TRIP: Trust-Limited Coercion-Resistant In-Person Voter Registration

Abstract

Remote electronic voting promises convenience and flexibility, but presents risks of coercion and vote buying. One promising mitigation strategy enables voters to give a coercer fake voting credentials, which silently cast votes that do not count. Specific proposals, however, make problematic assumptions such as a trusted registrar, trusted hardware, or expect voters to interact cryptographically with multiple registrars. We present TRIP, the first voter registration scheme that addresses these challenges by leveraging the physical security of in-person interaction. Voters use a kiosk in a privacy booth to print real and fake paper credentials, which appear indistinguishable to others after registration. Voters interact with only one authority, need no trusted hardware, and need not trust the registrar except when under coercion. For individual verifiability, each credential includes an interactive zero-knowledge proof, which is sound in real credentials and unsound in fake credentials. Voters learn the difference by observing the order of printing steps, but need not understand the technical details. To evaluate TRIP, we first prove security against integrity and coercion adversaries. We then evaluate two prototypes to show their suitability in low-power environments, with one prototype using 4.5 Wh for 17 consecutive registration sessions over 30 minutes. A user study on TRIP with 150 participants suggests that TRIP is usable, receiving a slightly-above-average score of 70.4 on the System Usability Scale. 47% of participants could identify and report a malicious kiosk after appropriate security education.

1 Introduction

State-of-the-art online voting systems can ensure that each vote is verifiable and private, but leave unsolved challenges such as voter coercion and the buying of votes [64, 30, 17, 47, 27, 49]. Together with the attractive freedom and convenience of voting in any physical setting on any device, comes the risk that a coercer may control the voter's physical setting and/or device [7, 33, 9, 47, 17]. Many reports show that voter coercion and vote buying remain widespread issues in democratic countries, which can reduce voter trust and participation [23, 40, 38, 36, 22]. Some systems allow a voter to override a coerced vote with a truthful vote cast later [42, 56], but this defense is readily defeated by coercing a last-minute vote or stealing the voter's credential.

The property of *coercion resistance*, as defined by Juels, Catalano, and Jakobsson (JCJ), is achieved when coercers cannot determine whether voters have complied with their demands [47]. To resist coercion and vote buying, JCJ proposes creating fake voting credentials, which appear identical to real credentials, but cast votes that do not count in the election. A voter under coercion may give (or sell) fake credentials to a coercer, while secretly using a real credential to cast their true vote at any time, either before or after coerced votes are cast.

Any remote voting system including JCJ, however, critically depends on some trust bootstrapping process, in which voters have an uncompromised interaction with the election authority: e.g., to establish their real and fake credentials. Most proposed voting systems, including JCJ, merely assume that some abstract trusted *registration* process exists, without detailing how such registration is to be achieved, or is to be made secure and usable in practice.

The slim prior work addressing registration leaves problematic assumptions and design gaps. Civitas [33], for example, assumes voters have an *untappable channel* with a trusted registration teller. To implement this untappable channel, Civitas suggests that voters either have tamper-resistant, trusted hardware such as electronic ID cards, distributed and managed via an unspecified process, or else interact in-person with a trusted registration teller, via another unspecified process. Krivoruchko splits the registration role [49], but assumes that voters use an uncompromised device to interact with a trusted "identifying authority" over an untappable channel, again leaving the concrete process unspecified. These proposals leave important practical questions unanswered such as: why can a coercer not simply demand that the voter hand over their electronic ID for use only under the coercer's supervision, or register in-person with a compromised device provided by

the coercer, or wear a recording device during registration enabling the coercer to monitor the "untappable" channel?

We present Trust-limited Registration In Person or TRIP, the first coercion-resistant registration scheme we are aware of that defines a concrete, real-world process taking usability into account. TRIP leverages an in-person process in which voters interact with only one registration authority, and are neither required nor permitted to have a trusted device during credential issuance. Voters use a kiosk in a privacy booth to obtain real and fake credentials on paper. All credentials may be verified and used subsequently on any device, to vote in multiple successive elections, amortizing the inconvenience of in-person registration. The voter learns in the privacy booth which credential is real, but cannot prove this to anyone after leaving. Voters under coercion need only hide their real paper credential (and not an electronic device) from the coercer at registration time.[1] This real credential may be used later to vote from any device the voter trusts – such as that of a trusted friend, if their own device(s) are under the coercer's control.

TRIP confronts three key practical security challenges. First, the credentialing process must be voter verifiable despite the voter being prohibited from bringing devices into the booth. For example, it must be detectable if a malicious kiosk issues a fake credential while claiming it is real. Second, real and fake credentials must be cryptographically indistinguishable after the voter leaves the booth. Third, all real (and fake) credentials must be securely—without relying on tamper-proof trusted hardware—and conveniently transferable to, verifiable by, and subsequently usable on any vote-casting device of the voter's (or coercer's) choosing.

To create voter-verifiable credentials, the voter and the kiosk carry out and print a transcript of an interactive Σ protocol, which in essence proves in zero knowledge that votes cast with this credential will be counted. The printing of real credentials follows the proper (commit, challenge, response) order for Σ protocols, and hence constrain the kiosk to produce a sound proof. The voter's device checks this proof later during credential activation, so that voters can identify and report any kiosk misbehavior. The printing of fake credentials follows a distinct (challenge, commit, response) order, however, allowing the kiosk to lie: i.e., to print a proof transcript indistinguishable from a real one but embodying a *false* claim that votes cast with this (fake) credential will be counted. Because only the printing sequence differentiates sound proofs from unsound proofs, the voter can distinguish the real credential as it is printed but cannot prove this distinction to others after leaving the booth. Finally, to transfer the credential securely, we devise a mechanical two-state envelope and paper receipt design. A *transport state* reveals public information required for an official to complete the voter's in-person registration session, while an *activation state* reveals the secrets required by the voter's device to cast votes with the credential. Related zero-knowledge proof-printing techniques have been proposed for receipt-free in-person voting [57, 24], but to our knowledge never for use in remote voting systems.

We evaluate TRIP by: (1) proving its security against integrity and coercion adversaries, enabling voters to verify the authenticity of their real credentials, while also protecting them against coercion, and (2) designing and measuring two low-power kiosk prototypes, one of which is three times cheaper than the other for use in regions where cost and power constraints are important [32, 6].

In addition, we briefly summarize a few key insights from a usability study with this design in the companion paper "E-Vote Your Conscience: Voter Perceptions on Coercion and Vote Buying, and the Usability of Fake Voting Credentials for Online Voting", which involves 150 participants in a suburban public park of a major metropolitan city. Results show that TRIP has an 83% success rate and receives a score of 70.4 on the System Usability Scale (SUS), which was slightly above the industry's average score of of 68. Participants found TRIP's full process just as usable as a simplified one involving only real credentials. Participants who received security education were able to detect and report a misbehaving kiosk in 47% of cases, while 10% could do so without security education.

This paper makes the following primary contributions:

- The design of TRIP, which is to our knowledge the first coercion-resistant, voter-verifiable voter registration scheme supported by a user study.
- The first use of printed transcripts of interactive zero-knowledge proofs to achieve individual verifiability and coercion resistance for voter registration.
- A novel mechanical two-state receipt-envelope design, which reveals only the information required in each stage, and avoids relying on trusted hardware.
- Security proofs that TRIP ensures voter verifiability of their real credentials as well as coercion resistance.
- An implementation and evaluation of two kiosk prototypes, one for use in regions with power and cost constraints.

2 Background

This section presents the motivations for secret ballots, highlights the drawbacks of existing "secret-ballot" e-voting systems and introduces the evolution to coercion-resistance.

2.1 Why Secret Ballots?

Throughout human history, vote-buying, voter intimidation and political violence have plagued democratic societies [39, 31, 72]. As far back as the Roman Empire, bribery was the norm at every political election and acts to intimidate voters were not uncommon [72]. More than a millennia later, many

[1] Extreme coercion scenarios – where the voter is unable or willing to hide even a paper credential from the coercer, or has no access to any device they trust outside the privacy booth – are beyond TRIP's threat model, but could be addressed with a vote-delegation extension outlined in Appendix B.

democratic societies are still plagued with voter intimidation and vote buying. Developing countries [23, 22] and those with a low democracy index [36, 38] are particularly vulnerable, but even those with a high index are not immune [41]. Coercers have included gangs, employers, and political parties.

In the 1850s, Australia introduced the modern practice of casting secret ballots in a private, supervised polling environment with independent observers [39, 15]. This also marked the introduction of uniform, government-supplied ballots to prevent malicious tracing of ballots to voters [29]. The Australian secret ballot's design emphasizes two key objectives: (1) ensuring voters cannot prove their vote to third parties, and (2) minimizing trust in government officials. TRIP's goal is to achieve these same objectives for e-voting registration.

Contrary to frequent claims that blockchains improve online voting security, blockchains can significantly exacerbate risks [30, 64]. Decentralized autonomous organizations or DAOs might pay voters to vote a certain way [30]. Future ransomware might demand votes, not just cryptocurrency, in exchange for unlocking ransomed data. Such attacks are efficiently and globally scalable, are often hard to trace, and are difficult to deter even if traced due to jurisdiction boundaries.

2.2 Remote Electronic Voting Systems

Electronic voting (e-voting) systems promise convenience and a higher voter turnout [33], as well as unparalleled integrity in election results through end-to-end verifiability [28]. For example, voters might in principle cast their votes on a public ledger, whose final tally can then be verified by anyone, not just by designated observers. Each voter can also readily verify that their vote was recorded accurately. However, public voting exposes voters to peer pressure, vote buying, and coercion, which can reduce trust in the electoral process [22]. E-voting systems must not only ensure integrity, but also protect against adversarial influence on voters' choices.

Secret ballot e-voting systems have been studied since the 1980s by Chaum [26], Benaloh [13], and others. Modern e-voting systems generally either have the client encrypt the ballots [69] or have voters use code voting [7]. A set of talliers, collectively trusted for voter privacy, tally the ballots and generate a correctness proof. These systems thus prevent an adversary from using the public ledger to verify how a particular voter voted. An adversary can, however, ask the voter to *prove* how they voted, as was often done before the introduction of the secret ballot (§2.1). As a result, despite their strong appeal, e-voting systems do not yet offer the same level of voter protections as traditional in-person voting.

2.3 Evolution to Coercion-Resistance

Receipt-freeness. The notion of receipt-freeness [17, 61] informally requires that voters be incapable of proving how they voted [45]. Several e-voting schemes achieve receipt-freeness [43, 62, 19, 48], but are nevertheless susceptible to at least one of the following attacks [47, 53]:

- *Simulation*: an adversary impersonates the voter by, for example, having the voter divulge their voting credential.
- *Randomization*: an adversary has the voter submit an invalid vote, which will not count in the final tally.
- *Forced abstention*: an adversary prevents the voter from casting a vote altogether.

Coercion-Resistance. JCJ proposes a stronger adversary model called coercion resistance, where the coercer cannot determine whether a targeted voter followed the coercer's demands [47, 52, 50]. JCJ suggests the use of fake credentials, which appear to act identically to a voter's real credential but do not affect the election's outcome. Voters can then give or sell fake credentials to a coercer, or cast votes under the coercer's supervision using fake credentials, thereby pretending to comply with the coercer's demands. Another common approach is deniable re-voting [56, 65, 8], where only the last vote cast using the voter's voting credential counts in an election. However, this approach exposes the voter to last-minute coercion or to the theft of the voter's (only) credential.

Voter Registration. One important gap that JCJ leaves is to define a concrete bootstrapping process for voters to obtain their real and fake credentials, beyond merely assuming that voters have access to an untappable channel with a trusted registrar. How does this untappable channel work? How can it withstand potential interference from a powerful adversary, including when the voter intends to comply? These are questions that TRIP aims to address in order to bridge this gap.

3 TRIP Design Overview

This section summarizes TRIP's design, detailing its key properties, system and trust model, assumptions, and outlining a voter's perspective on the registration process.

Preliminary TRIP is a coercion-resistant registration scheme designed to take in an electoral roll and output real and fake credentials for each voter. This scheme is integratable into a broader electronic voting system that supports fake credentials and handles both ballot casting and tallying. We adopt the necessary elements commonly used in e-voting schemes (e.g., actors, definitions) to show how TRIP fits into such a scheme and provide well-founded proofs. For simplicity, we generally ignore provisions pertaining to casting or tallying ballots.

Registration Workflow The four steps in the registration process (Fig. 1), excluding Setup, are Check-In, Credentialing, Check-Out and Credential Activation. At check-in, prospective voters present legacy authentication documents, such as an ID document, to a registration official to receive their check-in ticket. Voters then surrender their electronic devices to the registrar and enter a privacy booth. Inside, voters present their check-in ticket to a kiosk to obtain their real and fake

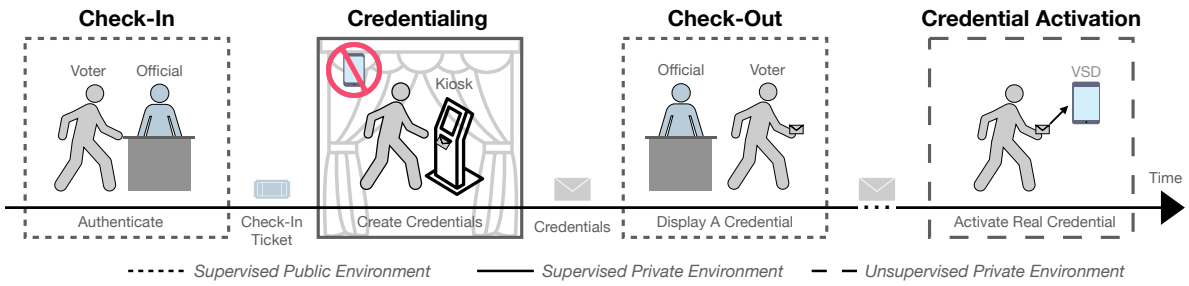


Figure 1: **TRIP Voter Workflow**.

The voter (1) checks in with an official by authenticating themselves and receiving a check-in ticket, (2) enters a supervised private environment to create their credentials, (3) checks out with an official by displaying one of their credentials, and (4) activates their credentials on their device.

credentials. After leaving the booth, voters retrieve their electronic devices and proceed to check-out. At check-out, voters display a credential (real or fake) to the official and exit the registrar. Voters can now activate their credentials at any time.

3.1 System Properties

TRIP aims to satisfy the following properties as a coercion-resistant voter registration scheme:

- **Coercion-Resistance** [47]: a (coercive) adversary cannot determine whether a targeted voter complied with the coercer's demands, even if the voter is willing to comply.
- **Individual Verifiability** [18, 7]: a voter is convinced that a ballot confirmed as coming from the voter contains their intended vote. With regards to TRIP, the system enables a voter to verify that their registrar-issued "real" voting credential is real, meaning that it casts votes that count.

3.2 System Model

TRIP consists of the following four actors present in most electronic voting schemes: a ledger, a registrar, an authority, and voters (Fig. 2). We now give an overview of each actor.

Ledger. The ledger $\mathbb{L}$ is an append-only, always available, publicly accessible data structure. To segment information for clarity, we refer to sub-ledgers whenever appropriate.

Registrar. The registrar $\mathbb{R}$ is responsible for enrolling voters deemed eligible on the given electoral roll. The registrar $\mathbb{R}$ consists of: (1) kiosks $\mathbb{K} = \{K_1, K_2, \ldots, K_{n_K}\}$, each located within a privacy booth, which provide voters their voting credentials; (2) envelope printers $\mathbb{P} = \{P_1, P_2, \ldots, P_{n_P}\}$, which supply envelopes used by voters while interacting with a kiosk; (3) registration officials $\mathbb{O} = \{O_1, O_2, \ldots, O_{n_O}\}$, represented by their *official supporting device* (OSD), who authenticate voters and authorize their credentialing sessions.

Authority. The tallying authority $\mathbb{A} = \{A_1, A_2, \ldots, A_{n_A}\}$ consists of n_A members who jointly process the ballots cast on the ledger to produce a publicly verifiable tally.

