# $d$-BAEV: Distributed Blockchain-Based Anonymous Electronic Voting

Jian Ren, Ehab Zaghloul and Tongtong Li
Michigan State University, East Lansing, MI 48824 USA. (email: {renjian, ebz, tongli}@msu.edu)

***Abstract*—Electronic voting (e-voting) presents a convenient and cost-effective alternative to current paper ballot-based voting. It provides many benefits such as increased voter turnout and accuracy in the decision-making process. While presenting many improvements, e-voting still faces serious security challenges that hinder its adoption, especially when designed to be run over mobile devices. In this paper, we propose $d$-BAEV, a novel remote e-voting model for large-scale elections by introducing a novel distributed trust model through participation of (at least) two conflicting parties to ensure election integrity and accountability. $d$-BAEV is secure and preserves voter privacy through secure multi-party computations performed by parties of differing allegiances. It also leverages a blockchain running smart contracts as a publicly accessible and tamper-resistant bulletin board to permanently store votes and prevent double-voting. Our performance analysis and simulation results show that the proposed $d$-BAEV scheme is practical for large-scale elections in smartphone devices.**

***Index Terms*—e-voting, distributed, voter anonymity, universal verifiability, coercion-resistant, unlinkability, blockchain**

## I. Introduction

Election is the foundation of a democratic government and nation. More than half the world's countries are classified as democratic nations, employing governments that enforce and secure their democracies. While these governments may vary in structure, they all grant eligible members the ability to exercise their power by voting. However, guaranteeing that a democratic election is *free* and *fair* still remains a challenge for most governments. Let alone, proving the freeness and fairness of the election to everyone, especially to the losing candidates, is an even bigger challenge.

The most fundamental principle defining credible elections is that they must reflect the free expression of the will of the people. To achieve this, elections should be transparent, inclusive, and accountable, and there must be equitable opportunities to compete in the elections. Unfortunately, the most widely utilized casting techniques, such as Direct-Recording Electronic (DRE), cannot provide voter verification. The voters can never know whether their votes are being counted and the votes being counted are legitimate. While incorporating computerized systems such as optical scanners and Direct-Recording Electronic (DRE) along with cryptographic primitives could help reduce the human factor intervention and may even offer verifiability, these systems still present computer vulnerabilities [1], [2]. Various schemes [3], [4] aim to minimize such vulnerabilities and provided end-to-end (E2E) verifiable voting. However, the mandatory requirement of voting at a poll-site using either technique interferes with the availability and accessibility fairness and overall voter turnout. To overcome this issue, some elections may permit remote voting through absentee/mail-in ballots allowing voters to vote at other poll-sites or return their ballots via mail. Yet, these method raise concerns that ballots may be subject to fraud/manipulation during the process of transmission.

There was a saying that went, *those who cast the votes decide nothing; those who count the votes decide everything*. The integrity of an election depends on how the entire electoral process is monitored and by whom. It is one of the safeguards that help to protect the viability and honesty of the election administration, as well as ensuring fair participation by election participants.

Unfortunately, this is extremely challenging, as had been made clear by President John Adams in 1798: "*Our Constitution was made only for a moral and religious people. It is wholly inadequate to the government of any other*." President Adams, and many others, believed that "*in a society riddled with vices, the various mechanisms created by the Constitution would not be able to function properly, with the result that the democratic republican order would be eventually supplanted by despotism* [5]." Therefore, it is vital to create a system that can ensure election is transparent to everyone, free of any moral or religions prerequisite.

Recent research has presented several remote electronic voting systems [6], [7]. Unfortunately, these schemes were not designed for large-scale elections and cannot ensure election freeness and fairness. In this paper, we introduce a novel voting scheme that enables free and fair large-scale elections. Our proposed scheme leverages the existence of at least two parties in an election with different allegiances, who engage in a multi-party computation along with the voters. We assume that it conflicts with the interests of these parties to collude or exchange any information during the election process that may sacrifice the winning chances of the candidates they support.

The model presented in this paper is primarily based on the US general election. It expands significantly on [8], [9] in that we show orthogonality between universal verifiability and coercion-resistance in our proposed scheme, allowing for a trade-off between the two. We also prove that it is computationally infeasible to link any vote to the voter or multiple votes cast by one voter.

The rest of the paper is organized as follows: In Section II, security model and security goals are described. We present the system components in Section III. The election system setup is discussed in Section IV Voter ballot request and casting vote are discussed in Section V and Section VI followed by casting votes in Section VI tabulation in Section VII. Security analysis is presented in Section VIII. We provide performance analysis in Section IX. The design trade-offs is presented in Section X and conclude the paper in Section XI.

## II. Distributed Trust Model and Design Goals

The trust model is one of the most crucial elements of an election, as it determines who will oversee the process and ultimately shape its outcome. In principle, the election administration should be distributed and not controlled by or

This work was supported in part by NSF under Grant CCF1919154 and Grant ECCS-1923409.

in favor of any particular party. Unfortunately, in practice, it is one of the most challenging issues due to human factors, and as a result, most elections are not fully distributed, which is the root cause of most issues in recent elections. A distributed trust model is indispensable in free and fair elections. It can also provide a solution to all election irregularities [10]–[12], regardless of the ruling or opposition parties.

In the traditional paper ballot-based voting, election integrity depends on the joint effort of multiple conflicting parties for voter authentication, ballot acquiring, election auditing, and monitoring. Unfortunately, due to human factors, it is practically impossible to ensure election accuracy.

