [14] is the blockchain platform chosen to develop this Proof of Concept (PoC) of an electronic voting system. Hyperledger Fabric is an open-source enterprise-grade permissioned distributed ledger technology (DLT) platform established under the Linux Foundation [17,18]. It focuses on the use of authorized blockchain networks for distributed commercial applications with the following elements: consensus

protocol between nodes; Certification Authority (CA); peer-to-peer protocol; common and distributed database whose integrity is maintained by all the nodes in the network; support for smart contracts authored in general-purpose programming languages such as Go, JavaScript/Typescript, Java; mechanism to establish different levels of privacy and visibility of information among members (companies, institutions, partnership, corporation, among others).

2.1.1. Assets

Hyperledger Fabric provides the ability to modify assets through smart contract transactions. These can range from the tangible (real estate and hardware) to the intangible (intellectual property) and are represented as a collection of key-value pairs in binary and/or JSON format, with state changes recorded as transactions in a ledger.

2.1.2. Chaincode or Smart Contract

Chaincodes are codes written in a general-purpose programming language that is not affected by the natural language ambiguities that traditional contracts have. These automate the verification of compliance with the agreements established between the parties but written in lines of executable code by the network itself (and because they are part of the network, they are unalterable). The code and the agreements contained in it exist throughout the distributed and decentralized blockchain network. Transactions are traceable and irreversible, thus building trust among members. This enables companies to make better decisions quickly, saving time and reducing costs and risks. In other words, it is software which defines an asset or assets and the transaction instructions for modifying the asset(s).

2.1.3. Identities in Hyperledger Fabric

Since HF is an authorized network, participants—such as components (nodes) and users that interact with the blockchain network—in the blockchain network need to prove their identity (authenticate) to operate. This is achieved through verifiable identities of the Public Key Infrastructure (PKI) through a chain of trust. In HF, there is the network component called CA, which generates these public and private key pairs.

2.1.4. Membership Service Provider (MSP)

Since a private key should never leave its correspondent wallet, a mechanism is needed to verify (authenticate) the identity of the network participant. If the identity is verified, then this mechanism must decide what privileges this participant has. This is how an MSP turns identity into a role. So the purpose of an MSP in a network blockchain aims to verify the identities of the participants and determine the privileges assigned to a specific participant.

2.1.5. Wallet

A wallet contains a set of users' identities. An application executed by a user selects one of these identities when it connects to the blockchain network. Access rights to resources, such as the ledger, are determined using this identity in combination with an MSP.

3. The Case of Electronic Voting

3.1. Challenges

It seems to be necessary to improve the process of voting and counting, as well as the voter registration and validation process. Some of the most frequent flaws in electoral systems are the hacking of electronic voting devices and the manipulation of votes. Some articles [1,3,6–13,19] have already been written on how to implement a fraud-proof electronic voting system. However, despite the benefits that blockchain can bring to voting, there is still some skepticism about such a model; the most significant doubt relates to accessibility since, in most cases, voters are required to have a digital device with

Internet access, such as a mobile phone, a tablet, or a computer. Although decentralization helps to solve many of these problems, the system cannot prevent all possible types of electoral fraud.

Another challenge is the scalability of blockchain-based solutions. Scalability is the basic feature of traditional distributed systems, but it is hard to reach in blockchain due to decentralization requirements. Some factors that affect the performance of blockchain transactions are consensus mechanisms, network size, and transaction verification. Bitcoin and Ethereum 1.0 technologies are limited with respect to the rate of transactions per second [20,21], in public domain scenarios. Such a limitation can degrade the performance of a system, which is why some research [7,10] has focused on permissioned blockchain. Given that performance and scalability are extremely important to the Hyperledger Consortium, they decided to create the Performance and Scalability Working Group (PSWG https://wiki.hyperledger.org/display/PSWG/Performance+and+Scale+Working+Group (accessed on 23 October 2021) to discuss, research, and identify key metrics related to the performance and scalability of blockchain and blockchain-related technologies. Several research papers have been published studying and testing the performance capabilities of Hyperledger Fabric. The latest advancement scaled HF to 20,000 transactions per second [22].

3.2. Why Hyperledger Fabric?

Hyperledger Fabric is a distributed database that records batches of network transactions in blocks organized in a chain. Sophisticated cryptographic algorithms reinforce the integrity of each block in the chain. The ledger is created and maintained by a peer-to-peer (P2P) network, where each new block is committed in the global ledger by the members of the P2P network after the successful completion of the decentralized consensus procedure, finally endowing the system with a log of fully auditable transactions.

Permissionless and permissioned traditional blockchain platforms follow a sequential execution style, whereby transactions in smart contracts are typically executed after consensus or entwined with it and where all participants execute all contracts. The order-execute architecture shown in Figure 3 limits scalability, requires sequential transactions, and endorsement by all peers [23].

