Receipt-Free Electronic Voting from zk-SNARK

Maryam Sheikhi, Rosario Giustolisi and Carsten Schuermann
IT University of Copenhagen, Copenhagen, Denmark

Keywords: Electronic Voting, Receipt-Freeness, Everlasting Privacy, Participation Privacy.

Abstract: In 2016, Locher and Haenni (Locher and Haenni, 2016) proposed an e-voting scheme that offers verifiability, everlasting vote privacy, and computational receipt-freeness, as well as an informal discussion of how the scheme achieves such properties. We advance this line of work by proposing a new cryptographic scheme that provably satisfies those properties as well as everlasting participation privacy and efficient tallying. Receipt-freeness relies on deniable vote updating and verifiable *null* ballot posting, generated from public knowledge stored on the bulletin board. The everlasting vote and participation privacy properties directly result from the hash-based commitment scheme and efficient zero-knowledge proofs (SNARKs). Finally, we provide mathematical proofs for all the properties, including a new game-based definition of participation privacy.

1 INTRODUCTION

In 2016, Locher and Haenni (Locher and Haenni, 2016) proposed an e-voting scheme that offers computational receipt-freeness, verifiability, and everlasting vote privacy with minimal trust assumptions. Verifiability guarantees a verification of the accuracy of the election outcome, even if not all election participants are honest. Vote privacy refers to what can be learned about the link between a vote and the identity of the voter. Most voting schemes guarantee vote privacy under standard cryptographic assumptions, which means that vote privacy depends directly on the choice of key sizes which are expected to be broken once computing power has caught up. In contrast, Locher and Haenni's scheme guarantees *everlasting vote privacy*, which is independent of key sizes and other computational assumptions. It also guarantees that voters are unable to convince a third party about the way they voted, even if they are willing to do so, a property that is called *receipt-freeness* and that prevents vote-buying and mitigates voter coercion.

In this work, we propose an e-voting scheme that introduces a credential protocol for voter registration and vote submission through a combination of hash-based commitment scheme and efficient zero-knowledge proofs (zk-SNARKs). Our scheme satisfies verifiability, everlasting vote privacy, and computational receipt-freeness. Furthermore, we prove that our scheme also guarantees *participation privacy*, which means that an adversary cannot learn from the information published on the bulletin board, if a voter has voted or not, i.e. participated in the election. Most e-voting schemes resort to digitally signed ballots, which protect the integrity of the vote from modification by malicious parties but usually reveal if a voter has participated in the election or not.

- We propose an e-voting scheme with everlasting vote privacy, everlasting participation privacy, receipt-freeness, and verifiability based on minimal trust assumptions.
- We provide a new definition of everlasting participation privacy and prove that our e-voting scheme satisfies this definition.
- We prove that our scheme also meets vote privacy and receipt-freeness.

2 RELATED WORK

Benaloh and Tuinstra (Benaloh and Tuinstra, 1994) proposed the first scheme achieving receipt-free voting. Their idea was later extended by Sako and Kilian (Sako and Kilian, 1995), who apply mix-networks that shuffle the order of encryption of yes/no votes and then send the order to the voter through an untappable channel. Different flavours of receipt-freeness properties based on untappable channels with various cryptographic schemes and efficiency guarantees were introduced in (Okamoto, 1997; Hirt and Sako, 2000; Ryan et al., 2016) and in coercion-resistance

Sheikhi, M., Giustolisi, R. and Schuermann, C.
Receipt-Free Electronic Voting from zk-SNARK.
DOI: 10.5220/0012140000003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 254-266
ISBN: 978-989-758-666-8; ISSN: 2184-7711

Table 1: Comparison of different receipt-free voting schemes. We use the notation from (Haines et al., 2023). A security property not relying on any trust assumption is denoted by +. $\mathcal{DT}_T$ denotes the distributed trust assumption on tally servers. $\mathcal{T}_P$ denotes trust on a third party. N/C denotes a property that is not claimed, while N/A denotes a not-applicable property.

Property	BelenoisRF	KTV-Helios	Locher & Haenni	**This scheme**
Computational vote privacy	$\mathcal{DT}_T$	$\mathcal{DT}_T$	+	+
Everlasting vote privacy	N/A	N/A	+	+
Computational participation privacy	N/A	$\mathcal{T}_P$	N/C	+
Everlasting participation privacy	N/A	N/A	N/C	+
Verifiability	$\mathcal{T}_P$	$\mathcal{T}_P$	+	+
Computational receipt-freeness	$\mathcal{T}_P$	$\mathcal{T}_P$	$\mathcal{T}_T$	$\mathcal{T}_P$

schemes (Juels et al., 2005; Bohli et al., 2007). In BeleniosRF (Chaidos et al., 2016), a trusted randomization server provides receipt-freeness. Kulyk et al. (Kulyk et al., 2015) extended Helios (Adida, 2008) allowing a voter to update their vote by revoting and nullifying the previous ballot. In this scheme, receipt-freeness is achieved thanks to dummy ballots that are cast by a trusted third party. The scheme also provides participation privacy, which is based on the indistinguishability of dummy ballots from ballots cast by voters. All the schemes above achieve computational vote privacy but not everlasting privacy.

(Moran and Naor, 2006) introduced a verifiable receipt-freeness scheme with everlasting privacy using an untappable channel that models a private polling booth. (Demirel et al., 2012; Demirel et al., 2013) proposed enhancements to Helios (Adida, 2008) and Prêt à voter (Ryan et al., 2009) with everlasting privacy, which is achieved by perfectly hiding and computationally binding commitment schemes and untappable channels. However, those schemes assume that the voting server is trusted (Demirel et al., 2012; Buchmann et al., 2013; Demirel et al., 2013). (Locher and Haenni, 2015) proposed a scheme that meets everlasting privacy without trusted authorities and computational hardness assumptions. They later enhanced their scheme to provide receipt-freeness (Locher and Haenni, 2016), which can be obtained by a trust assumption on the tally phase. However, they only provide an informal discussion of how the scheme achieves its properties.

Table 1 summarizes the comparison between our scheme and previous receipt-free e-voting schemes. BeleniosRF and KTV-Helios achieve receipt-freeness and verifiability, assuming a trusted third party. They are not designed to achieve everlasting privacy. KTV-Helios is one of the few e-voting schemes that provides computational participation privacy against a computationally bounded adversary. Locher and Haenni provide a scheme that can achieve everlasting privacy and verifiability without the need for trusted parties. Our scheme has the same trust assumption as their scheme and provably achieves everlasting participation privacy without the need for trusted parties.

3 CRYPTOGRAPHIC PRELIMINARIES

In this section, we present the definitions and the security properties of the cryptographic primitives that form the building blocks of our voting scheme. We also provide a foretaste of how we intend to use the cryptographic primitives in our scheme.

ElGamal Encryption Scheme. The ElGamal encryption scheme is a triple of PPT algorithms (KeyGen, Enc, Dec) defined as follows.

- KeyGen(λ) $\rightarrow (\mathcal{P}, pk, sk)$: on input of security parameter λ and $sk \in_U \mathbb{Z}_q$, it derives the public parameter $\mathcal{P} = (\mathcal{G}, q, g)$ and computes $pk = g^{sk}$. $\mathcal{P}$ contains a cyclic group $\mathcal{G}$ of prime order q generated by g. When derivable from the context, we omit $\mathcal{P}$ from the public and secret keys.
- Enc$(m, ((\mathcal{G}, q, g), pk)) \rightarrow (c_1, c_2)$: on input of a message m and public key pk, it chooses $r \in_U \mathbb{Z}_q$ and outputs (c_1, c_2), where $c_1 = g^r$, $c_2 = m \cdot pk^r$. The message m is of form $m = g^a$ where $a \in \mathbb{Z}_q$.
- Dec$((c_1, c_2), ((\mathcal{G}, q, g), sk)) \rightarrow m$: on input a ciphertext (c_1, c_2), it outputs $m = c_2 \cdot c_1^{-sk}$.

The ElGamal encryption scheme satisfies semantic security under the Diffie-Hellman assumption. For the encryption scheme, we use the NM-CPA securitr definition. The ElGamal ciphertext with a Schnorr proof is NM-CPA secure in the random oracle model.

For our election scheme, pk_T denotes the public encryption key and sk_T denotes the decryption key, such that $pk_T = g^{sk_T}$. The decryption key can be distributed using (k, n) Shamir secret sharing so that $k > 1$ out of n shares are required to decrypt the ciphertext (Brandt, 2005). A re-encryption of a given ciphertext (c_1, c_2) with a new $r' \in_U \mathbb{Z}_q$ can be simply computed by multiplying the ciphertext to the encryption of zero i.e g^0 (or $m = 1$), namely, $reEnc((c_1,c_2), pk) = (c_1,c_2) \cdot Enc(g^0, pk)$. The re-encryption result of a given ciphertext can be described as $Enc(g^m, pk)$ with randomness $r+r'$. As we shall see later on the description of our voting scheme, we denote the encryption of the message g^m with public key pk and randomness r by $enc_{pk}(m;r)$.