In this paper, our goal is to develop a distributed electronic voting scheme that can ensure the principles and values established by the founding fathers are reflected in modern elections. In particular, it satisfies the following design requirements:

*Distributed trust:* Unlike existing election schemes, our proposed election scheme is not fully controlled by any single party. Instead, we assume that an election will be organized by a registrar $\mathcal{R}$ and a moderator $\mathcal{M}$ with conflicting interests. They need to cooperate for both ballot distribution and vote casting even though neither $\mathcal{R}$ nor $\mathcal{M}$ trusts the other party and is fully trusted by all voters. However, due to the conflicting interests between them, they are unlikely to collude.

*Voter eligibility:* Voting is limited to eligible voters. This requires proper voter registration that determines and confirms the eligibility of voters, granting them voting rights.

*Double-voting resistant:* Each eligible voter is entitled to only a single vote counted toward the election. While submitting multiple votes is permitted, only one vote will be counted toward the tallied results as predefined by the election.

*Anonymity:* A cast vote cannot be linked to the identity of a voter. This protects voters and allows them to freely voice their desired opinions.

*Coercion-resistant:* Remote voting may expose voters to coercion. If subject to coercion, our goal is to provide voters the capability to cast votes as instructed under coercion, whilst still protecting their actual votes.

*Voter verifiability:* Voters can verify that their votes have been properly cast and counted towards the election results.

*Universal verifiability:* Anyone can verify and determine if all of the ballots have been counted correctly.

*Election result manipulation-resistant:* Last-minute voters cannot manipulate the election results in a close race. This requires that all information related to votes be concealed until the voting phase is over.

*Network adversary:* We assume adversaries are capable of monitoring public communication channels and performing general network-level attacks. Our proposed scheme that builds on cryptographic operations should be able to limit the impact of network adversaries to at most DoS attacks since all operations can all be verified by the voters.

## III. System Components

Our election system consists of six entities, where the first three entities are the active participants of the election and the rest three are the supporting entities involved in the election.

*Voters:* The eligible voters $\mathcal{V} = \{v_i \mid 1 \leq i \leq n\}$ can cast votes in an election. The set of eligible voters is public and subject to audits.

*Registrar $\mathcal{R}$:* The first election organizing entity that is responsible for generating digital ballots to be shared with voters anonymously.

*Moderator $\mathcal{M}$:* The second election organizing entity that has conflicting interests with the registrar and is responsible for concealing the identities of the voters, delivering the ballots to them anonymously, and performing secure multi-party computations with the registrar to decrypt all the ballots.

*Election candidates:* The eligible set of candidates $\{\mathfrak{c}_i \in \mathfrak{C} \mid 1 \leq i \leq c\}$ running in an election.

*Blockchain network:* In the proposed $d$-BAEV, we utilize a permissionless blockchain as a bulletin board where voters cast their votes by interacting with the election smart contracts deployed over the blockchain. A brief discussion of the main components of blockchains can be found in [13].

*Tallying authority:* A party that performs vote tabulation at the end of the casting phase. This task is performed and publicly monitored. It can be performed by anyone monitoring the blockchain.

## IV. Election System Setup

Voter registration is crucial in preventing election fraud by ensuring that only eligible voters can cast their ballots. In this paper, we can incorporate the current voter identity verification method into our voter registration process, which complies with both federal and state election laws.

Let $\mathbb{G}$ be a publicly chosen multiplicative cyclic group of prime order $p$ and $g$ is a generator of $\mathbb{G}$.

*1) Setup:* The registrar and the moderator select their private keys $x_r \in_r \mathbb{Z}_p^*$ and $x_m \in_r \mathbb{Z}_p^*$, then compute their corresponding public keys as:

$$y_r = g^{x_r} \bmod, \quad y_m = g^{x_m} \bmod p,$$

respectively, where $\in_r$ denotes select randomly.

*2) Voter Registration:* This is an offline process and similar to the existing voter registration and each voter is only required to do it once for all elections. At this phase, voters are required to prove their voting eligibility to the registrar by providing evidence, such as their identities before they are added to the electoral roll.

*a) Voter Key Generation and Registration:* Each voter $v_i$ selects a private key $x_i \in_r \mathbb{Z}_p^*$ and computes the corresponding public key as:

$$y_i = g^{x_i} \bmod p.$$

The public key $y_i$ is shared with the registrar during registration.

*b) Signing Voter's Public Key:* After verifying the eligibility of the voters, the registrar signs their corresponding public keys. It then selects $u_i$ randomly from $\mathbb{Z}_{p-1}^*$ such that $\gcd(u_i, p-1) = 1$ and computes the following:

$$w_i = g^{u_i} \bmod p, \quad s_i = (h(y_i) - x_r w_i) u_i^{-1} \bmod (p-1). \quad (1)$$

The signature of $y_i$ is defined as

$$\mathsf{sign}_{\mathcal{R}}(y_i) = (w_i, s_i). \quad (2)$$

After this process, the voter is added to the electoral roll.

