Security Proofs for Participation Privacy, Receipt-Freeness and Ballot Privacy for the Helios Voting Scheme

David Bernhard
University of Bristol
bernhard@cs.bris.ac.uk

Oksana Kulyk
Technische Universität Darmstadt
oksana.kulyk@secuso.org

Melanie Volkamer
Technische Universität Darmstadt
Karlstad University
melanie.volkamer@secuso.org

ABSTRACT

The Helios voting scheme is well studied including formal proofs for verifiability and ballot privacy. However, depending on its version, the scheme provides either participation privacy (hiding who participated in the election) or verifiability against malicious bulletin board (preventing election manipulation by ballot stuffing), but not both at the same time. It also does not provide receipt-freeness, thus enabling vote buying by letting the voters construct receipts proving how they voted. Recently, an extension to Helios, further referred to as KTV-Helios, has been proposed that claims to provide these additional security properties. However, the authors of KTV-Helios did not prove their claims. Our contribution is to provide formal definitions for participation privacy and receipt-freeness that we applied to KTV-Helios. In order to evaluate the fulfillment of participation privacy and receipt-freeness, we furthermore applied the existing definition of ballot privacy, which was also used for evaluating the security of Helios, in order to show that ballot privacy also holds for KTV-Helios.

1 INTRODUCTION

The Helios voting scheme has been introduced in [2] and subsequently implemented and used in several real-world elections such as the IACR elections [24]. Moreover, the research conducted on Helios led to the development of several extensions for the scheme [13–15, 21, 23, 42, 48], formal security definitions and proofs [8, 9, 15, 30] and usability evaluations [27, 41]. Due to these numerous scientific extensions and evaluations, the Helios scheme can be considered one of the most evolved e-voting scheme which provides ballot privacy and end-to-end verifiability. However, the current implementation of Helios does not provide verifiability against malicious bulletin board that can add or modify ballots on behalf of the voters who do not perform the necessary verification procedures. The extension proposed in [15] solves this issue by introducing digital signatures thus providing such verifiability against malicious bulletin board. It however, does not ensure participation privacy, meaning that the public available election data reveals whether a honest voter participated in the election or abstained. Although this information is usually potentially available in traditional paper-based elections, whereby anyone can observe people going into a polling station, an Internet voting system without participation privacy reveals the identities of the voters who cast their ballot in an election on a much larger scale by publishing them online. Hence, the lack of participation privacy in Internet voting is a violation of voter privacy that is more serious in comparison to paper-based elections. A further issue with voter privacy in Helios is the lack of receipt-freeness, that enables voters constructing receipts that prove to a third party which candidate the voter has voted for. Thus, such receipts could be used for vote buying.

Recently an extension to Helios has been proposed [31] (henceforth referred to as KTV-Helios) that adds probabilistic participation privacy and probabilistic receipt-freeness to the Helios voting scheme while, at the same time, ensuring verifiability against malicious bulletin board, assuming a reliable public-key infrastructure is in place. However, despite their conceptual contributions to the Helios scheme, the authors of [31] did not actually formally prove the security of their scheme. Furthermore, providing such proofs for KTV-Helios requires introducing new formal definitions for participation privacy as well as receipt-freeness: Although the existing formal definitions of ballot privacy can be extended and applied for evaluating participation privacy in some voting systems, no definition that addresses participation privacy specifically has been proposed, yet. The available definitions of receipt-freeness, on the other hand, do not fully encompass the available e-voting schemes and security models that ensure receipt-freeness.

The main contributions of our paper are new formal definitions for probabilistic participation privacy (see Section 3) and probabilistic receipt-freeness (Section 4), that we use to apply to KTV-Helios and evaluate its security claims. In order to evaluate the participation privacy and receipt-freeness of KTV-Helios we first prove that it ensures ballot privacy according to the definition in [8] in the random oracle model (Section 3.3.1).

2 DESCRIPTION OF KTV-HELIOS

We first describe the version of the *Helios* scheme, based upon the improvements in [9, 14, 15] that KTV-Helios extends upon. In this version, the eligible voters exchange their public signing keys with the registration authority, who then publishes these keys. In the setup phase, the tabulation tellers generate a pair of ElGamal keys used for encrypting the votes. During the voting, the voters encrypt and sign their chosen voting option, also computing the well-formedness proof[1]. The voters then have an option either to

ARES '17, Reggio Calabria, Italy
© 2017 ACM. 978-1-4503-5257-4/17/08…$15.00
DOI: 10.1145/3098954.3098990

[1] The proof consists of proving the plaintext knowledge, and in case homomorphic tallying is used, it also includes proving that the plaintext is one of the valid voting options.

verify that their vote has been encrypted correctly or to submit it to the bulletin board. During the tallying, the encrypted votes are anonymized, either via mix net shuffle or homomorphic tallying. The anonymized result is jointly decrypted by the tabulation tellers and published as the outcome of the election.