We use *Re-encryption mix servers* in the tally phase of our scheme. Given a set of El-Gamal ciphertexts $\{(c_{11},c_{12}),(c_{21},c_{22}),\ldots,(c_{n1},c_{n2})\}$ and a uniformly random secret permutation ρ, the mix servers output $\{(c'_{\rho(1)1},c'_{\rho(1)2}),(c'_{\rho(2)1},c'_{\rho(2)2}),\ldots,(c'_{\rho(n)1},c'_{\rho(n)2})\}$. This can be done by re-encrypting the ciphertexts, and it is infeasible for a computationally bounded adversary to match inputs and outputs. Indeed, $(c'_0,c'_1)_{\rho(i)}$ is a re-encryption of $(c_1,c_2)_i$ for a sequence of secret permutations and random re-encryptions. Each mix server generates a proof of correct computation of the output, which satisfies the input and the public key.

Commitment Scheme. A commitment scheme is a triple of PPT algorithms (Setup, Commit, Open) defined as follows.

- Setup$(\lambda) \to \mathcal{PP}$: on input a security parameter λ, it outputs the public parameters $\mathcal{PP}$, including a description of the message space $\mathcal{M}$, commitment space, and commitment key space.
- Commit$(\mathcal{PP}, m) \to (c,r)$: on input $\mathcal{PP}$, and a message $m \in \mathcal{M}$, it outputs a commitment c and the opening randomness r.
- Open$(\mathcal{PP},(c,m,r)) \to 0/1$: on input a commitment c on message m with randomness r, it outputs 1 if accept and 0 otherwise.

A secure commitment scheme satisfies correctness, binding, and hiding properties as follows:

- *Correctness*. For every $m \in \mathcal{M}$,

$$Pr\left[\begin{array}{l|l} \texttt{Setup}(\lambda) \to \mathcal{PP} & \texttt{Open}(\mathcal{PP},(c,m,r) \\ \texttt{Commit}(\mathcal{PP},m) \to & \quad \to 1 \\ \quad (c,r) & \end{array}\right] = 1$$

- *Hiding*. For all PPT adversaries $\mathcal{A}$, there exists a negligible ε such that $\forall \lambda \in \mathbb{N}$,

$$Pr\left[\begin{array}{l|c} \texttt{Setup}(\lambda) \to \mathcal{PP} & \\ \mathcal{A}(\lambda) \to (st,m_0,m_1) & \\ b \to \{0,1\} & b = b' \\ \texttt{Commit}(\mathcal{PP},m_b) \to & \\ \quad (c,r) & \\ \mathcal{A}(st,c) \to b' & \end{array}\right] \leq 1/2 + \varepsilon(\lambda)$$

- *Binding*. For all PPT adversaries $\mathcal{A}$, there exists a negligible ε such that $\forall \lambda \in \mathbb{N}$,

$$Pr\left[\begin{array}{l|c} \texttt{Setup}(\lambda) \to \mathcal{PP} & \\ \mathcal{A}(\lambda) \to (c,m,r,m',r') & m \neq m' \\ b \to \{0,1\} & \wedge \\ \texttt{Open}(\mathcal{PP},(c,m,r)) \to b & b = b' = 1 \\ \texttt{Open}(\mathcal{PP},(c,m',r')) \to b' & \end{array}\right] \leq \varepsilon(\lambda)$$

A commitment scheme is *perfectly hiding* against a computationally unbounded adversary if the scheme satisfies the hiding property with $\varepsilon = 0$. A commitment scheme is *statistical hiding* against a computationally unbounded adversary if the scheme satisfies the hiding property with the adversary advantage at most $\varepsilon(\lambda)$. A commitment scheme is *perfectly binding* against a computationally unbounded adversary if the scheme satisfies the binding property with $\varepsilon = 0$.

In our scheme, we use the SHA-commitment scheme. The public parameters are assumed to be generated in an initial setup phase and publicly known to all parties thereafter. We simply write $c = H(cr,t)$ for a hash commitment to a message cr with opening randomness t.

Digital Signature Scheme. A digital signature scheme is a triple of PPT algorithms (KeyGen, Sign, verify) are defined as follows.

- KeyGen$(\lambda) \to (sk_\sigma, pk_\sigma)$: on input a security parameter λ, it outputs a signing key pair (sk_σ, pk_σ), with pk_σ denoting the verification key and sk_T the signing key.
- Sign$(sk_\sigma, m) \to \sigma$: on input a message $m \in \{0,1\}^*$ and a signing key sk_σ, it outputs a signature σ on message m.
- Verify$(\sigma, m, pk_\sigma) \to 0/1$: on input a signature σ on message m and a verification key pk_σ, it outputs 1 if it accepts the signature and 0 otherwise.

A digital signature scheme satisfies correctness and existential unforgeability properties (Katz and Lindell, 2007). In our scheme, the registrar signs the list of registered voters and publishes the signed list on the bulletin board.

zk-SNARK. A Zero-Knowledge Succinct Non-interactive Argument of Knowledge (zk-SNARK) is a cryptographic proof primitive. To achieve the succinctness it uses pre-processing for arithmetic circuit satisfiability. It assumes an algorithm running as a one-time trusted setup for preprocessing. Proof generation and verification depend on the output of the preprocessing setup.

Let R_λ, $\lambda \in \mathbb{N}$ is security parameter, be a polynomial-time decidable relations R on pairs (x, ω) where x is the statement, and w is the witness. We denote $R(x, \omega) = 1$ to show that (x, ω) satisfies on R and

$R(x,\omega) = 0$ otherwise. A preprocessing zk-SNARK for R_λ is a triple of PPT algorithms (KeyGen, Prove, Verify) defined as follows.

- KeyGen$(R,\lambda) \rightarrow (pk,vk)$: on input a security parameter λ and a relation R represented as an arithmetic circuit of size polynomial in λ, it outputs pk as a proving key and vk as a verifying key.
- Prove$(pk,x,\omega) \rightarrow \pi$: on input pk, an evaluation key for a relation R, a statement x, and a witness w such that $R(x,\omega) = 1$, it outputs a proof π.
- Verify$(vk,x,\pi) \rightarrow 0/1$: on input a verification key vk, a statement x, and a proof π, it outputs 1 to indicate a valid proof and 0 otherwise.

zk-SNARKs are required to protect the prover from the disclosure of the secret witness, and the verifier from a forged proof. We now recall the security notions to define a zk-SNARK.

- *Completeness*. An honest verifier always accepts a proof made by an honest prover for a statment x using the valid witness ω. Formally,

$$Pr\left[\begin{array}{c}\forall\lambda \in \mathbb{N}, \forall(x,\omega) \in R\\ \texttt{KeyGen}(R,\lambda) \rightarrow (pk,vk)\\ \texttt{Prove}(pk,x,\omega) \rightarrow \pi\\ \texttt{Verify}(vk,x,\pi) \rightarrow 1\end{array}\right] \geq 1-\varepsilon(\lambda).$$

- *Perfect Zero-Knowledge*. The proof and the keys reveal no information about the secret witness ω. Formally, there is PPT algorithm $sim = (simGen, simPr)$ such that for all $\lambda \in \mathbb{N}$, $(x,\omega) \in R$, and PPT adversary $\mathcal{A}$, the following two distributions are statistically close:

$$D_0 = \left[\begin{array}{c}\texttt{KeyGen}(R,\lambda) \rightarrow (pk,vk)\\ \texttt{Prove}(pk,x,\omega) \rightarrow \pi_0\\ (pk,vk,x,\pi_0)\end{array}\right]$$

$$D_1 = \left[\begin{array}{c}\texttt{simGen}(R,\lambda) \rightarrow (pk,vk,td)\\ \texttt{simPr}(pk,x,td) \rightarrow \pi_1\\ (pk,vk,x,\pi_1)\end{array}\right],$$

where td denotes the simulation trapdoor.

- *Proof of Knowledge*. Intuitively, every prover generating valid proof must know a the corresponding secret witness. Formally, for any PPT prover, there exists a PPT Extractor and negligible function ε such that for all $\lambda \in \mathbb{N}$, R, and any auxiliary input $m \in \{0,1\}^*$,

$$Pr\left[\begin{array}{l|l}\texttt{KeyGen}(R,\lambda) & \\ \quad\rightarrow (pk,vk) & \texttt{Verify}(vk,x,\pi)\\ \texttt{Prove}(pk,m) & \quad\rightarrow 1\\ \quad\rightarrow (\pi,x) & \wedge(x,\omega) \notin R\\ \texttt{Extract}(pk,vk,m) & \\ \quad\rightarrow \omega & \end{array}\right] \leq \varepsilon(\lambda).$$

The Extractor has full access to the prover's state, including any random coins.