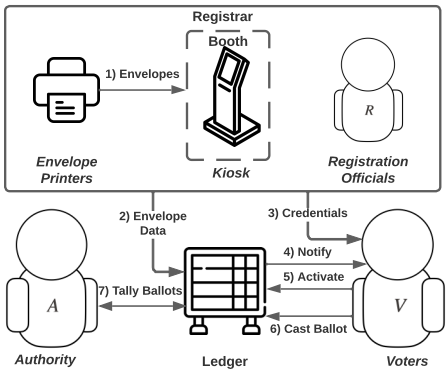


Figure 2: **TRIP System Actors**

The envelope printers (1) supply envelopes to booths, and (2) publish envelope data to ledger. Voters (3) obtain their credentials from registrar as shown in Fig. 1, (4) obtain a notification from their device via ledger, (5) activate their credentials, and (6) cast (real and fake) ballots. The authority (7) tallies ballots and publishes results.

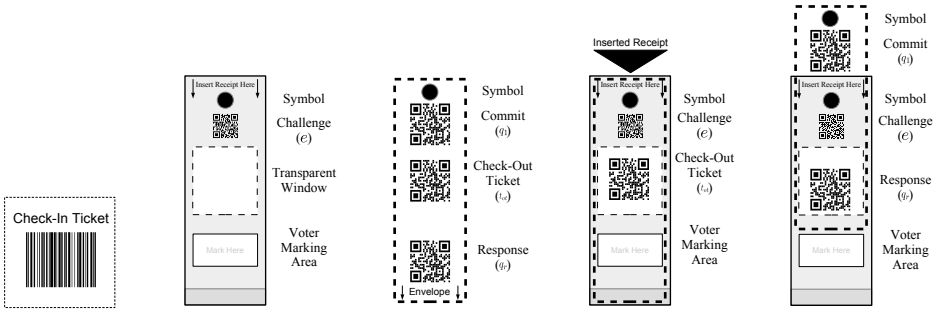
Voters. The set of voters on the electoral roll $\mathbb{V} = \{V_1, V_2, \ldots, V_{n_V}\}$ who receive their voting credential at the registrar, and then activate them on their *voter supporting device* (VSD) to cast their intended votes on the ledger. VSDs periodically monitor the ledger and inform voters of updates relevant to them, such as a completed registration session.

3.3 Actor Assumptions

To ensure the legitimacy of voters' credentials (I) and help them evade coercion (C), we present our actor assumptions.

Ledger. We assume the ledger is secure and ensures liveness (e.g., via Byzantine fault tolerant consensus). We assume that mechanisms exist for the ledger to certify the bindings between public keys and entities (except for voters).

Voters. We assume that for the integrity of all voters' real credentials, some percentage of voters can understand and visually follow the steps required to create their real credential and report any deviations or misbehavior (I1). We assume these voters typically pick envelopes randomly (I2). Finally,



(a) Check-In Ticket (b) Envelope (c) Receipt (d) Transport State (e) Activate State

Figure 3: **CheckIn, Receipt and Envelope Design.**
The check-in ticket is used to unlock a kiosk. An envelope contains a symbol (discussed in §7.1), a challenge QR code e, a transparent window and a voter marking area to help voters distinguish their credentials. A receipt contains three QR codes: commit, check-out ticket and response. The receipt inside an envelope forms the *transport* state; during VSD activation, the receipt is lifted 1/3 of its length to form the *activate* state.

we assume a significant number of voters activate their credentials before the tallying phase (I3).

We assume that for each voter under coercion, voters can attend a registrar office during open enrollment (C1), hide their real credential until activation (C2), have access to a device they trust somewhere (C3), open an anonymous channel with the election authority for online voting (C4), obtain an unsupervised coercion-free moment for activation and voting (C5), and lie to a coercer about having complied with the coercer's demands (C6). Assumptions C3 and C5 seem essential in any online private-voting system. Assumptions C2-5 (inclusive) could be relaxed in favor for stronger integrity assumptions via a vote-delegation extension discussed in Appendix B.

Registrar. We presume the existence of widespread information on the registration process, meaning that the details are widely recognized. In particular, we assume the instructional video shown at the registrar is also available online, and significant discrepancies between videos would be detected.

To maximize protection against coercion, we assume that the registrar protects voter privacy in the booth, and that the registrar correctly authenticates voters so that one voter cannot impersonate another. In addition, for each registration session, the registrar ensures each booth contains abundant envelopes. The thinness of envelopes ($<$ 1mm) makes it practical for each booth to contain a stack of hundreds or even thousands. We assume that a voter attempting to count the exact number of envelopes (without the aid of forbidden devices) would be time-consuming enough to risk raising questions by officials.

3.4 Threat Model

One threat-modeling challenge that e-voting systems encounter is that different realistic threats pertain to different situations. We cannot maximize system security simply by maximizing the power of a single conceptual adversary, but must model distinct adversaries around distinct situations. E-voting systems typically model separate (non-colluding) privacy and integrity adversaries, for example, so that some system elements can be trusted for privacy more (or less) than they are trusted for integrity.

Device / Adv.	Ledger	Authority	OSD	Kiosk	Envelope Printers	VSD w/ Cred. Real	VSD w/ Cred. Fake
Integrity	No	All	Yes*	Yes*	Yes*	No	Yes
Privacy	Yes	$n_A - 1$	Yes	Yes	Yes	No	Yes
Coercion	Yes	$n_A - 1$	No	No	No	No†	Yes

Table 1: **Threat Model**: This table depicts an adversary's capability to compromise a device and the entity or credential it represents.
* The risk of the adversary compromising voter registration, as discussed later in section 5, is small per individual registration session, and becomes negligible when considering the combined probability across all sessions.
† In cases where voters are unable to access a trusted device, they can delegate their vote to a trusted entity while interacting with the kiosk (Appendix B).

We build on this practice by further distinguishing between situations in which a voter is, or is not, under coercion. For the hopefully-common case of voters *not* under coercion, we wish to minimize the extent to which uncoerced voters must trust the system either for privacy or integrity, thereby maximizing the modeled adversary's control over system elements in these respects. However, for the hopefully-rare but important case of voters under coercion, we *must* consider the registrar to be trusted, at least by *this* (coerced) voter. Otherwise, the (coercion) adversary would effectively control both the voter and the registrar, leaving no plausible basis to bootstrap any kind of security. This trust relationship is consistent with in-person voting, where voters rely on the election authority to provide them with a safe space to cast their ballots, but may also scrutinize the election process to ensure its integrity.

We now informally define three distinct, non-colluding adversaries:[2] integrity $\mathcal{I}$, privacy $\mathcal{P}$ and coercion $\mathcal{C}$, and present in Table 1 a summary of which devices they may compromise. We assume that adversaries are computationally bounded and that the cryptographic primitives we use are secure.

Integrity. Integrity adversary $\mathcal{I}$'s goal is to manipulate the outcome of a voting event *without detection*, e.g., pre-empt,

[2]An availability adversary seeking to deny service is also relevant, but we assume this issue is addressed by the use of high-availability infrastructure.

alter or cancel votes. We assume there exists a well-known, accurate electoral roll that I cannot alter. We also assume this adversary cannot compromise the ledger nor the VSDs.

Privacy. Privacy adversary $\mathcal{P}$'s goal is to *reveal* a voter's real vote. We assume that this adversary cannot compromise VSDs containing voters' real credentials nor compromise all authority members. This latter assumption is common in privacy-enhancing systems [27, 51, 7, 37].

Coercion. $\mathcal{C}$'s goal is to determine whether a targeted voter complied with their demands. For example, through undue influence or vote-buying, $\mathcal{C}$ can demand voters to reveal their real credentials, cast $\mathcal{C}$-intended ballots, or even refrain from voting. $\mathcal{C}$ may control a subset of voters, who *comply* with $\mathcal{C}$'s demands. Similar to $\mathcal{P}$, $\mathcal{C}$ may not compromise all authority members nor the VSDs containing real credentials. In addition, $\mathcal{C}$ cannot compromise any of the registrar actors (kiosks, OSDs and envelope printers) nor prevent voters from performing voter registration. Finally, $\mathcal{C}$ cannot monitor the actions of voters while they are inside the booth at the registrar.[3]

3.5 Voter-Facing Design

We describe the TRIP voter registration process as experienced by a voter.

Instructional Video. On arrival at the registrar, the voter first views an instructional video illustrating the process from check-in to activation. The video introduces the notion of fake credentials, the processes to create real and fake credentials, and highlights the differences between the two processes.

Check-In. At check-in, the voter identifies themselves to the official and the official gives the voter a check-in ticket containing a bar code, and labeled "Check-In Ticket" (Fig. 3a).

Booth Entrance. Before entering the booth, the voter deposits any electronic devices in a lockable compartment, similar to those often found at a gym, museum, or embassy. Inside the booth, the voter finds a touchscreen kiosk with a built-in QR/Barcode reader, a receipt printer, a pen and a set of envelopes. Each envelope (Fig. 3b) is a hollow rectangle featuring a symbol, a QR code, a transparent window, and a rectangular outline for voter markings to allow voters to distinguish between credentials. Once a voter is ready to initiate the credentialing procedure, the kiosk displays instructional screens to guide the voter through the process. The voter is then prompted to scan their check-in ticket.

Real Credential Creation. The voter creates their real credential in 4 steps. Once the voter has scanned their check-in ticket (Step 1), the kiosk prints a symbol and a QR code. The voter is then asked to confirm on the screen the symbol printed along with a QR code (Step 2). Upon confirmation, the voter selects an envelope with the same symbol from a set of envelopes and scans it (Step 3). The kiosk then finalizes the paper receipt by printing two additional QR codes (Step 4). Finally, the voter tears the receipt from the printer and places it inside the envelope to complete their credential. To have the voter distinguish their real credential from fake credentials, the kiosk instructs the voter to use the provided pen to mark their real credential in some way the voter can remember. The reason for introducing symbols is explained in section §7.1.

We refer to the state in which the receipt is fully inserted as the *transport* state. In this state, the middle QR code on the receipt remains visible for an official to scan, while keeping the (sensitive) information in the top and bottom QR codes hidden. Maintaining the confidentiality of sensitive information is solely to prevent the voter from experiencing the inconvenience of re-registration in the event of a leak.

Education on Coercion and Vote Buying. The kiosk then educates the voter on coercion and vote buying, stating that "it is illegal for anyone to ask for your real credential, to offer money for it, or to pressure you to vote in any particular way." The kiosk then invites the voter to create a fake credential, stating that "a fake credential looks like a real credential, but votes cast with a fake credential do not count" and the voter "can give or sell a fake credential to anyone, including someone trying to buy or coerce your vote".

Fake Credential Creation. If the voter agrees to create a fake credential, this process happens in 2 steps. The voter picks and scans any envelope (Step 1) and the kiosk then prints the entire receipt containing three QR codes (Step 2). Similar to creating real credentials, the voter inserts the receipt into the envelope and marks the credential to distinguish it from their other credentials. The kiosk then invites the voter to create another fake credential and if the voter agrees, they repeat the steps outlined above. The voter can create as many fake credentials as desired within reason.[4] Once the voter has their credential(s), the kiosk reminds them of important information and instructs them to display one of their credentials at the check-out desk. Upon leaving the booth, the voter retrieves any deposited electronic devices.

Check-Out During the check-out process, the voter presents any one of their credentials (in its transport state) to the official, who then scans the check-out ticket visible through the credential's transparent window. After scanning the credential, the official informs the voter that their visit to the registrar is now complete. The voter will subsequently receive a notification of their visit on their VSD and via some out-of-band channel such as postal mail.

Activation Whenever the voter obtains their own private (coercion-free) moment, the voter can activate their real credential. They may activate their fake credentials at any time,

[3] We leave out of scope side channel attacks but discuss further in Appendix C

[4] We expect that registration officials will inquire if a voter spends an inordinate amount of time in a booth, thereby imposing an informal and nondeterministic upper bound on the number of fake credentials printed and resources consumed. An alternative solution would be for the kiosk to impose a per-voter randomized maximum quota of time or fake credentials.

including under coercion, following exactly the same steps. Upon opening the voting application, VSD instructs the voter to lift the receipt one third of its length out of the envelope, placing the credential in the *activate* state (Fig. 3e). This state reveals, in order, the receipt's top QR code outside the envelope, the envelope's QR code, and the receipt's bottom QR code inside the transparent window. The receipt's middle QR code is no longer visible. The device then prompts the voter to scan the three visible QR codes to complete activation.

After activation, the voter can cast ballots in current or future elections until a policy-determined expiration date, after which the voter must re-register in person. The device finally instructs the voter to discard the physical credential securely.[5] The device is unable to differentiate between real and fake credentials, as they are both activated and used in the same way. Should a voter lose their device, they would need to re-register for a new real credential, in effect converting their previous real credential into a fake credential.

4 TRIP Scheme Design Details

In this section, we present TRIP (Fig. 4), our coercion-resistant, voter-verifiable registration scheme.

Notation. For a finite set S, $s \leftarrow\!\$\ S$ denotes that s is sampled independently and uniformly at random from S. The symbol $\ominus$ represents exclusion from a collection of elements. We denote $a \Leftarrow b$ as appending b to a, $a \| b$ as concatenating b with a, $\mathbf{x}$ as a set of elements of type x, and $\mathbf{x}[i]$ as the ith entries of the vector $\mathbf{x}$. We use $\top$ and $\bot$ to indicate success and failure, respectively. Variables are summarized in Table 2.

Primitives. TRIP requires (1) the M-ElGamal scheme defined in JCJ MEG [47], (2) a distributed key generation scheme DKG, (3) a EUF-CMA signature scheme Sig, (4) a cryptographic secure hash function H, (5) a message authentication code scheme MAC, and (6) an interactive zero-knowledge proof of equality of discrete logarithms. These primitives are defined in Appendix D.

4.1 TRIP Functions

Setup. Setup (Fig. 9, Appendix A) initializes the core system actors (Ledger, Authority, and Registrar). Prior works often include registration as part of a general setup process, but we separate it to delineate registration cleanly:

- The ledger $\mathbb{L}$ becomes available and made accessible to all (including third) parties. We denote $\mathbb{L}_R, \mathbb{L}_E, \mathbb{L}_V$ the registration, envelope, and voting sub-ledgers, respectively.
- The authority members run DKG and outputs a private, public keypair for each authority member $(A_i^{\mathsf{sk}}, A_i^{\mathsf{pk}})$ and a collective public key $\mathbb{A}_{\mathsf{pk}}$ which is made available to all parties. $\mathbb{A}_{\mathsf{pk}}$ must be a generator of G_q.

[5] For voters not under coercion, the paper credential can be made immaterial after activation with the help of the tallying scheme (§8).