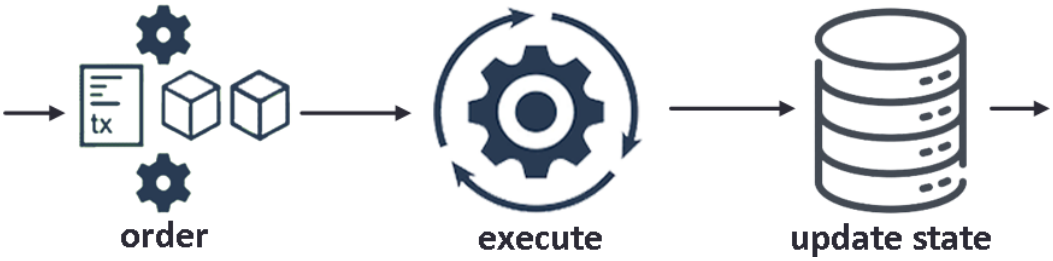

Figure 3. Traditional architecture for building blockchains.

HF was devised to use a different architecture that supports scalability and flexible trust assumptions. Rethinking the notion of permissioned blockchains occurs by introducing a cutting-edge approach that revamps the way blockchains manage non-deterministic events and security issues, such as resource exhaustion or Denial of Service (DoS) attacks [23]. Hyperledger Fabric uses a new execute-order-validate architecture, in which transactions are executed and endorsed first, before ordering them and validating that they do not conflict, as illustrated in Figure 4.

The execute-order-validate architecture departs radically from the order-execute paradigm because it separates the transaction flow into modular building blocks and includes elements of scalable replicated databases [23]. The platform supports pluggable consensus protocols that allow it to fit into specific trust models. One can take advantage of consensus protocols that do not require a native cryptocurrency. Avoidance of a cryptocurrency reduces some significant risk/attack vectors, and the absence of cryptographic mining operations means that the platform can be deployed with roughly the same opera-

tional cost as any other distributed system [24,25]. Cryptographic operations performed by HF nodes can be delegated to a Hardware Security Module (HSM) that protects private keys and handles cryptographic operations, allowing nodes to sign and validate transactions without exposing their private keys. HF currently leverages the PKCS11 standard to communicate with an HSM.

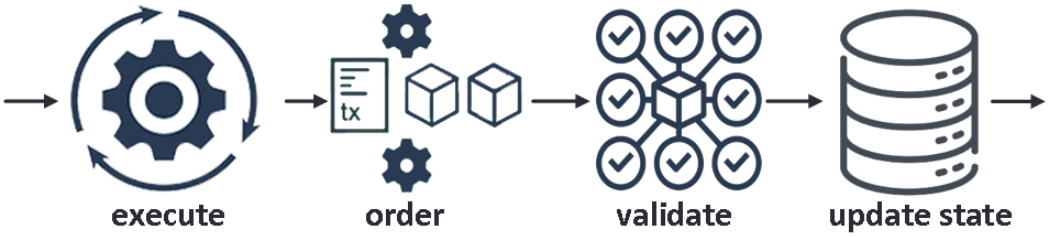

Figure 4. The transaction flow in Hyperledger Fabric.

In this way, HF allows organizations to collaborate in the creation of blockchain networks. Logically, an organization is a security domain and a unit of identity and credentials; additionally, each organization can deploy an unlimited number of nodes and its own CA to generate the identities of its nodes. "An organization can also be divided into multiple organizational units, each of which has a certain set of responsibilities, also referred to as affiliations. Think of an OU as a department inside an organization", which can be used in access control. Organizations/affiliations provide the ability to design custom network configurations customized to fit several scenarios. The combination of these features makes HF one of the better performing platforms [22,23,26] available today, both in terms of transaction processing and transaction confirmation latency; it also enables the use of smart contracts, privacy, and confidentiality of transactions [27], offering a versatile method to solve many of the problems of the electoral system by providing an efficient architecture for the execution, ordering, validation, and committing of transactions (votes) in blocks, the collection of data (vote count) and the declaration of results, while ensuring security.

4. System Proposal

In this section, we will describe an enterprise scenario involving two organizations using SuffrageNet, a voting network built on Hyperledger Fabric.

4.1. Introducing the SuffrageNet Network

The first step in determining a network structure for an application is to list the participating organizations. Logically, an organization governs one or more nodes in the network and depends on an MSP to issue identities for its participants.

Figure 5 shows the SuffrageNet network. This network consists of two organizations: SUFFRAGE, VALIDATOR. Each organization has at least one peer and one associated CA. In addition to the peers, the network consists of an MSP and an order service node (OSN) for each of the two organizations; the OSNs collaborate to create blocks and order them in a well-defined sequence. The ordering service—the ordering node and other orderer nodes form an ordering service—are distributed among all the organizations.