- *Succinctness*. For any $\lambda \in \mathbb{N}$, (pk,vk), and any binary relation R, the proof size is $poly(\lambda)$ and the verification time is $poly(\lambda) + |x|$.

Zero-knowledge proofs are the main tool in our protocol to achieve the voter's privacy. In our scheme, `KeyGen` samples a proving key pk and a verification key vk, where this preprocessing is publicly verifiable. Both keys are published as public parameters and can be used any number of times to prove/verify membership in L_R. Thanks to zk-SNARK, a voter can prove that they know the encryption randomness for a ciphertext $e_v = (c_1,c_2)$, the commitment c, the opening randomness t for a voting credential cr, and a list of commitments (a Merkle tree root) such that $c := H(cr,t)$. Nobody knows that the voting credential is assigned to which commitment in the voters' commitment list. In addition, it allows us to prove the voter's eligibility while protecting the unlinkability between voting credential and voter identity.

We also use zero-knowledge proofs to allow the voter to prove 1) knowledge of the secret information involved in the encryption of the vote, 2) knowledge of the secret commitment related to the public commitment list or Merkle tree root, and 3) knowledge of the opening randomness of the secret commitment related to the voting credential in the ballot. In the tally phase, we provide proof for the validity of mixing and decryption (Fiat and Shamir, 1986; Chaum and Pedersen, 1992; Hirt and Sako, 2000; Schnorr, 1991; Camenisch and Stadler, 1997).

It is worth to note that zk-SNARK requires a one-time trusted setup of public parameters, i.e., (pk,vk). The violation of the trust assumption might affect the soundness of the proofs though privacy continues to hold even if the setup trapdoor is revealed. In (Ben-Sasson et al., 2018), the authors proposed a *transparent zero-knowledge system* (zk-STARK) in which the setup does not rely on any trusted party, and it has no trapdoors that could be exploited by powerful parties to prove false witness.

4 ELECTION SCHEME AND ADVERSARY MODELS

Our e-voting scheme consists of the following participants: the election authority $\mathcal{E}$, the registrar $\mathcal{R}$, the voters $I = \{id_1, id_2, \ldots, id_m\}$, the talliers $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$ and a public append-only bulletin board $\mathcal{BB}$. The election authority provides the following election public information: the candidate list, the list of eligible voters, and the public parameters.

A voter interacts with the registrar via an authenticated channel to register for the election. The registrar authenticates eligible voters and publishes the voter identities on the $\mathcal{BB}$. It signs the list of registered voters. The $\mathcal{BB}$ lists the ballots, either cast by a voter or by the $\mathcal{BB}$ itself, as null ballots. The talliers publish the election result and the proofs of correct tallying on the $\mathcal{BB}$ during the tally phase. The talliers verify, mix, shuffle, and decrypt the selected ballots to compute the final result. Each tallier has a partial decryption key of a k-out-of-n encryption scheme.

Adversary Models. We first consider an adversary who aims to break the privacy of the voter by linking a vote to the voter. We consider the following adversarial capabilities: A computationally bounded adversary $\mathcal{A}$ can actively participate in the election, corrupt some voters and collect all data available during the election. To evaluate the everlasting properties, we consider a computationally unbounded adversary $\mathcal{A}'$ who can access any publicly available information and knowledge from corrupted voters in the future. As we shall see later, the cryptographic primitives in our scheme provide statistical hiding and zero-knowledge properties, therefore, a computationally unbounded adversary has a negligible advantage to break participation privacy and vote privacy. Another type of adversary is a vote buyer. A vote buyer aims to pay rewards to dishonest voters who can convince the adversary that they voted as instructed with a receipt. In general, receipt-freeness does not prevent an adversary from buying the voter's private key and voting on behalf of the voter. Our scheme is receipt-free under the assumption that the adversary is computationally limited and that the bulletin board and the voting device are trusted.

4.1 Definition of the E-Voting Scheme

Our scheme is defined in terms of eight functions, i.e. ES =(Setup, Registervoter, Register, Vote, Valid, Append, Tally, VerifyTally) and proceeds in five different phases: *setup*, *registration*, *voting*, *tally*, and *verification*. The eight functions are defined as follows.

- Setup$(\lambda, R) \rightarrow (\mathcal{PP}, sk_{\mathcal{T}}, sk_{\sigma})$: on input a security parameter λ and a relation R represented as an arithmetic circuit of size polynomial in λ, it generates the prover and verifier key pair $(pk, vk) \leftarrow$KeyGen(R, λ), the election encryption key pair $(pk_{\mathcal{T}}, sk_{\mathcal{T}}) \leftarrow$KeyGenE$(\lambda)$, the registrar's signing key pair $(sk_{\sigma}, pk_{\sigma}) \leftarrow$KeyGenS$(\lambda)$, the commitment parameters $CR \times T \leftarrow$SetupC(λ) from the commitment setup, and the public parameters $\mathcal{PP} = (\mathcal{G}, q, g, H, pk_{\mathcal{T}}, pk_{\sigma}, (pk, vk))$.
- Registervoter$(id) \rightarrow (c_{id}, cr_{id}, t_{id})$: on implicit input $\mathcal{PP}$ and voter identity id, it chooses a random pseudonym $cr_{id} \leftarrow CR$, computes $(t_{id}, c_{id}) \leftarrow$ Commit$(\mathcal{PP}, cr_{id})$, and returns $(c_{id}, cr_{id}, t_{id})$ where t_{id} is chosen randomly from T.
- Register$(id, c_{id}, L) \rightarrow (L, rt_L, \sigma)$: on input voter identity and commitment (id, c_{id}) and list L, it adds (id, c_{id}) to the list L, computes rt_L and signature σ on (L, rt_L) with secret register key sk_{σ}, it then returns (L, rt_L, σ).
- Vote$(id, sk_{id}, pk_{\mathcal{T}}, v) \rightarrow \beta$: on input voter identity id, election public key $pk_{\mathcal{T}}$, and voter secret key $sk_{id} = (t_{id}, cr_{id})$, it generates a ballot $\beta = (e_v, cr_{id}, \pi_{id})$ with pseudonym cr_{id} and vote v by computing $e_v = enc_{pk_{\mathcal{T}}}(v; r)$. In addition, it computes a disjoint proof with Prove$(pk_R, x, \omega) \rightarrow \pi$, where $\omega = (r, c_{id}, v, t_{id})$ and $x = (e_v, cr_{id}, rt_L)$, and simulates the null ballot proof.
- Valid$(\beta) \rightarrow 0/1$: on input a ballot $\beta = (e_v, cr_{id}, \pi_{id})$, it checks that it is valid, i.e., that the proof is correct and it is well-formed with Verify$(vk, (e_v, cr_{id}), \pi_{id}) \rightarrow 0/1$.
- Append$(\mathcal{BB}, \beta) \rightarrow \mathcal{BB}$: on input a ballot β, it appends β to $\mathcal{BB}$ based on D_t. It generates and appends a null ballot(s) (e_0, cr_{id}, π_{id}) with the pseudonym cr_{id} based on probability distribution D_r and D_t. It computes $e_0 = enc_{pk_{\mathcal{T}}}(0; r)$ and disjoint proof π_{id} with Prove$(pk_R, x, \omega) \rightarrow \pi_{id}$, where $\omega = (r, 0)$, $x = (e_0, cr_{id}, rt_L)$, and it simulates the other side of the voter's proof.
- Tally$(\mathcal{BB}, sk_{\mathcal{T}}) \rightarrow (s, \Pi)$: on input the public bulletin board, it computes the election result. It returns (s, Π), where s is the election result, and Π is proof of correct tallying, as follows.
  - Run Valid(β) and return 0 if it fails.
  - For each cr_{id} appearing in the ballots, computes $B_{cr_{id}} = \prod_{e_v \in B(cr_{id})} e_v$ where $B(cr_{id})$ is the set of (e_v, cr_{id}, π_{id}) identifying by cr_{id}.
  - Remove (cr_i, π_i) from each B_{cr_i} and mix the ballots $\{B_{cr_1}, B_{cr_2}, \ldots, B_{cr_k}\}$ where k is the number of distinct pseudonym cr_i, and return the mixed ballots $\{B'_1, B'_2, \ldots, B'_k\}$ with a proof of valid mixing.
  - For each B'_i and vote option $v \in \mathcal{V}$ apply a privacy equivalence test (PET) and provide the corresponding proof.
  - Compute the result s based on the PET for each vote v and publish the proofs.
- VerifyTally $(\mathcal{BB}, s, \Pi) \rightarrow 0/1$: on input (s, Π), it returns 1 if all the proofs are valid, otherwise 0.