### A. Election Ballots Generation

Before an election gets started, as the election official, the registrar needs to configure the election system based on the

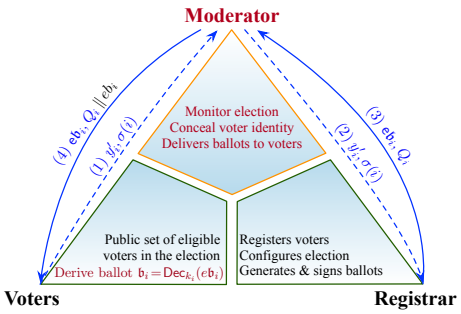



Fig. 1: The distributed ballot acquiring model.

election laws and also generates the election ballots. This process should be audited.

*1) Ballots Generation:* The registrar generates $n$ unique digital ballots $\mathcal{T} = \{\mathfrak{b}_i \,|\, i \in \mathbb{Z}_n\}$, where

$$\mathfrak{b}_i = \mathsf{Election}_{\mathsf{ID}} \,\|\, \mathsf{Election}_{\mathsf{TimeStamp}} \,\|\, \mathsf{Rand}_{\mathsf{i}}, \tag{3}$$

and $\mathsf{Rand}_i \in_r \mathbb{Z}_e$ for some integer to ensure ballots uniqueness and unpredictability. The registrar then digitally signs $\mathfrak{b}_i$ using ElGamal signature scheme. Let

$$\mathcal{B} = \{\mathfrak{b}_i \,\|\, \mathsf{Sign}(\mathfrak{b}_i) \,|\, i \in \mathbb{Z}_n\}$$

be the set of digitally signed ballots.

Next, it performs a permutation $\pi \colon \mathcal{B} \to \mathcal{B}$, so that it is infeasible for any other parties to link the ballots based on the orders.

## V. Voter Ballot Request and Distribution

In US general elections, voters get their ballots from their polling precincts or through mail-in absentee ballots. The difference between these two approaches is the security of in-person authentication versus signature authentication, as well as the implications for voter anonymity and attempts at multiple voting. It is clear that the in-person authentication model will always be more secure than any other options.

Our proposed $d$-BAEV scheme follows the traditional in-person voting model in voter authentication and ballot acquisition, which is a distributed model that requires participation of both the registrar and the moderator. It is computationally infeasible for anyone to link the voters and their ballots or acquire an illegal ballot. Therefore, it is much more secure than mail-in absentee ballots.

The ballot acquisition process includes three major steps: (i) the voters request their ballots through the moderator, who permutes the voters before sending the requests to the registrar to prevent anyone from linking the voters and their ballots based on the time or sequence of the ballot request; (ii) the moderator obscures the identity of the voters to make it infeasible for anyone to link the identity of the voters and their ballots; and (iii) the registrar assigns the ballots to the voters and sends them to the voters through the registrar, which makes the ballots assignment anonymous to everyone.

The distributed ballot acquisition model is shown in Figure 1.

### A. Voter Ballot Request

Voters only interact with the registrar during the ballot request. However, even for this process, the moderator acts as an intermediary and conceals voters' identities from the registrar during ballot distribution. Prior to this process, the moderator generates a one time permutation $\sigma$ on $\mathcal{V} = \{v_i \,|\, 1 \le i \le n\}$ so that it is infeasible for any other party to link voters' IDs to their ballots based on the electoral roll.

Voters initiate the request by sharing their public keys $y_i$ and the corresponding signature $\mathsf{Sign}_{\mathcal{R}}(y_i) = (w_i, s_i)$, defined in equation (1) and equation (2), with the moderator. The moderator validates the shared signature $(w_i, s_i)$ corresponding to $y_i$ in the electoral roll list through the following equation:

$$g^{h(y_i)} \equiv y_i^{w_i} \cdot w_i^{s_i} \bmod p. \tag{4}$$

The moderator can verify that the voter indeed possesses the private key associated with the public key in the electoral roll through a challenge-response based authentication, and/or other security credentials, such as DOB, SSN, driver's license number, etc., to ensure proper voter authentication.

### B. Voter Identity Obscuring (Moderator)

After receiving ballot requests from voters, and upon successful voter authentication, the moderator selects a random blinding factor $b_i \in_r \mathbb{Z}_p^*$ and conducts voter identity obscuring to conceal their identities, which is designed to ensure that the registrar cannot link the voters to the ballots assigned to them, as follows:

$$y_i' = y_i^{b_i} = g^{x_i b_i} \bmod p, \tag{5}$$

and sends $y_i'$ and $\sigma(i)$ to the registrar.

### C. Anonymous Ballot Assignment (Registrar)

To ensure anonymity in ballot assignment, voters must request their ballots from the moderator, who will then obscure their identity before passing on the ballot request to the registrar for assignment.

The registrar assigns ballots defined in equation (13) to the blinded voters as follows: $\mathfrak{b}_i = \pi(\sigma(i))$.

To conceal $\mathfrak{b}_i$ from the moderator as well as any other parties, the registrar encrypts the ballot as follows:

$$e\mathfrak{b}_i = \mathsf{Enc}_{k_i}(\mathfrak{b}_i), \tag{6}$$

where the encryption key $k_i$ is derived from:

$$k_i = (y_i')^{q_i} = g^{x_i b_i q_i} \bmod p, \tag{7}$$

and $q_i \in_r \mathbb{Z}_p^*$. $\mathsf{Enc}$ is a symmetric encryption such as AES encryption that conceals the ballot from the moderator and enables the registrar to share this ballot with the voter anonymously. The registrar also generates an ephemeral key $Q_i$ that would allow the voter to regenerate $k_i$ as follows:

$$Q_i = g^{q_i} \bmod p. \tag{8}$$

### D. Encrypted Ballot Transmission

The registrar sends the encrypted ballot $e\mathfrak{b}_i$ and the ephemeral key $Q_i$ to the moderator. Once received, the moderator forwards $e\mathfrak{b}_i$ and $Q_i$ to the voter along with the encrypted blind factor $eb_i$:

$$eb_i = (g^{r_m},\, b_i \cdot y_i^{r_m}) = (eb_{i,1},\, eb_{i,2}) \bmod p, \tag{9}$$

where $r_m \in_r \mathbb{Z}_p^*$.

