Efficient and Usable Coercion-Resistant E-Voting on the Blockchain

Neyire Deniz Sarier[1]

cosec, b-it, Bonn, Germany

Abstract

In [1], Sarier presents a practical biometric-based non-transferable credential scheme that maintains the efficiency of the underlying Brands credential. In this paper, we design a new Blockchain-Based E-Voting (BBEV) scheme that combines the system of [1] with encrypted Attribute Based Credentials for a non-transferable code-voting approach to achieve efficient, usable, anonymous, transparent, auditable, verifiable, receipt-free and coercion-resistant remote voting system for small/medium scale elections. To the best of our knowledge, the system is the first user-centric BBEV scheme that depends on the one-show Brands' Credential both for biometric authentication and pre-encrypted ballot generation leading to a natural prevention against double voting. Even though the system is instantiated with Bitcoin Blockchain due to its prevalence and various coin mixers available for anonymity, the system is designed to be generic, i.e. independent of the cryptocurrency. Thus, the new BBEV scheme can be extended to large-scale elections for public Blockchains with higher throughput/cheaper transaction fees. Finally, a cost analysis based on the last USA presidential election data shows that, the new BBEV is advantageous over the traditional one if implemented for three consecutive elections.

Keywords: E-voting, public Blockchain, Anonymity, Brands' Credential, Code-Voting, Pre-encrypted Ballots, encrypted Attribute Based Credential (ABC), Non-transferability, Biometrics, Fuzzy Extractors, Bitcoin Mixer, Smart card, user-centric

1. Introduction

Governmental organizations are investigating the opportunities to leverage Blockchain technologies for services beyond financial transactions that range from authentication/validation of electronic health records [2] to digital voting [3, 4, 5]. Previous work already described robust and verifiable electronic elections, where ballots and processing data are posted to a publicly accessible bulletin board. Regarding voting and citizen engagement, blockchain technology can enhance security, and ease transparency. Thus, recent work replaced the idea of append-only/publicly readable bulletin board by the blockchain to implement a fully decentralized voting system, but neglects several aspects related to the current implementation of identity, and the requirements on privacy/security.

Firstly, the notion of digital identity is challenging since the system should authenticate the voters anonymously to guarantee that no one can vote twice. In [4] the authors emphasize the main defect of many of the existent blockchain approaches in e-voting: *Successful authentication of the voters has already been performed without further addressing how this is accomplished in practice.* Thus, [4] suggests the use of Attribute Based Credentials (ABC), which have seldom been discussed in the context of Internet voting. Here, ABC based on *one-show credentials* like Brands' digital credential system could lead to a natural solution to efficient and secure e-voting since it prevents a malicious voter from double-voting due to its linkability during credential show. However, as noted in [6, 1], majority of the credential schemes (including ABC) do not guarantee true non-transferability. Binding the credential to the voter by means of biometrics is an effective solution for credential transfer. Hence, biometric-based credentials that require ownership of the credential owner's biometric on the fly ensures that voters are physically present when their credentials are in use, preventing credential sharing and abuse by theft. But, biometrics is assumed as sensitive data according to General Data Protection Regulation (GDPR) which requires provably secure biometric template protection schemes such as *Fuzzy Extractors* that hinder leakage of voter biometrics and thus preserve the voter's privacy [6, 1].

Secondly, in any voting system, the ballot links the voter's identity and their vote. Hence, to break this link, (1) the system either hides the identity, i.e. the voters cast their vote *anonymously* via a mixnet or by casting their ballots into a ballot box (2) or the system aims only for hiding the vote, where the voters cast their vote *encrypted* by using a tallying process (potentially homomorphic), (3) or the voters cast their vote both *encrypted and anonymously* [4]. Most of the e-voting protocols (except for code-voting and its flavors) anticipate to obtain from a voter an encrypted vote. Commonly, the encryption is

[1](e-mail: denizsarier@yahoo.com, deniz.sarier@gmail.com)

not done by the voter (due to the complexity of this task) but delegated to an external/untrusted voting device [7] leading to another security issue tackled by Code-voting.

Finally, practical solutions considering efficiency and usability allows for more widely adoption of the e-voting systems. Here, the latter is accomplished through the use of a tamper-proof smartcard for handling voter's credentials, even though smartcards bring their own problems, i.e. strong trust assumptions/limited computational capacity etc. In this context, the authors of [8] describe Blockchain-Based E-Voting (BBEV) systems that employ a tamper resistant device such as smart cards, which are already employed in e-voting schemes for secure authentication/identification, storage and for encrypting and signing messages and/or votes [8]. According to the election authorities, the smartcard delivered to each voter assures the authorization of each submitted ballot [8]. However, nothing prevents lending of the smart card to a coercer for vote-selling similar to lending of credit cards or other credentials.

1.1. Related Work on secure E-Voting

"*In this paper*, we focus on the current implementation of identity, and the requirements on privacy for secure Blockchain-Based E-Voting (BBEV) systems. Thus, the new system handles voter registration/valid ballot sheet distribution based on multi-factor authentication, and ballot/voter privacy during the ballot casting stage through the employment of the currently most efficient Brands credential scheme and their variations presented in [1, 9]".

1.1.1. Code Voting Systems with Pre-encrypted Ballots

Chaum introduced Code-Voting to deal with the Secure Platform Problem, namely, the untrusted voting computer [4]. Here, each voter receives prior to the voting phase an individual code sheet containing a personal vote code and the corresponding audit/verification code using a secure channel (e.g., by regular mail). The voter chooses his vote by means of the corresponding voting code. Using the verification code, voters can be assured of their votes to be recorded properly. Example pre-encrypted ballots are shown in Figure 1.

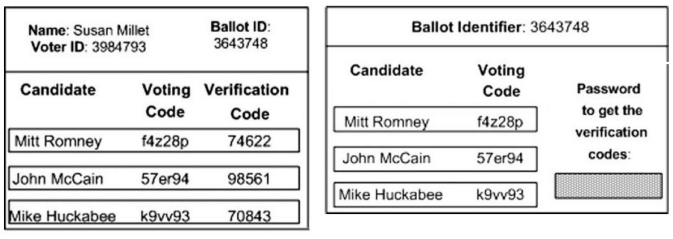

Figure 1: Example Pre-encrypted Ballots from [10]

After successful authentication, each eligible voter is given the code sheet that will be used during the ballot casting stage. In recent years, there has been several proposals for code voting systems. The reader is referred to [11] or to Section 6.5 of [4]. In this work, we focus on pre-encrypted ballots similar to [12].

1.1.2. Limited Number of Candidates/Voters

Current literature on e-voting can be classified based on the type of the elections:
(1) Yes/No Voting-Referendum limited to a single/two candidates [12, 13, 14, 15, 16, 17],
(2) Small-scale/boardroom voting [15],
(3) Elections based on a fixed [18]/small set of voters [17], and their complementary counterparts.

For most practical elections, the number of candidates is normally small [13]. For instance, in [14], every voter owns two voting tokens, which must be spent together on two distinct candidates to have a valid transaction.

The first class of e-voting schemes start with a single candidate and can be extended to multiple candidates, which is the same path we follow in this work. Besides, the Modified/Additive ElGamal encryption only manages small numbers because the number of voters, thus total number of votes is limited [17], i.e. less than 2^{30} and each voter votes Yes/1 or No/0 as in [17, 14] for each candidate. Therefore, the tally cannot be very large and an exhaustive search gives the result. As in [16, 19], a look-up table for the logs is employed.

"*In this paper*, we focus on BBEV systems with pre-encrypted ballots/voting sheets for two candidates running against each other. Our system assumes the voters of a medium scale country, where the total number of eligible voters does not exceed 100 million".

1.1.3. Blockchain-Based E-Voting (BBEV)

Elections are a critical component of democratic systems and there exists numerous initiatives designing/ implementing electronic/remote voting using Blockchain to achieve the challenging goal of a fully decentralized voting system. For a wider range of papers, the reader is referred to recent surveys of [5, 3, 4], and to [7, 14] for small surveys of Coercion-Resistant (CR) e-voting schemes. Here, [3] lists e-voting systems involving Biometric identification. [20] presents a small survey on BBEV, where existing Bitcoin-enabled e-voting protocols are classified as with or without coin mixing.

"*In this paper*, we follow the former approach, namely BBEV integrating an efficient Bitcoin-mixer such as Coinshuffle++ [21]".

1.1.4. Attribute-Based Credentials (ABC) in E-Voting

As stated in Section 6.6 of [4], Abendroth et al. describe a use case for ABC in the context of e-voting. Besides, [4] lists only one work, namely the work of Put et al. [22] on ABC-based remote voting. Here, the authors of [22] present avisPoll, which is specifically designed for elections that are arranged by parties not trusted by voters. Before a voter can cast a vote, she needs to obtain an anonymous Idemix credential. The main part of verifying a vote is to

verify the Idemix proof. However, as analyzed in [6, 1], CL-signature based systems such as IBM's Idemix are inefficient compared to Brands' credentials, where the latter is the basis for Microsoft's U-Prove [23]. Besides, IBM's Idemix and Microsoft's U-Prove [23] form practical implementations of Attribute Based Credentials (ABC) [4]. However, Brands' scheme/U-Prove outperforms the other constructions as shown in [6, 1] and summarized in Appendix based on the work of [24, 25].

"*In this paper*, we focus only on efficient credential schemes that are based on Brands."

1.1.5. Coercion Resistant (CR) E-Voting

Section 1.1.1 of [7] presents previous work on CR schemes that deliver the required secret keys in advance, that require a system for real/fake credentials and anonymous voting channels in addition to trust in the voting device and passwords memorized by voters, where the latter has a negative effect on the usability. Alternatively, CR is defined by challenging the attacker to distinguish between two ballots: the fake ballot produced for the coercer's preference and the real one generated for the true intent.