4.2 Phases of the E-Voting Scheme

We now describe how each function is executed in each phase. The detailed steps are in Figure 1.

Setup Phase: given a security parameter λ and relation R, $\mathcal{E}$ runs Setup(λ, R) that:

1: generates $(\mathcal{G}, q, g)$, threshold tuple (k,n) election key pair $(pk_{\mathcal{T}}, sk_{\mathcal{T}})$, Registrar key pair$(pk_{\sigma}, sk_{\sigma})$, commitment function and parameters $H : CR \times T \rightarrow C$, and zero-knowledge proof key pair (pk, vk) for the relation R.

2: publishes $\mathcal{PP} = (\mathcal{G}, q, g, H, pk_{\mathcal{T}}, pk_{\sigma}, (pk, vk))$, eligible voters I, and candidate list $\mathcal{V}$ (as an ElGamal message g^v) on $\mathcal{BB}$.

Registration Phase: a voter id runs Register(id) which

1: selects voting pseudonym $cr_{id} \in CR$;

2: computes $c_{id} = H(cr_{id}, t_{id}) \in \{0,1\}^{O(\lambda)}$ with $t_{id} \in T$ and stores (cr_{id}, t_{id}) locally.

3: sends c_{id} to $\mathcal{R}$ via an authentic channel.

4: $\mathcal{R}$ appends (id, c_{id}) to the list L;

5: computes the Merkle tree root rt_L from the order commitments in the list L;

6: signs L and rt_L and publishes on $\mathcal{BB}$.

7: The voter verifies $c_{id} \in L$ and the Merkle tree root rt_L on $\mathcal{BB}$.

Voting Phase: to cast a vote v, the voter runs Vote$(id, sk_{id}, pk_{\mathcal{T}}, v)$, with $sk_{id} = (cr_{id}, t_{id})$, which

1: computes $e_v = enc_{pk_{\mathcal{T}}}(v; r)$ where $r \in \mathbb{Z}_q$ is the encryption randomness. In the case of nullifying the previous vote v_{pre} and voting for v_{new}, the voter sets $v = v_{new} - v_{pre}$;

2: computes a zero-knowledge proof π_{id} with the proving key pk:

$$\pi_{id} = \{(r, c_{id}, v, t_{id}) | (e_v = (g^r, g^v pk_{\mathcal{T}}^r) \wedge c_{id} = H(cr_{id}, t_{id}) \wedge c_{id} \in rt_L)\} \vee \{(r, g^0) | e_v = (g^r, g^0 pk_{\mathcal{T}}^r)\};$$

3: submits $\beta = (e_v, cr_{id}, \pi_{id})$ to the $\mathcal{BB}$ via an anonymous channel.

4: $\mathcal{BB}$ runs Valid(β) which

5: checks the validity of the proof on the ballot and verifies the ballot does not already exist on $\mathcal{BB}$;

6: runs Append$(\mathcal{BB}, \beta)$ which appends the ballot β and null ballots to $\mathcal{BB}$.

7: The voter verifies that β is appended to $\mathcal{BB}$.

8: $\mathcal{BB}$ generates the null ballot as follows:

9: computes $e_0 = enc_{pk_{\mathcal{T}}}(0; r)$;

10: selects a cr_{id} from a β on $\mathcal{BB}$;

11: computes π_{id} as:

$$\pi_{id} = \{(r, c_{id}, v, t_{id}) | (e_0 = (g^r, g^v pk_{\mathcal{T}}^r) \wedge c_{id} = H(cr_{id}, t_{id}) \wedge c_{id} \in rt_L)\} \vee \{(r, g^0) | e_0 = (g^r, g^0 pk_{\mathcal{T}}^r)\}.$$

Tally Phase: $\mathcal{T}$ runs Tally$(\mathcal{BB}, sk_{\mathcal{T}})$ which

1: verifies the ballots on $\mathcal{BB}$ and select ballots with valid proof ;

2: computes a final ballot for each cr_{id} on $\mathcal{BB}$ by applying the homomorphic property of the encryption scheme;

3: shuffles, mixes, and publishes the final ballots without cr_{id} and provides proof of correctness;

4: applies PET on the final ballots and selects ballots that encrypt a vote;

5: decrypts the ballots and publishes the result with proof of decryption on $\mathcal{BB}$.

Verification Phase:

1: any party can verify that tallying is performed correctly by running VerifyTally$(\mathcal{BB}, s, \Pi)$ which verifies the result and all proofs posted on tallying.

Figure 1: Election Process.

Setup Phase. $\mathcal{E}$ runs the Setup algorithm with security parameter 1^{λ} and relation R that generates a threshold tuple (k,n), the candidate list $\mathcal{V}$, the encryption parameters $(\mathcal{G}, q, g)$, the election encryption key pair $(pk_{\mathcal{T}}, sk_{\mathcal{T}})$, the registrar key pair $(pk_{\sigma}, sk_{\sigma})$, the commitment scheme public parameters, and the setup function for zk-SNARKs that results in a key pair (pk, vk) for generating and verifying proofs. Once completed, $\mathcal{E}$ publishes $I = \{id_1, id_2, \ldots, id_m\}$, (pk, vk), $pk_{\mathcal{T}}$, pk_{σ}, the encryption and commitment public parameters, and the candidate list $\mathcal{V}$ on $\mathcal{BB}$. In addition, $\mathcal{E}$ defines the discrete probability distributions D_r and D_t used respectively to sample the number of null ballots for each pseudonym and to determine the time to cast each of them in the voting phase. While D_r can be a uniform distribution, D_t is a distribution that represents typical vote casting behaviour[1]

Registration Phase. Every voter id is assumed to se-

[1]If D_t is a uniform distribution, an adversary might be able to distinguish revoting from null ballots due to vote casting behaviour.

lect a pseudonym cr_{id} with opening value t_{id} and generates a commitment $c_{id} = H(cr_{id}, t_{id})$. Each voter submits commitment c_{id} to a registrar $\mathcal{R}$ via an authentic channel. $\mathcal{R}$ publishes the commitments and the list of voters on $\mathcal{BB}$.

$\mathcal{R}$ also publishes a Merkle tree root rt_L of the pairs of commitments and the corresponding voter identities: $L = \{(c_{id_1}, id_1), \ldots, (c_{id_m}, id_m)\}$ under the registrar signature σ. The voter can verify the published commitment next to their identity. The voter with the authentication path, which can be provided by the registrar, can verify that their commitment is a leaf of the Merkle tree root rt_L. Anyone can verify that rt_L is correctly built thanks to the signed list L. Note that publishing L does not affect participation privacy because the list contains the identity and commitment of the voters who are registered for the election.

Voting Phase. The voter submits a ballot β to the bulletin board through an anonymous channel using the Vote algorithm. The ballot $\beta = (e_v, cr_{id}, \pi_{id})$ contains an encrypted vote $e_v = enc_{pk_T}(v; r)$, the voting pseudonym cr_{id}, and a disjunctive zero-knowledge proof π_{id} that proves that either the ballot β was created by a voter id in possession of the secret witness about the voter pseudonym and the encryption randomness r, or it is a null ballot that reuses a voting pseudonym from a ballot previously cast on $\mathcal{BB}$. More precisely, π proves the knowledge of r, v, and t such that $H(cr_{id}, t_{id})$ is a leaf on rt_L and $e_v = enc_{pk_T}(v; r)$, or that the β is a null ballot. Note that cr_{id} is unique; thus, nobody can generate a new non-null ballot without knowing t_{id} to generate π_{id}. However, one can generate a null ballot (i.e., an encryption of zero) from a known voting pseudonym taken from a ballot on $\mathcal{BB}$.

A trusted server or the $\mathcal{BB}$ regularly generates and appends null ballots to distract a potential adversary observing the posts on $\mathcal{BB}$ from learning voting behaviors and enabling receipt-freeness. The $\mathcal{BB}$ checks the validity of the proof π, then verifies that the β does not already exist on $\mathcal{BB}$. A valid β is then published on $\mathcal{BB}$.

Tally Phase. All ballots with the same voting pseudonym are added together. Then, the respective voting pseudonym and proof are removed from the ballots, which are shuffled and mixed. Only ballots that pass a plaintext equality test (PET) are decrypted. The talliers publish on $\mathcal{BB}$ the result alongside the proofs of correct shuffle and decryption.

Verification Phase. Any party can verify the result and the proofs on $\mathcal{BB}$.

5 SECURITY PROPERTIES

We prove participation privacy, receipt-freeness, and vote privacy. Note that while we model in the proof the tallier as a single trusted party, the result also holds when the talliers are distributed.