Symbol	Description
G, q, g	A cyclic group G of order q with generator g
$\mathbb{A}, \mathbb{O}, \mathbb{K}, \mathbb{P}, \mathbb{V}$	Authority, Officials, Kiosks, Envelope Printers, Electoral roll
n_A, n_O, n_K, n_P	Number of Authorities, Officials, Kiosks, Envelope printers
$\mathbb{L}, \mathbb{L}_R, \mathbb{L}_E$	Ledger, Registration & Envelope (sub-)ledgers
$V_{id}, c_{\mathsf{pc}}, c_{\mathsf{pk}}, c_{\mathsf{sk}}$	Voter's identifier, Public Credential, Public & Private Keys
$c, \tilde{c}_{\mathsf{pk}}, \tilde{c}_{\mathsf{sk}}$	A credential, A fake credential's Public and Private Keys
$\mathbf{E}, n_E, n_c$	Envelope challenges and number of envelopes, credentials
τ, s_{rk}	MAC signature tag, Official & kiosk shared secret key
OSD, VSD	Registration official and Voter supporting device, respectively
$t_{in}, t_{out}, q_c, q_r$	Check-In & Check-Out Tickets, Commit & Response Codes

Table 2: **Scheme Notations.**

$\mathrm{TRIP}(\lambda, (G, q, g_1, g_2), V_{id}, \mathbb{L}, \mathbb{A}, \mathbb{O}, \mathbb{K}, \mathbb{P}, \mathbf{E}, s_{rk})$

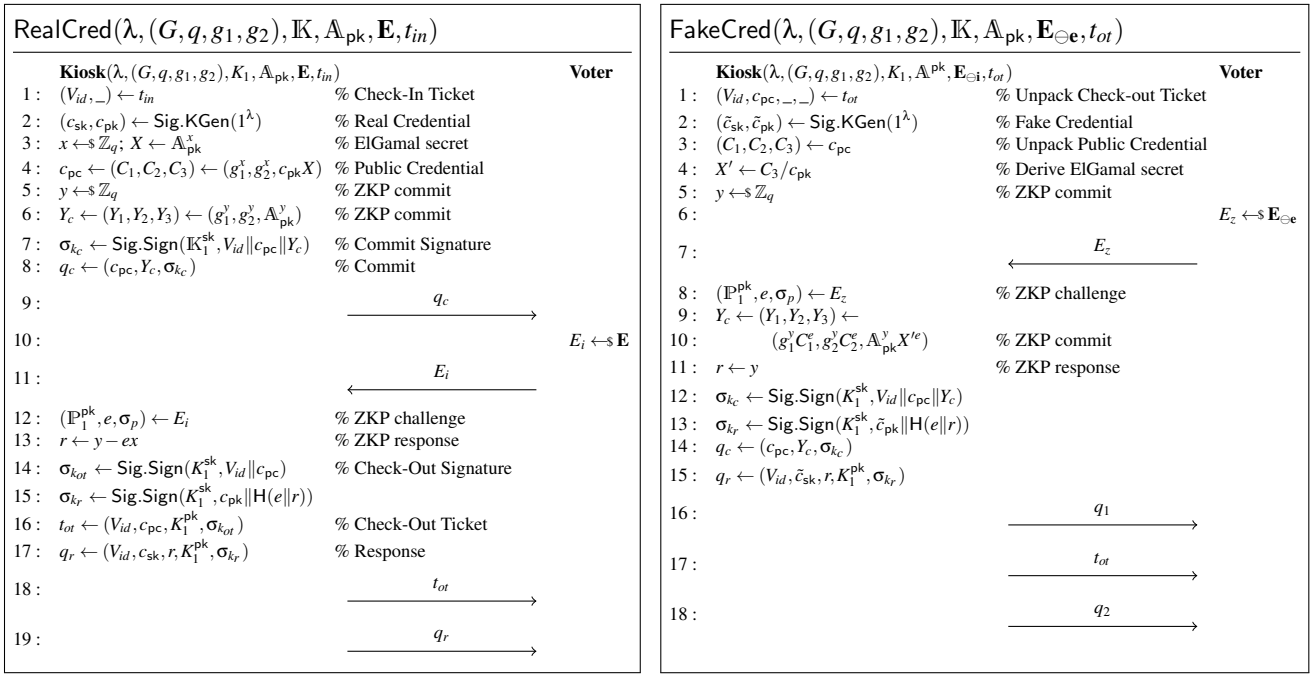
1: $t_{in} \leftarrow (V_{id}, \_) \leftarrow \mathsf{CheckIn}(\mathbb{O}, \mathbb{K}, s_{rk}, V_{id})$
2: $c \leftarrow (q_c, t_{ot}, q_r, e) \leftarrow ((c_{\mathsf{pc}}, Y_c, \sigma_{kc}), (V_{id}, c_{\mathsf{pc}}, K_1, \sigma_{k_{ot}}), (V_{id}, c_{\mathsf{sk}}, r, K_1, \sigma_{kr}), (P, \mathsf{H}(c), \sigma_p)) \leftarrow \mathsf{RealCred}(\lambda, (G, q, g_1, g_2), \mathbb{K}, \mathbb{A}_{\mathsf{pk}}, \mathbf{E}, t_{in})$
3: $\mathbf{e} \leftarrow \{e\}$; $\mathbf{c} \leftarrow \{c\}$ % Used challenges & vector of credentials
4: **for** 1 to $(n_c \leftarrow V()) - 1$ **do** % Chosen number of envelopes
5: $\quad \tilde{c} \leftarrow \mathsf{FakeCred}(\lambda, (G, q, g_1, g_2), \mathbb{K}, \mathbb{A}_{\mathsf{pk}}, \mathbf{E}_{\ominus \mathbf{e}}, t_{ot})$
6: $\quad \mathbf{c} \Leftarrow \tilde{c}$; $\mathbf{e} \Leftarrow \tilde{c}[e]$
7: **endfor**
8: $c_a \leftarrow\!\$\ \mathbf{c}$
9: $\mathbb{L} \leftarrow \mathsf{CheckOut}(\mathbb{L}, \mathbb{O}, \mathbb{K}^{\mathsf{pk}}, c_a[t_{ot}])$
10: **for** i to n_c **do**
11: $\quad \mathbb{L} \leftarrow \mathsf{Activate}(\mathbb{L}, V, \mathbf{c}_i)$
12: **endfor**

Figure 4: **TRIP** Registration process for a prospective voter.

- Each registrar actor (OSDs, kiosks & printers) generates their own private and public key pair using $\mathsf{Sig.KGen}(1^\lambda)$. The registrar uses the electoral roll $\mathbb{V}$ to populate $\mathbb{L}_R$ with each voter's unique identifier $\mathbb{V}_i^{id}$. Since actors are logical entities we use the first OSD (O_1), kiosk (K_1) and envelope printer (P_1) to register a voter $V \in \mathbb{V}$ (Fig. 4). The printer prints n_E (a number $\gg |\mathbb{V}|$) envelopes $\mathbf{E}$, where each envelope contains the printer's public key P, a cryptographic nonce $e \leftarrow\!\$\ \mathbb{Z}_q$, and a signature on this nonce $\sigma_p \leftarrow \mathsf{Sig.Sign}_p(\mathsf{H}(e))$. For each envelope, the printer also publishes $(P_1, \mathsf{H}(e), \sigma_p)$ to the ledger $\mathbb{L}_E$. The officials $\mathbb{O}$ and kiosks $\mathbb{K}$ generate a shared secret key s_{rk} to create and verify MAC authorization tags.

Check-In. At Check-In (Fig. 10, Appendix A), after successful authentication, OSD issues the voter a check-in ticket t_{in} consisting of the voter's identifier V_{id} and an authorization tag τ_r on V_{id}. When the voter presents their ticket to the kiosk, the kiosk verifies the tag τ_r. We use a MAC instead of a signature due to storage constraints in a barcode and a barcode instead of a QR code for usability reasons discussed in §7.

Real Credential. After verifying the authorization tag τ_r, the kiosk issues the voter their real credential while proving its correctness (Fig. 5). The kiosk first generates the voter's real credential's private and public keys $(c_{\mathsf{sk}}, c_{\mathsf{pk}})$ and M-ElGamal encrypts c_{pk} using the authority's public key $\mathbb{A}_{\mathsf{pk}}$ to obtain the voter's public credential c_{pc}. To prove that c_{pc} encrypts c_{pk} without revealing the M-ElGamal randomness secret x (to later enable the construction of fake credentials), the kiosk, as the prover, and the voter, as the verifier, run an interactive



(a) **Real Credential Creation Process.** Voter and kiosk follow a sound zero-knowledge proof construction.

(b) **Fake Credential Creation Process.** Voter and kiosk follow an unsound zero-knowledge proof construction. Envelopes cannot be reused.

Figure 5: **Voting Credential Creation Process.**

zero-knowledge proof of equality of discrete logarithms:[6] $\mathsf{ZKPoE}_{C_1,C_2,X}\{(x) : C_1 = g_1^x \wedge C_2 = g_2^x \wedge X = \mathbb{A}_{\mathsf{pk}}^x\}$. The kiosk first computes the commits $Y_1 = g_1^y$, $Y_2 = g_2^y$ and $Y_3 = \mathbb{A}_{\mathsf{pk}}^y$ for $y \leftarrow\!\!\$\ \mathbb{Z}_q$ and prints the commit q_c containing the voter's public credential c_{pc}, the commits (Y_1, Y_2, Y_3), and a signature σ_{k_c} on $(V_{id} \| c_{\mathsf{pc}} \| Y_c)$. The voter then supplies the kiosk with an envelope $E_i \leftarrow\!\!\$\ \mathbf{E}$ containing the challenge e. The kiosk finally computes the response $r = y - ex$, signatures $\sigma_{k_{ot}}$ and σ_{k_r}, and prints the check-out ticket t_{ot} and response q_r.

At this stage, the voter observes that the process follows the Σ-protocol sequence: commit, challenge, response. If the voter detects and reports an anomaly, we expect the registrar to direct the voter to another kiosk and inspect the one reported. The voter's device verifies computational correctness later.

Fake Credentials. To create a fake credential, the kiosk generates a new credential $(\tilde{c}_{\mathsf{sk}}, \tilde{c}_{\mathsf{pk}})$ and falsely proves that the public credential c_{pc} encrypts $\tilde{c}_{\mathsf{pk}}$. The kiosk first derives the M-ElGamal secret $X' \leftarrow C_3/\tilde{c}_{\mathsf{pk}}$. Evidently, the kiosk has no knowledge of an x' that satisfies $X' = \mathbb{A}_{\mathsf{pk}}^{x'}$, requiring one to solve the discrete logarithm problem. Instead, the kiosk and the voter follow an incorrect proof construction sequence that violates soundness without affecting the correctness of the computations. In this sequence, the voter first supplies a new envelope $E_z \leftarrow\!\!\$\ \mathbf{E}_{\ominus \mathbf{e}}$ to the kiosk, where $\mathbf{e}$ are the previously used envelopes/challenges. Then, the kiosk uses the new challenge e to compute a ZKP commit $(Y_1, Y_2, Y_3) \leftarrow (g_1^y C_1^e, g_2^y C_2^e, \mathbb{A}_{\mathsf{pk}}^y X'^e)$ for some $y \leftarrow\!\!\$\ \mathbb{Z}_q$ and the ZKP response $r \leftarrow y$. The kiosk finishes by computing signatures σ_{k_c} and σ_{k_r} and printing the commit q_c, check-out t_{ot}, and response q_r sequentially, where t_{ot} is identical (both in content and visually) to the one in the real credential process. The voter can repeat this process for any number of desired fake credentials (within reasonable limits mentioned in §3.3).

Check-Out. At check-out (Fig. 11, Appendix A), the registration official uses their OSD to scan one of the voter's credentials. This credential is shown in transport state (Fig. 3d), which reveals the contents of the check-out ticket t_{ot} but not the secrets to be used in activation.

The OSD first checks the credential's authenticity by checking the kiosk's public key $K_1^{'\mathsf{pk}} \in \mathbb{K}^{\mathsf{pk}}$, and verifying the signature $\sigma_{k_{ot}}$. OSD then provides their stamp of approval with a digital signature σ_o on the voter's identifier V_{id}, the voter's public credential c_{pc} and the kiosk's signature $\sigma_{k_{ot}}$. Finally, OSD updates the ledger entry V_{id} with $(c_{\mathsf{pc}}, K_1^{\mathsf{pk}}, \sigma_{k_{ot}}, O_1^{\mathsf{pk}}, \sigma_o)$. Once updated, the ledger $\mathbb{L}$ performs the necessary checks and the VSD notifies the voter about their recent voter registration session with information on how to report any irregularities.

Activation. During activation (Fig. 12, Appendix A), the voter uses their VSD to scan a credential in activate state (Fig. 3e). This reveals the commit q_c, envelope e, and response q_r; the check-out ticket t_{ot} is not visible. VSD then verifies the integrity of the credential by (1) verifying the signatures

[6] For cryptographic purposes, we equate the voter with the verifier. However, in reality, the voter only observes the order in which QR codes are printed on the receipt without needing to understand them. The voter's device is responsible for checking the actual proof transcripts.

Notation	*Description*
$\mathbb{R}$	Registrar (combines kiosks, envelope printers and officials)
$\mathbb{L}_V, X, P, \mathsf{st}$	Voting (sub-)ledger, Tally & Tally Proofs, (Adversarial) state
D^c, D^v	Probability distribution of fake credentials and votes
$\mathcal{C}, \mathbf{V}_C, j, \beta$	Coercer, $\mathcal{C}$-controlled voters, $\mathcal{C}$-target voter, $\mathcal{C}$-intended ballot
β, n_j^c, M	Target Vote, Target number of total credentials, Voting options
n_V, n_C, n_M	Number of voters, controlled voters and voting options

Table 3: **Additional Variables for Proofs.**

$(\sigma_{k_c}, \sigma_{k_r}, \sigma_p)$, (2) deriving the ElGamal secret X and verifying the ZKP, (3) checking whether the public credential c_{pc} matches the public credential on the ledger c'_{pc}, and (4) checking that the challenge e has not been used via the ledger $\mathbb{L}_E$. Upon success, the device publishes the challenge e on $\mathbb{L}_E$ and stores the credential c_{sk} for future voting. The device publishes the envelope challenge on the ledger for integrity, ensuring that the challenges are not re-used and are statistically random. Upon failure, VSD reports the offending actor based on the failure check, and instructs the voter to re-register.

5 Security Analysis

This section analyzes the security of TRIP against the adversaries outlined in our threat model (§3.4): integrity, privacy, and coercion. We presume that voters comply with our assumptions and defer discussion on human factors to §7.

Coercion. This section shows that $\mathcal{C}$ cannot determine whether the victim succumbed to the coercer's demands, thereby making coercion meaningless. We prove that TRIP is coercion-resistant by showing that the difference between $\mathcal{C}$'s winning probability in a real game (representing the adversary's interactions with our system) and in an ideal game (representing the desired level of coercion-resistance) is negligible. In both games, $\mathcal{C}$'s goal is to determine whether the target created only the real credential and handed it over, or also created and handed over a fake credential. Like the total number of votes cast in an election, we treat the total number of credentials created as public information: $\mathcal{C}$ could trivially win if it knew exactly how many (fake) credentials *all other* voters created. The adversary's uncertainty about the target voter thus derives from the other honest voters, each of whom creates an unknown (to the adversary $\mathcal{C}$) number of fake credentials, which we model as a probability distribution D^c. To achieve statistical uncertainty also on the voting choice, we adopt the same approach for the content of the ballot with the distribution D^v. This "anonymity among the honest voters" is exactly in line with standard reasoning by which votes themselves are considered to be (statistically) protected once anonymized. We present the ideal game in Fig. 6 to highlight TRIP's level of coercion-resistance and defer the proof to Appendix F.2.

In the game, adapted from JCJ [47], the coercer chooses the target voter j and $n_C < n_V$ controlled voters who abide by the coercer's demands. All voters create their real credential and

Game C-Resist-TRIP-Ideal$^{\mathcal{C}',b}(\lambda, \mathbb{V}, \mathbb{R}, \mathbb{A}, M, n_{\mathcal{C}})$

1: $\mathbf{V}_C, \{n_c^i\}_{1\le i\le|\mathbb{V}_C|}, \mathsf{st}_{\mathcal{C}'} \leftarrow \mathcal{C}'(\mathbb{V}, \text{"Choose controlled voter set"})$
2: $\mathbb{L}_R^{JCJ}, \mathbb{L}_E, n_V, n_M, n'_C \leftarrow \emptyset, \emptyset, |\mathbb{V}|, |M|, |\mathbf{V}_C|$
3: $(j, \beta, \mathsf{st}_{\mathcal{C}'}) \leftarrow \mathcal{C}'(\mathsf{st}_{\mathcal{C}'}, \text{"Pick target voter and their parameters"})$
4: **if** $n'_C \neq n_C$ or $j \notin \{1, 2, ..., n_V\} \setminus \mathbf{V}_C$ **then abort**
5: **for** $i = 1$ **to** n_V **do**
6: $\quad c_{\mathsf{sk}}^i, c_{\mathsf{pc}}^i, \_, \mathbb{L}_R^{JCJ} \leftarrow \mathsf{RealCred}(\mathbb{L}_R^{JCJ}, \mathbb{R}_{\mathsf{sk}}, \mathbb{V}_{id}^i, \lambda)$
7: $\quad$ **if** $i \in \mathbb{V} \setminus (\mathbf{V}_C \cup \{j\})$ **do** *% Honest voters create & activate fake creds*
8: $\quad\quad \tilde{\mathbf{c}}_{\mathsf{sk}}^i, \_ \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}}, \mathbb{V}_{id}^i, c_{\mathsf{pc}}^i, D^c_{1,[0,\inf]}, \lambda)$
9: $\quad\quad \_, \mathbb{L}_E \Leftarrow \left\{\mathsf{Activate}(\mathbb{L}_E, \tilde{c}_{\mathsf{sk}}^i)\right\}_{1\le i\le|\tilde{\mathbf{c}}|}$
10: $\quad$ **endif**
11: **endfor**
12: **if** $b = 0$ **then** *% Target voter evades — casts real vote*
13: $\quad \tilde{c}_{\mathsf{sk}}^j, \_ \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}}, \mathbb{V}_{id}^j, c_{\mathsf{pc}}^j, 1, \lambda)$
14: $\quad \_, \mathbb{L}_E \Leftarrow \left\{\mathsf{Activate}(\mathbb{L}_E, c_{\mathsf{sk}}^j), \mathsf{Activate}(\mathbb{L}_E, \tilde{c}_{\mathsf{sk}}^j)\right\}$
15: $\quad \mathbb{L}_V^{JCJ} \Leftarrow \mathsf{Vote}(c_{\mathsf{sk}}^j, \mathbb{A}_{\mathsf{pk}}, M, \beta, \lambda)$
16: **endif**
17: $c_{\mathcal{C}'} \leftarrow c_{\mathsf{sk}}^j$ *% Target voter always releases their real credential*
18: $\mathbb{L}_V^{JCJ} \Leftarrow \left\{\mathsf{Vote}(c_{\mathsf{sk}}^i, \mathbb{A}_{\mathsf{pk}}, M, D^v_{n_V - n_C, n_O}, \lambda)\right\}_{i\in\mathbb{V}\setminus\{\mathbf{V}_C\cup j\}}$
19: $\mathbb{L}_V^{JCJ} \Leftarrow \mathcal{C}'(\mathsf{st}_{\mathcal{C}'}, \mathbb{L}_R^{JCJ}, c_{\mathcal{C}'}, \{c_{\mathsf{sk}}^i\}_{i\in\mathbf{V}_C}, \lambda, \text{"Coercer casts ballots"})$
20: $(\mathbf{X}, \_) \leftarrow \mathsf{Ideal\text{-}Tally}(\mathbb{A}_{\mathsf{sk}}, \mathbb{L}_V^{JCJ}, M, \mathbb{L}_R^{JCJ}, \lambda)$
21: $b' \leftarrow \mathcal{C}'(\mathsf{st}_{\mathcal{C}'}, \mathbf{X}, \mathbb{L}_E, \text{"Guess } b\text{"})$
22: **return** $b' = b$

Figure 6: **Game C-Resist-TRIP-Ideal.** The TRIP ideal game for coercion-resistance modified from JCJ to take into account the adversary's probabilistic knowledge of honest voters' fake credentials.

honest voters create and activate fake credentials. The envelope ledger $\mathbb{L}_E$'s content (line 21) releases to $\mathcal{C}$ the number of total credentials created during Activate. The distribution D^c models this number, i.e., the uncertainty, of fake credentials *created and activated* by honest voters, which can be zero.