In this context, [14] designs a BBEV system, where the authorities generate v-tokens for each eligible voter together with the masks for the candidates. Specifically, each voter generates a wallet without involvement/control of anyone else, and registers it with the two authorities, who generate a valid and a fake v-token, both of which are indistinguishable and controlled by this wallet. Here, the voter is informed on which token is fake and which is valid without a receipt. Hence, the voter cannot officially prove the validity of a v-token, requiring the zero-knowledge proofs (ZKPs) to be interactive, so that any transcript of the ZKPs is unworthy for an outsider. Besides, the authorities prove with ZKPs the correctness of the tokens. The voter and authorities interact privately in such a way that one authority is unaware of the other's message to the voter (i.e. via untappable channels) [14]. In summary, every voter owns two voting tokens, i.e. one fake, one valid, however, only the voter knows which v-token is which. During voting, each voter assigns the valid v-token to the chosen/preferred candidate and the fake one to the remaining one. A vote receipt is returned to the voter that shows the two transactions. In the final tally, the fake v-tokens are thrown-away resulting in a publicly auditable, completely transparent, fully verifiable, and CR-process [14].

"*In this paper*, we will follow a similar approach except for the two v-tokens and a pseudonymous wallet of [14]." Specifically, [14] uses pseudonymous wallets and assumes complete privacy due to the fact that pseudonymous wallets cannot be linked to voter's real identity, which is a wrong assumption for Permissionless Blockchains like Bitcoin and Ethereum [26, 18].

1.2. Motivation

The main motivation of this paper arises from two recent papers that point out the following two main research gaps in e-voting literature:

- In many papers including the recent work of [14], registration and authentication of voters are assumed as very hard to solve, thus, the protocols start to work only once the users become voters.
- Since current schemes do not consider how users are identified and leave voter identification out of scope, [4] suggests the use of Attribute Based Credentials (ABC) that have seldom been discussed in the context of Internet voting.
- Most of the current CR e-voting schemes prioritize security over usability [7], where the latter can be improved through biometric smart cards that can handle the complexity of the pre-encrypted ballots/voter credentials.
- If designed generically, a BBEV (with an abstract model) can be implemented with any available/suitable blockchain system [14].

For the first item, the authors of [5] also point out the weakness of identification systems in remote voting. Thus, without the use of a Biometric system, which ensures the authentication/identification of each voter, one can never be sure that the vote is cast by an eligible (the right registered) person [5]. For elections, it is necessary to authenticate the voters anonymously to assure that no one can vote twice. To the best of our knowledge, the only work that involves biometrics in BBEV is [27], which is only a proposal of an architectural framework lacking any analysis.

As a solution to the first item, the authors of [14, 8] suggests to use a Permissioned/Consortium Blockchain, which re-introduces trust in the authorities that run these Blockchains, thus, the requirement for fair observers. Therefore, e-voting systems designed for Permissioned/Consortium Blockchains are left out of the scope of the paper. Even though the above suggestion overcomes the limitations/ scalability issues of public Blockchains like Bitcoin, -if employed for large-scale elections-, for regional elections taking place in medium-scale countries or national elections in small-scale countries, Bitcoin Blockchain is a suitable platform. Moreover, the computation of the tally starts with the local result, followed by the regional one and finally ends up with the national election result in case of a large scale election, which requires computations at different levels. Similar to the multi-candidate extension, the election scheme can be extended to large groups of voters after decrypting the intermediate result at each local election site and summing up those partial tally computations to obtain the regional and national results.

As a solution to the second item, we see that the only work on encrypted Attribute Based Credentials (ABC)

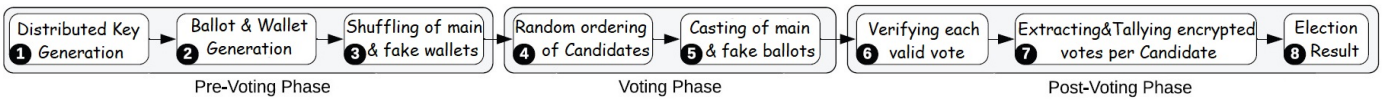

Figure 2: Flowchart of the New Voting Scheme

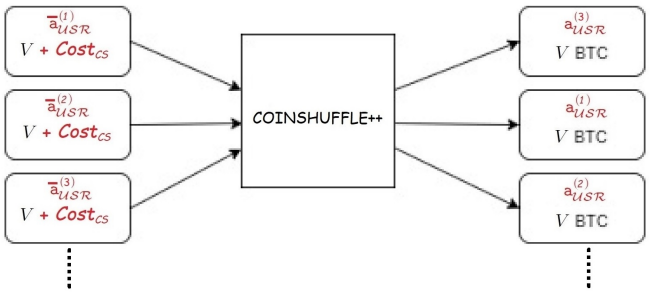

Figure 3: Example application of Coinshuffle++ in the proposed system

that is efficient and one-show, which results in a possible platform for code-voting systems preventing double voting is [9]. Even though there exists many e-voting systems based on various crypto-primitives, [9] has only been investigated further for anonymous credential systems (with 3 citations) and applications in Private Contact Tracing and Attribute Based Access Control (each with a single citation) summing up to 8 citations [28]. Thus, there exists a research gap for the design of generic e-voting schemes guaranteeing eligibility, based on different crypto-primitives and new technologies involving biometrics/smartcards.

Finally, one of our main design principle for e-voting involves usability, thus, we choose the most widely used/ common Blockchain, namely Bitcoin Blockchain to instantiate our generic BBEV system. Besides, we integrate a biometric smartcard and a practical biometric-based voter authentication module [1] that is based on the same (but modified) ABC system employed in the main voting stage.

1.3. Contributions

As stated in the previous sections in quotation marks, we describe a novel BBEV system that tries to fill in the research gaps in order to achieve a more efficient, secure, transparent, user-centric solution for small/medium scale elections with small number of candidates, which constitutes the majority of current elections. As different from current e-voting proposals, the new BBEV is designed on top of the recent practical work of [1] involving biometrics/smartcards to guarantee eligibility and the most efficient digital credential/ABC system that forms the basis of the biometric identification/code-voting feature of the new BBEV, namely one-show (Brands) Credential. To the best of our knowledge, (and also as verified in Section 1.2 and in Table 2 of [4]), the proposed system is the first BBEV scheme constructed on top of one-show encrypted ABC. Specifically, the new BBEV combines non-transferable, (hidden) biometric digital credential scheme of [1] and encrypted ABC of [9], both of which are based on the same one-show Brands credential, which results in a natural prevention for double-voting due to its linkability. We only focus on code-voting systems with pre-encrypted ballots for small/medium scale elections involving biometric smartcards and well-exploited Bitcoin Blockchain, which is the most common cryptocurrency for use, similar to traditional currencies. Even though our system is instantiated for a two-candidate election, the system can be extended to multi-candidate elections since the (linear) complexity of the pre-encrypted ballots will be handled by the smartcard. Similarly, Bitcoin Blockchain is not a requirement for the new system to operate. Public blockchains that outperform Bitcoin Blockchain in terms of throughput, security, privacy, usability could be preferred over Bitcoin, since the generic framework is independent of the Blockchain platform.

We consider the highest notion of security in e-voting, namely coercion resistance without sacrificing usability, efficiency, and election costs as shown in Table 1. Due to the employment of the widely used and well-exploited Bitcoin Blockchain in conjunction with smart cards, the new system is auditable, universally verifiable and as different from current systems, it guarantees eligibility due to the non-transferable, (hidden) biometric-based credential stored in the smartcard. Besides, privacy is protected even if the smartcard is lost/tamper-proofness is eliminated, or when a coercer tries to take the smart card away from the voter to cast ballots on his own or to force abstention, since the new system does not employ biometric data directly, instead it requires a Fuzzy Extractor as in [6, 1]. Finally, the lack of anonymity of Bitcoin is compensated with the use of a suitable mixer. Thus, our system guarantees ballot and identity privacy due to the pre-encrypted ballots and anonymity, where the latter is achieved through the use of a Bitcoin mixer.

Table 1: Comparison of the new BBEV to traditional elections in North Dakota, USA in terms of the election costs in 2022 [29]. †: consecutive, ‡: Initial Cost, i.e. valid only if BBEV is employed once

	Cost per Vote	Total cost per vote of 3† elections	Average cost per vote of an election
Traditional	13.6$	40.8$	13.6$
New BBEV	24$‡$	33$	11$

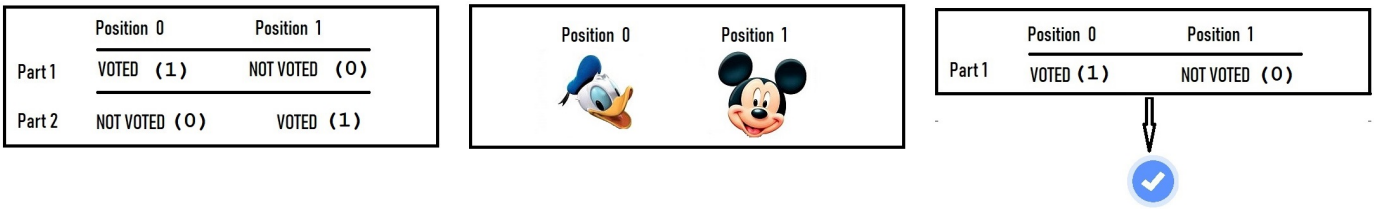

Figure 4: (a) Delivered ballot in Pre-Voting (b) Random Candidate-Position assignment in Voting (c) Verifying Vote for D. Duck in Post-Voting

2. Background

In this section, we review Bitcoin Mixers for anonymity and biometric-based non-transferable credential scheme of [1] for voter identification. The reader is referred to Appendix for Bitcoin, Fuzzy Extractors, ElGamal Encryption, Brands' Credential scheme and the scheme presented in [1] and intermediate computations of pre-encrypted ballots based on the credential issuance phase of encrypted ABC scheme of [9].