5.1 Participation Privacy

A voting scheme ensures participation privacy if the scheme only reveals the number of participants and the results of the election. Our definition of participation privacy is inspired by a game-based definition of ballot privacy (Bernhard et al., 2015). Kulyk et al. (Kulyk et al., 2015) propose a quantitative definition of participation privacy. Their definition captures participation privacy for a voter based on dummy ballots, which are inserted by a trusted party, to make a voter who participates in the election indistinguishable from one who abstains. We propose a new definition that does not rely on a trusted party casting dummy ballots and can be also used to prove everlasting participation privacy.

Definition. Given a PPT adversary $\mathcal{A}$, we define the experiment $Exp_{\mathcal{A},ES}^{p-priv,b}$, which models an indistinguishability game involving two bulletin boards being tracked simultaneously. Only one of these bulletin boards is accessible to $\mathcal{A}$ depending on the bit $b \in \{0,1\}$ of $Exp_{\mathcal{A},ES}^{p-priv,b}$, as defined in Fig.2. The challenger flips a coin and executes the Setup phase where $\mathcal{PP}$ represents the public information of an election scheme (ES) in the Setup phase. The adversary $\mathcal{A}$ can make multiple queries to the oracle OVote to let an honest voter with identity id to cast a vote for candidate v on $\mathcal{BB}_0$ and $\mathcal{BB}_1$. The adversary $\mathcal{A}$ can call the oracle OCast to cast a ballot on behalf of any voter. By using these oracles, $\mathcal{A}$ populates both bulletin boards with additional contents so that both bulletin boards have the same ballots except the ballots for id_0 and id_1. In order to model vote abstention versus vote participation, we provide $\mathcal{A}$ with an oracle OAbstain to allow an honest voter id_0 to participate to the election by voting for a candidate v, while an honest voter id_1 abstains on $\mathcal{BB}_0$ or vice versa on $\mathcal{BB}_1$. The election result is computed on $\mathcal{BB}_0$. The adversary can make a query to the oracle OTally to access the result on $\mathcal{BB}_b$. The adversary $\mathcal{A}$ is allowed to query all oracles multiple times. Note that the computationally unbounded adversary $\mathcal{A}'$ has access through $view_{\mathcal{A}}$ to the knowledge of $\mathcal{A}$, but not on information derived from the communication channels. This is also known as *practical everlasting privacy*.

Definition 1. *An election scheme ES achieves participation privacy if for all adversaries $\mathcal{A}$ and $\mathcal{A}'$, where*

$Exp^{p-priv,b}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$:

$(\mathcal{PP}, sk_{\mathcal{T}}, sk_{\sigma}) \leftarrow$Setup$(\lambda, R)$
$(id_0, id_1) \leftarrow \mathcal{A}(\mathcal{PP}, L, \mathcal{V})$
$b \leftarrow \mathcal{A}^{O}(\mathcal{PP}, L, \mathcal{V}, id_0, id_1)$
$b \leftarrow \mathcal{A}'(\mathcal{PP}, L, \mathcal{V}, id_0, id_1, view_{\mathcal{A}})$
output b'

OVote(id, v) :
if $v \notin \mathcal{V}$ or $id \notin L \setminus \{id_0, id_1\}$ **then** return $\perp$
end if
$\beta \leftarrow$ vote$(id, sk_{id}, pk_{\mathcal{T}}, v)$
append$(\beta, \mathcal{BB}_0)$
append$(\beta, \mathcal{BB}_1)$

OAbstain(v) :
if $v \notin \mathcal{V}$ **then** return $\perp$
end if
$\beta_0 \leftarrow$ vote$(id_0, sk_{id_0}, pk_{\mathcal{T}}, v)$
$\beta_1 \leftarrow$ vote$(id_1, sk_{id_0}, pk_{\mathcal{T}}, v)$
append$(\beta_0, \mathcal{BB}_0)$
append$(\beta_1, \mathcal{BB}_1)$

OCast$(\beta_{\mathcal{A}})$:
if valid$(\mathcal{BB}_b, \beta_{\mathcal{A}}) = 0$ **then** return $\perp$
end if
append$(\mathcal{BB}_0, \beta_{\mathcal{A}})$
append$(\mathcal{BB}_1, \beta_{\mathcal{A}})$

OBoard() :
return $\mathcal{BB}_b$

OTally() :
$(s, \Pi_b) \leftarrow$ tally$(\mathcal{BB}_b, sk_{\mathcal{T}})$
return (s, Π_b)

Figure 2: Participation Privacy $Exp^{p-priv,b}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$.

$\mathcal{A}$ is PPT,

$$\Big| Pr\Big[Exp^{p-priv,0}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R) = 1\Big] - Pr\Big[Exp^{p-priv,1}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R) = 1\Big]\Big|$$

is negligible in security parameter λ.

To evaluate the participation privacy of our scheme, we specify the current capabilities of $\mathcal{A}$ and the future capabilities of $\mathcal{A}'$ as follows: the computationally bounded adversary $\mathcal{A}$ might be able to vote, corrupt some voters, and has access to communication channels. However, the computationally unbounded adversary $\mathcal{A}'$ cannot access the (anonymous) communication channels, but can access to any other information in possession of $\mathcal{A}$. Honest voters do not actively prove participation or abstention to $\mathcal{A}$.

Theorem 1. *Our scheme has participation privacy.*

Proof. We define a sequence of games, starting with $\mathcal{A}$ interacting with the participation privacy challenger with $b = 0$, and ending with $\mathcal{A}$ interacting with the participation privacy challenger with $b = 1$. Each transition will be noticed by the $\mathcal{A}$ with a negligible probability. Therefore, we will be able to show that $\mathcal{A}$ has a negligible distinguishing advantage.

Let G be the participation privacy game corresponding to $Exp^{p-priv,0}_{\mathcal{A},ES}(\lambda, L, \mathcal{V})$. This experiment simulates the voting scenario in our scheme where id_0 participates while id_1 does not. $\mathcal{A}$ sees $\mathcal{BB}_0$ and the result (s, Π) is returned by oracle OTally() on $\mathcal{BB}_0$. The adversary $\mathcal{A}'$ has access to $\mathcal{BB}_0$ via $view_{\mathcal{A}}$.

Let G_0 be the game obtained by modifying the game G. More precisely, we modify G so that challenger that has access to the trapdoor and programming random oracle provided in the setup phase simulates all zk-SNARK proofs. Since the zk-SNARK system is perfect zero knowledge, the distribution of the simulated π is identical to that of the proofs computed in G. Hence the advantage of $\mathcal{A}$ and $\mathcal{A}'(view_{\mathcal{A}})$ in distinguishing these two games is zero.

Game G_1 is obtained by changing game G_0 as follows: the challenger replaces the output of OAbstain(v) by swapping the ballots of the voters (id_0, c_0) and (id_1, c_1). In this case, a ballot (e_v, cr_1, π_{cr_1}) is placed on $\mathcal{BB}_0$ instead of (e_v, cr_0, π_{cr_0}) as a result of an OAbstain(v) query.

Based on the hiding property of the commitment scheme and on zk-SNARK proof system, we argue that the distinguishing probability of the participation privacy adversary between game G_0 and G_1 is negligible in security parameter λ. To this end, we consider a computationally unbounded adversary $\mathcal{B}$ against a statistical hiding property of the commitment scheme that makes use of the distinguishing advantage of $\mathcal{A}$ between games G_0 and G_1. The adversary $\mathcal{B}$ simulates the games for $\mathcal{A}$: let $\{c_b, c_{1-b}\}$ be the response of the hiding-challenger to query $\{cr_0, cr_1\}$. The adversary $\mathcal{B}$ adds (id_b, c_b) and (id_{1-b}, c_{1-b}) to the list L. The adversary $\mathcal{B}$ simulates the answer of the oracle OAbstain(v) to $\mathcal{A}$ as follows: $\mathcal{B}$ computes the encryption of the vote v and simulates the proof π corresponding to the ballot information $\beta = (e_v, cr_0, \pi)$. The adversary $\mathcal{B}$ can compute the result which is equal to G_0. If c_b is a commitment on cr_0, $\mathcal{B}$ exactly simulates the game G_0, and G_1 otherwise. In the game G_0, the voter pseudonym cr_0 in the ballot β_0 on $\mathcal{BB}_0$ is related to (id_0, c_0). However, in G_1, the voter pseudonym cr_1, in the ballot β_1, is placed on $\mathcal{BB}_0$ where pseudonym cr_1 is related to the commitment c_1. As a result, the advantage of $\mathcal{A}$ in distinguishing game G_1 from G_0 is negligible in security parameter λ. Moreover, differences in the advantage between these games are negligible even if the computationally unbounded adversary $\mathcal{A}'$ is able to view $\mathcal{A}$'s information except the communication channels.