If $b = 0$, the voter uses the real credential to cast a vote. The challenger then gives $\mathcal{C}$ the real credentials of the controlled voters and the target voter. The honest voters and the coercer then proceed to cast ballots, where honest votes—and thus the adversarial statistical uncertainty—are drawn from the distribution D^v. The degree of coercion-resistance is bounded by the adversary's uncertainty about the voting behavior of honest voters. For tallying, we use Ideal-Tally, an algorithm adopted from JCJ [47], which tallies all votes from honest and controlled voters, but only tally the adversary's vote using the voter's real credential if $b = 1$. Given the tally along with the previously given credentials, the adversary guesses the bit b, i.e., whether the victim voter has cast a ballot. The real game and proof of the following theorem are in Appendix F.2.

Theorem 1 (Coercion-resistance, informal)**.** *The TRIP registration scheme (within the JCJ e-voting scheme [47]) is coercion-resistant under the decisional-Diffie-Hellman assumption in the random oracle model.*

Integrity. The goal of the integrity adversary $\mathcal{I}$ is to manipulate the outcome of an election *without being detected* (§3.4). For a voter registration system, this means that voters must be able to verify that their real credential will cast votes that count in elections. We first show that $\mathcal{I}$'s winning probabil-

Game $\mathsf{IV}^I(\mathbb{V}, j, pars)$

1 : $(\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}}, \mathbb{R}_{\mathsf{sk}}, \mathbf{c}_{\ominus j}, \mathbb{L})$
2 : $\quad \leftarrow \mathsf{Setup}_I(pars)$
3 : $\mathbf{c}_j \leftarrow \mathsf{Register}_I($
4 : $\quad \mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}\ominus 1}, \mathbb{R}_{\mathsf{sk}}, \mathbb{V}_j)$
5 : $s \leftarrow I(\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}\ominus 1})$
6 : $I^{V(c_j,s),\mathbb{L},E(\mathbb{A}_1^{\mathsf{sk}})}(\mathbb{A}_{\mathsf{pk}}, \mathbf{c})$
7 : **if** $\mathsf{bad}_{\mathsf{iv}}$ **return** 1

(a) **Individual Verifiability (Ind-Ver).** Definition of individual verifiability adapted from [18] to account for registration and adversary's access to the voter's real credential.

Game $\mathsf{I\text{-}IV}^I(\mathbb{V}, \lambda_v, pars)$

1 : $r \leftarrow 0; I(\lambda_v)$
2 : **for** $i = 1$ to $|\mathbb{V}|$ **do**
3 : $\quad r \mathrel{+}= \mathsf{IV}^I(\mathbb{V}, i, pars)$
4 : **endfor**
5 : **if** $r \geq \lambda_v$ **return** 1

(b) **Iterative Ind-Ver** Definition of iterative individual verifiability to account for an adversary targeting a group of voters to change an election outcome with respect to the winner of an election.

Figure 7: **Games for Defining (Iterative) Individual Verifiability**

ity in individual verifiability is small but not negligible, and then that the probability is negligible under our new iterative individual verifiability definition.

Our definition builds on the work of Bernhard et al. [18], which considers vote casting: a voter is convinced that their vote was cast as intended. This definition is however unsuitable for a registration process as we aim to detect whether an adversary tampered with the construction of a voter's credential. We present our definition in Fig. 7a.

In the game, after the Setup function, Register returns the real credential of the voter; both functions are under partial adversarial control. After the real credential creation, the adversary chooses and casts a vote. Event $\mathsf{bad}_{\mathsf{iv}}$ (line 7) occurs when statements (1) $\mathsf{Extract}(\mathbb{L}, c_{\mathsf{pc}}) \neq c_j \vee_* \mathsf{check}_*$ and (2) $\mathsf{Extract}(b, \mathbb{A}_{\mathsf{pk}}) = s$ are true. The first is true if the public credential c_{pc} on the ledger is not an encryption of the real credential c_j and the voter (and their VSD) has approved all checks. In other words, the statement is true if the adversary successfully tricks the user in accepting a wrong real credential. The second statement is true if the ledger does not contain a ballot b that encrypts the adversarially-chosen vote s.

For a voter registration scheme we focus on the first statement to verify that the public credential c_{pc} is a correct encryption of c_j and leave vote-casting verifiability to the voting scheme. In TRIP, a voter together with their VSD accepts their real credential only if they approve check_1: the (visual) construction of their real credential (steps described in §3.5), check_2: their voter registration session (VSD notification described in §3.2), and check_3: the successful activation of their real credential as outputted by VSD (§4.1). We define the advantage of the adversary I against individual verifiability is $\mathsf{Adv}_I^{\mathsf{IV}} = \Pr[\mathsf{IV}^I = 1]$. In Theorem 2 (and formally in Theorem 5), we show that the advantage is small but not negligible.

Theorem 2 (Integrity, informal)**.** *For a security parameter $\lambda \in \mathbb{N}$ and the integrity adversary I against the* TRIP *registration scheme the following holds:* $\mathsf{Adv}_I^{\mathsf{IV}} \leq (\mathbf{e}[e]/n_e)\,(n_u/n_e)^{n_c} + \mathsf{negl}(\lambda)$*, where $\mathbf{e}[e]$ is the number of times that the same challenge e appears in the set of envelopes* $\mathbf{e}$*, and n_e, n_u, n_c is the number of envelopes, unique printed challenges and voter-created credentials, respectively.*

Many verifiability schemes, such as Benaloh challenge [14], rely on the voter repeating a task multiple times to achieve negligibility, but we cannot rely on n_c as a security parameter, expecting each voter to, create at least 128 credentials. For comparison, TRIP achieves a 0.5% advantage with 50 unique challenges, 100 envelopes, and one fake credential, while the adversary in Benaloh always holds a 50% advantage. In Theorem 5 (Appendix F.3), we prove that I's advantage is negligible in the security parameter λ_v under the new definition of *iterative individual verifiability* (Fig. 7b), where λ_v represents the number of times that I must win the game IV to successfully win against $\mathsf{I\text{-}IV}$. This game suits schemes that aim to detect the manipulation of an election's result instead of single voter's vote.

TRIP enables voters to verify the integrity of their real credential, but the registrar, who creates these credentials, has access to them and therefore may cast votes without the voters' consent. We propose two solutions in §8.

Privacy. The goal of a privacy adversary $\mathcal{P}$ is to reveal the content of a voter's real vote. We only provide an informal analysis as privacy is solely relevant to the *ballot's contents*, but voting and tallying are out of scope. To win, $\mathcal{P}$ needs to (1) identify and (2) decrypt the ciphertext that encrypts the real vote. Given our threat model for $\mathcal{P}$ commonly used in e-voting (§3.4 & §2.2), $\mathcal{P}$ can complete the first objective by compromising the kiosk but fails to complete the second: it cannot compromise VSDs to view the plaintext vote nor compromise all talliers to decrypt the encrypted vote.

6 Implementation & Evaluation

We implemented and measured the energy consumption of two TRIP prototypes to assess their suitability for deployment in cost and power constrained environments, especially in regions with high corruption [6, 32] where the use of TRIP could help improve election integrity.

Software. We implemented TRIP in 750 lines of Go [2] and leveraged Kyber [3] and gozxing for cryptographic and QR code operations, respectively. We employ M-ElGamal encryption, Schnorr signatures [67], and SHA-256. Our implementation uses the Ed25519 elliptic curve group for signatures and encryption as it produces QR code-friendly outputs.

Prototypes. We developed two prototypes for our project. The first prototype is a Raspberry Pi with a touchscreen LCD, costing approximately $270.66, with the majority of the cost being the 7" touchscreen. The second prototype is a factory-made touchscreen kiosk with a QR/Barcode reader along with a dedicated EPSON receipt printer (**??**). A complete list of components for both prototypes can be found in Appendix E.

Power Usage. We measured the power consumption of the first prototype using a USB power meter (in mWh), and the

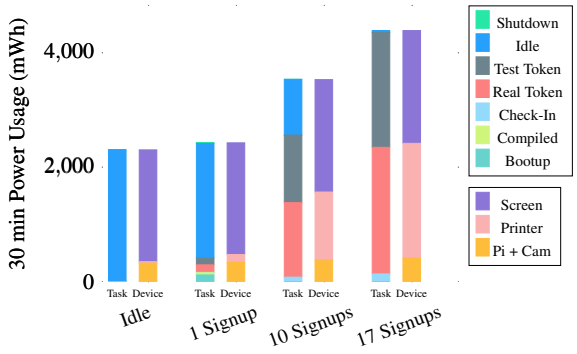


Figure 8: **Kiosk Power Usage** Task and Device.

second prototype using a socket power meter (in Wh). We performed four scenarios in our evaluation: (1) The idle scenario as the control, where the device powered on with the screen displaying the desktop and the printer remaining on standby; (2) The "1 Signup" scenario, where the kiosk powered on, compiled the voter registration program and performed a 2-credential voter registration session and then powered off at 30 minutes from startup; (3) The "10 Signup" scenario, where the kiosk performed ten 2-credential consecutive registrations without performing overhead tasks and then powered off at 30 minutes from startup; and (4) The maximum scenario, where the kiosk performed the maximum number of registrations in 30 minutes. We provide the measurements for the first prototype grouped by task and device usage in mWh over 30 min in §6. The measurements for the second prototype in Wh, which combined both the kiosk and the printer, were (1) 4.0Wh, (2) 4.0Wh, (3) 4.1Wh and (4) 359 signups at 10.5Wh.

Our results indicate that, when on idle, prototype 2's power consumption is about twice as much of prototype's 1 power consumption. However, at 10 signups, both prototypes show comparable power results (3.5 vs. 4 Wh). While prototype 2 can handle a substantial 359 signups compared to prototype 1's 17 signups, it consumes over twice as much power (10.5 vs 4.5 Wh). Prototype 2's receipt printer can print 20x faster than the prototype 1's receipt printer, but comes at a 3x increase in financial costs. In addition, it is also unrealistic to have 359 signups occur in a 30-minute period, while 17 signups is more realistic within that timeframe or even less.

7 Registration Process User Study

This section presents a summary of a user study conducted with TRIP. See the following paper for full details: "E-Vote Your Conscience: Voter Perceptions on Coercion and Vote Buying, and the Usability of Fake Voting Credentials for Online Voting." This study evaluated whether ordinary voters can understand and use TRIP by conducting a study with 150 paid participants, recruited over 3 months at a suburban park of a major metropolitan city. This study consists of a diverse group of adults, where age ranges from 19 to 83, with a median and mean of 36.5 and 44, respectively. We do not include full details of this study, but briefly summarize its findings on two key questions: system usability and the rate at which participants identify and report a malicious kiosk.

Groups	(C)ontrol	(F)ake Creds.	(M)alicious Kiosk	(S) & (F)	(S) & (M)
Scales			*System Usability Scale*		
Overall	69.6	70.4	69.9	67.3	62.7
SD	18.6	18.6	17.4	19.8	21.9
N	29	28	29	30	29
Percentile	55.1	57.8	56.1	47.8	35.0
Usability	69.5	69.8	68.4	67.7	62.5
Learnability	69.8	73.2	75.9	65.8	63.4
Reporting			*Malicious Kiosk Detection*		
Facilitator	0%	0%	10%	0%	47%
Survey	10%	7%	20%	7%	57%

Table 4: **System Usability Scale & Malicious Kiosk Detection** The top half displays the System Usability Scale (SUS), including standard deviation (SD) and sample size (N). SUS has two subscales: Usability and Learnability. The Percentile Rank measures the group's usability relative to 446 studies [66] The bottom half presents the proportion of participants who reported the kiosk's misbehavior to the facilitator or on the survey. Security priming significantly increases reporting rate (chi-squared $p = 0.004$) but at the cost of a perceived decrease in system usability.

7.1 Study Design

The study simulated voter registration according to §3.5, with variations for A/B testing: The 150 participants were randomly divided equally into 1 control and 4 experimental groups: (C) only real credential, (F) real and fake credentials, (M) group F but with a misbehaving kiosk, (SF) group F but with security priming, (SM) group SF but with a misbehaving kiosk. The control group only received a real credential known as a "voting credential". In group F, participants experienced the anticipated voter registration process (§3.5): real and fake credentials and a well-intentioned kiosk. In group M, participants encountered a "malicious" kiosk that instructed them to scan an envelope before printing a symbol and QR code. The difference between F and SF along with M and SM is that they exposed participants to an instructional video that explicitly stated that the kiosk can, in the unlikely event, misbehave. This video includes the visual cues of a misbehaving kiosk that wants to violate ZKP soundness.

7.2 Voting

After check-out, the study asked participants to identify and cast a vote using their real credential. From the possible 120 participants who could create a fake credential, 92 did so, with 90% of these 92 successfully casting a mock vote using their real credential. Overall, TRIP had a success rate of 83%, after taking into account those who required assistance during the study.

7.3 System Usability Scale

The study evaluated usability with the commonly used System Usability Scale (SUS) [21, 12, 58], which measures an individual's subjective assessment of the system's appropriateness for its purpose. SUS consists of ten standardized statements, where participants rate their agreement with them from *Strongly Disagree* to *Strongly Agree* (a 5 point Likert Scale [55]). Table 4 shows each group's SUS score. The average SUS score from 446 studies of diverse systems, including websites and hardware, is 68 with a standard deviation of 12.5 [66]. TRIP (Group F) achieved a slightly-above-average SUS score of 70.4, which falls within the 57.8th percentile of these studies and according to Bangor et al. [11], is "Acceptable" and earns an adjective rating of "Good", despite having participants unknowingly participate in a cryptographic protocol. This suggests that TRIP is usable, and the slight increase in the SUS score of participants who were exposed to both real and fake credentials compared to the control group only exposed to real credentials, defies the expectation of a perceived drop in usability. The noticeable difference in SUS scores between group F and groups SF and SM suggests that the participants perceived the system as less usable due to the exposure to a video on system misbehavior. However, it may be the exposure and detection of this misbehaving kiosk that brings down the SUS score the most since the difference is two times greater with SM than with SF. Such malicious behavior should (hopefully) not occur in practice.