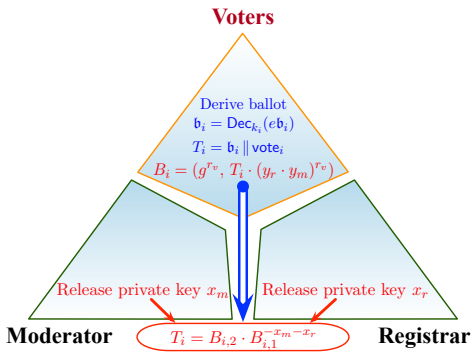



Fig. 2: The distributed vote casting model.

### E. Deriving Ballot

The voter decrypts $eb_i$ and derives the blind factor $b_i$:

$$b_i = eb_{i,2} \cdot eb_{i,1}^{-x_i} = (b_i \cdot y_i^{r_m}) \cdot (g^{r_m})^{-x_i} \bmod p. \tag{10}$$

Next, the voter derives key $k_i$:

$$k_i = Q_i^{x_i b_i} = g^{q_i x_i b_i} \bmod p. \tag{11}$$

Finally, the voter decrypts $e\mathfrak{b}_i$ from symmetric encryption:

$$\mathfrak{b}_i = \mathsf{Dec}_{k_i}(e\mathfrak{b}_i). \tag{12}$$

## VI. Casting Votes

In our proposed $d$-BAEV system, the ballots associated with the votes are double-encrypted, which requires cooperation between the registrar and the moderator. They need to perform a multi-party computation to decrypt the double-encryption and count the votes.

*1) Vote Double-Encryption:* The ballot associated with each vote defined as

$$T_i = \mathfrak{b}_i \,\|\, \mathsf{vote}_i, \tag{13}$$

where $\mathsf{vote}_i = (\mathfrak{c}_{i,1}, \cdots, \mathfrak{c}_{i,m})$ is a sequence of bits representing each candidate such that:

$$\mathfrak{c}_k = \begin{cases} 1, & \text{if voting for } \mathfrak{c}_k, \\ 0, & \text{if voting against } \mathfrak{c}_k. \end{cases} \tag{14}$$

$T_i$ is encrypted under the public keys of both the registrar $y_r$ and the moderator $y_m$ as follows:

$$B_i = (g^{r_v},\, T_i \cdot (y_r \cdot y_m)^{r_v}) = (B_{i,1},\, B_{i,2}) \bmod p. \tag{15}$$

*2) Submit Vote:* To submit the encrypted ballot, the voter calls the election vote smart contract that takes $B_i$ as input. A vote is permanently cast once the result of the smart contract is appended to the blockchain. This process involves importing the blockchain account of the voter that allows her to call the smart contract deployed over the blockchain and integrate the encrypted ballot as input. Once the smart contract transaction is appended to the blockchain, the voter receives a confirmation of the vote along with the transaction hash reference. The transaction hash is used to verify that the cast vote is stored permanently over the blockchain.

*3) Publishing Requested Ballots:* After the casting votes phase is completed, the registrar will publish the set of requested ballots $\mathcal{B}' \subset \mathcal{B}$ while the moderator will publish the list voters $\mathcal{V}' \subset \mathcal{V}$ who requested the ballots on the blockchain bulletin board. This process enables public audit of the ballot distribution while preventing any possible irregularities, including requesting duplicate ballots or using unrequested ballots to sway an election in a specific direction. Additionally, this design also provides ballot universal verifiability, as voters can confirm that the ballots they received are legitimate.

## VII. Votes Tabulation

Votes that appear on the blockchain within the casting votes phase are collected to be validated and counted. Both the moderator and registrar publicly disclose their private keys $x_m$ and $x_r$, which guarantees that all valid ballots will be treated uniformly, as shown in Fig. 2.

The tallying authority, which may consists of the registrar and the moderator working collaboratively, decrypts the collected ballots appearing on the blockchain as:

$$\begin{aligned} T_i &= B_{i,2} \cdot B_{i,1}^{-x_m} \cdot B_{i,1}^{-x_r} \\ &= T_i \cdot (y_r \cdot y_m)^{r_v} \cdot (g^{r_v})^{-x_m} \cdot (g^{r_v})^{-x_r} \bmod p. \end{aligned} \tag{16}$$

Based on equation (13), we can extract ballot $\mathfrak{b}_i$ from $T_i$. If $\mathfrak{b}_i \in \mathcal{B}'$, the tallying authority examines the associated vote sequence and increments the counter of each candidate accordingly. Each election may specify its own rules that disqualify votes which are cast incorrectly. For example, voting *yes* for two opposing candidates.

## VIII. Security and Privacy Analysis

In this section, both security and voter privacy of the proposed $d$-BAEV scheme will be analyzed.