2.1. Bitcoin Mixer

Previous literature on BBEV cannot assure anonymity or employ anonymous communication paths, since Bitcoin transactions are not anonymous, but only pseudonymous, where all data are disclosed. The exchanges of a person with a Bitcoin address is publicly readable by anyone. For the e-voting use case, this means which voter made what vote is revealed to anyone [18]. Coin mixing is a new technique to prevent the linkability problem without losing compatibility with Bitcoin protocol. In coin mixing, the coins belonging to a set of peers are sent to freshly created addresses to assure that these fresh addresses are not linkable to the previous addresses of those peers. A P2P mixing protocol denoted as CoinShuffle++, which is based on a mixnet run by the peers in order to guarantee the unlinkability of input/output addresses in a CoinJoin, which is a mixing transaction created jointly by multiple users with multiple inputs and outputs [21]. CoinJoin is the core idea underlying CoinShuffle (and other mixing techniques), which is improved by CoinShuffle++.

In this paper, we require a practical decentralized mixing protocol for Bitcoin users to enhance anonymity without requiring modifications to the Bitcoin Blockchain. Hence, we choose, CoinShuffle++, which is a coin mixing protocol based on a more efficient anonymous communication protocol, which is compatible with a P2P trust model [21]. Alternatively, [18] propose an e-voting system to solve the anonymity problem by using Zerocoin, which is a Bitcoin mixing/laundery [26].

2.2. Practical Biometric-Based Non-transferable One-Show Digital Credential Scheme

[1] presents a simple modification to the original Brands digital credential as summarized in Appendix C. Briefly, [1] describes an efficient non-transferable digital credential stored on a smart card that is only responsible for capturing a fresh biometrics reading (and erasing it subsequently). Even if an attacker accesses all data on the smartcard, no biometric data is leaked because of the fuzzy extractor, which enables a user/voter to extract and reproduce a random string (biometric attribute) from noisy biometrics. Thus, the requirements on the smart card are minimum as in [30]. The reader is referred to Appendix B and C for the details.

3. The New BBEV Scheme

In this section, the reviewed crypto-primitives and Bitcoin Blockchain that works with a suitable Mixer are combined to describe the new BBEV scheme. First, we present the modified version of the encrypted Attribute-Based Credential (ABC) scheme using Fiat-Shamir Transform to obtain Pre-encrypted Ballots and their verification in Post-voting stage. A simple flowchart of each stage and the corresponding Bitcoin Transactions is described in Figure 4 and Figure 6, respectively.

3.1. Participants

There are four roles apart from the voter $\mathcal{V}$:

Government ($\mathcal{G}$): The Central Authority that issues permanent National ID Cards and handles election costs. The National ID is a smartcard that integrates a multi-show unlinkable anonymous/digital credential such as [25, 24, 6, 1].

Polling Authority ($\mathcal{PA}$): A non-colluding (and independent) Central Authority that issues the pre-encrypted ballots and one-show credentials with biometric data and other attributes defining the one-show identity of the voter for non-transferability. The smartcard storing the credential and pre-encrypted ballots that are signed by $\mathcal{PA}$ is trusted by everyone.

Verifying Authority ($\mathcal{VA}$): A non-colluding (and independent) Authority, which checks the records that each eligible $\mathcal{V}$ casts on the blockchain. It is also responsible for the announcement of each verified (valid) vote according to the ballot verification info via a (final) Bitcoin Transaction.

Independent Observer/Notary ($\mathcal{N}$): Only responsible from the random assignment of candidate to position on the ballot sheet after the delivery of each voter's smartcard by $\mathcal{PA}$ and the Pre-Voting phase is completed.

Except for the voter, the four participants are assumed to be independent and non-colluding.

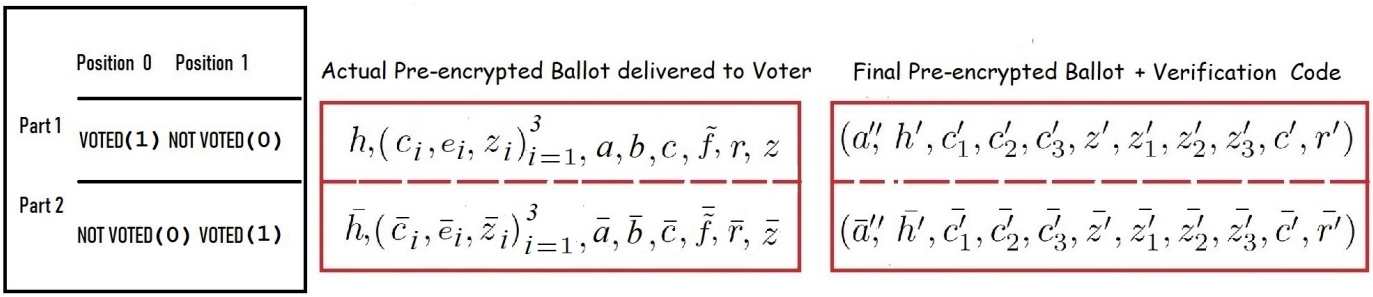

Figure 5: (a)Abstracted Vote *vs. (b)Pre-encrypted Ballot delivered to each voter vs. (c)Final Pre-encrypted Ballot with Verification code a'' both computed by the voter: Part 1 (top row) is followed by Part 2 (bottom row)*

3.2. Platforms and devices

The new BBEV system requires specific devices for processing/storage of sensitive/personal data. Firstly, each voter is assumed to have a smartphone/computer to prepare final ballots for voting and to connect a biometric smartcard reader/biometric device. This is a natural assumption for citizens who use their biometric National ID Card for other e-government applications on their electronic devices. The biometric device capturing the biometrics of the voter, extracts the biometric features and generates the biometric template of the voter. The device is assumed to be trusted by any party and it erases the biometrics once finished with all the operations. Finally, it returns only the required data as in [1, 6, 2]. The Bitcoin wallet and the smartcard(s) are assumed as separate entities.

Two different platforms, i.e. the Bitcoin blockchain for on-chain storage and a public IPFS for offchain data storage take place in the new BBEV system, where only a reference to the encrypted biometric template is stored on-chain. As in [1, 2, 26, 6], the reference is a data_pointer that can be a hash. In summary, the public ledger functionality provides integrity of data, an IPFS is used for storage of large amount of data via a Merkle tree, and anonymity is guaranteed using Bitcoin mixers.

3.3. Trust Assumptions and Adversary Model

Each participant is assumed as independent and non-colluding, and the smart cards are assumed as tamper-proof and trusted by all entities of the system. Except for the voter and coercer, the remaining parties (i.e. authorities) are assumed as semi-honest.

Here, the government-issued permanent National ID with an integrated multi-show unlinkable credential and $\mathcal{PA}$-generated smart card storing the non-transferable one-show credential of [1] and pre-encrypted ballot is trusted by everyone. Since receipt-freenes is a security goal for the new system, the verification of the pre-encrypted ballot that encodes Part 1 and Part 2 correctly, is performed interactively as in [14], as a result, any transcript of the zero-knowledge proofs is unworthy for an outsider. The voter-authority interaction is private and untappable as in [14]. Every voter owns two Bitcoin wallets: one associated for the valid vote, the other for the fake one, but only the voter knows which is which. The most critical interaction occurs between the voter and the $\mathcal{PA}$ during the notification of the main/valid wallet address after shuffling, thus we assume an anonymous channel that is also required later during the ballot casting phase so that that an adversary cannot ascertain whether a voter cast a ballot or not. For any e-voting scheme to be fully CR, this assumption is a requirement: If an adversary can determine whether or not a given voter cast a ballot, then the adversary can easily mount a forced-abstention attack [31]. Voters are assumed to cast their ballots through anonymous communication channels such as the anonymity network TOR, where a network connection remains anonymous. If the voter's IP address is visible during voting then it can be easily linked to the voter's identity presented during the registration/pre-voting phase (this is a standard assumption in e-voting where voters may directly interact with an adversarial authority [8]). Besides, due to the employment of encrypted ABC of [9], where the voters don't know the credential attributes, hence, they cannot render ZKPs for those attributes as part of the verification protocol.

Adapted to our setting, i.e. to the pre-encrypted ballots encoding Part 1= (1 0) and Part 2 = (0 1), the voter cannot prove to the coercer even which Part corresponds to which Position on the ballot sheet regardless of the assignment of the candidates to each position later by the Notary. A similar assumption is also valid for the Polling Authority $\mathcal{PA}$ who records the intermediate versions of each pre-encrypted ballot on the smartcards before offline delivery. Since the independent Notary randomly assigns each candidate to a position on the ballot sheet after the smartcards are delivered to each voter, $\mathcal{PA}$ cannot cheat in the elections via delivering modified/biased pre-encrypted ballots in favor of a particular candidate.

3.4. Digital Credential Schemes with Encrypted Attributes

In [9], the authors introduced a new crypto-primitive called as "Anonymous Credential Schemes with Encrypted Attributes" that use the Brands' credential scheme based on the blind Chaum-Pedersen (CP) signature scheme and ElGamal Encryption. For simplicity, we generate the code sheet for two-candidates, hence, they are represented by 2

attributes (x_1^*, x_2^*), where $x_1^*, x_2^* \in \{0,1\}$. The code sheet we use in our system is based on the scheme of [9] designed for the case that the issuer knows the attributes (i.e. only the position of the candidates on the ballot sheet) in the clear since the random assignment of which candidate corresponds to which position on the ballot sheet is performed publicly after all non-complete ballot sheets are delivered securely to each potential voter, who finalize the pre-encrypted ballots. Here, the choice of a voter based on the position of the candidates on the ballot sheet is represented by $x_1^*, x_2^* \in \{0,1\}$, where a voted candidate is assigned the value 1, the remaining one is assigned the value 0 by generating the encryptions (c_1, c_2) of them, while the verifier does not learn these.

Since the voters do not generate the attributes of a credential (and the position of the candidates on the ballot sheet), as stated before, they cannot render ZKPs for these attributes/candidates as part of the verification protocol, which results in receipt-freeness. Instead, the voters only know that the delivered first part of the ballot corresponds to a valid vote for the first candidate, i.e. encoding of $(x_1^* = 1\ x_2^* = 0)$, whereas the second part of the ballot corresponds to a valid vote for the second candidate, i.e. encoding of $(x_1^* = 0\ x_2^* = 1)$. The identities of the candidates represented by Position 0 and Position 1 on each ballot will be announced before the start of the voting, where each voter only submits one part of the ballot sheet corresponding to his desired candidate. An example flow of a two-candidate election process, i.e. generation of pre-encrypted ballots, random assignment of each candidate to the position on the pre-encrypted ballot, a valid vote for the first candidate is presented in Figure 4.