The basic idea of *KTV-Helios* is the introduction of so-called dummy ballots that are meant to obfuscate the presence of ballots cast by the voters[2]. The dummy ballots are cast on behalf of each voter by the new type of entity, the posting trustee[3], during the whole voting phase and are published next to that voter's name. Each dummy ballot consists of an encryption of a null vote accompanied with the well-formedness proof that is constructed in the same way as the proofs for non-dummy ballots. The well-formedness proof ensures that only the voter herself can cast non-dummy ballots. Before the tallying, for each voter the ballots that are published next to the voter's name are aggregated into the final ballot. Due to the homomorphic property of the cryptosystem, and due to the fact that the dummy ballots contain the encryption of a null vote, this final ballot encrypts the sum of all non-dummy votes cast by the voter. The final ballots of all voters are being anonymized via mix net shuffle. Afterwards, each anonymized ballot is either assigned to a valid voting option or discarded without revealing its value.

Similar to the proposal in [15], KTV-Helios achieves verifiability against malicious bulletin board and prevents ballot stuffing by publishing the identities of the eligible voters and using public-key infrastructure to authenticate the voters during voting. It, however, requires trusting the device that holds the private signing key of the voter. This assumption, however, might be realistically expected in some settings, e.g. in case of a national eID infrastructure with tamper-resistant smartcards. While the discussion of verifiability in KTV-Helios is out of scope for this paper, we provide the security model and the formal proofs in the full version of our paper [19]. Futhermore, the dummy ballots in KTV-Helios serve to achieve participation privacy and receipt-freeness.

In the subsections that follow we provide a formal description of KTV-Helios with more details.

2.1 Building Blocks of KTV-Helios

In this section, we describe the building blocks (i. e. the cryptographic primitives and the probability distributions) of the KTV-Helios scheme. The scheme uses the following *cryptographic primitives*:

- Signed ElGamal [9], a NM-CPA secure encryption scheme (the same one is used in Helios). Its algorithms are KeyGen, Enc, Dec. The encryption of a message $m \in \mathbb{Z}_q$ with a public key $(g, h) \in \mathbb{G}^2$ is $((g^r, g^m h^r), \pi_{PoK})$ where $r \leftarrow_{\$} \mathbb{Z}_q$ is randomly sampled and π_{PoK} is a Schnorr proof of knowledge of r. To decrypt a ciphertext $((c^{(1)}, c^{(2)}), \pi_{PoK})$ with a secret key sk, first check the PoK and if successful set $m = c^{(2)} \cdot (c^{(1)})^{(-sk)}$.
- An existentially unforgeable digital signature scheme consisting of algorithms SigKeyGen, Sign and Verify, for example Schnorr signatures.
- The Chaum-Pedersen NIZK proof $\mathsf{EqProof}(g_1, g_2, h_1, h_2)$ that proves the equality of discrete logarithms $\log_{g_1} h_1 = \log_{g_2} h_2$ as described in [11]. This proof can be simulated in the random oracle model, for which we write $\mathsf{SimEqProof}(g_1, g_2, h_1', h_2')$ (see e. g. [8]).
- A NIZK disjunctive proof $\mathsf{DisjProof}(\mathsf{pk}_{id}, \mathsf{sk}_{id'} \in \{\mathsf{sk}_{id}, 0\}, g_1, g_2, h_1, h_2, t)$ that given $(\mathsf{pk}_{id}, \mathsf{sk}_{id}) \leftarrow_{\$} \mathsf{SigKeyGen}$ and $g_1, g_2, h_1, h_2 \in \mathbb{G}_q$ and timestamp t proves either the knowledge of $s = \mathsf{Sign}(\mathsf{sk}_s, g_1 || g_2 || h_1 || h_2 || t)$[4], or the equality of discrete logarithms $\log_{g_1} h_1 = \log_{g_2} h_2$.
- A re-encryption mix-net for ElGamal ciphertexts $\mathsf{Mix}(c_1, ..., c_N)$, for example the one of Wikström and Terelius [47].
- A plaintext equivalence test (PET) to decrypt ElGamal ciphertexts. On input a ciphertext c, a secret key sk and a message m it creates a decryption factor d that is 1 if c is an encryption of m under sk and random in $\mathbb{Z}_q$ if not. It also creates a proof π_{PET} that it operated correctly (this is another Chaum-Pedersen EqProof).