7.4 Malicious Kiosk Detection

Since TRIP assumes that a reasonable percentage of voters can detect and report a malicious kiosk (§3.3), the study estimated this reporting rate with ordinary voters. As expected, no participant who interacted with an honest kiosk (Group C, F, and SF) reported any misbehavior to a facilitator. A small number of participants reported oddities with the kiosk in the survey, but their responses were related to confusion with the credentialing process. Participants in Group M and SM interacted with a misbehaving kiosk that instructs voters to scan an envelope after scanning their check-in ticket and removes any related material around the correct process. In group M, 10% and 20% of participants reported the kiosk's misbehavior to the facilitator and on the survey, respectively. This discrepancy between the reporting rates may have been due to being uncomfortable in expressing their doubt, but the survey gave a medium to express such doubts, as shown in the non-security priming groups. In group SM, 47% of participants reported the misbehavior to the facilitator and an additional 10% reported it on the survey. In fact, the security priming in group SM resulted in a statistically significant improvement over group M (chi-squared test, $p < 0.01$) with the SUS score near the average SUS score of 68.

8 Limitations and Future Work

This section discusses limitations and areas for future work.

Mitigating Real Credential Exposure. TRIP registrar, which issues voters' real credentials, may cast real votes without the voters' consent. One potential solution[7] is for the kiosk to produce not the voting credential itself but a single-use token that the VSD "spends" at activation time to create a fresh voting credential. The voter would know if this token has been spent before activation, and if not, only the VSD ever holds the private key for vote casting.

Enhancing Voting and Tallying. The JCJ scheme suffers from a simple but powerful denial of service attack where anyone can freely generate virtually unlimited fake credentials and subsequently cast a ballot with each. The system has no choice but to accept and tally *every* ballot, otherwise a rejected ballot reveals information as to whether a ballot contains a real or fake credential: it is impossible to reject a ballot containing a real credential and meet individual verifiability. With TRIP, the kiosk could generate a distinct signature on each credential issued, which voters must release to have their vote count, allowing the ledger to only accept kiosk-issued credentials.

Electoral Roll Integrity Dilemma. Practically all e-voting systems rely on electoral rolls for bootstrapping. In reality, electoral rolls are not always accurate, revealing instances of millions of ghost and duplicate voters [5]. In turn, for transparency some countries have publicly made available their electoral rolls [4]. However, the disclosure of such information could lead to coercion and vote-buying, such as preventing voters from registering to vote or targeting voters in swing districts. Moreover, companies [63] and political parties [68] have also used this information to influence voters. Unfortunately, addressing this problem is challenging, seemingly requiring yet another delicate balance between transparency and resistance to undue influence but remains necessary to ensure free and fair elections.

9 Related Work

Voter registration plays a critical role in maintaining election legitimacy, yet most works on coercion-resistance focus on tallying [27, 70, 10, 16, 71, 56]. To address this gap, works explored solutions such as distributing trust among multiple registrars [27] using trusted hardware [60, 59, 35, 34], or having the voter generate the real credential [49]. In Civitas, voters need to interact with multiple registration tellers to create their real credential and *a trusted for coercion* teller to create fake credentials. In TRIP, we also assume a trusted for coercion entity (kiosk), but interacts with the kiosk to create both their real and fake credentials, eliminating the

[7] A second potential solution would be to record a voter-recognizable event on the ledger when a ballot is cast, so that the VSD can notify the voter of any ballots cast unexpectedly.

need to interact with (and repeat the same steps) with multiple tellers. In Krivoruchko's work [49], voters generate their real credential and bring its public key to the registrar, requiring a device and private moment beforehand. TRIP eliminates these requirements prior to activation, and offers voters the ability to give their real paper credential to a trusted third party instead of using a device.

Several works [60, 59, 35, 34] suggest using smart cards for credential issuance. However, their usage presents dependence on trusted hardware assumptions and the risk of seizure by a coercer. Allowing voters to obtain multiple smart cards is a possible solution for the latter, but could incur high cost and environmental impact. Instead, TRIP uses QR codes on receipt paper for credential issuance and protects sensitive information through its envelope-receipt design.

The challenge with remote bootstrapping of real and fake credentials is the lack of a mechanism for voters to signal whether they are acting under coercion. The advantage of in-person bootstrapping of credentials inside a private booth is that it ensuresthat voters are acting under their own free will. Now, when a voter is under coercion in a remote setting, they have the means by using their real or fake credential to "signal" to the election system to count or not count this vote.

Prior work has used interactive Zero-Knowledge Proofs (ZKPs) for receipt-free, voter-verifiable voting with Direct Recording Electronic (DRE) machines [57, 24]. In particular, Moran and Noar's approach [20] relied on the DRE machine to preserve an opaque shield over part of the receipt and asked voters to enter random words as a challenge. In contrast, TRIP uses interactive ZKPs for voter registration and simplifies the process for users with a design that involves selecting an envelope and scanning its QR code. In TRIP, we don't consider necessary to shield the printed commitment from the voter's view: it is hard to interpret or compute cryptographic functions without electronic devices.

10 Conclusion

TRIP is a voter registration scheme designed for a coercion-resistant remote e-voting systems using fake credentials. TRIP enables voters to verify the authenticity of their real credentials, while also protecting them against coercion. To evaluate TRIP, we proved security against integrity and coercion adversaries and evaluated two prototypes In addition, we summarized results from a study on TRIP which suggests that TRIP is usable. We discussed mitigation strategies to TRIP's exposure of the real credential, improvements to JCJ-style voting and tallying schemes with TRIP, and presented an electoral roll integrity dilemma.

References

[1] Failure to Vote. Western Australian Electoral Commission.

[2] The Go Programming Language.

[3] Kyber Advanced Crypto Library for Go.

[4] Open Election Data Initiative.

[5] 85 million fake or duplicate names on electoral rolls: EC, February 2015.

[6] Electricity Sector Corruption Perceptions Index 2017: Promoting Accountability in Nigeria's Electricity Sector. Technical report, Stakeholder Democracy Network, 2017.

[7] Protocol of the Swiss Post Voting System: Computational Proof of Complete Verifiability and Privacy. Technical Report 1.0.0, Swiss Post, 2021.

[8] Dirk Achenbach, Carmen Kempka, Bernhard Löwe, and Jörn Müller-Quade. Improved coercion-resistant electronic elections through deniable re-voting. *USENIX Journal of Election Technology and Systems (JETS)*, August 2015.

[9] Ben Adida. Helios: Web-based Open-Audit Voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[10] Roberto Araújo, Sébastien Foulle, and Jacques Traoré. A Practical and Secure Coercion-Resistant Scheme for Internet Voting. In *Towards Trustworthy Elections: New Directions in Electronic Voting*, pages 330–342. Springer, Berlin, Heidelberg, 2010.

[11] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, 2009.

[12] Aaron Bangor, Philip T. Kortum, and James T. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, July 2008.

[13] Josh Benaloh. *Verifiable Secret-Ballot Elections*. Ph.D. Thesis, September 1987.

[14] Josh Benaloh. Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In *2007 USENIX/ACCURATE Electronic Voting Technology Workshop*, Boston, MA, August 2007.

[15] Josh Benaloh. Rethinking Voter Coercion: The Realities Imposed by Technology. *USENIX Journal of Election Technology and Systems*, 1(1), 2013.

[16] Josh Benaloh, Tal Moran, Lee Naish, Kim Ramchen, and Vanessa Teague. Shuffle-Sum: Coercion-Resistant Verifiable Tallying for STV Voting. *IEEE Transactions on Information Forensics and Security*, 4(4):685–698, December 2009.

[17] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the twenty-sixth annual ACM symposium on Theory of Computing*, STOC '94, pages 544–553, New York, NY, USA, May 1994.

[18] David Bernhard, Véronique Cortier, Pierrick Gaudry, Mathieu Turuani, and Bogdan Warinschi. Verifiability Analysis of CHVote, October 2018.

[19] David Bernhard, Oksana Kulyk, and Melanie Volkamer. Security Proofs for Participation Privacy, Receipt-Freeness and Ballot Privacy for the Helios Voting Scheme. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 1–10, New York, NY, USA, August 2017.

[20] Matthew Bernhard, Allison McDonald, Henry Meng, Jensen Hwa, Nakul Bajaj, Kevin Chang, and J. Alex Halderman. Can Voters Detect Malicious Manipulation of Ballot Marking Devices? In *2020 IEEE Symposium on Security and Privacy*, pages 679–694, May 2020.

[21] John Brooke. SUS: A 'Quick and Dirty' Usability Scale. In *Usability Evaluation In Industry*. CRC Press, 1996.

[22] Miguel Carreras and Yasemin İrepoğlu. Trust in elections, vote buying, and turnout in Latin America. *Electoral Studies*, 32(4):609–619, December 2013.

[23] Eleno Castro and Randy Kotti. Saving Democracy: Reducing Gang Influence on Political Elections in El Salvador. Master's thesis, March 2022.

[24] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, January 2004.

[25] David Chaum and Torben Pryds Pedersen. Wallet Databases with Observers. In *Advances in Cryptology — CRYPTO' 92*, pages 89–105, Berlin, Heidelberg, 1993.

[26] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981.

[27] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. In *2008 IEEE Symposium on Security and Privacy*, pages 354–368, May 2008.

[28] Véronique Cortier, David Galindo, Ralf Küsters, Johannes Müller, and Tomasz Truderung. SoK: Verifiability Notions for E-Voting Protocols. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 779–798, May 2016. ISSN: 2375-1207.

[29] Malcolm Crook and Tom Crook. Reforming Voting Practices In a Global Age: The Making and Remaking of the Modern Secret Ballot in Britain, France and the United States c. 1600—c. 1950. *Past & Present*, (212):199–237, 2011.

[30] Philip Daian, Tyler Kell, Ian Miers, and Ari Juels. On-Chain Vote Buying and the Rise of Dark DAOs, July 2018.

[31] Jason Daley. Lessons in the Decline of Democracy From the Ruined Roman Republic. *Smithsonian Magazine*, November 2018.

[32] Kumar Biswajit Debnath and Monjur Mourshed. Corruption Significantly Increases the Capital Cost of Power Plants in Developing Contexts. *Frontiers in Energy Research*, 6, 2018.

[33] Piret Ehin, Mihkel Solvak, Jan Willemson, and Priit Vinkel. Internet voting in Estonia 2005–2019: Evidence from eleven elections. *Government Information Quarterly*, 39(4):101718, October 2022.

[34] Ehsan Estaji, Thomas Haines, Kristian Gjøsteen, Peter B. Rønne, Peter Y. A. Ryan, and Najmeh Soroush. Revisiting Practical and Usable Coercion-Resistant Remote E-Voting. In *E-Vote-ID 2020: Electronic Voting*, pages 50–66, 2020.

[35] Christian Feier, Stephan Neumann, and Melanie Volkamer. Coercion-Resistant Internet Voting in Practice. In *Informatik 2014*, pages 1401–1414, Bonn, 2014.

[36] Dragan Filipovich, Miguel Niño-Zarazúa, and Alma Santillán Hernández. Voter coercion and pro-poor redistribution in rural Mexico. July 2021.

[37] David Froelicher, Patricia Egger, João Sá Sousa, Jean Louis Raisaro, Zhicong Huang, Christian Vincent Mouchet, Bryan Ford, and Jean-Pierre Hubaux. UnLynx: A Decentralized System for Privacy-Conscious Data Sharing. In *Proceedings on Privacy Enhancing Technologies*, volume 4, pages 152–170, 2017.

[38] Timothy Frye, Ora John Reuter, and David Szakonyi. Hitting Them With Carrots: Voter Intimidation and Vote Buying in Russia. *British Journal of Political Science*, 49(3):857–881, July 2019.

[39] Livia Gershon. Why Do We Vote by Secret Ballot?, October 2020.

[40] Ezequiel Gonzalez-Ocantos, Chad Kiewiet de Jonge, Carlos Meléndez, David Nickerson, and Javier Osorio. Carrots and sticks: Experimental evidence of vote-buying and voter intimidation in Guatemala. *Journal of Peace Research*, 57(1):46–61, January 2020.

[41] Michael Graff and Nick Ochsner. 'This Smacks of Something Gone Awry': A True Tale of Absentee Vote Fraud, November 2021.

[42] Sven Heiberg and Jan Willemson. Verifiable internet voting in Estonia. In *2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*, pages 1–8, October 2014.

[43] Martin Hirt and Kazue Sako. Efficient Receipt-Free Voting Based on Homomorphic Encryption. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*. Springer, 2000.

[44] Markus Jakobsson and Ari Juels. Mix and Match: Secure Function Evaluation via Ciphertexts. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, Lecture Notes in Computer Science, pages 162–177, Berlin, Heidelberg, 2000. Springer.

[45] Hugo L. Jonker and Erik P. de Vink. Formalising Receipt-Freeness. In *Information Security*, 2006.

[46] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections (extended abstract). In *Workshop on Privacy in the Electronic Society (WPES)*, November 2005.

[47] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In *Towards Trustworthy Elections: New Directions in Electronic Voting*, pages 37–63. Berlin, Heidelberg, 2010.

[48] Dalia Khader, Qiang Tang, and Peter YA Ryan. Proving prêt à voter receipt free using computational security models. In *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*, 2013.

[49] Taisya Krivoruchko. Robust Coercion-Resistant Registration for Remote E-voting (extended abstract). In *Proceedings of the IAVoSS Workshop on Trustworthy Elections*, 2007.

[50] Oksana Kulyk and Stephan Neumann. Human Factors in Coercion Resistant Internet Voting–A Review of Existing Solutions and Open Challenges. In *Proceedings of the Fifth International Joint Conference on Electronic Voting*, page 189. TalTech press, 2020.

[51] Ralf Küsters, Julian Liedtke, Johannes Müller, Daniel Rausch, and Andreas Vogt. Ordinos: A Verifiable Tally-Hiding E-Voting System. In *2020 IEEE European Symposium on Security and Privacy*, pages 216–235, September 2020.

[52] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A Game-Based Definition of Coercion-Resistance and Its Applications. In *2010 23rd IEEE Computer Security Foundations Symposium*, pages 122–136, July 2010. ISSN: 2377-5459.

[53] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Verifiability, Privacy, and Coercion-Resistance: New Insights from a Case Study. In *2011 IEEE Symposium on Security and Privacy*, pages 538–553, May 2011. ISSN: 2375-1207.

[54] Corentin Lavaud, Robin Gerzaguet, Matthieu Gautier, Olivier Berder, Erwan Nogues, and Stephane Molton. Whispering Devices: A Survey on How Side-channels Lead to Compromised Information. *Journal of Hardware and Systems Security*, 5(2):143–168, June 2021.

[55] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22 140:55–55, 1932.

[56] Wouter Lueks, Iñigo Querejeta-Azurmendi, and Carmela Troncoso. VoteAgain: A scalable coercion-resistant voting system. In *29th USENIX Security Symposium*, pages 1553–1570, 2020.

[57] Tal Moran and Moni Naor. Receipt-Free Universally-Verifiable Voting with Everlasting Privacy. In *Advances in Cryptology - CRYPTO 2006*, pages 373–392, Berlin, Heidelberg, 2006.

[58] André Silva Neto, Matheus Leite, Roberto Araújo, Marcelle Pereira Mota, Nelson Cruz Sampaio Neto, and Jacques Traoré. Usability Considerations For Coercion-Resistant Election Systems. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, IHC 2018, pages 1–10, New York, NY, USA, October 2018.

[59] Stephan Neumann, Christian Feier, Melanie Volkamer, and Reto Koenig. Towards a practical JCJ / civitas implementation. In *INFORMATIK 2013 – Informatik angepasst an Mensch, Organisation und Umwelt*, pages 804–818, Bonn, 2013.

[60] Stephan Neumann and Melanie Volkamer. Civitas and the Real World: Problems and Solutions from a Practical Point of View. In *2012 Seventh International Conference on Availability, Reliability and Security*, pages 180–185, August 2012.

[61] Valtteri Niemi and Ari Renvall. How to prevent buying of votes in computer elections. In *Advances in Cryptology — ASIACRYPT'94*, pages 164–170, Berlin, Heidelberg, 1994.

[62] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols*, 1998.

[63] Mark Pack. Cambridge Analytica, big data, poor journalism and possible Electoral Commission probe, March 2017.

[64] Sunoo Park, Michael Specter, Neha Narula, and Ronald L Rivest. Going from bad to worse: from Internet voting to blockchain voting. *Journal of Cybersecurity*, 7(1), February 2021.

[65] Gerald V. Post. Using re-voting to reduce the threat of coercion in elections. *Electronic Government, an International Journal*, 7(2):168–182, January 2010.