### A. Double-Voting

The term *double-voting* refers to the malicious act of attempting to cast more than one vote in order to swing the outcome of an election. In the United States, while the majority of states prohibit voting twice in the same election, only a few of them prohibit voting in more than one state for the same election [14]. This means that eligible voters could potentially register in more than one state and attempt to double-vote.

The process of detecting and penalizing such voters becomes expensive and challenging. In remote electronic-voting systems, voters receive digital ballots and cast their votes remotely rather than visiting a polling station. Since our proposed scheme uses the blockchain as a bulletin board to permanently store votes, it becomes simple to identify double-voting attempts that reuse digital ballots. However, the reuse of digital ballots is not the only method to attempt double-voting. Adversaries may attempt to double-vote by obtaining undeserved voting credentials giving them the right to cast more votes. We formally prove that our proposed electronic voting scheme is secure against such attempts.

**Definition 1** (Secure against double-voting)**.** *A voting scheme is said to be secure against double-voting if no PPT adversary is able to forge a digital ballot that is digitally signed by the registrar.*

**Theorem 1.** *It is infeasible for any adversary to commit double-voting if the DDH assumption holds.*

*Proof.* To perform double-voting, the adversary must be able to forge a legitimate ballot signed by the registrar using discrete logarithm, which conflicts with the DDH assumption. Therefore, it is infeasible for any adversary to vote more than once, or perform double-voting. □

### B. Voter Anonymity

Voter anonymity in elections is a cornerstone of the democratic process. It is also essential to the integrity of the electoral process and guards against coercion. To preserve the anonymity of voters, an adversary should not be able to link any vote to a specific voter.

**Definition 2** (Voter anonymity)**.** *A voting scheme is said to preserve the anonymity of voters if no PPT adversary is able to get a non-negligible advantage in deriving the identity of the voter of any cast ballot.*

The proposed $d$-BAEV scheme relies on multi-party computation performed by parties of different allegiances for ballot distribution. The moderator and the registrar would need to collude for ballots to be linked to the identities of voters. We assume collusion is not in the best interest of any of these parties, therefore, voter anonymity can be preserved if (i) the probability of the registrar to distinguish a public key $y_i$ that has been blinded and a randomly selected public key chosen by the adversary does not exceed $\frac{1}{2}$ significantly, and (ii) the moderator cannot derive an encrypted ballot.

We would like to introduce the following concept:

**Definition 3** (Indistinguishability under Chose-Plaintext Attack (IND-CPA))**.** *Let* $\mathcal{O}_1 = (\mathsf{BFGen}, \mathsf{Blind}, \mathsf{Rec})$ *be a DDH oracle, where* $\mathsf{BFGen}$ *is the blind factor generation function,* $\mathsf{Blind}$ *is the blind function, and* $\mathsf{Rec}$ *is the recovery function. The IND-CPA game consists of a set of interactions between a PPT adversary* $\mathcal{A}$ *and a challenger* $\mathcal{C}$ *acting as the moderator.*

1) $\mathcal{C}$ *computes a blind factor* $b = \mathsf{BFGen}(1^k)$ *and keeps it secret.*
2) $\mathcal{A}$ *does not have access to* $\mathcal{O}_1$*, however it may request that* $\mathcal{C}$ *blinds for it as many public keys as it likes during any time of the game.* $\mathcal{A}$ *then computes two public addresses* $y_0$ *and* $y_1$ *to be challenged against and sends them to* $\mathcal{C}$*.*
3) $\mathcal{C}$ *uniformly and randomly selects* $\mu \in_r \{0,1\}$ *then computes* $y' = \mathsf{Blind}(b, s, y_\mu)$*, where* $s$ *represents a randomness state to diversify the blind process and is a value that has not been used in any of the previously computed ciphertexts. Next,* $\mathcal{C}$ *sends* $y'$ *to* $\mathcal{A}$*.*
4) $\mathcal{A}$ *outputs a guess* $\mu'$ *of* $\mu$*.* $\mathcal{A}$ *wins the security game if* $\mu' = \mu$ *and loses otherwise.*

A scheme is IND-CPA secure if no adversary has a non-negligible advantage in winning the above game.

**Theorem 2.** *The proposed d-BAEV scheme preserves the voter anonymity if the DDH assumption holds.*

*Proof.* First, using the IND-CPA security game, the DDH assumption implies that the adversary is unable to get a non-negligible advantage from the IND-CPA security game in determining the public key that is blinded.

Assume there is an adversary that has non-negligible advantage $\varepsilon$ to distinguish a public key, that is $\mathsf{Adv}_{\mathcal{A}} > \frac{1}{2} + \varepsilon$. We construct a simulator $\mathcal{A}$ that can distinguish a DDH element from a random element with advantage $\varepsilon$. The DDH challenger begins by selecting the random parameters: $a, b, c \in_r \mathbb{Z}_p^*, \mu \in_r \{0,1\}$. Let $g \in \mathbb{G}$ be a generator and $Y$ is defined as $Y = g^{ab} \bmod p$ if $\mu = 0$, and $Y = g^c \bmod p$ otherwise. The simulator acts as the challenger in the IND-CPA game as defined before.