Each organization plays a specific role in the voting process. SUFFRAGE is the only one authorized to mint the Ballots to manage identity and issue voter transactions, while VALIDATOR is the only one authorized in the validation of the entire electoral process. The SUFFRAGE organization has both SUFFRAGE.MINTER and SUFFRAGE.VOTER OUs to reflect separate lines of business (minting and voting, respectively) and implement access control on a chaincode level. When organization CA issues X.509 certificates, the certificate OU-field specifies the affiliation for that identity. Defining affiliations for an organization can be useful when an organization has too many responsibilities or specifies some complex policies to restrict access even at the chaincode level (as in this proposal). In SuffrageNet, ev-

ery invokes/query transaction should be signed valid certificate issued by the organization CA (see Figure 6).

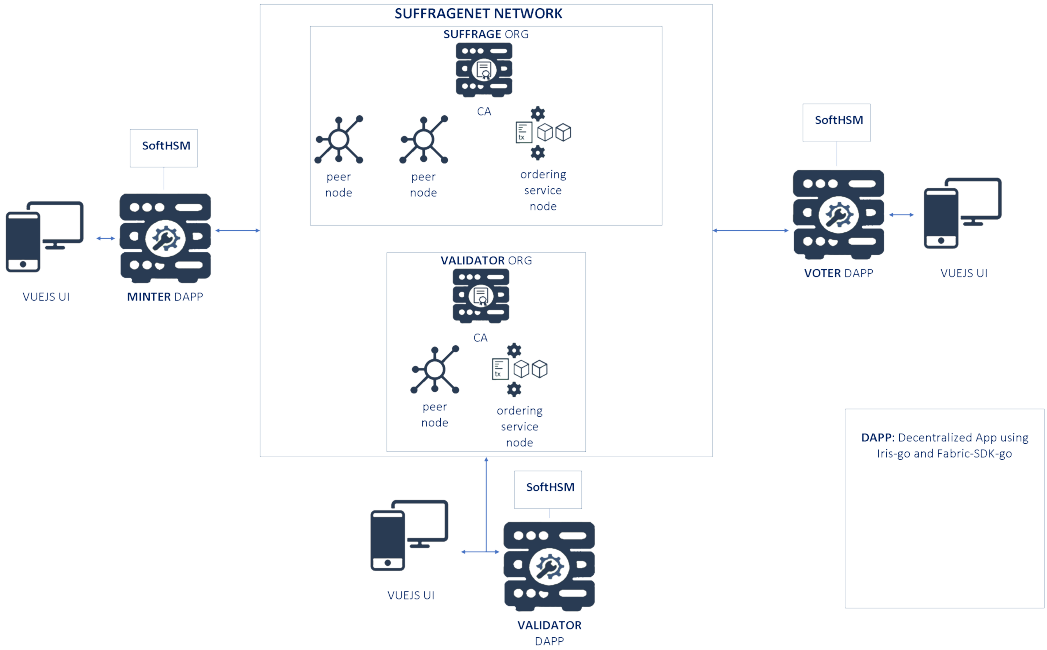

Figure 5. Proposed network structure.

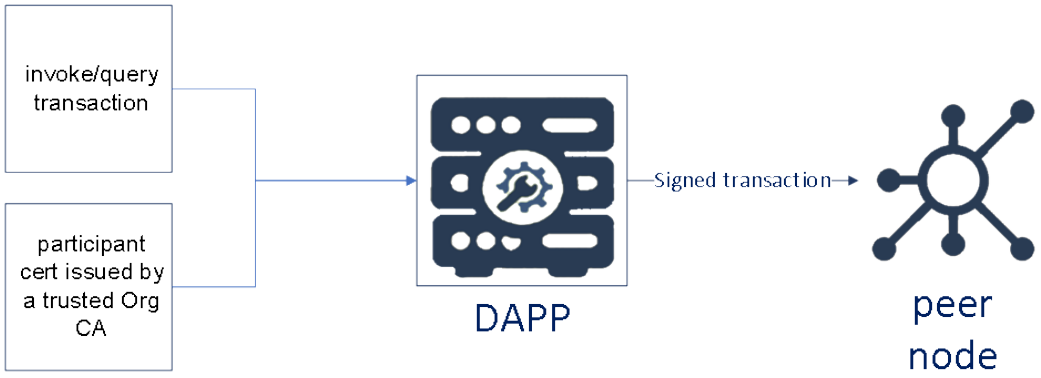

Figure 6. Invoke/Query transaction to peer via dApp.

The following sample from a signing certificate shows how the OUs are represented in this certificate, the affiliation is coded inside certificate subject, in OU. For example, subject for SUFFRAGE.MINTER.