The secret key used for verification is distributed among multiple parties. To achieve unlinkability for the credential/ballot sheet, honest voters blind the encrypted attributes/candidates (c_1, c_2) to (c_1', c_2') as provided by the issuer, by performing random re-encryptions of these attributes/candidates resulting in ballot privacy. Since Brands' credential is a single-use credential scheme, the scheme of [9] does so as well, which prevents double-voting.

Given a security parameter k, system parameters $(q; g)$, with prime $q > 2^k$ are generated. Then, the public key $(h_0, f, \hat{f}, g_1, g_2, g_3, f_1, f_2)$ is generated jointly, corresponding to the secret keys of the issuer and verifier adapted from [9]. Specifically, the voter's secret parameter is denoted by α and the issuer's secret parameters are generated using randomly selected $x_0, (\phi_1, \phi_2) \in \mathbb{Z}_q$. Here, the issuer certify the attributes $(x_i = x_i^* + \phi_i)$ for $i = 1, 2$ unknown to the voter, and an additional random $x_3 \in \mathbb{Z}_q$.

The verifier's secret key is also generated randomly $\lambda, y_1, y_2, y_3 \in \mathbb{Z}_q$ with $h_0 = g^{x_0}, f = g^{\lambda}, \hat{f} = f^{x_0} = h_0^{\lambda}$, and $g_i = g^{y_i}$ for $i = 1, 2, 3$ and $f_i = g^{\phi_i}$ for $i = 1, 2$.

3.4.1. Credential Issuance: Pre-Voting

Part 1 of the voter's ballot sheet is signed indirectly via the group element $h = g_1^{x_1} g_2^{x_2} g_3^{x_3} h_0 \neq 1$, the attributes $(x_1^* = 1\ x_2^* = 0)$ are provided to the voter in encrypted form only, and are blinded by the voters by random re-encryption as in [9]. Hence, a credential on $(x_1^* = 1\ x_2^* = 0)$, consists of a tuple $(h', c_1', c_2', c_3', \alpha, z', z_1', z_2', z_3', c', r')$ satisfying Equation 2a and 2b of [9]. Here, α is the voter's secret parameter for the Part 1 of his pre-encrypted ballot. The same is repeated for the Part 2 of the voter's pre-encrypted ballot, namely $(\bar{h}', \bar{c}_1', \bar{c}_2', \bar{c}_3', \bar{\alpha}, \bar{z}', \bar{z}_1', \bar{z}_2', \bar{z}_3', \bar{c}', \bar{r}')$. Again, $\bar{\alpha}$ is the voter's secret parameter for the Part 2 of his pre-encrypted ballot. As in [9], we assume that an issuer can use the same tuple ϕ_1, ϕ_2 for the executions of the issuance protocol.

After completing the interactive intermediate computations summarized in Appendix D, the issuer, namely the $\mathcal{PA}$ records the computed variables of his side on the smart card of the voter as shown in Figure 5.

Together with the one-show non-transferable digital credential of [1] also recorded on the smartcard, the issuer sends the Pre-encrypted Ballots Part 1 and 2 offline to each voter. The final version of the ballots will be computed by the voter before casting their votes as follows.

3.4.2. Credential Verification: Voting

Since the ballot verification is achieved non-interactively, we slightly modify the Verification protocol of the encrypted credential scheme presented in Figure 2 of [9] by employing Fiat-Shamir transformation [32], which allows to replace the interactive step 2 in Figure 2 of [9] with a non-interactive random oracle access, namely a cryptographic hash function $\mathcal{H}$. For better readability (and to avoid confusion between the repeating variable names in Figure 1 and 2 of [9]), the randomly picked a in Figure 2 of [9] is replaced by a'' and the randomly picked c is replaced by $c = \mathcal{H}(a'', h', c_1', c_2', c_3', z', z_1', z_2', z_3', c', r')$. This way, the voter can now also compute r as in the interactive step 3 in Figure 2 of [9]. Later, anyone (i.e. Verifying Authority) can use these values to calculate c and complete verification. Hence, the final Part 1 of pre-encrypted ballot is:
$(a'', h', c_1', c_2', c_3', z', z_1', z_2', z_3', c', r')$.
Similarly, the final Part 2 of pre-encrypted ballot is:
$(\bar{a}'', \bar{h}', \bar{c}_1', \bar{c}_2', \bar{c}_3', \bar{z}', \bar{z}_1', \bar{z}_2', \bar{z}_3', \bar{c}', \bar{r}')$.
As one notices, the final version of the pre-encrypted ballots involves also a verification code info a'', c, where $c = \mathcal{H}(a'', h', c_1', c_2', c_3', z', z_1', z_2', z_3', c', r')$ can be computed easily from the final pre-encrypted ballot. Here, $\mathcal{H}$ denotes a cryptographic hash function.

3.4.3. Credential Verification Continued: Post-Voting

Finally, each voter publishes the hashes of the proof-ref link to the actual vote Vote1 and Vote2 on the blockchain as shown in Figure 6.

For simplicity, we assume that proof-ref$_1$ represents Part 1 of the actual Pre-encrypted Ballot described in Figure 4(c) and Figure 5(c).

3.5. Security Properties

In this section, the security notions of ballot privacy, verifiability, and coercion-resistance (CR) [33] are reviewed.

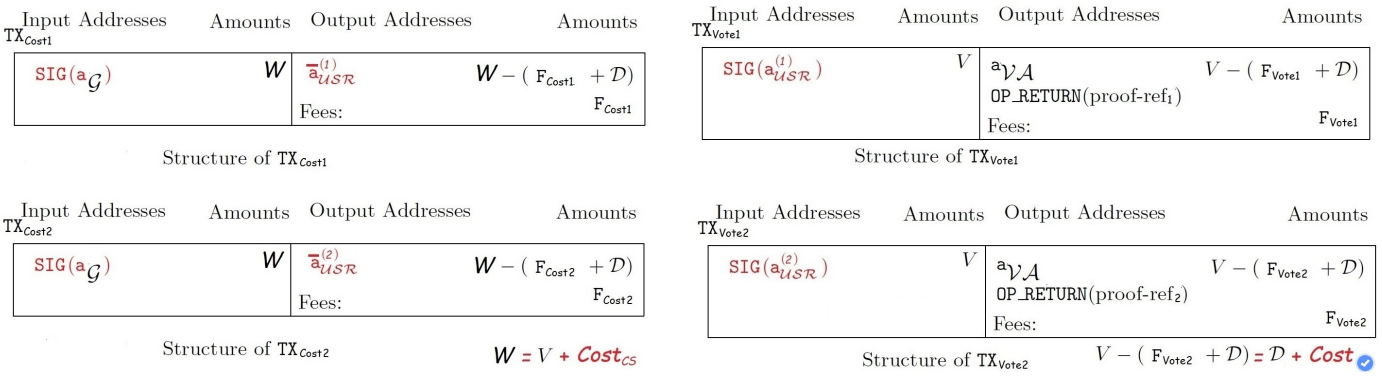

Figure 6: Transaction Flow of the new BBEV: $Cost_{CS}$ is shown in Figure 3 and the final verification transaction is shown in Figure 7

Algorithm 1: Pre-voting

Input: National ID Card with a digital credential, two fresh Bitcoin wallet addresses main/coercion Bitcoin wallets
Output: Smart card for Code Voting with a non-transferable one-show credential, main/coercion Bitcoin wallets
- Key Generation Protocol is run by Verifying Authority $\mathcal{VA}$ and Polling Authority $\mathcal{PA}$
- Announcement of eligible voters by $\mathcal{PA}$
- $\mathcal{PA} \leftrightarrow \mathcal{V}$ Pre-encrypted ballot and non-transferable one-show credential generation by $\mathcal{PA}$
- Offline sending of the smartcard recording the pre-encrypted ballot (with Part 1 and Part 2) and the non-transferable one-show credential to each voter $\mathcal{V}$
- $\mathcal{V} \rightarrow \mathcal{G}$ Sending of two fresh Bitcoin wallet addresses from each eligible $\mathcal{V}$ to Government $\mathcal{G}$
- $\mathcal{G} \leftrightarrow \mathcal{V}$ If the voter's digital credential in National ID card verified, transfer of equal voting cost to each wallet via Bitcoin Transactions
- $\mathcal{V} \leftrightarrow \mathcal{V}$ CoinShuffle++ each wallet (of equal voting cost) for anonymity
- $\mathcal{V} \leftrightarrow \mathcal{PA}$ Shuffled main/coercion Wallet Address notification to $\mathcal{PA}$ (offline/anonymous channel)
- Announcement of the list of all wallets (shuffled and notified main/coercion wallets) in random order that will participate in the elections by $\mathcal{PA}$

Algorithm 2: Voting

Input: Smart card for Code Voting with a non-transferable one-show credential, main/coercion Bitcoin wallets
Output: Ballot casting via two Bitcoin Transactions
- Ballot Position-Candidate Assignment by Independent Observer/ Notary
- Computing the final version of the pre-encrypted ballot by the voter $\mathcal{V}$
- Arrangement of Part 1 or Part 2 of pre-encrypted ballot as Bitcoin Transactions according to main/real wallet address by $\mathcal{V}$
- Arrangement of Part 1 or Part 2 of pre-encrypted ballot as Bitcoin Transactions according to coercion/fake wallet address by $\mathcal{V}$
- $\mathcal{V} \rightarrow \mathcal{PA}$ Casting the real vote via a Bitcoin Transaction from the main/real wallet address before deadline by $\mathcal{V}$
- $\mathcal{V} \rightarrow \mathcal{PA}$ Casting the fake vote via a Bitcoin Transaction from the coercion/fake wallet address before deadline by $\mathcal{V}$

Algorithm 3: Post-Voting

Input: Main Bitcoin wallets, threshold decryption keys
Output: Verification of each ballot via a Bitcoin Transaction, Election Result
- Announcement of valid votes according to main wallet info by Polling Authority $\mathcal{PA}$
- Announcement of verified (valid) votes according to ballot verification info by Verifying Authority $\mathcal{VA}$ via a Bitcoin Transaction to each main wallet address
- Extracting each encrypted candidate vote of each voter by $\mathcal{VA}$
-Homomorphic tallying of all encrypted votes per candidate by $\mathcal{VA}$ to recover the number of votes from the plaintext by brute-forcing the exponent, which is the number of votes given for the candidate.
-Announcement of the election result by $\mathcal{PA}$

Ballot privacy is guaranteed when outside/inside observers (voting authorities) cannot determine the voters' plain choices by examining the published election data such as voters' ballots, talliers' proofs of integrity, etc.). Hence, data leakage on how voters voted is not more than what can be derived from the final election result.