[66] Jeff Sauro. *A Practical Guide to the System Usability Scale*. Measuring Usability LLC, 2011.

[67] Claus Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.

[68] Shivam Shankar Singh. A former BJP data analyst reveals how the party's WhatsApp groups work, February 2019.

[69] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security Analysis of the Estonian Internet Voting System. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 703–715, New York, NY, USA, November 2014.

[70] Oliver Spycher, Reto Koenig, Rolf Haenni, and Michael Schläpfer. A New Approach towards Coercion-Resistant Remote E-Voting in Linear Time. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, pages 182–189, Berlin, Heidelberg, February 2011.

[71] Stefan G. Weber, Roberto Araujo, and Johannes Buchmann. On Coercion-Resistant Electronic Elections with Linear Work. In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 908–916, April 2007.

[72] Edward Whelan. A Short History of Voting in the Ancient World | Classical Wisdom Weekly, November 2020.

A TRIP Formal Function Definitions

This section contains the formal definitions of the internal TRIP's functions.

$\mathsf{Setup}(\lambda, \mathbb{V}, (G,p,g), n_A, n_O, n_K, n_P, n_E)$

1: $\mathbb{L} \leftarrow \emptyset$

2: $\left\{A_i^{\mathsf{sk}}, A_i^{\mathsf{pk}}\right\}, \mathbb{A}_{\mathsf{pk}} \leftarrow \mathsf{DKG}(G,p,g,n_A)$

3: $\left\{O_i, K_i, P_i \leftarrow \mathsf{Sig.KGen}(1^\lambda)\right\}_{0\le i<n_O, 0\le i<n_K, 0\le i<n_P}$

4: $\mathbb{L}_R \leftarrow \left\{\mathbb{V}_i^{id}\right\}_{i\in\mathbb{V}}$

5: $\mathbf{E} \leftarrow \left\{e_i \leftarrow\!\$\, \mathbb{Z}_q; \mathbb{L}_E \Leftarrow (P_1^{\mathsf{pk}}, \mathsf{H}(e_i), \mathsf{Sig.Sign}(P_1^{\mathsf{sk}}, \mathsf{H}(e_i)))\right\}_{0\le i<n_E}$

6: $s_{rk} \leftarrow \{0,1\}^\lambda$

Figure 9: **Setup** Procedure for the ledger, the authority members, the officials, the kiosks and the envelope printers with envelope issuance. The secret s_{rk} is shared between the officials and the kiosk.

$\mathsf{CheckIn}(\mathbb{O}, \mathbb{K}, s_{rk}, V_{id})$

	$\mathbf{OSD}(O_1, s_{rk}, V_{id})$	$\mathbf{Kiosk}(K_1, s_{rk}, t_{in})$
1:	$\tau_r \leftarrow \mathsf{MAC.Sign}(s_{rk}, V_{id})$	$(V_{id}, \tau_r) \leftarrow t_{in}$
2:	$t_{in} \leftarrow (V_{id}, \tau_r)$	$\mathsf{MAC.Vf}(s_{rk}, \tau_r, V_{id}) \stackrel{?}{=} \top$

Figure 10: **Check-In.** The offical's device issues check-in ticket and Kiosk verifies authenticity of check-in ticket.

B Vote-Delegation

A limited number of voters may face challenges in concealing their real voting credential from a coercer, who might conduct a search post-registration. Additionally, some voters may not have access to a device that is not under the control of the coercer. In these cases of extreme coercion, where the voter can still visit a registrar, they have the option of delegating their voting rights to a well-known entity, such as a political party. In countries like Australia [1], voting is compulsory and non-compliance incurs a fine, providing voters facing extreme coercion an opportunity to exercise their right to vote. Vote-delegation provides an alternative to trust assumptions C2 through C5 (§3.3) by having these voters rely on the kiosk to delegate their voting rights for them. However, by relying on the registrar, the voter relinquishes their ability to verify the integrity of their real credential (and subsequently votes). In the following paragraph, we provide a concise overview of how vote-delegation could work, which requires collaboration between the registration and tallying schemes.

During the voter registration setup phase, the registration officials supply the kiosks with the public key of these well-known entities. When a voter chooses to delegate their vote to a particular well-known entity, the kiosk encrypts that entity's public key (rather than the public key of a new credential) and the voter and kiosk proceed with the *fake* credential creation process to produce a paper credential with a randomly

$\mathsf{CheckOut}(\mathbb{O}, \mathbb{L}, \mathbb{K}^{\mathsf{pk}}, t_{ot})$

	$\mathbf{OSD}(O_1, \mathbb{L}, \mathbb{K}^{\mathsf{pk}}, t_{ot})$	
1:	$(V_{id}, c_{\mathsf{pc}}, K_1^{'\mathsf{pk}}, \sigma_{k_{ot}}) \leftarrow t_{ot}$	% Check-Out Ticket
2:	$K_1^{'\mathsf{pk}} \stackrel{?}{\in} \mathbb{K}^{\mathsf{pk}}$;	% Authorized?
3:	$\mathsf{Sig.Vf}(K_1^{'\mathsf{pk}}, \sigma_{k_{ot}}, V_{id} \Vert c_{\mathsf{pc}}) \stackrel{?}{=} \top$	% Verify Signature
4:	$\sigma_o \leftarrow \mathsf{Sig.Sign}(O_1^{\mathsf{sk}}, V_{id} \Vert c_{\mathsf{pc}} \Vert \sigma_{k_{ot}})$	% Official Approval
5:	$\mathbb{L}_R[V_{id}] \leftarrow (c_{\mathsf{pc}}, K_1^{\mathsf{pk}}, \sigma_{k_{ot}}, O_1^{\mathsf{pk}}, \sigma_o)$	% Update Ledger on V_{id}
	$\mathbf{Ledger}(\mathbb{L}_R, \mathbb{O}^{\mathsf{pk}}, \mathbb{K}^{\mathsf{pk}}, (V_{id}, c_{\mathsf{pc}}, \sigma_{k_{ot}}, \sigma_o))$	
6:	$K_1^{'\mathsf{pk}} \in \mathbb{K}^{\mathsf{pk}}$; $\mathsf{Sig.Vf}(K_1^{'\mathsf{pk}}, \sigma_{k_{ot}}, V_{id} \Vert c_{\mathsf{pc}})$	% Verify Kiosk Signature
7:	$O_1^{'\mathsf{pk}} \in \mathbb{O}^{\mathsf{pk}}$; $\mathsf{Sig.Vf}(O_1^{'\mathsf{pk}}, \sigma_o, V_{id} \Vert c_{\mathsf{pc}})$	% Verify Official Signature
	$\mathbf{VSD}(\mathbb{L}, V)$	
8:	$\mathsf{Notify}(V_{id})$	% Notify Voter

Figure 11: **Check-Out Process**. The official confirms and approves the voter's registration session, publishing on the ledger their signature, the kiosk signature and the voter's public credential. The ledger verifies signatures and VSD monitoring the ledger notifies the voter.

$\mathsf{Activate}(\mathbb{L}, V, c)$

	$\mathbf{VSD}(\mathbb{L}, V, c)$	
1:	$((c_{\mathsf{pc}}, Y_c, \sigma_{k_c}), \_, (V'_{id}, c_{\mathsf{sk}}, r, K_1^{\mathsf{pk}}, \sigma_{k_r}),$ $(P_1^{\mathsf{pk}}, e, \sigma_p)) \leftarrow (q_c, \_, q_r, e) \leftarrow c$	% Unpack Credential
2:	$c_{\mathsf{pk}} \leftarrow \mathsf{Sig.PubKey}(c_{\mathsf{sk}})$	% Get Public Key
3:	$\mathsf{Sig.Vf}(K_1^{\mathsf{pk}}, \sigma_{k_c}, V'_{id} \Vert c_{\mathsf{pc}} \Vert Y_c) \stackrel{?}{=} \top$	% Receipt Integrity Check 1
4:	$\mathsf{Sig.Vf}(K_1^{\mathsf{pk}}, \sigma_{k_r}, \mathsf{H}(e \Vert r)) \stackrel{?}{=} \top$	% Receipt Integrity Check 2
5:	$\mathsf{Sig.Vf}(P_1^{\mathsf{pk}}, \sigma_p, H(e)) \stackrel{?}{=} \top$	% Envelope Integrity Check
6:	$(C_1, C_2, C_3) \leftarrow c_{\mathsf{pc}}$; $X \leftarrow C_3 / c_{\mathsf{pk}}$	% Derive ElGamal Secret
7:	$Y_1 \stackrel{?}{=} g_1^r C_1^e$; $Y_2 \stackrel{?}{=} g_2^r C_2^e$; $Y_3 \stackrel{?}{=} A^y X^e$	% Verify ZKP on C_1, C_2, C_3
8:	$(c'_{\mathsf{pc}}, K_1^{'\mathsf{pk}}, \sigma_{k_{ot}}, \mathbb{O}_1^{\mathsf{pk}}, \sigma_o) \leftarrow \mathbb{L}_R[V_{id}]$	% Voter Reg. Session
9:	$c'_{\mathsf{pc}} \stackrel{?}{=} c_{\mathsf{pc}} \wedge K_1^{'\mathsf{pk}} \stackrel{?}{=} K_1^{\mathsf{pk}} \wedge V'_{id} \stackrel{?}{=} V_{id}$	% Verify Public Cred & Actors
10:	$e \stackrel{?}{\notin} \mathbb{L}_E[\mathsf{H}(e)]$; $\mathbb{L}_E[\mathsf{H}(e)] \leftarrow e$	% Challenge Unused & Append

Figure 12: **Credential Activation.** Verifies the integrity of the credential: if any procedure with fails, then the process aborts. Upon success, VSD stores the credential's private key c_{sk}.

generated, useless key. This is necessary to ensure that the voter leaves with at least one credential for check-out. During the tallying phase, encrypted real credentials with the same plaintext will exist, and thus their corresponding votes after shuffling and mixing should be counted the same number of times as there are encrypted real credentials.

C Side Channels

We provide an overview of some potential side channel attacks that could give an adversary an advantage in determining the number of credentials held by a voter and propose some mitigation strategies. We also refer the reader to a survey on side channels in [54].

Timing Attack. In this attack, the adversary may be able to estimate the number of credentials a voter obtains by measuring the time taken to create each credential and the time spent inside the booth. To mitigate this risk, we recommend that the kiosk introduces artificial delays during the credentialing process, which increases uncertainty and helps thwart the attacker's efforts. These delays could include changing screens, introducing printing delays, or even displaying a "Please Wait..." screen while doing nothing behind the scenes. Additionally, at the expense of convenience, the kiosk can impose a minimum number of fake credentials that the voter must create.

Sound. The printer noise may also pose a threat, as it can indicate the number of prints made by the kiosk. To mitigate this, we recommend using a low-noise printer or placing the voter in a separate room or using a device that simulates printer noises within the booth or registrar environment, thus making it harder to determine the source of the noise.

D Cryptographic Primitives

This section introduces the cryptographic primitives used in TRIP.

Distributed Key Generation Scheme. TRIP uses a distributed key generation protocol DKG [37] that takes in a group parameter (G, q, g)—a cyclic group G of order q with generator g—and the number of parties n and creates a private key and public key pair for each party $(P_i^{\mathsf{sk}}, P_i^{\mathsf{pk}})$ and a collective public key P^{pk}:

$$\left\{P_i^{\mathsf{sk}}, P_i^{\mathsf{pk}}\right\}, P^{\mathsf{pk}} \leftarrow \mathsf{DKG}(G, p, g, n),$$

such that $P_i^{\mathsf{pk}} = g^{P_i^{\mathsf{sk}}}$ where $P_i^{\mathsf{sk}} \leftarrow\!\!\$\ \mathbb{Z}_q$ and $P^{\mathsf{pk}} = P_1^{\mathsf{pk}} + P_2^{\mathsf{pk}} + \cdots + P_n^{\mathsf{pk}}$.

M-ElGamal Encryption Scheme. TRIP employs the modified ElGamal (M-ElGamal) encryption scheme as defined in JCJ [47]. This scheme is parameterized by a cyclic group G of prime order q and two distinct, random generators g_1, g_2; and consists of the following algorithms: $\mathsf{EG.KGen}(G, q, g_1, g_2)$ which takes as input the group definition and outputs a public key pk along with two private keys $\mathsf{sk}_1, \mathsf{sk}_2$ such that $\mathsf{pk} = g_1^{\mathsf{sk}_1} g_2^{\mathsf{sk}_2}$ where $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow\!\!\$\ \mathbb{Z}_q$; a randomized encryption algorithm $\mathsf{EG.Enc}(\mathsf{pk}, m)$ which inputs a public key pk and a message $m \in G$ and outputs a ciphertext $C = (C_1, C_2, C_3) = (g_1^r, g_2^r, \mathsf{pk}^r m)$ for $r \leftarrow\!\!\$\ \mathbb{Z}_q$; and, a deterministic decryption algorithm $\mathsf{EG.Dec}(\mathsf{sk}_1, \mathsf{sk}_2, C)$ which takes as input two private keys $\mathsf{sk}_1, \mathsf{sk}_2$ and a ciphertext C and outputs a message $m = C_3 (C_1^{\mathsf{sk}_1} C_2^{\mathsf{sk}_2})^{-1}$.

Signature Scheme. TRIP uses a EUF-CMA signature scheme defined by the following three algorithms: a randomized key generation algorithm $\mathsf{Sig.KGen}(1^\lambda)$ which takes as input the security parameter and outputs a signing key pair $(\mathsf{sk}, \mathsf{pk})$; a signature algorithm $\mathsf{Sig.Sign}(\mathsf{sk}, m)$ which inputs a private key and a message $m \in \{0,1\}^*$ and outputs a signature σ; a signature verification algorithm $\mathsf{Sig.Vf}(\mathsf{pk}, m, \sigma)$ which outputs $\top$ if σ is a valid signature of m and $\perp$ otherwise; and, an algorithm $\mathsf{Sig.PubKey}(\mathsf{sk})$ that takes as input a private key sk and outputs the corresponding public key pk.

Hash. TRIP utilizes a cryptographic secure hash function H, for which the output is 2λ, for security parameter λ.

Message Authentication Code. TRIP uses a message authentication code scheme defined by the following two algorithms: a probabilistic signing algorithm $\mathsf{MAC.Sign}(k,m)$ which takes as input a (secret) key k and a message m and outputs an authorization tag τ; and a deterministic verification algorithm $\mathsf{MAC.Vf}(k,m,\tau)$ which takes as input the (secret) key k, the message m and the authorization tag τ and outputs either $\top$ for accept or $\bot$ for reject.

Zero-Knowledge Proof of Equality. TRIP employs an interactive zero-knowledge proof of equality of discrete logarithms [25] ZKPoE so that a prover P can convince a verifier V that P knows x, given messages $y \equiv g_1^x \pmod p$ and $z \equiv g_2^x \pmod p$ without revealing x. In *interactive* zero knowledge proofs, the verifier V must provide the challenge only *after* the prover P has computed and provided the commit to V.

E Kiosk Prototypes

Component	*Quantity*	*Total Cost (USD)*
Raspberry Pi Zero WH (Wireless/Headers)	1	14.00
16GB SD Card with Buster Lite	1	9.95
USB-A to Micro-B cable	1	2.95
Raspberry Pi Camera Module v2.1	1	29.95
Raspberry Pi Zero Camera Cable	1	5.95
Mini Thermal Receipt Printer	1	49.95
Thermal paper roll - 50' long, 2.25" wide	1	1.95
Female DC Power adapter - 2.1mm jack	1	2.95
USB to 2.1mm Male Barrel Jack Cable - 22AWG & 1 meter	1	2.75
7" black frame universal HDMI LCD with capacitive multi-touch	1	124.99
Mini HDMI to mini HDMI Cable	1	6.38
USB to 2.7mm/0.7mm DC cable	1	4.99
5V 1A (1000mA) USB port power supply	2	5.95
5V 2A (2000mA) Switching Power Supply w/ USB-A Connector	1	7.95
Total	15	270.66

Table 5: **Kiosk Prototype 1**: Components with Pricing Information

Component	*Quantity*	*Total Cost (USD)*
Factory-Made Kiosk purchased on Alibaba	1	694.00
Epson TM-T20III Thermal Receipt Printer	1	150.00
Total	2	844.00

Table 6: **Kiosk Prototype 2**: Components with Pricing Information

Kiosk Prototype 1 Setup The Pi camera is connected to the camera serial interface port on the Pi Zero using a ribbon cable, enabling both power and data transmission. The thermal receipt printer is connected to the Raspberry Pi's GPIO headers for data transmission and is connected directly to power via a DC jack → USB (5-9 volts). The touchscreen display is connected to the Raspberry Pi's Mini HDMI port and is connected directly to power via a DC jack → USB (5 volts). The entire prototype is capable of running off a 5V Power Bank with 3 USB ports for a max draw of ∼3A (e.g., RAVPower 22000mAh Portable Charger 3-Port Power Bank with a total max draw of 5.8A for $20.99).