'subject':'CN=33932069-6fce-4b16-8072-6715d591f35b,
OU=client+OU=SUFFRAGE+OU=MINTER,O=SUFFRAGE'

HF has a library known as client identity chaincode (CID), which was used within the suffrage chaincode to make access control decisions based on the client's identity (i.e., the invoker of the chaincode). In particular, access control decisions can be made based on any or a combination of the following information associated with the client: the client identity's MSP ID, an attribute associated with the client identity or an OU value associated with the client identity. Figure 7 shows how to check-in suffrage chaincode that the invoker of the chaincode is a dApp using an identity that belongs to the SUFFRAGE.MINTER affiliation.

```
// Demonstrate the use of Access Control by checking
// to see if the caller has the "SUFFRAGE.MINTER" OUValue;
// if not, return an error.
//
found, err := cid.HasOUValue(ctx.GetStub(), OUValue: "SUFFRAGE.MINTER")
if err != nil {
    return err
}
if !found {
    return fmt.Errorf( format: "submitting client not authorized to perform this action, does not have SUFFRAGE.MINTER OU")
}
```

Figure 7. Access control in MINTER affiliation.

The proposed structure allows organizations to validate the information before committing it to the ledger, making the vote more challenging to manipulate. Likewise, it allows more flexibility in the network configuration and adapts it to various scenarios.

4.2. The Suffrage Chaincode

The proposed solution includes a chaincode called suffrage for the purpose of automating the electoral process, in which various assets are implemented.

4.2.1. Ballot

It is a non-fungible token (NFT) and they can only be minted by an authorized organization (see Figure 8). Basically, this implies that the NFT is created on the blockchain.

```
const dataPrefix = "identity~tokenId"

const minterMSPID = "MINTER"

// Ballot MUST emit when a single ballot is transferred
// The from field MUST be the minter organization address.
// The to field MUST be the voter anonymous identity.
// The result field MUST be voter response.
// When minting/creating tokens, the from field MUST be set to minter organization address.
type Ballot struct {
    From   string `json:"from"`
    To     string `json:"to"`
    Result rune   `json:"result"`
}
```

Figure 8. Ballot written in GO.

Identity Management and Access Control

For each user account, an anonymous client identity is issued by the SUFFRAGE CA. When the ballot is issued, the smart contract obtains the client identity that submitted the request, and stores the identity and token ID (random uuid) as the owner in the asset key/value in the public chaincode world state (see Figure 9). Each ballot has access control based on MSP ID (organization name) and affiliation. Only identities from MINTER affiliation of SUFFRAGE organization can mint ballots (see Figure 10).

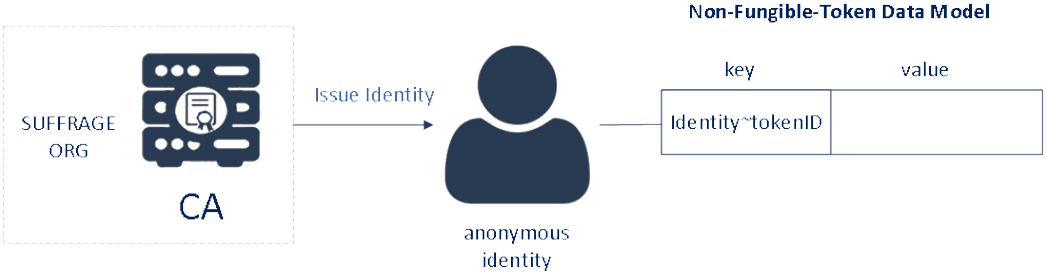

Figure 9. NFT Data Model.

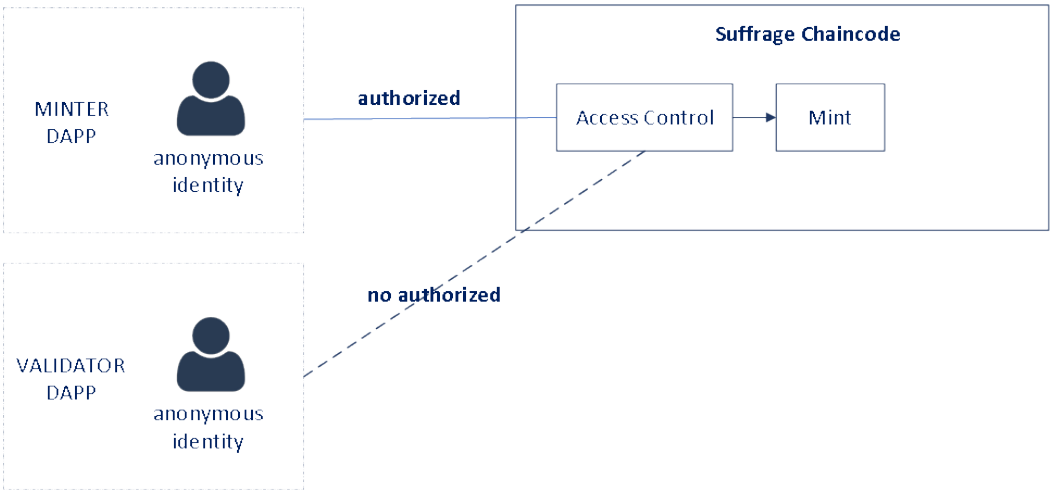

Figure 10. NFT Control Access.