We have replaced the view of the adversary in game G namely $\mathcal{BB}_0$ to $\mathcal{BB}_1$ in $Exp^{p-priv,1}_{\mathcal{A},ES}$ through a sequence of the games. The advantage of the $(\mathcal{A}, \mathcal{A}')$ in distinguishing the transition over the game is negligible.

□

5.2 Receipt-Freeness

Definitions of receipt-freeness in the computational model usually consider indistinguishability games defined by oracles. We extend the receipt-freeness definition by Bernhard et al. (Bernhard et al., 2017) by allowing $\mathcal{A}$ to access the oracle OCast for casting a ballot, which is the same as the one defined in the participation privacy experiment. We refer to (Bernhard et al., 2017) for the definitions of the oracles which $\mathcal{A}$ can query in the experiment $Exp^{rf,b}_{\mathcal{A},EA}$.

Definition 2. *An election scheme ES achieves receipt-freeness if there exists an algorithm sim-proof such that for all PPT adversaries $\mathcal{A}$,*

$$\left| Pr\left[Exp^{rf,0}_{\mathcal{A},ES}(\lambda, L, \mathcal{V}, R) = 1\right] - Pr\left[Exp^{rf,1}_{\mathcal{A},ES}(\lambda, L, \mathcal{V}, R) = 1\right]\right|$$

is negligible in security parameter λ.

We assume that the tallier and the voting device are trustworthy. We assume an anonymous communication channel between the voter and the voting scheme. Also, the bulletin board that generates the null ballots is trustworthy, and voters can cast their ballots without being observed by the adversary $\mathcal{A}$.

We show that our scheme satisfies receipt-freeness, as defined in the receipt-free definition in (Bernhard et al., 2017). The oracle OVoteLR in $Exp^{rf,b}_{\mathcal{A},ES}$ models an honest voter id with two potential votes v_0 and v_1 in our scheme. The oracle OReceipt models the behavior of a voter when asked to provide a receipt by the adversary $\mathcal{A}$. The coerced voter can provide receipts to the adversary and does not change their votes, as illustrated on $\mathcal{BB}_0$. The case where a voter decides to update their vote is modeled on $\mathcal{BB}_1$. The function update-vote$(id, sk_{id}, pk_{\mathcal{T}}, v - v_{\mathcal{A}})$ models a coerced voter who changes their vote by encrypting the vote $v - v_{\mathcal{A}}$. The obfuscate function simulates the function of casting null ballots in our scheme.

Theorem 2. *Our scheme is receipt-free under the DDH assumption in the random-oracle model.*

Proof. We consider a sequence of games, starting from $Exp^{rf,0}_{\mathcal{A},ES}$ and step by step change the view of the adversary $\mathcal{A}$ from $\mathcal{BB}_0$ to $\mathcal{BB}_1$ in the $Exp^{rf,1}_{\mathcal{A},ES}$. We will demonstrate that the adversary $\mathcal{A}$ distinguishes the transition through all these games with a negligible advantage. In our scheme, the tally oracle simulates the proofs for the tally when $b = 1$ using a programmable random oracle. The function sim-proof takes as input the board $\mathcal{BB}_1$ and the result R from $\mathcal{BB}_0$, and returns the simulated proof Π_1. This proof is indistinguishable to PPT adversaries from the proof of correct tallying.

Let G_0 be the first game corresponding to $Exp^{rf,0}_{\mathcal{A},ES}(\lambda, L, \mathcal{V}, R)$. This experiment simulates the voting scenario in our scheme where the coerced voter id submits $\beta_{v_{\mathcal{A}}}$ and does not update their vote. In this game, the adversary $\mathcal{A}$ sees $\mathcal{BB}_0$, and the result (s, Π) is returned by oracle OTally() on $\mathcal{BB}_0$.

Let G_1 be the game obtained by modifying G_0 with the following change: the zero-knowledge proof in the tally phase is simulated by the algorithm sim-proof. Thanks to the ZK property of the proof system, the distinguishing probability of the receipt-free adversary $\mathcal{A}$ between G_0 and G_1 is negligible.

Let G_2 be the game obtained from game G_1 with the following change on $\mathcal{BB}_0$: the output of the oracle OVoteLR is replaced. Precisely, the ballot β_0 on $\mathcal{BB}_0$ is replaced with the corresponding ballot β_1 on $\mathcal{BB}_1$. The result s is equal to the result of the game G_1 and the proof is simulated by the sim-proof on the current bulletin board, namely, $\mathcal{BB}_{G_2}$.

Let G_3 be the game obtained from G_2 with a change on the output of the oracle OReceipt on $\mathcal{BB}_{G_2}$. A null ballot on $\mathcal{BB}_{G_2}$ is replaced by the update ballot β_v on $\mathcal{BB}_1$. The obfuscate function generated the null ballots on $\mathcal{BB}_0$ in the game G_0. The result s is equal to the result of the game G_2, and the proof is simulated by the sim-proof on $\mathcal{BB}_{G_3}$. In this game, the adversary view of the content of the bulletin board is corresponding to Experiment $Exp^{rf,1}_{\mathcal{A},ES}(\lambda, L, \mathcal{V}, R)$. This experiment is equivalent to the voting scenario in our scheme, where the voter id casts an additional update ballot β_v.

We now prove that the adversarial advantage in distinguishing between the output of G_1 and G_2 is negligible. Let $\mathcal{B}$ be an adversary against the non-malleable CPA property of the ElGamal scheme. The adversary $\mathcal{B}$ simulates the games for $\mathcal{A}$ and uses the distinguishing advantage of $\mathcal{A}$ to output a bit in NM-CPA game. Assume that β^* is the answer of the NM-CPA challenger to the adversary $\mathcal{B}$ on a query $\{v_0, v_1\}$. The adversary $\mathcal{B}$ sets β^* on the bulletin board view of $\mathcal{A}$ as a result of the oracle $\mathcal{O}\mathrm{VoteLR}(id, v_0, v_1)$. The adversary $\mathcal{B}$ computes the tally result by querying the decryption oracle in the NM-CPA game for all ballots on the bulletin board except for β^*. The result s is computed by the output of decryption oracle and vote v_0 for the ballot β^*. Note that we use the decryption oracle of the NM-CPA challenger to decrypt the ballots before computing the result s. The proof Π is simulated by sim-proof on the ballots on the visible bulletin board. The adversary $\mathcal{B}$ exactly simulates the G_1, if β^* be an encryption of v_0 and G_2 otherwise. As a result, the advantage of the adversary $\mathcal{A}$ in distinguishing the G_1 from G_2 is equal to the advantage of adversary $\mathcal{B}$ in the NM-CPA game. We proceed to show that G_2 and G_3 are indistinguishable. Again, a reduction to the NM-CPA security game of the ElGamal encryption scheme proves that the advantage of the adversary $\mathcal{A}$ in distinguishing between two games G_2 and G_3 are negligible. This reduction is simulated by the adversary $\mathcal{B}$ as before by setting β^* on the visible bulletin board. β^* is an answer of the NM-CPA challenger to the adversary $\mathcal{B}$ on a query $\{v_{\mathcal{A}}, v\}$.

In the games G_0, G_1, G_2, and G_3, we step by step replace all the ballots that depend on the bit $b = 0$ with the corresponding ballots depend on $\mathcal{BB}_1$. In particular, we prove that the advantage of the adversary $\mathcal{A}$ through the transition from $Exp^{rf,0}_{\mathcal{A},ES}$ to $Exp^{rf,1}_{\mathcal{A},ES}$ is negligible in security parameter λ.

□

5.3 Vote Privacy

We analyze the privacy of our scheme with the vote privacy game defined by $Exp^{vpriv,b}_{\mathcal{A},\mathcal{A}',ES}$, between the challenger and the adversary $\mathcal{A}$. We extend the ballot privacy definition for a permutation of honest votes (Benaloh and Yung, 1986; Bernhard et al., 2015) with several oracles, and model a game such that the adversary should not be able to distinguish if the identity of two voters with the votes v_0 and v_1 are swapped.