Given $\mathcal{A}$, if $Y = g^{ab} \bmod p$, then $y^*$ is a valid ciphertext, $\mathsf{Adv} = \varepsilon$ and

$$\Pr\left[\mathcal{A}\left(g, g^a, g^b, Y = g^{ab}\right) = 1\right] = \frac{1}{2} + \varepsilon. \tag{17}$$

If $Y = g^c \bmod p$ *or* $Y \neq g^{ab} \bmod p$ then $y^*$ is nothing more than a random value to the adversary. Therefore,

$$\Pr\left[\mathcal{A}\left(g, g^a, g^b, Y = g^c\right) = 1\right] = \frac{1}{2}. \tag{18}$$

From equation (17) and equation (18), we can conclude that

$$\left|\Pr\left[\mathcal{A}\left(g, g^a, g^b, Y = g^{ab}\right) = 1\right] - \Pr\left[\mathcal{A}\left(g, g^a, g^b, Y = g^c\right) = 1\right]\right| = \varepsilon.$$

The simulator plays the DDH game with a non-negligible advantage, which contradicts the DDH assumption. Therefore, neither the registrar nor any other adversary can get any advantage $\varepsilon$ to derive the identity of a voter that has been assigned a known ballot.

Second, we prove that the moderator cannot derive an encrypted ballot by the registrar.

Since each $\mathfrak{b}_i$ is encrypted by the registrar as shown in equation (6) prior to being shared with the moderator and the encryption key $k_i$ is generated by the registrar based on the blinded public key $y_i^{'}$ and a randomly select value $q_i \in_r \mathbb{Z}_p^*$, the moderator has no advantage in deriving the assigned ballot it transmits to the voter during the ballot transmission process. Therefore, our proposed scheme preserves voter anonymity against the registrar, moderator, and any other adversaries. □

### C. Coercion-Resistance and Ballot Unlinkability

In the proposed $d$-BAEV scheme, casting a vote traces back to the voters encrypting their desired votes as shown in equation (15). To prove that the proposed $d$-BAEV scheme is coercion-resistant, we use the IND-CPA game consisting of a DDH oracle $\mathcal{O}_2 = (\mathsf{KeyGen}, \mathsf{D\text{-}Enc}, \mathsf{D\text{-}Dec})$, where $\mathsf{KeyGen}$ is a public key pair generator, $\mathsf{D\text{-}Enc}$ and $\mathsf{D\text{-}Dec}$ are the encryption and decryption functions shown in equations (15) and (16), respectively.

The IND-CPA game is a set of interactions between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

1) $\mathcal{C}$ computes two pairs of keys $\mathsf{KeyGen}(1^k) \rightarrow (y_0, x_0)$ and $\mathsf{KeyGen}(1^k) \rightarrow (y_1, x_1)$ then sends the public keys $y_0$ and $y_1$ to $\mathcal{A}$ while keeping $x_0$ and $x_1$ private.
2) $\mathcal{A}$ has access to $\mathcal{O}_2$ and can encrypt as many $T = \mathfrak{b} \,\|\, \mathsf{vote}$ of its choice. Next, $\mathcal{A}$ chooses a $T_0 = \mathfrak{b} \,\|\, \mathsf{vote}_0$ and $T_1 = \mathfrak{b} \,\|\, \mathsf{vote}_1$ then sends them to $\mathcal{C}$.
3) $\mathcal{C}$ uniformly and randomly selects $\mu \in_r \{0,1\}$ then computes $c^* = \mathsf{D\text{-}Enc}(g^v, T_\mu \cdot (y_r \cdot y_m)^v)$. Next, $\mathcal{C}$ sends $c^*$ to $\mathcal{A}$.
4) $\mathcal{A}$ outputs a guess $\mu'$ of $\mu$. $\mathcal{A}$ wins the security game if $\mu' = \mu$ and loses otherwise.

In our proposed scheme, only the last cast vote with a legitimate ballot is counted towards the election. As a result,

TABLE I: Computational costs of our proposed distributed e-voting scheme.

| | Voter | Moderator | Registrar |
|---|---|---|---|
| **Setup** | | $\mathsf{E}$ | $\mathsf{E}$ |
| **Voter registration** | | | $2n\mathsf{M} + n\mathsf{E}$ |
| **Acquiring ballots** | $2\mathsf{E} + 2\mathsf{M}$ | $6n\mathsf{E} + 2n\mathsf{M}$ | $2n\mathsf{E}$ |
| **Casting ballot** | $2\mathsf{E} + 2\mathsf{M}$ | | |
| **Tabulation** | | $2n\mathsf{E} + 2n\mathsf{M}$ (or Tally authority) | |

the adversary may be able to discover whether the coerced voter has behaved as instructed. The DDH assumption implies that the adversary is unable to get a non-negligible advantage from the IND-CPA security game in determining the $T$ that was encrypted.

**Definition 4** (Coercion-resistance)**.** *A voting scheme is said to be coercion-resistant if no PPT adversary is able to get a non-negligible advantage by performing the IND-CPA security game, i.e.* $\mathsf{Adv}_{\mathcal{A}} = Pr[\mu' = \mu_i] < \frac{1}{2} + \varepsilon$ *for any negligible* $\varepsilon$*.*

**Theorem 3.** *The proposed d-BAEV scheme is coercion-resistant if the DDH assumption holds.*

Due to space limitation, the proof is omitted here.