Result Field

The NFT that represents a ballot has a result field r to store the voter's response. According to the solution proposal [19], this field is defined in order to not constrain the solution, so in this way the system can be adapted to several use cases of an electoral process.

We call the answers to the question q_i as A_i, and $|A_i|$ the number of all possible answers for the question q_i. Then the first $\lceil \log A_1 \rceil$ bits of the field will be reserved for the response q_1, the next $\lceil \log A_2 \rceil$ bits will be reserved for the answer to question q_2, and so on. For example, suppose that the ballot contains three questions q_1, q_2, q_3. For q_1 we have 10 possible answers, for q_2—eight possible answers, and q_3—three possible answers. Therefore, the first $\lceil \log 10 \rceil = 4$ bits will be reserved to encode the response A_1, the next $\lceil \log A_1 \rceil = 3$ bits will be reserved for A_2, and the next $\lceil \log 3 \rceil = 2$ bits for A_3. This gives us $4 + 3 + 2 = 9$ bits reserved for encoding responses. Therefore, this voting representation mechanism is a matter of bits interpretation, which means that it can be adapted to different data schemas. Then r is encrypted to avoid calculating the results during the election process, and is done using the public keys of the organizations.

4.2.2. Election

The election defines information about the electoral process, considering aspects such as number of voters, duration of the process, country, state, and locality (see Figure 11).

```
// Election
// The docType field is used to distinguish the various types of objects in state database
// The payload field is used to store extra data
type Election struct {
    DocType   string `json:"docType"`
    ID        string `json:"id"`
    Name      string `json:"name"`
    Country   string `json:"country"`
    Locality  string `json:"locality"`
    StartTime string `json:"startTime"`
    EndTime   string `json:"endTime"`
    State     string `json:"state,omitempty" metadata:",optional"`
    Payload   string `json:"payload"`
}
```

Figure 11. Election asset written in GO.

4.3. Procedure

The system must prevent unauthorized users from participating in the vote while providing the highest level of transparency. To identify a user, the identity provider (IdP)

must be used to certify its identity, IdPs are mechanisms widely used in authentication processes. During the elections, the authorization server may be the national digital identity provider available in many countries. For smaller or more specific scenarios, email can be used as a provider of signed tokens (e.g., JWT; JSON Web Token is an open standard based on JSON proposed by the IETF for the creation of access tokens) for addresses of eligible users.

The CA of the SUFFRAGE organization is integrated with the IdP; this allows it to generate a cryptographic identity (public and private key pair) for each voter previously verified by the IdP; it is also integrated with the SoftHSM—a software emulation of an HSM—where module where said identities are stored, once inside SoftHSM, cryptographic material can never be extracted or viewed. In addition, the use of SoftHSM makes it possible to take advantage of some security functionalities that a true HSM offers, making the proposal more viable in the sense of cost, simplicity and ease of maintenance.

The MINTER affiliation is the only one authorized at the chaincode level to mint the fixed number of ballots necessary for the voting process; these ballots can only be generated on a single occasion per election. In short, the entire sub-process means that the only participants authorized to mint the ballots are those governed by the SUFFRAGE organization; in the same way, only transactions issued by MINTER dApp will be allowed.

The Election asset (see Section 4.2.2) stores some public data so all organizations can consult those data. The fields to be defined for the asset depend on each voting process. In this proposal, it is assumed that there is no objection in that the number of voters has information in the public domain and is stored in the ledger when a specific electoral process is created. For example, the number of voters in an election process for a particular professional or labor organization may be irrelevant to the rest of the citizenry, but the fact that these data are in the public domain is not a problem.

Voting is done by issuing an anonymous transaction to the network blockchain that modifies some token states. A new state is stored by linking the identifier of the ballot with the transaction ID. This last step guarantees Verifiability, as this ID allows the system to track any transaction.

4.4. Phases of the Proposed Protocol

The protocol is classified into three phases, in which each phase depends on another.

4.4.1. Pre-Voting Phase

1. The network operators of each organization (see Figure 12) are authorized to deploy their nodes, join the blockchain network, and install the suffrage chaincode.
    a. The suffrage chaincode implements the InitLedger function to populate the ledger (Election asset) with some initial data.
2. The MINTER dApp interacts and initiates the electoral process.
    a. The user coordinator authorized identity to perform administration operations at the application level. Figure 12 interacts with the MINTER dApp to mint the necessary Ballots

4.4.2. Voting Phase

The only ones on the SuffrageNet authorized to send voting requests (transactions) and interact with the ledger are the SUFFRAGE.VOTER organization. It is valid to clarify that each transaction issued will be executed and endorsed transaction signed by a peer by each network peer (see Section 3.2) before being committed in the ledger.