Verifiability enables voters or external observers/auditors to verify whether the published election data are correct, even if, inside/outside adversaries try to modify the election data/result without being detected.

Receipt-freenes (RF) prevents any information on how the voter voted that is used to prove, to a vote-buyer, when the voter wishes to sell her vote.

Coercion-resistance (CR) [31] is defined as a voter cannot interact with a coercer to provide information, which can be used to prove how she voted. CR allows a voter to apply a counter-strategy, where the coerced voter successfully votes for her favorite candidate, however, the coercer cannot distinguish whether the coerced voter followed his instructions or tried to accomplish her own goal. In this work, we use the most widely used technique denoted as "Fake Credentials", where each voter is provided with a unique and secret credential to submit her actual and valid vote. However, if a voter is under coercion, fake credential is used which is an invalid credential indistinguishable from the valid one. Later, the voting authorities will remove the the invalid vote. CR is a stronger privacy guarantee than RF, where the latter is implied by the former. The reader is referred to [31, 33, 4] for the details.

4. Security Analysis of the New System

In this section, we evaluate the new BBEV schema based on the basic requirements which are relatively easy

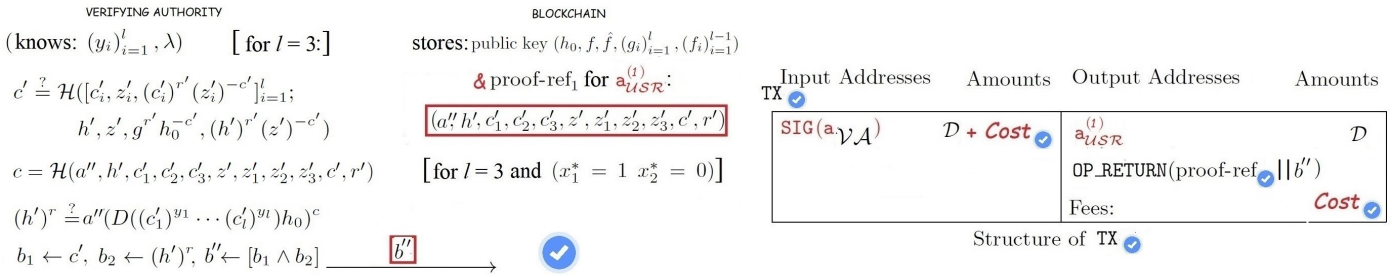

Figure 7: Post-Voting: Verification of a valid vote for Donald Duck stored on the Blockchain and its corresponding final verification transaction

to implement: Privacy, Unreusability, Verifiability, Completeness, Soundness and Eligibility and extended requirements: Universal verifiability, RF and CR, where the latter two properties are very important and must be handled in a fair and democratic election.[8].

Lemma 4.1. *If the encrypted ABC system of [9] and CoinShuffle++ protocol are secure, then the new proposal guarantees Ballot Privacy.*

The attributes, (i.e. the encoding of each candidate) are provided to the voter in encrypted form only, and the voters blinds them via random re-encryption. Indeed, [9] restricts the voters to random re-encryptions of the encrypted attributes by using additional parameters as described in Section 4.2 of [9]. Besides, the candidates' positions on the ballots are announced at the beginning of Voting-stage as shown in Algorithm 2. Hence, $\mathcal{PA}$ cannot link neither the intermediate nor the final version of the one-show credential/pre-encrypted ballot to the voter. Next, CoinShuffle++ breaks the link between the Bitcoins transferred to each eligible voter by $\mathcal{G}$ to cover the election costs and the ballots cast by the receivers of those Bitcoins. Hence, $\mathcal{G}$ cannot link the one-show credential/pre-encrypted ballot -whose proof-ref link is stored as an OP_RETURN output on the blockchain- to the voter. Hence, ballot privacy is achieved against the non-colluding $\mathcal{PA}$ and $\mathcal{G}$.

Lemma 4.2. *The new proposal is unreusable, i.e. prevents double-voting.*

The Lemma is due to the linkability of Brands' Credential based systems employed in the new proposal, namely, encrypted ABC system of [9] and the scheme in Section 2.2 of [1]. That is why they are also called as one-show credentials. Besides, the number of eligible voters e who intend to participate in e-voting are announced by $\mathcal{PA}$ at the end of the Pre-Voting stage as shown in Algorithm 1. This number is smaller than half of the total number of Bitcoin Transactions g signed by $\mathcal{G}$, where $\mathcal{G}$ and $\mathcal{PA}$ are assumed as non-colluding. Finally, the total number of valid votes associated to the total number of final verification transactions f is less than e.

Lemma 4.3. *The new proposal is individually verifiable.*

In fact, the new BBEV proposal is user-centric, similar to Bitcoin-based identity management systems summarized in [6, 1]. The voter has control over each step of the voting life cycle, starting with the Pre-voting phase:

(1) generating two random Bitcoin wallets and informing $\mathcal{G}$, (2) mixing of main/fake Bitcoin wallets for anonymity against $\mathcal{G}$, (3) keeping which Bitcoin wallet corresponds to valid vote as secret after informing $\mathcal{PA}$, and continuing with the Voting phase as:

(4) computing the final version of pre-encrypted ballots and their verification codes for ballot privacy against $\mathcal{PA}$, (5) casting of real and fake ballots to achieve CR against a coercer, and finishing with Post-Voting phase as:

(6) verifying the final verification transaction of the valid vote for completeness, etc. are all performed by $\mathcal{V}$.

Lemma 4.4. *The new proposal is complete, sound, eligible and fair.*

All valid votes are counted correctly since each valid vote is individually verifiable and any observer can compute the election result from the total number of final verification transactions after verifying each verification proof at the end of the Post-Voting phase.

Invalid votes can easily be detected and discarded by any outsider/observer since $\mathcal{PA}$ announces at the end of the Pre-Voting phase all the main/fake wallet addresses. Next, $\mathcal{PA}$ announces each main wallet info at the start of the Post-Voting phase, which are already listed/announced in random order by $\mathcal{PA}$. Hence, the total number of actual voters that participated in the elections are less than half of the total number of entries in the randomly ordered list of potential participants announced at the end of Pre-Voting phase. Besides, any observer can verify each final verification transaction and eliminate invalid ballots, and compute the election result.

Eligibility is the strongest security property of the new BBEV, since the only work involving biometrics in BBEV is [27], which is limited to an architectural framework.

Finally, to prevent factors influencing the vote of the remaining voters, the new BBEV does not allow for any early/partial result before Post-Voting phase is completed since verification of each ballot can only be started after the members of $\mathcal{VA}$ (i.e. a set of parties with keys generated using threshold cryptography) construct each

secret key in Figure 7 required for verification. $\mathcal{VA}$ is assumed as semi-honest, hence, each secret key of $\mathcal{VA}$ is distributed among the parties (i.e. threshold shared) using a distributed key generation scheme as in [9].

Lemma 4.5. *The new proposal is universally verifiable.*

Any observer/auditor can track the Bitcoin transaction flow starting from the election cost transactions covered by $\mathcal{G}$ ending with final verification transaction of each valid vote. The total election cost is transparent similar to the whole e-voting life cycle, the total number of potential participants with shuffled main/fake wallets are publicly known (before and after shuffling) and announced by $\mathcal{PA}$ after each voter informed $\mathcal{PA}$ (privately/anonymously) about their final main/fake wallets. Besides, each voter casts his ballot using his fake/real wallets according to the announcement of $\mathcal{PA}$ performed at the end of Algorithm 1. Finally, the announcement of main wallet info before tallying and the total number of submitted valid votes is on the Blockchain together with their proofs. The only missing info is who voted for whom, which is the main goal of any secure e-voting scheme.

Lemma 4.6. *The new proposal is Receipt-Free (RF).*

The receipt-freeness property is due to the interactive credential issuance/pre-encrypted ballot generation based on the encrypted ABC scheme. Since the intermediate computations for the pre-encrypted generation is interactive as shown in Appendix D, any ZKP's transcript (i.e. the transcripts of any statement/proof) is unworthy for an attacker/outsider.

Besides, the two fresh Bitcoin wallet addresses are randomly generated by each voter, hence, every voter owns two Bitcoin addresses: one valid and one fake, however, only the voter knows which one is which. After shuffling, the voter only shares this information with $\mathcal{PA}$, -namely which Bitcoin address is fake and which is valid-, without any receipt and through anonymous communication channels such as the anonymity network TOR, so the voter cannot officially prove the validity of a Bitcoin address to a coercer. Finally, the voter and $\mathcal{PA}$ interact privately and via an untappable channel, hence, after shuffling, only $\mathcal{PA}$ knows the main Bitcoin address that will be used in the final tally. We assume that a coercer targets a voter after $\mathcal{PA}$ announces the actual voters that will participate in the elections as shown in the last item of Algorithm 1.