**Corollary 1** (Ballot unlinkability)**.** *It is computationally infeasible to link any vote to its voter, or link multiple votes cast by one voter (including the ones created for coercion).*

*Proof.* It follows directly from Theorem 2 and Theorem 3. □

### D. Election Results Manipulation

Before casting their votes to the blockchain, voters are required to encrypt their votes under public keys of both the registrar and the moderator. This encryption acts as a secure envelope as it conceals the actual vote during the voting phase. It ensures that all encrypted votes can only be decrypted when both parties disclose their private keys.

$d$-BAEV employs a blockchain as a public bulletin board for voters to cast their votes which is secure against election result manipulation. Votes that pass the validation are counted towards the election results while those that fail are discarded. Valid votes are posted to the blockchain along with their corresponding ballots to announce election results and allow voter validation. Voters can individually recognize that their votes have been counted correctly towards the final election results.

## IX. Performance Evaluation

### A. Computational Costs

We evaluate the number of multiplication ($\mathsf{M}$) and exponentiation ($\mathsf{E}$) operations of each voting stage.

*Setup:* The moderator and the registrar need to generate their public key pairs at the start of the election process, which requires a single $\mathsf{E}$ operation for each.

*Voter Registration:* Our proposed distributed voting scheme relies on a registrar to facilitate voter registration. Voters begin by generating their key pairs. After verifying the eligibility of voters, the registrar signs their public keys and adds them to the electoral roll. The total operations for registering all voters are $2n\mathsf{M}$ and $n\mathsf{E}$ operations, where $n$ is the number of voters.

*Acquiring Ballots:* The proposed $d$-BAEV scheme involves voter engagement with the moderator and registrar. Initially, the registrar generates the set of random and digitally signed ballots. Next, the voters interact with the moderator, who obscures their identities from the registrar while facilitating ballot distribution. The total operations performed by a voter, the moderator, and the registrar are $2\mathsf{M}$+$2\mathsf{E}$, $2n\mathsf{M}$+$6n\mathsf{E}$, and $2n\mathsf{E}$, respectively.

*Casting Votes:* The proposed $d$-BAEV scheme requires a voter to double encrypt $\mathfrak{b}_i$ along with the votes $\mathsf{vote}_i$ using ElGamal encryption, where $\mathsf{vote}_i$ is a sequence of bits representing each candidate, 1 if voting for $\mathfrak{c}_k$ and 0 if not voting for $\mathfrak{c}_k$. The total operations performed by a voter are $2\mathsf{M}$ and $2\mathsf{E}$.

*Tabulation:* For the proposed $d$-BAEV scheme, the tallying authority is required to decrypt the double-encryption of each vote using the revealed moderator and registrar private keys, $x_m$ and $x_r$. This imposes a computational cost of $2n\mathsf{M}$ and $2n\mathsf{E}$ operations.

Table I summarizes the number of these two operations for each of these processes.

### B. Scalability

Scalability is one of the most critical yet challenging practical issues for large scale election. There are multiple possible candidates of blockchains that can be considered for this design.

Solana [15] is one of the blockchains that has the highest scalability. According to its development team, Solana can process up to 60,000 transactions per seconds (TPS) [16]. Running an election for the approximately 150 million Americans that voted in the 2020 U.S. presidential election utilizing the proposed $d$-BAEV scheme over the Solana will suffice to complete the casting votes phase in approximately 42 minutes. Sloana requires a minimum of 128 GB of RAM for each server and some of the most expensive enterprise processors currently available.

NEAR [17], [18] is another possible candidate. It is a layer one, sharded, Proof-of-Stake (PoS) blockchain. The key difference between the NEAR blockchain and many others is that it relies on commodity servers, whereas other blockchains like Solana require expensive enterprise-grade hardware. The scalability of NEAR is reported to be up to 100,000 transactions per second. Running the election for 150 million voters using the NEAR blockchain, the voting phase can be completed in 25 minutes.

### C. Empirical Results

We implement the proposed $d$-BAEV scheme in both a MacBook Air using Maple v16 and a iPhone XR. The MacBook Air is equipped with 2 cores, 1.8 GHz Intel Core i5, and 8 GB 1600 MHz DDR3. The iPhone is equipped with the Apple A12 Bionic, a 64-bit ARMv8.3-A six-core CPU, with two cores running at 2.49 GHz. The iPhone application is developed over Xcode version 11.2.1 and is written in Swift 5. We use the BigInt library[1] to perform large number cryptographic computations. We measure the time costs to perform the double encryption computation presented in equation (15) as follows:

[1] https://github.com/attaswift/BigInt

TABLE II: Implement time (Second).

| **Key size** (Bits) | 512 | 1024 | 1536 | 2048 | 2560 | 3072 | 3548 | 4096 |
|---|---|---|---|---|---|---|---|---|
| **Time** (iPhone) | 0.220 | 1.172 | 3.554 | 8.396 | 15.313 | 25.99 | 40.47 | 59.34 |
| **Time** (Mac) | 0.001 | 0.003 | 0.008 | 0.018 | 0.032 | 0.052 | 0.078 | 0.114 |

## X. Design Trade-offs

### A. Universal Verifiability vs. Coercion-Resistance

In the case that universal verifiability is preferred, at the end of the tabulation phase, the registrar and the moderator both disclose their private keys and the tallying authority publishes votes along with their corresponding ballots of the election on the blockchain, which allows any voters to verify whether their votes have been counted properly toward the election and the legitimacy of the election results. This design also prevents the registrar from attempting to issue unassigned ballots to voters in favor of a particular candidate. However, once the election results are disclosed, the coercers would be able to find out whether the coerced votes are being counted toward the election.