1. The voter interacts with the application Figure 5 using one of the known methods, for example, email/password, one-time code, email confirmation, OAuth2, among others.
2. The IdP validates the voter as a valid user.
3. The SUFFRAGE CA registers and enrolls the anonymous identity in the SoftHSM.
4. Voter selects candidate on ballot.

5. The voter casts the vote:
    a. The voter signs the result, this operation returns a signed-result;
    b. The result-signed is encrypted with the public keys of the member organizations.

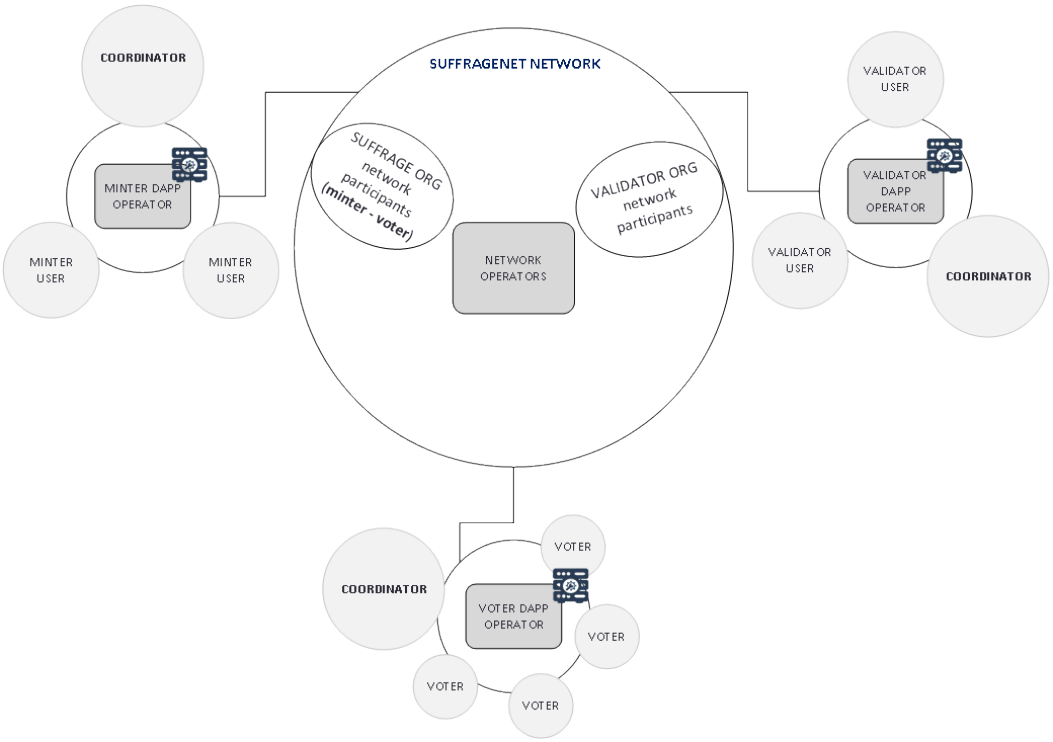

Figure 12. Depiction of users and operators of the blockchain network.

4.4.3. Post-Voting Phase

1. A coordinator user of the "MINTER" dApp ends the electoral process.
2. An event is broadcast to all organizations to approve the vote count:
    a. If they approve the count, the private key used to encrypt the result of each ballot is issued;
    b. The system processes all ballots and counts.
3. The final result is stored in the ledger.

5. Discussion and Future Work

The system proposed in this paper can satisfy several requirements of a voting system. The use of Hyperledger Fabric technology and the implementation of a chaincode guarantees each vote's immutability, integrity, and traceability, since actions such as vote counting are implemented at the chaincode level and do not require a central authority. In addition, the proposal allows the possibility of integrating the solution with Hyperledger Explorer, a simple, powerful, well-maintained, open-source utility to browse activity on the underlying blockchain network. In addition, the proposed network structure allows for more flexibility in the network configuration and adapts it to various scenarios. Intermediate CAs (ICAs) are not mandatory, but to reduce the risk of the organization's CA (root) being compromised, one or more ICAs in SuffrageNet may be included. For example, one ICA for both MINTER and VOTER affiliations of the SUFFRAGE organization may be added. The network design supports it. Additionally, due to the treatment given to the ballot as NFT and the inclusion of a bitmap field in a said ballot, the system can be adjusted with relative ease to more than one voting scheme.