- $c^i_{\mathsf{sk}}, c^i_{\mathsf{pc}}, P^i, \mathbb{L}_R \leftarrow \mathsf{RealCred}(\mathbb{L}_R, \mathbb{R}_{\mathsf{sk}}, \mathbb{V}^{id}_i, \lambda)$: takes as input the ledger $\mathbb{L}_R$, the registrars' private key $\mathbb{R}_{\mathsf{sk}}$, a voter's identifier $\mathbb{V}^{id}_i$ and a security parameter λ and outputs the voter's private and public credential c^i_{sk} and c^i_{pc}, correctness proofs P^i and an updated registration ledger $\mathbb{L}_R$. For simplicity, $\mathsf{RealCred}$ incorporates the $\mathsf{CheckIn}$ and $\mathsf{CheckOut}$ processes required in TRIP.
- $\mathbf{c}^i_f, \mathbf{P}^i \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}}, \mathbb{V}^{id}_i, c^i_{\mathsf{pc}}, n_f, \lambda)$: takes as input the registrar's private key R_{sk}, the voter's identifier $\mathbb{V}^{id}_i$, the voter's public credential c^i_{pc}, the number of fake credentials $n_f \in \mathbb{N}$, and a security parameter λ and outputs n_f fake credentials $\mathbf{c}^i_f$ and n_f proofs of correctness $\mathbf{P}^i$.
- $out, \mathbb{L}_R \leftarrow \mathsf{Activate}(\mathbb{L}_R, c, P)$: takes as input the registration ledger $\mathbb{L}_R$, a credential c and the credential's correctness proof P and outputs $out \in \{\top, \bot\}$ and an updated registration ledger $\mathbb{L}_R$.
- $\mathbb{L}_V \leftarrow \mathsf{Vote}(\mathbf{c}_{\mathsf{sk}}, T_{\mathsf{pk}}, n_M, D_{n_U, n_C}, \lambda)$: takes as input the set of credentials $\mathbf{c}_{\mathsf{sk}}$, the talliers' public key T_{pk}, the candidate list n_M, and a probability distribution D_{n_H, n_M} over the possible (voter, candidate) pairs[a]. It appends a TRIP-formatted ballot for each credential in $\mathbf{c}_{\mathsf{sk}}$ to the ledger $\mathbb{L}_V$.
- $(X, P) \leftarrow \mathsf{Tally}(T_{\mathsf{sk}}, \mathbb{L}, n_M, \lambda)$: takes as input the talliers' private key T_{sk}, the ledger $\mathbb{L}$, the candidates n_M, and a security parameter λ and outputs the tally X and a proof P showing that the talliers computed the tally correctly.
- $(X, P) \leftarrow \mathsf{Ideal\text{-}Tally}(T_{\mathsf{sk}}, \mathbb{L}, n_M, \lambda)$: takes as input the talliers' private key T_{sk}, the ledger $\mathbb{L}$, the candidates n_M, and a security parameter λ and outputs an ideal-tally X and a proof P. This algorithm, as defined in JCJ and only used in the ideal game, differs from Tally by not counting any ballots cast by the adversary if the bit $b = 0$.

[a] This distribution captures the uncertainty of honest voters' choices, which impairs the adversary's ability to detect coercion.

Figure 13: **TRIP API.**

F Security Proofs

In this section, we prove that TRIP maintains coercion-resistance and meets individual verifiability under our definitions in §5. Table 2 and Table 3 show our notation and summarize the variables.

F.1 TRIP API

We first redefine the TRIP API (Fig. 13), where algorithms output an appended ledger instead of submitting to it. Since the registrar is either all malicious (integrity, privacy) or all honest for coercion, we denote $\mathbb{R}$ to represent the kiosks $\mathbb{K}$, registration officials $\mathbb{O}$ and the envelope printers $\mathbb{P}$.

F.2 Coercion-Resistance

In this section we prove that TRIP is coercion-resistant.

Game $\mathsf{C\text{-}Resist}^{\mathcal{C},b}(\lambda,\mathbb{V},\mathbb{R},\mathbb{A},M,n_C)$

$\mathbf{V}_C,\{n_f^i\}_{i\in\mathbf{V}_C},\mathsf{st}_C \leftarrow \mathcal{C}(\mathbb{V},\text{“Choose controlled voter set”})$
$\mathbb{L},n_V,n_M,n'_C \leftarrow \emptyset,|\mathbb{V}|,|M|,|\mathbf{V}_C|$
$(j,n_f^j,\beta,\mathsf{st}_C) \leftarrow \mathcal{C}(\mathsf{st}_C,\text{“Pick target voter and their parameters”})$
if $n'_C \neq n_C$ or $j \notin \{1,2,...,n_V\}\setminus\mathbf{V}_C$ **then abort endif**
for $i=1$ **to** n_V **do**
  $c_{\mathsf{sk}}^i,c_{\mathsf{pc}}^i,P_c^i,\mathbb{L} \leftarrow \mathsf{RealCred}(\mathbb{L},\mathbb{R}_{\mathsf{sk}},V_i^{id},\lambda)$
  if $i=j$ and $b=0$ **then** *% Target voter evades coercion*
    $\tilde{\mathbf{c}}_{\mathsf{sk}}^j,\tilde{\mathbf{P}}_c^j \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}},\mathbb{V}_{id}^j,c_{\mathsf{pc}}^j,n_f^j+1,\lambda)$
  elseif $i\in(\mathbf{V}_C\cup j)$ **then** *% $\mathcal{C}$-Controlled Voters; target voter submits*
    $\tilde{\mathbf{c}}_{\mathsf{sk}}^i,\tilde{\mathbf{P}}_c^i \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}},\mathbb{V}_{id}^i,c_{\mathsf{pc}}^i,n_f^i,\lambda)$
  else *% Honest voters*
    $\tilde{\mathbf{c}}_{\mathsf{sk}}^i,\tilde{\mathbf{P}}_c^i \leftarrow \mathsf{FakeCreds}(\mathbb{R}_{\mathsf{sk}},\mathbb{V}_{id}^i,c_{\mathsf{pc}}^i,D^c_{1,[0,\text{inf}]},\lambda)$
  endif
  $\mathbf{c}^i \leftarrow (\tilde{\mathbf{c}}_{\mathsf{sk}}^i,\tilde{\mathbf{P}}_c^i);\ \mathbf{c}^i \Leftarrow (c_{\mathsf{sk}}^i,P_c^i)$
endfor
$\mathbf{c}_C \leftarrow \mathbf{c}^i$ *% Target voter releases all credentials*
if $b=0$ **then** *% Target voter evades coercion*
  $\_,\mathbb{L}_E \leftarrow \mathsf{Activate}(\mathbb{L}_E,c_{\mathsf{sk}}^j)$
  $\mathbb{L}_V \Leftarrow \mathsf{Vote}(c_{\mathsf{sk}}^j,\mathbb{A}_{\mathsf{pk}},M,\beta,\lambda)$
  $\mathbf{c}_C \leftarrow (\tilde{\mathbf{c}}^j,\tilde{\mathbf{P}}^j)$ *% Target voter releases only fake credentials*
endif
for $i\in\mathbb{V}\setminus(\mathbf{V}_C\cup\{j\})$ **do** *% Honest voters cast vote*
  $\_,\mathbb{L}_R \leftarrow \{\mathsf{Activate}(\mathbb{L}_R,c_{\mathsf{sk}}^i)\}_{1\le i\le|\mathbf{c}^i|}$
  $\mathbb{L}_V \Leftarrow \left\{\mathsf{Vote}(c_{\mathsf{sk}}^i,\mathbb{A}_{\mathsf{pk}},M,D^v_{n_V-n_C,n_M},\lambda)\right\}_{1\le i\le|\mathbf{c}^i|}$
endfor
$\mathbb{L} \Leftarrow \mathcal{C}(\mathsf{st}_C,\mathbb{L},\mathbf{c}_C,\{\mathbf{c}^i\}_{i\in\mathbf{V}_C},\lambda,\text{“Activate and cast votes”})$
$(\mathbf{X},P_t) \leftarrow \mathsf{Tally}(\mathbb{A}_{\mathsf{sk}},\mathbb{L},n_C,\lambda)$
$b' \leftarrow \mathcal{C}(\mathsf{st}_C,\mathbf{X},P_t,\mathbb{L},\text{“guess }b\text{”})$
return $b'=b$

Figure 14: **Game C-Resist.**

JCJ E-Voting Scheme. To demonstrate the coercion-resistant properties of TRIP, we need voting and tallying protocols. To do so, we will adopt those relevant protocols from the JCJ e-voting scheme, but we first provide an overview of the complete JCJ scheme for the reader's understanding. Their scheme comprises four distinct phases: Setup, Voter Registration, Voting, and Tallying. During the setup phase, the system actors, including the talliers, the registrar, and the ledger, are established. In voter registration, eligible voters undergo authentication, receive their real credentials, and have their encrypted real credentials published on the ledger. During the voting phase, voters use their device to cast their vote which creates a 2-tuple ballot, including a fresh encryption of their real credential and the actual vote. This ballot is then published on the ledger. In the presence of coercion, voters use any freshly generated credential. During the tallying phase, the talliers verifiable shuffle the encrypted real credentials published by the registrar, and the encrypted 2-tuple ballots cast by each voter. Talliers then determine the actual ballots by conducting privacy equivalence tests [44] between each encrypted real credential published by the registrar and the first element of each 2-tuple ballot on the ledger. Finally, the talliers work together to decrypt the votes and uncover the actual results.

Definition 3 (Coercion-resistance)**.** *A scheme is coercion-resistant if for all PPT adversaries $\mathcal{C}$, all security parameters $\lambda\in\mathbb{N}$, and all parameters $\mathbb{V},\mathbb{R},\mathbb{A},M,n_C$, the following holds:*

$$\mathsf{Adv}^{coer}_{\mathcal{C},\mathsf{C\text{-}Resist}}(\lambda,\cdot) = \left|\Pr_{\mathcal{C},b}[\mathsf{C\text{-}Resist}^{\mathcal{C},b}(\cdot)=1] - \Pr_{\mathcal{C}',b}[\mathsf{C\text{-}Resist\text{-}Ideal}^{\mathcal{C}',b}(\cdot)=1]\right| \le \mathsf{negl}(\lambda),$$

where the probability is computed over all the random coins used by the algorithms in the scheme.

We introduce the following three major changes from JCJ games to model TRIP's behavior.

Change #1: Voter Registration Algorithms. To model information beyond the credential and the entries on the ledger (e.g., proofs of correctness), we replace JCJ algorithms register and fakekey with RealCred and FakeCreds, respectively. Further, we use Activate to model additional registration information placed on the ledger (e.g., envelope data). Activate always returns *out* as $\top$ in our C-Resist games since we trust the registrar $\mathbb{R}$ for coercion-resistance.

Change #2: Fake Credentials Issuance In the JCJ game, voters only give their real credential but in TRIP, an adversary can demand voters before voter registration to provide $\mathcal{C}$-designated credentials. We then give the coercer the ability to instruct the coerced voter, before voter registration, to create $n_f\in\mathbb{N}$ fake credentials. Similar to how JCJ models the uncertainty of how honest voters vote as D^v, we model the uncertainty of fake credentials *created and activated* by honest voters as another probability distribution D^c[6]. We refer the reader to Section 5 for the reasoning behind these distributions.

Alteration #3: Ledger Entries JCJ performs registration using a trusted algorithm which outputs a voting roster containing each voter's public credential c_{pc}. On the other hand, TRIP has a more elaborate protocol for individual verifiability, requiring additional information on the ledger $\mathbb{L}$. In particular, the envelope challenge is revealed on the envelope ledger during Activate, permitting the adversary to have knowledge on the number of fake credentials. In addition, the registration ledger also contains digital signatures from the kiosk and officials responsible with issuing the credential to the voter. We show that this additional information makes for a negligible difference on the coercer's winning probability, beyond the additional probability distribution of fake credentials as represented in the ideal game. As in JCJ, the winning

[6] In practice, to increase this uncertainty, envelope printers can post challenges on the ledger *without* printing a corresponding envelope and gradually release these values, similar to the JCJ option of voting authorities or third parties intentionally injecting fake votes to add noise.

probability of the ideal game $\gg 0$ since coercion-resistance is bounded by adversary's uncertainty over the behavior of the honest voters [47].

We present our formal definition for coercion-resistance under Definition 3, and use the ideal (C-Resist-Ideal) and real (C-Resist) games in figures 6 and 14, respectively. The coercer wins if they can correctly guess the bit b, representing whether or not the targeted voter gives in to coercion. While C-Resist represents the coercive adversary $\mathcal{C}$, to prove security we must compare $\mathcal{C}$ with another adversary $\mathcal{C}'$ who plays C-Resist-Ideal, which embodies the security we want to achieve against coercion. We show that the difference between the real and ideal games is negligible.

Proof. We use three hybrid games to transition from the real to the ideal game, with each game involving a protocol change:

1. **Eliminate Voting Ledger View:** Eliminate $\mathcal{C}$'s access to the TRIP voting ledger $\mathbb{L}_V$, as $\mathcal{C}$ is incapable of differentiating between a ledger filled with honest voter ballots and a randomly generated set of ballots, assuming the decisional-Diffie-Hellman assumption holds.
2. **Number of Fake Credentials:** $\mathcal{C}$'s ability to demand voters to create a specific number of fake credentials is equal to that of an adversary $\mathcal{C}'$ who *cannot* demand voters to create a specific number of fake credentials, given the distribution of the honest voters' fake credentials.
3. **Eliminate Registration Ledger View**: Eliminate $\mathcal{C}$'s access to TRIP roster $\mathbb{L}_R$ by introducing a new ledger $\mathbb{L}_R^{JCJ}$ where we can decouple V_i^{id} and $(c_{\mathsf{pc}}^i, K, \sigma_k, R, \sigma_r)$ via semantic security:
   - $\mathbb{L}_R$: $(V_i^{id}, c_{\mathsf{pc}}^i, K, \sigma_k, R, \sigma_r)$
   - $\mathbb{L}_R^{JCJ}$: (c_{pc}^i)

Hybrid 1. We replace Tally (Fig. 14, l. 27) with Ideal-Tally (Fig. 6, l. 20) by proving that $\mathcal{C}$ has no longer access to $\mathbb{L}_V$. We use a simulation-based approach to show that if an adversary with access to $\mathbb{L}_V$ has a non-negligible advantage over an adversary who does not have access to $\mathbb{L}_V$, then the decisional-Diffie-Hellman assumption is broken. The simulator gets as input a tuple of group elements that are either a Diffie-Hellman tuple or uniformly random and must output a guess. It interacts with $\mathcal{C}$, simulates the honest parties of the protocol, and makes a guess based on its output.