A practical coercion-resistant voting system should be *receipt-free*, allowing voters to evade proving to their coercers how they voted. Our proposed scheme requires voters to encrypt their votes as shown in equation (15). This vote associated with the ballot can served as a receipt and may allow the coercer to verify how the voters have voted. Therefore, the choice between universal verifiability and receipt-freeness must be made because they are orthogonal to each other.

If strong coercion-resistance is required, ballot verification can be delegated to only the registrar and the moderator, who jointly decrypt and publish the results without revealing individual ballots. Given that the registrar and the moderator are unlikely to collude, and that their verifications must match, voters can trust that the integrity of the election result is maintained. However, this approach does limit voter and universal verifiability.

### B. Receipt-Freeness

Our proposed $d$-BAEV scheme may be modified to offer receipt-freeness at an additional computational cost. Once a vote is cast as described in equation (15), the moderator can decrypt then re-encrypt the vote as follows:

$$B_i' = (B_{i,1}g^u, B_{i,2}B_{i,1}^{-x_m}y_r^u) = (g^{r_v+u}, T_i y_r^{r_v+u}) = (B_{i,3}, B_{i,4}),$$

where $u \in_r \mathbb{Z}_p^*$. This method conceals the ballots of voters, preventing them from being identified during the vote casting phase. Simultaneously, once tallying is performed and the election results are disclosed, voter and universal verifiability can still be performed since ballots are unconcealed.

### C. Counting First Ballot vs. $n^{th}$ Ballot

In $d$-BAEV, voters receive unique digital ballots which they use to cast their votes to the blockchain. This design grants voters the ability to cast their votes multiple times while identifying those trying to double vote. However, only one cast vote corresponding to each legitimate voter is counted toward the final election results. Therefore it becomes a trade-off between selecting the first ballot vs. $n^{th}$ ballot, which makes it more difficult for coercers to monitor and detect how voters have voted since they must be physically present in a timely manner among the voters to force them to give up their credentials.

## XI. Conclusion

In this paper, we proposed $d$-BAEV, a novel blockchain-based anonymous electronic voting scheme. $d$-BAEV is designed to run large-scale elections and aims at improving voter turnout. Our security and privacy analyses show that the proposed $d$-BAEV scheme is secure, preserves voter privacy, protects voters against coercers, and maintains the integrity of election results. In our empirical results, we present the results of running various simulations for the proposed $d$-BAEV scheme over both a desktop machine and a smartphone. Our results show that it is feasible to run the proposed $d$-BAEV scheme over a mobile device, hence improving the accessibility of elections.

## References


[1] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach, "Analysis of an electronic voting system," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. IEEE, 2004, pp. 27–40.

[2] R. Gonggrijp and W.-J. Hengeveld, "Studying the nedap/groenendaal es3b voting computer: A computer security perspective," in *Proceedings of the USENIX workshop on accurate electronic voting technology*. USENIX Association, 2007, pp. 1–1.

[3] P. Y. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia, "Prêt à voter: a voter-verifiable voting system," *IEEE transactions on information forensics and security*, vol. 4, no. 4, pp. 662–673, 2009.

[4] C. Culnane, P. Y. Ryan, S. Schneider, and V. Teague, "vvote: a verifiable voting system," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 1, pp. 1–30, 2015.

[5] L. F. Network. John adams on religion and the constitution. [Online]. Available: https://oll.libertyfund.org/quote/john-adams-religion-constitution

[6] B. Adida, "Helios: Web-based open-audit voting," in *USENIX security symposium*, vol. 17, 2008, pp. 335–348.

[7] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1995, pp. 393–403.

[8] E. Zaghloul, T. Li, and J. Ren, "Anonymous and coercion-resistant distributed electronic voting," in *2020 International Conference on Computing, Networking and Communication*. IEEE, 2020.

[9] ——, "$d$-BAME: Distributed blockchain-based anonymous mobile electronic voting," *IEEE Internet of Things Journal*, no. 22, pp. 16 585–16 597, Nov. 15 2021.

[10] X. Landen. (1/6/22) Only 20 percent of americans say they're 'very confident' in u.s. election integrity: Poll. [Online]. Available: https://www.newsweek.com/only-20-percent-americans-say-theyre-very-confident-us-election-integrity-poll-1666411

[11] S. Kulke. (December 23, 2020) 38% of Americans lack confidence in election fairness. [Online]. Available: https://news.northwestern.edu/stories/2020/12/38-of-americans-lack-confidence-in-election-fairness/

[12] A. Lardieri. (Feb. 13, 2020) Majority of americans don't trust elections, poll finds.

[13] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "Bitcoin and blockchain: Security and privacy," *IEEE Internet of Things Journal*, 2020.

[14] National Conference of State Legislatures, "Double voting," http://www.ncsl.org/research/elections-and-campaigns/double-voting.aspx, 2018.

[15] A. Yakovenko. Solana: A new architecture for a high performance blockchain v0.8.13. [Online]. Available: https://solana.com/solana-whitepaper.pdf

[16] What is Solana (SOL). [Online]. Available: https://cryptos724.com/en/what-is-solana/

[17] The NEAR white paper. [Online]. Available: https://near.org/papers/the-official-near-white-paper/

[18] NEAR protocol (NEAR). [Online]. Available: https://learn.bybit.com/altcoins/what-is-near-protocol/