To use a SoftHSM in the proposed system, Hyperledger Fabric's nodes were modified concerning the BCCSP (Blockchain Cryptographic Service Provider) configuration section

and were recompiled. This configuration prepares the cryptographic standards and algorithms of Hyperledger Fabric's nodes so it can be integrated with a Hardware Security Module (HSM). This feature provides a higher security level in the solution. It is worth mentioning that a SoftHSM is not more secure than an actual HSM. In contrast, an actual HSM is designed with physical security in mind. For example, an HSM is equipped with tamper-detection circuitry for destroying any keys it contains when someone tries to open the hardware case and sniff out the keys with a logic probe. To ensure voters' privacy in their actions, the authentication of the cryptographic identity generated by the CA is decoupled. Each identity is stored in a SoftHSM, protected with the voter's credentials. A voter's signature is inserted into each vote and stored concerning the transaction ID to achieve verifiability. To improve voting untraceability, it was planned to replace the current MSP certificates format (X.509) with Identity Mixer. "Idemix is a cryptographic protocol suite, which provides strong authentication as well as privacy-preserving features such as anonymity, the ability to transact without revealing the identity of the transactor, and unlinkability, the ability of a single identity to send multiple transactions without revealing that the transactions were sent by the same identity" [28]. Regarding unlinkability, it is important to note that in X.509 certificate, all attributes have to be revealed to verify the certificate signature. This implies that all certificate usages for signing transactions are linkable. New X.509 certificates need to be used every time to avoid such linkability, which results in a complex key management process. Idemix helps avoid linkability concerning both the CA and verifiers since even the CA cannot link proofs to the original credential. Neither the issuer nor a verifier can tell whether two proofs were derived from the same credential (or from two different ones).

The solutions for voting systems using blockchain technology are still immature. The blockchain network can verify the validity of a vote, avoid double-counting, guarantee that votes are cast from an authorized device, but cannot guarantee that the device from which a vote is cast is free of malicious software. Everything indicates that the use of blockchain in voting solutions is evolving in systems similar to the direct recording electronic voting machine (DRE voting machine); the DREs have been successful due to their ease of use and the ability to vote. On the other hand, DREs have been frequently attacked by cybersecurity experts [29,30]. Some of those security issues can be prevented or mitigated through a blockchain.

6. Conclusions

The proposed system consists of a secure and straightforward architecture that uses enterprise blockchain technology. It is designed to drastically reduce the vote-counting time and provide a high traceability capacity, without requiring the high energy consumption of other blockchain technologies, to comply with the network's consensus during the realization of the vote. As a result of the proposal, validators can monitor the entire election process without violating voters' privacy. All this together allows companies, organizations, or institutions to carry out voting processes with high standards of auditability and security.