The simulator then proceeds as follows:

1. **Setup:** Choose two group elements x_1, x_2 uniformly at random and broadcast the public key $(g_1, g_2, g_1^{x_1} g_2^{x_2})$. Broadcast a list of candidate identifiers $\{M_i\}_{i=1}^{n_M}$ where each identifier is a random group element.
2. **Coin Flip**: Flip the coin b.
3. **Adversarial Corruption**: The adversary chooses the set $\mathbf{V}_C$ of controlled voters and a target voter j. For each controlled voter and the target voter, the adversary chooses the number of fake credentials n_f to create. For the target voter, $\mathcal{C}$ sets the target vote β. If the number of controlled voters is not equal to n_C, or j is not an appropriate index then the simulator aborts.
4. **Registration**: For each voter i, the simulator runs the TRIP registration process while acting as the registrar. The simulator first issues each voter their real credential c_{sk}^i along with their public credential c_{pc}^i. The simulator then continues with generating fake credentials for each group of voters. For each controlled voter i in $\mathbf{V}_C$, the simulator issues n_f^i fake credentials as specified by the adversary. For the target voter, if $b = 0$, the simulator generates $n_f^i + 1$ fake credentials for the target voter; otherwise, it generates n_f^i fake credentials. For each uncontrolled honest voter, the simulator creates fake credentials, for which the amount is sampled from a probability distribution that models the adversary's knowledge of the number of fake credentials that these voters intend to create. Finally, the simulator carries out check-out for all voters.
5. **Credential Release:** The simulator gives $\mathcal{C}$ the real and fake credentials of voters in $\mathbf{V}_C$. If $b = 0$, the simulator gives $\mathcal{C}$ the target voter's $n_f^j + 1$ fake credentials; otherwise, the simulator gives $\mathcal{C}$ the voter's real credential and n_f^j fake credentials.
6. **Honest Ballot Casting**: For each honest voter i, the simulator samples a random vote $\beta_i \leftarrow\!\!\$\, D_{n_V - n_C, n_M}$ and posts a ballot for this vote on the voting ledger $\mathbb{L}_V$. The simulator forms the ballot using the input (g_1, g_2, h_1, h_2) as follows. The simulator generates two random group elements r_i, k_i. Next, the simulator computes the encryption of the public credential c_{pc}^i as $E_1 = (h_1^{r_i}, h_2^{r_i}, h_1^{x_1 r_i} h_2^{x_2 r_i} c_{\mathsf{pc}}^i)$, and the encryption of the vote as $E_2 = (h_1^{k_i}, h_2^{k_i}, h_1^{x_1 k_i} h_2^{x_2 k_i} \beta_i)$ This way, if the input is Diffie-Hellman, the result is a valid encryption, whereas if the input is random, the ballot is random and contains no information about the vote or the credential used to cast it. The simulator then simulates the required NIZK proofs using standard techniques.
7. **Adversarial Ballot Posting:** The adversary now posts a set of ballots onto $\mathbb{L}$.
8. **Decryption of Ballots:** The simulator can now check the NIZK proofs and discard the ballots with incorrect proofs. Then, since the simulator plays the role of the honest talliers, it can decrypt the ballots to prepare for the tallying process.
9. **Tallying Simulation:** This is carried out as in JCJ's simulator [46]. Namely, the simulator eliminates duplicates, mixes, removes fake votes and finally decrypts the remaining real votes:
   - *Duplicate elimination:* The simulator removes duplicates.
   - *Mixing:* To simulate the MixNets from the real tally protocol, the simulator outputs an equal-length list of random ciphertexts.
   - *Credential Validity:* The simulator now checks, for each ballot, whether it was cast with a valid creden-

tial. This check is possible since the simulator can decrypt all the entries in $\mathbb{L}_R$ and $\mathbb{L}_V$.
- *Output final count:* The simulator can then use the decrypted values to compute the final tally and output it.

10. **Output Guess**: The simulator uses $\mathcal{C}$'s output to make its guess about whether the input was a DDH triplet or a random triplet.

If we can show that $\mathcal{C}$ has a non-negligible advantage in the real game over hybrid 1, then this implies that the simulator can break the DDH assumption. The key to this argument lies in how the simulator constructs the ballots in Step 6. When the input is DDH, where $h_1 = g_1^a$ and $h_2 = g_2^a$, we have $E_1 = (g_1^{ar_i}, g_2^{ar_i}, g_1^{x_1 ar_i} g_2^{x_2 ar_i} c_{\mathsf{pc}}^i)$ which is a valid M-ElGamal encryption. The same holds for E_2, but with k_i instead of r_i. Thus, when the input is DDH, the view of $\mathcal{C}$ corresponds to the actual game and the adversary receives the contents of $\mathbb{L}_V$. However, if the input tuple is a random one, then E_1 and E_2 are not valid M-ElGamal encryptions, but just random tuples. In this scenario, the vote and credential are perfectly concealed, making it equivalent to denying the adversary access to the ledger. Hence, if the adversary $\mathcal{C}$ holds a significant advantage in the real game, then the probability that the simulator correctly guesses will also be significant, leading to a contradiction.

Hybrid 2. In this Hybrid, we demonstrate that the adversary's ability to demand a specific number of fake credentials from voters is lost, as compared to Hybrid 1. text In C-Resist (Fig. 14, l. 16 and 20), the voter gives their credentials to the adversary while in C-Resist-Ideal, the adversary always gets the voter's single real credential. (Fig. 6, l. 17).

First, we show that real and fake credentials are indistinguishable: both real and fake credentials contain a ZKP transcript, although the ZKP transcript for the fake credentials are simulated. The zero-knowledge property of the proof system implies indistinguishability.

Next, from hybrid 1, $\mathcal{C}$ does not get access to $\mathbb{L}_V$, thus $\mathcal{C}$ cannot use any real or fake credentials to cast valid ballots, or see ballots cast with these. As a result, since real and fake credentials are indistinguishable, always giving the adversary the real credential gives them no advantage since the credentials cannot influence the tally outcome with it. Now the value of b only determines whether or not the target voter casts a ballot.

For the same reason, $\mathcal{C}$ can only use the n_f fake credentials for determining whether the target voter cast a ballot or not and to influence the number of envelope challenges on $\mathbb{L}_E$.

Yet, the signing key pair $(\tilde{c}_{\mathsf{sk}}, \tilde{c}_{\mathsf{pk}})$ corresponding to each fake credential is sampled independently from the real pair $(c_{\mathsf{sk}}, c_{\mathsf{pk}})$, and these cannot be used by the adversary or the target voter to cast a valid vote. Regarding the number of challenges on $\mathbb{L}_E$, the only difference in the case where the target voter resists coercion is that they create an additional fake credential. As a result, the number of challenges will only differ by one. To detect this, the adversary needs to distinguish between the following distributions:

- Number of envelope challenges when target complies: $n_C + n_T + n_H + n_f + D_{n_H}^{fake}$
- Number of envelope challenges when target resists: $n_C + n_T + n_H + n_f + D_{n_H}^{fake} + 1$

Since everything but $D_{n_H}^{fake}$ is known to the adversary, this is equivalent to just distinguishing between $D_{n_H}^{fake}$ and $D_{n_H}^{fake} + 1$, so they get no advantage from requesting a certain number of fake credentials. Therefore, the adversary does not gain an advantage from specifying the number of fake credentials, since these fake credentials will not help them identify whether the voter gave them a real credential.

Hybrid 3. In this hybrid, we replace the TRIP roster $\mathbb{L}_R$ initialized on the first line of Figure 14 with the JCJ Roster $\mathbb{L}'_R$ from the Ideal Game in Figure 6.

To prove that the advantage of the adversary is negligible between these hybrids, we show that given a JCJ roster, a simulator can output a TRIP roster that is indistinguishable from a real one. This is possible due to the semantic security of m-ElGamal.

We describe the simulator: **Input:** JCJ roster $\mathbb{L}'_R$, List of voter IDs V_{id}

1. Create a kiosk key pair (k, K) and a registrar key pair (r, R).
2. Initialize $\mathbb{L}'_R$ to contain each V_{id} in a different entry, along with a random timestamp d.
3. Apply a random permutation to the JCJ roster.
4. Append one entry V_e of the JCJ roster to each entry of $\mathbb{L}_R$.
5. For each entry, add the necessary signatures. First, use k to simulate the kiosk signature $\sigma_2 = \mathsf{Sig.Sign}_k(V_{id}^i || d || V_e)$ and append K, σ_{k_2} to the entry. Then, use r to simulate the registrar's check-out signature $\sigma_r = \mathsf{Sig.Sign}_r(V_{id} || d || V_e || \sigma_{k_2})$ and append R, σ_r to the entry.
6. Output $(K, R, \mathbb{L}'_R)$. Recall that we consider a single logical entity for all the registration actors.

Now, as stated earlier, L'_R is indistinguishable from a real TRIP roster for the same list of voters due to the semantic security of m-ElGamal. If we start with a real TRIP roster, we could create intermediate rosters by swapping two encryptions at a time and updating the digital signatures until we get a random permutation. Each of these swaps will yield indistinguishable rosters due to the semantic security of Elgamal. At the end, the distribution will be the same as that of our simulator.

Then, recall that the view of $\mathcal{C}$ is just the list of valid verification keys for the kiosks and registrars, along with the TRIP roster. Since we have shown that the additional elements contained in the TRIP roster but not in the JCJ roster

Game $\mathsf{IV}^I(\mathbb{V}, j, pars)$

$(\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}}, \mathbb{R}_{\mathsf{sk}}, \mathbf{c}_{\ominus j}, \mathbb{L})$
$\leftarrow \mathsf{Setup}_I(pars)$
$\mathbf{c}_j \leftarrow \mathsf{Register}_I($
$\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}\ominus 1}, \mathbb{R}_{\mathsf{sk}}, \mathbb{V}_j)$
$s \leftarrow I(\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}\ominus 1})$
$I^{V(c_j,s),\mathbb{L},E(\mathbb{A}_1^{\mathsf{sk}})}(\mathbb{A}_{\mathsf{pk}}, \mathbf{c})$
if $\mathsf{bad}_{\mathsf{iv}}$ **return** 1

Figure 15: **Individual Verifiability (Ind-Ver).** Definition of individual verifiability adapted from [18] to account for registration and adversary's access to the voter's real credential.

can be simulated, this means that the advantage of $\mathcal{A}$ must be negligible.

With this, we have reached the Ideal JCJ game and have thus shown that the advantage of $\mathcal{A}$ in the C-Resist game is negligible over the advantage in C-Resist-Ideal.

□

F.3 Individual Verifiability

Our definition of verifiability builds on the work on the Swiss Post E-Voting System [7] and CH-Vote [18]. As is traditional with individual verifiability, their focus is on "voter receiving conclusive evidence that the vote has been cast and recorded as intended" [18]. However, since TRIP is a *registration* scheme, we define individual verifiability as *the voter receiving conclusive evidence that their voting credential is real*, a prerequisite to voting.

We present the original individual verifiability game in Fig. 7a, but for reader's convenience we report the same game in Fig. 15. In the IV game, the adversary has complete control over the Setup function and partial control over the Register function, which we present in Fig. 16. In particular, the adversary controls the creation of real and fake credentials, but *does not* control the number n_c of credentials that the voter decides to create, nor Activate, i.e., the procedure that the voter uses to activate all the credentials.

During the execution of the Register function the voter performs several checks to verify that procedures, some of which are under adversarial control, are performed correctly. To formally prove verifiability we make these checks explicit: the voter interacts with the possibly malicious kiosk and produces explicit confirmation checks (denoted with check in the Register function). Moreover, in §5, we define a inefficient *extractor* that, given the ledger, extract either the plain real credential or the unencrypted ballot. This follows the approach of Bernhard et al. [18]. The adversary wins if all the checks passes and the real credential is incorrect, or if the ledger does not contains the correct adversarially-selected vote s.

$\mathsf{Register}_I(\mathbb{A}_{\mathsf{pk}}, \mathbb{A}_{\mathsf{sk}\ominus 1}, \mathbb{R}_{\mathsf{sk}}, \mathbb{V}_j)$

$c, \mathsf{check}_1, \mathsf{st}_I \leftarrow I(\mathsf{st}_I)$
$\mathbf{e} \leftarrow \{e\}$; $\mathbf{c} \leftarrow \{c\}$ % Used challenges & vector of credentials
$n_c \leftarrow V()$ % Voter picks number of envelopes
for 1 to $n_c - 1$ **do**
$\tilde{c}, \mathsf{st}_I \leftarrow I(\mathsf{st}_I, t_{ot})$
$\mathbf{c} \Leftarrow \tilde{c}$; $\mathbf{e} \Leftarrow \tilde{c}[e]$
endfor
$c_a \leftarrow V()$
$\mathbb{L}, \mathsf{check}_2 \leftarrow I(\mathsf{st}_I, \mathbb{L}, \mathbb{O}, \mathbb{K}^{\mathsf{pk}}, c_a[t_{ot}])$
for i to n_c **do**
$\mathbb{L}, \mathsf{check}_{3,i} \leftarrow \mathsf{Activate}(\mathbb{L}, V, \mathbf{c}_i)$
return c

Figure 16: Register function.

Theorem 4 (Individual verifiability of TRIP)**.** *Assume that all the cryptographic primitives that* TRIP *uses are secure. Under the discrete-logarithm assumption in the random oracle model, for a PPT adversary I, the following holds:*

$$\mathsf{Adv}_I^{\mathsf{IV}} = \Pr[\mathsf{IV}^I = 1] \leq \left(\frac{n_u}{n_e}\right)^{n_c} + \mathsf{negl}(\lambda),$$

where the probability is computed over all the random coins that the algorithms use and n_e is the number of envelopes $|\mathbf{E}|$ that the adversary produces, n_u is the number of unique challenges that the adversary produces in the n_e envelopes and n_c is the number of credentials (one real, the rest fake) that the voter decides to create and activate.

Proof. We give a sketch of the proof. The adversary's winning probability depends on the number of envelopes n_c that the voter decides. Note that this choice is not under adversarial control. Let n_e be the total number of envelopes $|\mathbf{E}|$ and n_u be the total number of *unique* challenges that the adversary prints on the envelopes. Let $\mathbf{e}[e]$ be the number of times that the same challenge e appears in $\mathbf{e}$. "the adversary correctly guesses the challenge on the envelope e that the voter uses for the real credential" and let F be the event "the voter does not pick the same challenge twice". Then the advantage of the integrity adversary I is

$$\begin{aligned}\mathsf{Adv}_I^{\mathsf{IV}} &= \Pr[\mathsf{IV}^I = 1] = \Pr[\mathsf{E}] \cdot \Pr[\mathsf{F}] + \mathsf{negl}(\lambda) \\ &= \frac{\mathbf{e}[e]}{n_e} \cdot \Pr[\mathsf{F}] + \mathsf{negl}(\lambda) \leq \frac{\mathbf{e}[e]}{n_e} \cdot \frac{\binom{n_u}{n_c}}{\binom{n_e}{n_c}} + \mathsf{negl}(\lambda),\end{aligned}$$

where an information-theoretic argument indicates the last inequality as follows. The best strategy for the adversary to maximize $\Pr[\mathsf{F}]$ is to distribute the n_u unique challenges equally among the n_v envelopes to maximize the entropy, and thus minimize the probability does pick twice the same

challenge, which implies that the adversary is detected and loses the game due to a false check returned by Activate. By manipulating the fraction of binomial coefficients we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IV}}_{I} = \Pr[\mathsf{IV}^{I} = 1] &\leq \frac{\mathbf{e}[e]}{n_e} \cdot \frac{\binom{n_u}{n_c}}{\binom{n_e}{n_c}} + \mathsf{negl}(\lambda) \\ &\leq \frac{\mathbf{e}[e]}{n_e} \cdot \left(\frac{n_u \cdot e}{n_c}\right)^{n_c} \cdot \left(\frac{n_e \cdot e}{n_c}\right)^{-n_c} + \mathsf{negl}(\lambda) \\ &= \frac{\mathbf{e}[e]}{n_e} \cdot \left(\frac{n_u}{n_e}\right)^{n_c} + \mathsf{negl}(\lambda)\end{aligned}$$

where we use the inequality $\binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^{k}$ to remove the binomial coefficients from the advantage's expression. □

Individual verifiability does not achieve a negligible advantage against the integrity adversary since we cannot use n_c as a security parameter (i.e., expect each voter to create, for example, 128 or 256 credentials). Many individual verifiability schemes, such as Benaloh challenge [14], also present with this issue, where they rely on the voter repeating a task over and over again to achieve negligibility.

Instead, we achieve negligible advantage when considering the combined probability across voter registration sessions. In this work we introduce iterative individual verifiability in Fig. 7b. The definition of iterative individual verifiability is parametrized by a security parameter λ_v, which indicates the *minimum number of votes* that the adversary must change to sway the final outcome of an election.

Theorem 5 (Iterative individual verifiability of TRIP)**.** *Assume that all the cryptographic primitives that* TRIP *uses are secure. Under the discrete-logarithm assumption in the random oracle model, for a PPT adversary I, the following holds:*

$$\mathsf{Adv}^{\mathsf{I\text{-}IV}}_{I} = \Pr[\mathsf{I\text{-}IV}^{I} = 1] \leq \mathsf{negl}(\lambda_v),$$

where the probability is computed over all the random coins that the algorithms use.

Proof. We give a sketch of the proof. In order to win the I-IV game, the adversary I must win at least λ_v times the IV game of Fig. 7a. The theorem follows. □