Lemma 4.7. *The new proposal is Coercion-Resistant (CR).*

As stated in Lemma 4.6, a coercer targets a voter after the registration of voters is complete, namely $\mathcal{PA}$ announces the actual voters that will participate in the elections as shown in the last item of Algorithm 1. Here, we assume that the coercer and $\mathcal{PA}$ are non-colluding. Specifically, Pre-voting phase proceeds without any corruption of voters as in [31], which assumes this as a requirement for a CR election, since an adversary that can corrupt and seize the credentials of a voter in the Pre-Voting stage can mount a simulation attack [31]. Besides, as stated in Lemma 4.6, the two fresh Bitcoin wallet addresses are randomly generated by each voter and once the Pre-Voting phase is over, a coercer can obtain the information on which Bitcoin address is valid and which is fake only from the voter. As a result, a coercer cannot distinguish which Part of the pre-encrypted ballot corresponds to a valid vote and which to a fake one. Finally, as stated in section 3.3, we assume that a coercer cannot determine whether a voter cast the ballot due to the anonymous communication channels such as the anonymity network TOR, again, this is a requirement for any election scheme to be fully CR: If an adversary can determine whether a voter cast a ballot or not, then the adversary can mount a forced-abstention attack [31].

5. Discussion

In this section, we analyze the new construction in terms of its limitations and compare it to other recently introduced e-voting systems, each selected according to a particular criteria. Specifically, Table 2 lists each e-voting scheme that share at least one property of our new system.

5.1. Extention to Multi-Candidates

Our new system can easily be extended to multi-candidates since the pre-encrypted ballot construction depends on the encrypted ABC scheme of [9], which is designed for a list of attributes (each representing a different candidate) having fixed values of either 0 or 1. The encoding for more than two-candidates prolongs the ballot sheet by additional Parts and, thus resulting in increased complexity. Moreover, if a voter wants to sell its vote to multiple coercers each representing a different candidate, then the number of voting transactions increases linearly similar to the election cost handled by the government.

Similar to [14], where complexity increases linearly with the number of candidates, one can generalize BBEV so that it allows for multiple candidates, where the computational cost scales again linearly with the number of candidates.

5.2. Performance Analysis of the New System

A comparison of various e-voting schemes based on computational cost and other properties is presented in Table 2. Here, the main selection criteria of the schemes in Table 2 is their universal verifiability, their publication date (5 out of 7 items are published in last 5-years) and their restriction on the number of voters and/or candidates due to Bitcoin/Zerocoin, exhaustive search to solve Discrete log, etc.. Based on the comparison summary charts in Figure 8 and 9, our system outperforms the only ABC based e-voting of [22] scheme as listed in [4]. For the only remaining Anonymous BBEV scheme of [18], the computational cost arises from the Zerocoin, which relies on the

Strong RSA assumption and Double-Discrete Logarithm (DDL) proofs with known performance restrictions compared to Schnorr-type ZKPs. Finally, the only remaining BBEV that has a high computational cost is due to the complexity of the protocol.

5.3. Cost Analysis of the New System

In 2020, nearly two-thirds of eligible U.S. voters cast ballots for 2020 presidential election, casting nearly 158.4 million ballots that approximately cost 1.5 Billion US Dollars nationwide [34]. Since the new system is designed for small/medium scale countries, one can focus on a local state such as North Dakota, which spent a total of 3.3 Million US Dollars for 2020 elections [34]. Specifically, for Primary Elections in 2022, 106,168 ballots cast corresponding to 564,935 eligible voters results in an election cost of 1,444,739USD, where cost per vote is calculated as 13.61USD [29]. If North Dakota employs the new system, since pre-voting and voting stages have different time schedules, considering only the election day, each voter cast their real and fake votes by submitting two Bitcoin Transactions. Bitcoin generally has a higher transaction activity than other cryptocurrencies, and in May 2023, it reached its highest transaction volume of 670,000 coins on the same day. The final verification transaction is assumed to follow the next day, which results in total $2 \times 106{,}168 = 212{,}336$ ballots = 212,336BTC transactions for an election in North Dakota.

For BBEV, the initial cost of a single vote consists of the Cost of the smart card + Postal Cost + Total Bitcoin Transaction Cost. For simplicity, we assume $F_{Vote} = Cost_{CS} = Cost_{\checkmark}$, where the latter is in fact the Bitcoin Transaction Fee of the final verification transaction.

$$\begin{aligned} 2W &= 2V + 2Cost_{CS} \\ &= 2(2\mathcal{D} + F_{Vote} + Cost_{\checkmark}) + 2Cost_{CS} \\ &= 4\mathcal{D} + 2F_{Vote} + 2Cost_{\checkmark} + 2Cost_{CS} \\ &\approx 4\mathcal{D} + 6F_{Vote} \end{aligned} \tag{1}$$

The initial cost of a smart card with an embedded processor ranges between 7.00 to 15.00 (USD), where some biometric card companies charge up to 20 US Dollar per card [35]. We assume that those digital ballots delivered in form of a tamper-proof smart card are re-programmable/reloadable for the next election (returned to the Polling Authority before the next election date for secure storage) as opposed to their paper-based counterparts, which are one-time usable per election. Under the assumption that a biometric smart card is used at least three times in various elections (with at most two-year intervals), the cost per vote decreases below 7USD per election, whereas the total cost for three elections are up to 40.83USD per voter for paper-based ones as shown in Table 1.

Besides, Bitcoin Average Transaction Fee is as low as 0.64USD for August 20, 2023 [36], which remains almost identical to the Transaction Costs in US Dollars analyzed for November 2019 in [6]. Bitcoin Tumblers take a percentage transaction fee of the total coins mixed to turn a profit, typically 1–3% [37]. In Bitcoin, the dust value limit $\mathcal{D}$ is assumed as 546 satoshis, which equals 0.14USD in August 2023. In fact, the magical "546 satoshis" value is simply the most commonly known one to represent $\mathcal{D}$, for a p2pkh output. Hence, according to Equation 1, the cost of each vote to the state sums up to $(2W+7)USD \approx (4\mathcal{D}+6F_{Vote}+7)USD = (0.56+3.8+7)USD \approx 11USD$, which is less expensive compared to the $13.61USD$ cost per vote, when paper-based ballots are used at least for three consecutive elections, as in the e-voting case.

6. Conclusion

In this paper, a new BBEV scheme is proposed, which aims to solve two main issues in e-voting: -authentication of voters for eligibility and CR for highest security-, without sacrificing efficiency and usability. The new BBEV scheme is by design generic: it can work with existing public Blockchains with a suitable mixer for anonymity such as Bitcoin, which does not require any implementation and results in a reduced cost per vote compared to traditional elections in spite of the additional smartcard cost. Thus, we leave it as a future work to implement BBEV using other public Blockchains that can outperform Bitcoin, thus improve the overall performance of BBEV.

References

[1] N. D. Sarier, Comments on Biometric-based Non-transferable Credentials and their Application in Blockchain-based Identity Management, Computers & Security 105 (2021) 102243.

[2] N. D. Sarier, Privacy preserving biometric authentication on the blockchain for smart healthcare, Pervasive Mob. Comput. 86 (2022) 101683.

[3] M. V. Vladucu, Z. Dong, J. Medina, R. R. C. Vladucu, E-voting meets blockchain: A survey, IEEE Access 11 (2023) 23293–23308.

[4] M. P. Heinl, S. Gölz, C. Bösch, Remote electronic voting in uncontrolled environments: A classifying survey, ACM Comput. Surv. 55 (8).

[5] A. Benabdallah, A. Audras, L. Coudert, N. E. Madhoun, M. Badra, Analysis of blockchain solutions for e-voting: A systematic literature review, IEEE Access 10 (2022) 70746–70759.

[6] N. D. Sarier, Efficient biometric-based identity management on the blockchain for smart industrial applications, Pervasive and Mobile Computing 71 (1) (2021) 101322.

[7] T. Finogina, J. Herranz, On remote electronic voting with both coercion resistance and cast-as-intended verifiability, Journal of Information Security and Applications 76 (2023) 103554.

[8] T. Dimitriou, Efficient, coercion-free and universally verifiable blockchain-based voting, Computer Networks 174 (2020) 107234.

[9] J. Guajardo, B. Mennink, B. Schoenmakers, Anonymous credential schemes with encrypted attributes, in: CANS'10, Vol. 6467 of LNCS, Springer, 2010, p. 314–333.

[10] V. Morales-Rocha, M. Soriano, J. Puiggalí, New voter verification scheme using pre-encrypted ballots, Computer Communications 32 (7) (2009) 1219–1227.

[11] K. Marky, M. L. Zollinger, P. Roenne, P. Y. A. Ryan, T. Grube, K. Kunze, Investigating usability and user experience of individually verifiable internet voting schemes, ACM Trans. Comput.-Hum. Interact. 28 (5).

Table 2: Selected Usable/Practical e-voting schemes based on either ABC, Code-voting, Bitcoin with Coin Mixing, Zerocoin, Ethereum or Permisssioned/Consortium† Blockchain. NC: Non-comparable, *: if modified as in [1, 30], DDL$^{\times}$: Double Discrete Log, $^{+}$: Complexity of the Protocol, ‡: Commitment or ElGamal combined with Schnorr/Groth-Sahai Zero Knowledge Proof (ZKP)

Scheme	Code-Voting	Blockhain & Anonymity	Non-transferablity/ Biometric ABC	Receipt free	Coercion Resistant	Universally Verifiable	Limited #Voters or Candidates	Computational Cost
[22]	No	NC	Yes*	Yes	Yes	Yes	Yes	High: Idemix
[18]	No	Yes	No	No	No	Yes	Yes	High: DDL$^{\times}$
[14]	No	No†	No	Yes	Yes	Yes	Yes	High^{+}
[12]	Yes	NC	No	Yes	Yes	Yes	Yes	Low:Enc‡+ ZKP
[15]	No	No†	No	No	No	Yes	Yes	Low:Enc‡ +ZKP
[13]	No	NC	No	No	No	Yes	Yes	Low: Enc‡ +ZKP
[20]	No	Yes	No	Yes	No	Yes	Yes	Low: joint-RSS
New Sys.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Low: Brands CL

[12] V. Cortier, A. Filipiak, J. Lallemand, Beleniosvs: Secrecy and verifiability against a corrupted voting device, in: CSF'19, 2019, pp. 367–36714.

[13] F. Hao, P. Ryan, P. Zieliński, Anonymous voting by two-round public discussion, IET Information Security 4 (2010) 62–67.

[14] C. Spadafora, R. Longo, M. Sala, A coercion-resistant blockchain-based e-voting protocol with receipts, Advances in Mathematics of Communications 17 (2) (2023) 500–521.