Author Contributions: Conceptualization, C.D.G., D.F.M., A.M.M. and G.S.-G.; Formal analysis, C.D.G.; Investigation, C.D.G., D.F.M. and G.S.-G.; Methodology, C.D.G., O.R. and G.S.-G.; Project administration, O.R. and G.S.-G.; Supervision, A.M.M.; Validation, C.D.G., D.F.M. and A.M.M.; Writing—original draft, C.D.G., D.F.M., A.M.M. and G.S.-G.; Writing—review and editing, O.R. and G.S.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, K.M.; Arshad, J.; Khan, M.M. Secure digital voting system based on blockchain technology. *Int. J. Electron. Gov. Res.* **2018**, *14*, 53–62. [CrossRef]
2. Alam, K.R.; Maruf, A.; Rakib, R.R.; Ali, G.G.N. An Untraceable Voting Scheme Based on Pairs of Signatures. *Int. J. Netw. Secur.* **2018**, *20*. [CrossRef]
3. Jonker, H.L.; De Vink, E.P. Formalising receipt-freeness. *Lect. Notes Comput. Sci.* **2006**, *4176*, 476–488. [CrossRef]
4. Delaune, S.; Kremer, S.; Ryan, M.D. Receipt-freeness: Formal definition and fault attacks. In Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005), Milan, Italy, 15–16 September 2005.
5. Benaloh, J.; Tuinstral, D. Receipt-free secret-ballot elections (Extended abstract). *Proc. Annu. Acm Symp. Theory Comput.* **1994**, *1295*, 544–553. [CrossRef]
6. Poniszewska-Marańda, A.; Pawlak, M.; Guziur, J. Auditable blockchain voting system—The blockchain technology toward the electronic voting process. *Int. J. Web Grid Serv.* **2020**, *16*, 1–21. [CrossRef]
7. Khan, K.M.; Arshad, J.; Khan, M.M. Investigating performance constraints for blockchain based secure e-voting system. *Future Gener. Comput. Syst.* **2020**, *105*, 13–26. [CrossRef]
8. Taş, R.; Tanrıöver, Ö.Ö. A systematic review of challenges and opportunities of blockchain for e-voting. *Symmetry* **2020**, *12*, 1328. [CrossRef]
9. Hjálmarsson, F.Þ; Hreiðarsson, G.K.; Hamdaqa, M.; Hjálmtýsson, G. Blockchain based e-voting system. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018.
10. Daramola, O.; Thebus, D. Architecture-centric evaluation of blockchain-based smart contract e-voting for national elections. *Informatics* **2020**, *7*, 16. [CrossRef]
11. Alvi, S.T.; Uddin, M.N.; Islam, L. Digital voting: A blockchain-based e-voting system using biohash and smart contract. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 228–233.
12. Arnob, M.U.M.S.; Sarker, N.; Haque, M.I.U.; Sarwar, M.G. Blockchain-based secured e-voting system to remove the opacity and ensure the clarity of election of developing countries. *Int. Res. J. Eng. Technol. (IRJET)* **2020**, *7*, 1826–1831.
13. Kamil, M.; Bist, A.S.; Rahardja, U.; Santoso, N.P.L.; Iqbal, M. Covid-19: Implementation e-voting Blockchain Concept. *Int. J. Artif. Intell. Res.* **2021**, 25–34. [CrossRef]
14. Gaur, N.; O'Dowd, A.; Novotny, P.; Desrosiers, L.; Ramakrishna, V.; Baset, S.A. *Blockchain with Hyperledger Fabric: Build Decentralized Applications Using Hyperledger Fabric 2*; Packt Publishing Ltd.: Birmingham, UK, 2020.
15. Bulut, R.; Kantarcı, A.; Keskin, S.; Bahtiyar, Ş. Blockchain-based electronic voting system for elections in Turkey. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; pp. 183–188.
16. Al-Maaitah, S.; Qatawneh, M.; Quzmar, A. E-Voting System Based on Blockchain Technology: A Survey. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; pp. 200–205.
17. Burke, J.J. Distributed Ledger Technology. In *Financial Services in the Twenty-First Century*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 131–154.
18. Aggarwal, S.; Kumar, N. Hyperledger. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2021; Volume 121, pp. 323–343.
19. Barański, S.; Szymański, J.; Sobecki, A.; Gil, D.; Mora, H. Practical I-voting on stellar blockchain. *Appl. Sci.* **2020**, *10*, 7606. [CrossRef]
20. Sallal, M.; Owenson, G.; Adda, M. Security and Performance Evaluation of Master Node Protocol in the Bitcoin Peer-to-Peer Network. In Proceedings of the IEEE Symposium on Computers and Communications, Rennes, France, 7–10 July 2020. [CrossRef]
21. Dabbagh, M.; Kakavand, M.; Tahir, M.; Amphawan, A. Performance Analysis of Blockchain Platforms: Empirical Evaluation of Hyperledger Fabric and Ethereum. In Proceedings of the IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAIET 2020, Kota Kinabalu, Malaysia, 26–27 September 2020. [CrossRef]
22. Gorenflo, C.; Lee, S.; Golab, L.; Keshav, S. FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. *Int. J. Netw. Manag.* **2020**, *30*. [CrossRef]
23. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
24. Barger, A.; Manevich, Y.; Meir, H.; Tock, Y. A Byzantine Fault-Tolerant Consensus Library for Hyperledger Fabric. In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021; pp. 1–9.
25. Fu, W.; Wei, X.; Tong, S. An Improved Blockchain Consensus Algorithm Based on Raft. *Arab. J. Sci. Eng.* **2021**, *2021*, 1–13. [CrossRef]
26. Ayub Khan, A.; Laghari, A.A.; Shaikh, A.A.; Bourouis, S.; Mamlouk, A.M.; Alshazly, H. Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher Education Commission. *Appl. Sci.* **2021**, *11*, 10917. [CrossRef]
27. Ma, C.; Kong, X.; Lan, Q.; Zhou, Z. The privacy protection mechanism of Hyperledger Fabric and its application in supply chain finance. *Cybersecurity* **2019**, *2*, 1–9. [CrossRef]

28. Karagiannidis, N.; Nadales, D.; Truu, A.; Voutsinas, N.; Zacharias, T. Report on Tools for Secure Ledger Systems; PRIViLEDGE Consortium. 2021. Available online: https://media.voog.com/0000/0042/1115/files/D4.4%20%E2%80%93%20Report%20on%20Tools%20for%20Secure%20Ledger%20Systems.pdf (accessed on 23 October 2021).
29. Filiol, E. Unconventional attack against voting machines enlarging the scope of cybersecurity risk analysis. In Proceedings of the 7th International Conference on Information Systems Security and Privacy, Online Streaming, 11–13 February 2021; pp. 763–770. [CrossRef]
30. Dunn, M.; Merkle, L. Overview of Software Security Issues in Direct-Recording Electronic Voting Machines. In Proceedings of the ICCWS 2018 13th International Conference on Cyber Warfare and Security, Washington, DC, USA, 8–9 March 2018; p. 182.