Our vote privacy game tracks two bulletin boards $\mathcal{BB}_0$ and $\mathcal{BB}_1$ (see Fig. 3). Only one bulletin board is accessible to the adversary by calling the oracle $\mathcal{O}\mathrm{Board}()$. The adversary can make calls to the oracle $\mathcal{O}\mathrm{Vote}(id, v)$ to let a voter with id to cast a vote v on $\mathcal{BB}_0$ and $\mathcal{BB}_1$; and to the oracle $\mathcal{O}\mathrm{Cast}(\beta_{\mathcal{A}})$ to cast ballots $\beta_{\mathcal{A}}$ generated by the adversary on $\mathcal{BB}_0$ and $\mathcal{BB}_1$. To model the voter indistinguishability, we provide the adversary with the oracle $\mathcal{O}\mathrm{VoterIND}(v_0, v_1)$. The adversary goal is to distinguish between two bulletin boards with the following change: the oracle $\mathcal{O}\mathrm{VoterIND}(v_0, v_1)$ lets a voter id_0 to cast a vote for v_0, and a voter id_1 to cast a vote for v_1 on $\mathcal{BB}_0$. The $\mathcal{O}\mathrm{VoterIND}$ appends the ballot β_{10} and β_{01} which denote id_1 with a vote for v_0, and id_0 with a vote for v_1 on $\mathcal{BB}_1$ respectively. More precisely, the identities of the ballots are swapped on $\mathcal{BB}_1$ in comparison with $\mathcal{BB}_0$. We use the notation $\mathcal{O}\mathrm{VoterIND}(v_0, v_1)$ to let $\mathcal{A}$ make queries multiple times on different vote options for two honest voters id_0 and id_1. The adversary can query the oracle $\mathcal{O}\mathrm{Tally}()$ to see the result of the election. For the everlasting privacy property, we define a computationally unbounded adversary $\mathcal{A}'$ receiving the view of the PPT adversary $\mathcal{A}$ except for some auxiliary information such as network communication, timestamps, etc. Therefore, $\mathcal{A}'$, who has access to $view_{\mathcal{A}}$ attempts to guess a bit b.

Definition 3. *An election scheme ES achieves vote privacy if there exists an algorithm sim-proof such that for all adversary $\mathcal{A}'$ and adversary $\mathcal{A}$, where $\mathcal{A}$ is PPT adversary,*

$$\left| Pr\left[Exp^{vpriv,0}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R) = 1\right] - Pr\left[Exp^{vpriv,1}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R) = 1\right] \right|$$

is negligible in λ.

Theorem 3. *Our scheme provides vote privacy under the statistical hiding property of the hash-based commitment scheme in the random oracle model.*

Proof. In this proof, we will step by step replace the view of the adversary in the game with $b = 0$ to the game $b = 1$. We show the advantage of the adversary $(\mathcal{A}, \mathcal{A}')$ is negligible in distinguishing these games.

Game G_0 is as $Exp^{vpriv,0}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$. This experiment is equal to the voting situation, where the voter id_0 submits β_{v_0} and the voter id_1 submits β_{v_1}. The oracle $\mathcal{O}\mathrm{Vote}$ allows $\mathcal{A}$ to let an honest voter id vote for a candidate v, while $\mathcal{O}\mathrm{Cast}$ allows voting on behalf of the corrupted voters and adds null ballots. In this game, $\mathcal{A}$ has access to $\mathcal{BB}_0$ by calling the oracle $\mathcal{O}\mathrm{Board}$, and the result (s, Π) is returned by oracle $\mathcal{O}\mathrm{Tally}()$ on $\mathcal{BB}_0$. The everlasting adversary $\mathcal{A}'$ takes the state of $\mathcal{A}$ and the oracle output as an input.

Game G_1 is equal to G_0 apart from the following change: the zero-knowledge proofs are simulated by the challenger who has access to the trapdoor and programming random oracle with the simulation set up.

$Exp^{vpriv,b}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$:

$(\mathcal{PP}, sk_{\mathcal{T}}, sk_{\sigma}) \leftarrow$Setup$(\lambda, R)$
$(id_0, id_1) \leftarrow \mathcal{A}(\mathcal{PP}, pk_{\mathcal{T}}, L, \mathcal{V})$
$b \leftarrow \mathcal{A}^{O}(\mathcal{PP}, L, \mathcal{V})$
$b \leftarrow \mathcal{A}'(\mathcal{PP}, L, \mathcal{V}, (id_0, id_1), view_{\mathcal{A}})$
output b'

OVote(id, v) :
if $v \notin \mathcal{V}$ or $id \notin L \setminus \{id_0, id_1\}$ **then** return $\perp$
end if
$\beta \leftarrow$ vote$(id, sk_{id}, pk_{\mathcal{T}}, v)$
append$(\beta, \mathcal{BB}_0)$
append$(\beta, \mathcal{BB}_1)$

OVoterIND(v_0, v_1) :
if $v_0 \notin \mathcal{V}$ or $v_1 \notin \mathcal{V}$ **then** return $\perp$
end if
$\beta_{00} \leftarrow$ vote$(id_0, sk_{id_0}, pk_{\mathcal{T}}, v_0)$
$\beta_{11} \leftarrow$ vote$(id_1, sk_{id_1}, pk_{\mathcal{T}}, v_1)$
$\beta_{10} \leftarrow$ vote$(id_1, sk_{id_1}, pk_{\mathcal{T}}, v_0)$
$\beta_{01} \leftarrow$ vote$(id_0, sk_{id_0}, pk_{\mathcal{T}}, v_1)$
append$((\beta_{00}, \beta_{11}), \mathcal{BB}_0)$
append$((\beta_{10}, \beta_{01}), \mathcal{BB}_1)$

OCast$(\beta_{\mathcal{A}})$:
if valid$(\mathcal{BB}_b, \beta_{\mathcal{A}}) = 0$ **then** return $\perp$
end if
append$(\mathcal{BB}_0, \beta_{\mathcal{A}})$
append$(\mathcal{BB}_1, \beta_{\mathcal{A}})$

OBoard() :
return $\mathcal{BB}_b$

OTally() :
$(s, \Pi_0) \leftarrow$ tally$(\mathcal{BB}_0, sk_{\mathcal{T}})$
$\Pi_1 \leftarrow$ sim-proof$(\mathcal{BB}_1, s)$
return (s, Π_b)

Figure 3: Vote privacy $Exp^{vpriv,b}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$.

G_0 and G_1 are indistinguishable by the simulatability of the zero-knowledge proof system.

Let G_2 be the game obtained from game G_1 with the following change on $\mathcal{BB}_0$: the output of the oracle OVoterIND is replaced as follows: the ballot $\beta_{00} = (e_{v_0}, cr_0, \pi_{c_0})$ on $\mathcal{BB}_0$ is replaced with the ballot $(e_{v_0}, cr_1, \pi_{c_0})$. Similarly, the ballot $\beta_{11} = (e_{v_1}, cr_1, \pi_{c_1})$ is replaced by the ballot $(e_{v_1}, cr_0, \pi_{c_1})$. As the zero-knowledge proofs are simulated by the challenger, this change is equal to the replacement of the voting pseudonym in $\beta_{00} = (e_{v_0}, cr_0, \pi_{c_0})$ and $\beta_{11} = (e_{v_1}, cr_1, \pi_{c_1})$ on $\mathcal{BB}_0$. More precisely, the voting pseudonyms cr_0 and cr_1 are swapped, and the related proofs are simulated by the challenger. Other ballots on $\mathcal{BB}_0$, which are either the output of OVote or OCast remain the same as in G_1. The result s is equal to the result of the game G_1 and the proof Π is simulated by the sim-proof on the current bulletin board, namely, $\mathcal{BB}_{G_2}$. At the end of this game, the adversary's view of the content of the bulletin board is corresponding to $Exp^{vpriv,1}_{\mathcal{A},\mathcal{A}',ES}(\lambda, L, \mathcal{V}, R)$. This experiment is equivalent to the game on $\mathcal{BB}_1$, where the voter id_1 with pseudonym cr_1 submits vote v_0 and the voter id_0 submits v_1. In this game, $\mathcal{A}$ has direct access to all oracles and channels while $\mathcal{A}'$ takes all information of the view of $\mathcal{A}$ excluding the communication.

We prove that the adversarial advantage in distinguishing between the output of G_1 and G_2 is negligible. Let $\mathcal{B}$ be an adversary against the hiding property of the commitment scheme. $\mathcal{B}$ simulates the games G_1 and G_2 for $\mathcal{A}$ and use the distinguishing advantage of $\mathcal{A}$ to output a bit in the hiding game. Assume that $\{c_b, c_{1-b}\}$ be the answer of the hiding challenger to $\mathcal{B}$ on a query $\{cr_0, cr_1\}$. $\mathcal{B}$ adds c_b and c_{1-b} to the list L, and simulates the output of the oracle OVoterIND(v_0, v_1) as follows: $\beta^b = (e_{v_0}, cr_0, \pi_{c_b})$ and $\beta^{1-b} = (e_{v_1}, cr_1, \pi_{c_{1-b}})$ on the bulletin board view of $\mathcal{A}$ as a result of the oracle OVoterIND(v_0, v_1). $\mathcal{B}$ computes the tally result of all ballots on the bulletin board and returns the result. Note that $\mathcal{B}$ can decrypt the ballots on the bulletin board. $\mathcal{B}$ exactly simulates G_1, if $b = 0$ in the hiding game related to the commitments $\{c_b, c_{1-b}\}$, and G_2 if $b = 1$. Hence, the advantage of $\mathcal{A}$ in distinguishing G_1 from G_2 is equal to the advantage of $\mathcal{B}$ in the statistically hiding game. In particular, we prove that the advantage of the PPT adversary $\mathcal{A}$ and the computationally unbounded adversary $\mathcal{A}'$, who has access to the state of $\mathcal{A}$ through the games that transfers from $Exp^{vpriv,0}_{\mathcal{A},\mathcal{A}',ES}$ to $Exp^{vpriv,1}_{\mathcal{A},\mathcal{A}',ES}$, is negligible in the security parameter λ.