[15] P. McCorry, S. F. Shahandashti, F. Hao, A smart contract for boardroom voting with maximum voter privacy, in: FC'17, Vol. 10322 of LNCS, Springer, 2017, p. 357–375.

[16] P. Ryan, S. Schneider, Prêt à voter with re-encryption mixes, in: ESORICS'06, Vol. 4189 of LNCS, Springer, 2006, p. 313–326.

[17] Y. Li, W. Susilo, G. Yang, Y. Yu, D. Liu, X. Du, M. Guizani, A blockchain-based self-tallying voting protocol in decentralized iot, IEEE TDSC 19 (1) (2022) 119–130.

[18] Y. Takabatake, Y. Okabe, An anonymous distributed electronic voting system using zerocoin, in: ICOIN'21, 2021, pp. 163–168.

[19] F. Tang, G. Ling, C. Cai, J. Shan, X. Liu, P. Tang, W. Qiu, Solving small exponential ecdlp in ec-based additively homomorphic encryption and applications, IEEE TIFS 18 (2023) 3517–3530.

[20] N. Lu, X. Xu, C. Choi, T. Fei, W. Shi, Bevote: Bitcoin-enabled e-voting scheme with anonymity and robustness, Security and Communication Networks 2021 (Article ID 9988646).

[21] T. Ruffing, P. Moreno-Sanchez, A. Kate, P2P mixing and unlinkable bitcoin transactions, in: NDSS'17, The Internet Society, 2017.

[22] A. Put, I. Dacosta, M. Milutinovic, B. D. Decker, An anonymous, verifiable internet service poll system, in: Report CW 669, KU Leuven, Technical Report, 2014, pp. 1–12.

[23] C. Paquin, G. Zaverucha, U-prove cryptographic specification v1.1, in: Technical Report, Microsoft Corporation, 2011.

[24] M. Chase, S. Meiklejohn, G. Zaverucha, Algebraic macs and keyed-verification anonymous credentials, in: ACM SIGSAC'14, ACM, 2014, pp. 1205–1216.

[25] F. Baldimtsi, A. Lysyanskaya, Anonymous credentials light, in: ACMCCS'13, ACM, 2013, pp. 1087–1098.

[26] N. D. Sarier, Privacy preserving biometric identification on the bitcoin blockchain, in: CSS'18, Vol. 11161 of LNCS, Springer, 2018, pp. 254–269.

[27] M. J. H. Faruk, M. Islam, F. Alam, H. Shahriar, A. Rahman, Bie Vote: A Biometric Identification Enabled Blockchain-Based Secure and Transparent Voting Framework, in: BCCA'22, IEEE, 2022, pp. 253–258.

[28] GoogleScholar, Articles 8 results, https://scholar.google.com/scholar?oi=bibs&hl=en&cites=6817698115327438857,5335497885852728578 (Consulted on September, 2023).

[29] N. D. S. of State, Summary of North Dakota Election Statistics, 1980 – Present (2020), https://vip.sos.nd.gov/pdfs/Portals/statistics-turnout.pdf (Consulted on August, 2023).

[30] M. Blanton, W. M. P. Hudelson, Biometric-based non-transferable anonymous credentials, in: ICICS'09, Vol. 5927 of LNCS, Springer, 2009, pp. 165–180.

[31] A. Juels, D. Catalano, M. Jakobsson, Coercion-resistant electronic elections, in: Towards Trustworthy Elections'10, Vol. 6000 of LNCS, Springer, 2010, p. 37–63.

[32] A. Fiat, A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, in: CRYPTO'86, Vol. 263 of LNCS, Springer, 1986, pp. 186–194.

[33] T. Haines, J. Müller, I. Querejeta-Azurmendi, Scalable coercion-resistant e-voting under weaker trust assumptions, in: SAC'23, ACM, 2023, p. 1576–1584.

[34] C. S. III, The cost of conducting elections, https://electionlab.mit.edu/sites/default/files/2022-05/TheCostofConductingElections-2022.pdf (Consulted on 2023).

[35] E. Inc., Would you want a biometric credit card?, https://www.linkedin.com/pulse/would-you-want-biometric-credit-card-eromnet (Consulted on 2023).

[36] YCHARTS, Bitcoin average transaction fee (i:batf), https://ycharts.com/indicators/bitcoin_average_transaction_fee (Consulted on August 22, 2023).

[37] Wikipedia, Cryptocurrency tumbler, https://en.wikipedia.org/wiki/Cryptocurrency_tumbler (Consulted on August, 2023).

[38] S. A. Brands, Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy, MIT Press, 2000.

Author Bio: N. Deniz Sarier received her M.Sc. degree in Media Informatics from RWTH Aachen, and PhD degree in Computer Science from b-it, cosec of Bonn University, Germany in 2007 and 2013, respectively. As an Assoc. Prof. of Computer Science, her research interests include biometric security, Blockchain, identity management, public-key cryptography, in particular, integration of biometrics into cryptographic/blockchain applications.

Appendix

This section is summarized from [1, 6, 9, 17, 19, 38, 24, 25, 26, 32] and functions as a supplementary material. To avoid confusion of the variables, $\mathcal{V}$ denotes the verifier instead of the voter. The reader may skip Appendix, if familiar with the primitives.

7. Appendix A

7.1. Bitcoin Blockchain

Although our system is instantiated using Bitcoin, the system can employ any blockchain platform (preferably with a higher transaction rate) that allows the sender of a transaction to store arbitrary data in a transaction as in [6]. For the Bitcoin instantiation of the system shown in Figures 6 and 7, we have a single standard Bitcoin transaction type that requires OP_RETURN outputs as in [6]. Each such output contains up to 80 bytes of space in which the sender of a transaction can store arbitrary information. As each of these transactions performed once every 10 min is mainly to record the votes of each entity, they are assigned values slightly larger than the minimum amount $\mathcal{D}$ of standard Bitcoin transaction as in [6].

7.2. ElGamal Encryption Scheme

In this section, we review additively homomorphic ElGamal encryption and distributed ElGamal cryptosystem for verifiable distributed protocols. Briefly, an authority chooses and publishes a cyclic group $\mathbb{G} =< g >$ of prime order q together with a generator g of the group. Here, $k = |q|$ denotes the security parameter. The secret key is $sk = \lambda \leftarrow \mathbb{Z}_q$ and the corresponding public key $pk = f = g^{\lambda}$.

To encrypt a message $x \in \mathbb{Z}_q$, one randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $c = (g^r, g^x f^r) = [[g^x]]$. The homomorphic property is additive as $[[g^a]] \times [[g^b]] = [[g^{a+b}]]$.

Additive ElGamal encryption using a finite field and Elliptic Curve (EC) version of ElGamal encryption are both additively homomorphic schemes, which can be quite useful in data aggregation. However, the former is hampered by a difficult discrete logarithm computation for proper decryption and the latter requires to solve an elliptic curve discrete logarithm problem (ECDLP). Although there are methods for solving ECDLP, they are only efficient when the length l of the plaintext $x \in \{0,1\}^l$ is short , i.e. $l < 32$ [19]. Thus, both schemes appears to be useful only if all plaintexts, i.e. total number of votes for each candidate is limited to a small range, i.e. less than 2^{30}. The reader is referred to [19] for the details of Exp-ElGamal, in particular, its main efficiency advantages in encryption, homomorphic addition, scalar multiplication apart from the short ciphertext length compared to other additively homomorphic systems such as Paillier.

In [17], distributed ElGamal cryptosystem is employed for a self-tallying e-voting system. Since the verifying authorities of our system is assumed as semi-honest as in [9], we also employ them as a set of parties using threshold cryptography. Indeed, the secret keys can be threshold shared among the parties using a suitable distributed key generation protocol. The reader is referred to [17, 9] for such schemes. In this work, we also require verifiable distributed protocols for key generation and for decryption.

8. Appendix B

Brands [38] presented the digital credentials scheme, where the same credential, signatures and parameters are used in each instance of the showing protocol resulting in a single-show credential system. Thus, Brands credentials are linkable, and identical to Bitcoin, pseudonymity instead of anonymity can be achieved for them. Similarly, U-Prove [23] does not allow unlinkable reuse of credentials: To unlinkably use a credential again, a user must get it reissued. However, from the efficiency point of view, Microsoft U-Prove [23] (based on Brands' work [38]) is evaluated as the most efficient construction.

8.1. Brands selective disclosure scheme (DLRep)

Brands selective disclosure scheme [38] enables selective disclosures involving an identity with $n-1$ fields, $(X_1, ..., X_{n-1})$. Let q be a prime number and $\mathbb{G}$ a group of order q, which could be the same group used for the Bitcoin signature protocol.

8.1.1. Credential Issuance:

Let $g_0, g_1, ..., g_n \in \mathbb{G}$. X_0 prevents an attacker with a priori knowledge on the X_j attributes of an entity from performing a dictionary attack where she guesses values for the remaining X_js [6, 1]. The tuple $(X_0, X_1, ..., X_{n-1}) \in \mathbb{Z}_q^n$ is called a Discrete Logarithm Representation (DLRep) of $h = \prod_{j=0}^{n-1} g_j^{X_j}$ with respect to $(g_0, g_1, ..., g_{n-1})$.

8.1.2. Credential Showing:

The showing protocol is a presentation of a credential with selective disclosure. This scheme is a proof of knowledge of those attributes the entity does not want to reveal. The proof can be carried out if those attributes are indeed the same ones that are committed in the credential stored in the smart card. To prove knowledge of a DLRep of h to a verifier $\mathcal{V}$, a prover $\mathcal{P}$ performs the following protocol steps [38]. The disclosed attributes with their indexes (j, X_j) for the subset of indexes $j \in D \subseteq \{1, ..., n-1\}$. Let the set of concealed attributes be $C = \{1, ..., n-1\} \backslash D$.

For convenience, we introduce the following notations for the products of DL commitments to the closed and disclosed attributes: $h^C = \prod_{j \in C} g_j^{X_j}$ and $h^D = \prod_{j \in D} g_j^{X_j}$ and $h = g_0^{X_0} h^C h^D$. Knowing (j, X_j) for $j \in D$, both $\mathcal{P}$ and $\mathcal{V}$ can calculate h^D. The following protocol proves to $\mathcal{V}$ the DL representation of $H = h(h^D)^{-1} = g_0^{X_0} h^C$ with respect to g_i's where $i \in C$ is known by $\mathcal{P}$.

1. $\mathcal{P}$ generates random, secret numbers $a_0 \in \mathbb{Z}_q, a_j \in \mathbb{Z}_q$ for $j \in C$. Let $A = g_0^{a_0} \prod_{j \in C} g_j^{a_j}$. $\mathcal{P}$ sends A to $\mathcal{V}$.
2. $\mathcal{V}$ provides a challenge number c.
3. $\mathcal{P}$ computes $b_0 = a_0 + cX_0$, and $b_j = a_j + cX_j$ for $j \in C \subseteq \{1, ..., n-1\}$ and sends them to $\mathcal{V}$.

4. The verifier $\mathcal{V}$ checks that $A = g_0^{b_0} \prod_{j \in C} g_j^{b_j} H^{-c}$ holds.

$\mathcal{P}$ knows all of the X_js to perform step 3. Here, X_0 is generated randomly by the voter to be used as a secret key. Assuming that the $\mathcal{P}$ does not know two DL representations with respect to this set, the verifier can be convinced that the claimed revealed attributes are indeed embedded in the certificate. All the other attributes remain hidden due to the selective disclosure protocol.

	Brands		CL credentials			
			Rsa		Pairings	
Efficiency	S^1	U^2	S	U	S	U
Signing[3]	2	12	10	8	15	14p
Verification	7	0	NC[4]		5+6p	9+6p

[1] Signer, [2] User,

[3] In number of exponentiations

[4] Non-comparable

Figure 8: Comparison of Brands Credential to CL-Credential [25]

	Time (in ms) when $(n, c, r) =$ (10,2,2)	(10,10, 0)	Credential size (in bits)
U-Prove	3.38	12.43	1024
Idemix	71.72	226.79	5369
Bilinear CL	20.98	28.32	$512n + 768$

estimated presentation proof generation cost. U-Prove and bilinear CL use 256-bit elliptic curve parameters, Idemix uses a 2048-bit modulus. n is the number of attributes, r is the number of *revealed* attributes in a presentation proof, and c is the number of *committed* attributes.

Figure 9: Comparison of U-Prove to Idemix [24]

8.2. Non-Interactive Zero Knowledge (NIZK)

This notion consists of a $\mathcal{P}$ who tries to convince $\mathcal{V}$ of the validity of some assertion in one move, i.e. without interaction with $\mathcal{V}$. The basic zero knowledge requirement for such proofs consists in exhibiting an efficient simulator outputting messages indistinguishable from $\mathcal{P}$'s. The definition of the zero knowledge requirement for these proofs is simplified because $\mathcal{V}$ cannot affect $\mathcal{P}$'s actions. The most famous technique to obtain NIZK from their interactive variants is known as the Fiat-Shamir paradigm [32]. It consists of letting $\mathcal{P}$ compute $\mathcal{V}$'s challenge himself as a hash of the statement to be proved and of the first message. The security of this construction is provided only in the random oracle model.

9. Appendix C

To achieve multi-factor authentication of each voter, credential systems with embedded biometrics data that require the direct use of biometrics is employed to compare the fresh biometrics to the stored template based on a distance measure. Example schemes of this category are summarized in [6, 1].

9.1. Fuzzy Extractors

Fuzzy extractors [39] allow one to extract randomness from biometrics w (for use in cryptographic schemes) and later reproduce it exactly using different w' from any value close to the original w. A fuzzy extractor is generated by Gen that, on input w, outputs extracted random string R and a helper string P; and can be reproduced by Rep that on input w' and P outputs R that was generated using Gen if $\mathsf{dis}(w, w') \leq d$. The generating function Gen executes a secure sketch scheme $S \leftarrow \mathsf{SS}$ and applies a strong extractor Ext to w to extract a random string R. Here, S and random coins used by Ext form the helper data P. Let r_2 denote the random coins used by Ext (i.e., execution is of the form $\mathsf{Ext}(w; r_2)$). We obtain $P = (S, r_2)$. The reproduction function Rep uses S from P to recover the original w given that $\mathsf{dis}(w, w') \leq d$ and extracts R by computing $\mathsf{Ext}(w; r_2)$.

9.2. Practical and Efficient Biometric-Based Non-transferable Digital Credential Scheme

In [1], the author modifies Brands' DLRep scheme by directly encoding the biometric attribute as a private attribute that is never going to be revealed to any party but its existence guarantee the non-transferability of the credential during the showing/presentation protocol with $\mathcal{V}$. Brands selective disclosure scheme [38] enables selective disclosures involving an identity with $n - 1$ fields, $(X_1, ..., X_{n-1})$. In the simple modification of [1], X_1 represents a user's biometric attribute, X_2 her name,..., and X_{n-1} be the nationality.

In [1], the modified system also needs a smartcard that requires a fresh biometric reading on each authentication attempt. The card stores the credential and helper data for fuzzy extractor. The voter's biometric data is not stored on the card, hence there is no check whether the biometrics of the current holder of the card match those stored inside it before an authentication request. [1] reserves X_1 for the biometric attribute, which is extracted by the credential issuer from the voter's biometrics using a biometric key binding/generation technique based on fuzzy extractors described in section 9.1. Specifically, the random string R is hashed using a cryptographic hash function H to assign R to X_1 via $X_1 = \mathsf{H}(R)$, which represents the biometric attribute of the voter as shown in Figure 3 of [1]. Also, $P = (S, r_2)$ is stored in the smartcard of the voter. No biometric template data, or biometric image is stored in this device and one cannot obtain any information about X_1 or biometrics of the voter either from P or from the

stored credential. The smart card extracts R by computing $\mathsf{Ext}(w; r_2)$) to compute the biometric attribute X_1 required during the showing protocol to prove knowledge of undisclosed attributes based on the interactive ZKPoK protocol with selective disclosure. If the credential issuer is not assumed to be trusted, then, the voter can employ a smart card with integrated sensor that captures the fresh biometrics, extracts the features and computes the hidden attribute X_1 using a fuzzy extractor, under the supervision of the credential issuer, i.e. the issuer can observe that the user applied his own biometrics. This way, the authority cannot learn the extracted attribute (i.e. the fuzzy extracted secret attribute) X_1, although the raw biometrics is assumed as public data. Here, the smartcard of the voter is trusted to process the biometrics correctly and generating a unique key X_1 identical for each session of showing phase with the help of the hidden helper data.

Thus, the Modified Credential Issuance phase is almost identical to the issuance protocol of section 8.1.1 except for reserving X_1 for the biometric attribute and X_0 as the secret parameter of the voter. Assuming the voter's encoded secret key $g_0^{X_0} g_1^{X_1}$ similar to the voter's encoded secret key g^{α} of [38] summarized in section 8.1, the set of disclosed attributes, which is a subset of $(X_2, ..., X_{n-1})$, are common input to the parties and they can both construct DLRep of H identical to the issuance protocol of section 8.1.1.

Modified Credential Showing phase is almost identical to the showing protocol of section 8.1.2 except that the voter must regenerate the biometric attribute X_1 from the fresh biometrics using his smartcard *before executing the showing protocol* of section 8.1.2. We note that the biometric attribute X_1 remains hidden for each verification of the credential but its existence guarantees the non-transferability of the credential, since a fresh reading is required on each credential show. Hence, the verification cost is identical to [24] for Brands credential based U-Prove. Finally, we emphasize that no biometric template/data is stored on the smartcard of the voter, only the credential h and helper data for the fuzzy extractor is required to be stored. This way, even if the credential h is stored on the smart card that is lost/compromised, there is no way to link the (unstored) biometric data to the identity/credential of the voter although Brands digital credentials do not provide multi-show unlinkability.

Appendix D

In this section, we review the Issuance protocol of the encrypted credential scheme of [9], where Figure 10 is in fact identical to Figure 1 of [9]. The only modification is the underlined variables that constitute the intermediate computations for the final pre-encrypted ballot prepared by the voter.

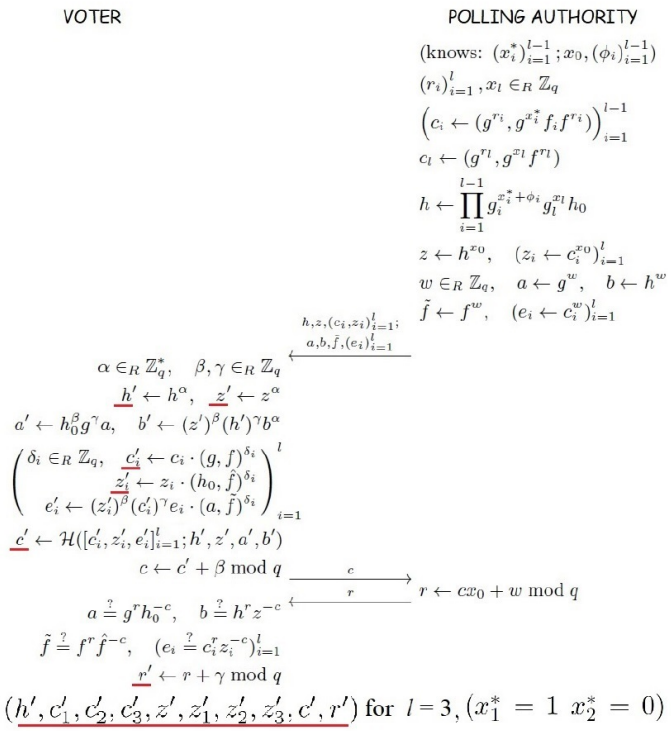

Figure 10: Intermediate computations for Part 1 of Pre-encrypted Ballot generation based on the work of [9]. The same procedure is repeated for $(x_1^* = 0 \; x_2^* = 1)$, i.e. Part 2. The final Pre-encrypted Ballot is generated according to the additional computations presented in Section 3.4.2