□

5.4 Verifiability

A verifiable voting scheme guarantees that the result of the election is computed on the votes of the following groups: 1) the group of honest voters, who have verified their ballots on the bulletin board after casting their ballots; 2) the group of corrupted voters, who are fully controlled by the adversary; 3) the group of honest voters, who did not check their ballots on the bulletin board. For the last group, the adversary should not be able to modify the corresponding votes, but they can still be dropped or replaced by earlier ballots, if these exist.

In the tally phase, ballots with the same voting pseudonym are grouped. Let $\mathcal{BB} = \{\mathcal{B}_{cr_0}, \mathcal{B}_{cr_1}, \ldots, \mathcal{B}_{cr_{n_i}}\}$, where $\mathcal{B}_{cr_i}$ denotes the ballots with the same pseudonym cr_i. Given a voting pseudonym cr_i, the final ballot is the result of multiplying all ballots in $\mathcal{B}_{cr_i}$. The proof Π ensures that the result r is computed from $\{\mathcal{B}_{cr_1}, \mathcal{B}_{cr_0}, \ldots, \mathcal{B}_{cr_{n_i}}\}$. Indeed, the soundness of the proof Π prevents the manipulation of an adversary such as removing, adding, or modifying a ballot during the tally process. Let $|C|$ be the number of corrupted voters. We show that the adversary cannot corrupt more voters than $|C|$. Each ballot on $\mathcal{BB}$ is either a ballot generated by the knowledge of the secret pseudonym of a voter or it is a null ballot generated by the bulletin board. The proof π on a ballot verifies that the ballot is generated by the secret knowledge of the credential or the knowledge of the null ballot. The computationally binding property of the commitment scheme protects against the adversary who wants to generate a valid ballot with different (cr', t') such that $H(cr', t') = c$ and $(cr', t') \neq (cr, t)$. In addition, the soundness property of SNARKs and the trusted setup protect against the forgery of a ballot proof by an adversary. Hence, the adversary cannot cast a new non-null ballot without knowing the secret credential information or the zero-knowledge trapdoor. The adversary cannot modify the vote of an honest voter by revoting. So, non-null ballots on the $\mathcal{BB}$ that does not belong to honest voters must belong to corrupted voters. Thus, the bulletin board contains the ballots of the three groups mentioned above.

6 CONCLUSION

In this paper we proposed a new receipt-free e-voting scheme that also provides everlasting guarantees for participation privacy and vote privacy. Our scheme relies on hash-based commitments and zk-SNARK proofs to achieve everlasting guarantees while not compromising verifiability. In our scheme, the voter pseudonyms are not concealed. This may help to address a robustness issue (Haines et al., 2023) in previous receipt-free voting schemes with everlasting properties and minimal trust assumptions. For example, in Locher and Haenni (Locher and Haenni, 2016) a voter can submit an arbitrary number of ballots causing the bulletin board to be flooded with them. Such an attack cannot be avoided since the credentials are encrypted. If a voter casts a large number of ballots, the tallying phase can be costly. In our scheme, such an attack would be noticeable by the bulletin board (and anyone else looking at it), which can eventually refuse to add ballots coming from the same pseudonym. We can further limit the generation of null ballots to a (distributed) party with a known public key. In this case, the party casting a ballot should prove that they know the secret information of the public key and the null ballot. Therefore, extending our zero-knowledge proof for null ballots can fully prevent the bulletin board from getting flooded with null ballots.

We provide mathematical proofs that our scheme meets the privacy properties with minimal trust assumptions. For most properties, the sole assumption is the existence of an anonymous channel between the voter and the bulletin board. For verifiability, it is worth noting that zk-SNARK requires a one-time trusted setup of public parameters. However, such a requirement can be removed by replacing zk-SNARK with zk-STARK (Ben-Sasson et al., 2018) at the cost of less efficient proofs. Privacy and integrity are dependent on the security of the commitment scheme and ZK proof, while encryption is necessary for other properties such as fairness and receipt-freeness. Therefore, we note that hash-based commitment schemes that possess post-quantum secure binding properties, in conjunction with zk-STARK, can be used to build post-quantum secure e-voting systems with everlasting privacy.

ACKNOWLEDGEMENTS

This work is supported by the Villum Foundation, within the project "Enabling User Accountable Mechanisms in Decision Systems".

REFERENCES

Adida, B. (2008). Helios: Web-based open-audit voting. In *USENIX*.

Ben-Sasson, E., Bentov, I., Horesh, Y., and Riabzev, M. (2018). Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive*.

Benaloh, J. and Tuinstra, D. (1994). Receipt-free secret-ballot elections. In *STOC*.

Benaloh, J. C. and Yung, M. (1986). Distributing the power of a government to enhance the privacy of voters. In *PODC*.

Bernhard, D., Cortier, V., Galindo, D., Pereira, O., and Warinschi, B. (2015). Sok: A comprehensive analysis of game-based ballot privacy definitions. In *IEEE Symposium on Security and Privacy*.

Bernhard, D., Kulyk, O., and Volkamer, M. (2017). Security proofs for participation privacy, receipt-freeness and ballot privacy for the Helios voting scheme. In *ARES*.

Bohli, J.-M., Müller-Quade, J., and Röhrich, S. (2007). Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *E-VOTE ID*.

Brandt, F. (2005). Efficient cryptographic protocol design based on distributed el gamal encryption. In *Int. Conf. on Information Security and Cryptology*.

Buchmann, J., Demirel, D., and Graaf, J. v. d. (2013). Towards a publicly-verifiable mix-net providing everlasting privacy. In *Financial Cryptography and Data Security*.

Camenisch, J. and Stadler, M. (1997). Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, pages 410–424. Springer.

Chaidos, P., Cortier, V., Fuchsbauer, G., and Galindo, D. (2016). Beleniosrf: A non-interactive receipt-free electronic voting scheme. In *CCS*.

Chaum, D. and Pedersen, T. P. (1992). Wallet databases with observers. In *Annual international cryptology conference*, pages 89–105. Springer.

Demirel, D., Henning, M., Graaf, J. v. d., Ryan, P. Y., and Buchmann, J. (2013). Prêt à voter providing everlasting privacy. In *E-VOTE ID*.

Demirel, D., Van De Graaf, J., and dos Santos Araújo, R. S. (2012). Improving helios with everlasting privacy towards the public. *Evt/wote*.

Fiat, A. and Shamir, A. (1986). How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer.

Haines, T., Müller, J., Mosaheb, R., and Pryvalov, I. (2023). Sok: Secure e-voting with everlasting privacy. *PoPETs*.

Hirt, M. and Sako, K. (2000). Efficient receipt-free voting based on homomorphic encryption. In *Int. Conf. on the Theory and Applications of Cryptographic Techniques*.

Juels, A., Catalano, D., and Jakobsson, M. (2005). Coercion-resistant electronic elections. In *Towards Trustworthy Elections*.

Katz, J. and Lindell, Y. (2007). Private key encryption and pseudorandomness. *Introduction to Modern Cryptography, Chapman & Hall/CRC Cryptography and Network Security*, pages 47–109.

Kulyk, O., Teague, V., and Volkamer, M. (2015). Extending helios towards private eligibility verifiability. In *E-VOTE ID*.

Locher, P. and Haenni, R. (2015). Verifiable internet elections with everlasting privacy and minimal trust. In *E-VOTE ID*.

Locher, P. and Haenni, R. (2016). Receipt-free remote electronic elections with everlasting privacy. *Annals of Telecommunications*.

Moran, T. and Naor, M. (2006). Receipt-free universally-verifiable voting with everlasting privacy. In *Annual International Cryptology Conference*.

Okamoto, T. (1997). Receipt-free electronic voting schemes for large scale elections. In *Workshop on Security Protocols*.

Ryan, P. Y., Bismark, D., Heather, J. A., Schneider, S. A., and Xia, Z. (2009). The prêt à voter verifiable election system. *IEEE TIFS*.

Ryan, P. Y., Rønne, P. B., and Iovino, V. (2016). Selene: Voting with transparent verifiability and coercion-mitigation. In *Financial Cryptography and Data Security*.

Sako, K. and Kilian, J. (1995). Receipt-free mix-type voting scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*.

Schnorr, C.-P. (1991). Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174.