The next building blocks are the *probability distributions*. They are used by the posting trustees in order to cast a random number of dummy ballots at random times next to each voter's id. In order to specify the dummy ballot casting algorithm for the posting trustee, we use two probability distributions $\mathbb{P}_d$ and $\mathbb{P}_t$. The first probability distribution $\mathbb{P}_d$ is used to sample a number of dummy ballots for each voter. This distribution therefore has a support $[x, y]$ with x, y as the minimal and maximal number of dummy ballots that the posting trustee is going to cast for each voter (i. e., $x \in \mathbb{N}_0$, $y \in \mathbb{N}_0 \cup \{\infty\}$). The parameters x and y, as well as the exact $\mathbb{P}_d$ needs to be defined by the election authorities when setting up a corresponding system, i. e. their optimal trade-off between security and efficiency[5]. The second probability distribution $\mathbb{P}_t$ is used to determine the time to cast each dummy ballot. Thus, this distribution has a support $[T_s, T_e]$ with T_s denoting the timestamp at the start of the voting phase and T_e the timestamp at the end of the voting phase. In order to obfuscate the ballots cast by voters, $\mathbb{P}_t$ should resemble the distribution of times at which the voters cast their ballots. For this, e. g. the information from the previous elections could be used.

2.2 Formal Description of KTV-Helios

We are now ready to provide the formal description of the KTV-Helios scheme. This description is based upon the syntax proposed in [8], adjusted to the context of the KTV-Helios scheme. For the sake of simplicity, we assume a single tabulation teller and a single posting trustee[6]. We first specify the various functions in place, i.e.:

- $\mathsf{RegisterVoter}(1^\lambda, id)$ is run by the voter id. The voter id generates a pair of keys $(\mathsf{pk}_{id}, \mathsf{sk}_{id}) \leftarrow_{\$} \mathsf{SigKeyGen}(1^\lambda)$ and sends the public key pk_{id} to the registration authority.

[2] A similar concept of dummy ballots has also been used in [45] which extends the JCJ/Civitas voting scheme [12]

[3] Note that the description in [31] assumes that each voter can take over the role of a posting trustee. In this paper, however, we consider it as a separate entity for the sake of formal proofs, and assume that such an entity would be appointed by the election organizers.

[4] Methods for proving the knowledge of a digital signatures via Σ-proof are described by Asokan et al. [4] for common signature schemes; the general method of constructing NIZK disjunctive proofs is described by Cramer et al. in [18].

[5] We provide further information on how the choice of $\mathbb{P}_d$ affects the security of the scheme in Sections 3 and 4.

[6] We discuss extending the proofs towards several of those entities in the full version of our paper [19].

• $\mathsf{RegisterRA}(1^{\lambda}, id, \mathsf{pk}_{id})$ is run by the registration authority. The registration authority adds (id, pk_{id}) to the list of registered voters' public keys I_{pk} if $id \in I$, and returns $\perp$ otherwise.

• $\mathsf{Setup}(1^{\lambda})$ is run by the tabulation teller. It runs $(\mathsf{pk}, \mathsf{sk}) = \mathsf{KeyGen}$ to create the election keys and returns the public key pk.

• $\mathsf{Vote}((id', \mathsf{sk}_{id'}), id, v, t)$ creates a ballot $b = (id, c, \pi_{PoK}, \pi, t)$ for voter $id \in I$ and voting option v that is cast at a timestamp[7] t. If $id = id'$ (a voter casting her own ballot) then it computes $(c, \pi_{PoK}) = \mathsf{Enc}(\mathsf{pk}, v)$ where $c = (c^{(1)}, c^{(2)})$ and $\pi = \mathsf{DisjProof}(\mathsf{pk}_{id}, \mathsf{sk}_{id'}, g, h, c^{(1)}, c^{(2)}, t)$ using a signature $\mathsf{Sign}(\mathsf{sk}_{id'}, g||h||c||t)$. If $id' = \hat{id}$ (the posting trustee is casting a ballot on behalf of voter id) then $\mathsf{sk}_{id'}$ is not required but v must be 0. Note that the challenges used in π_{PoK} and π should include the statements and commitments from both π_{PoK} and π in order to prevent that the voter signs and casts the ballot she did not compute herself.

• $\mathsf{Validate}(b)$ parses the ballot b as $(id, c = (c^{(1)}, c^{(2)}), \pi_{PoK}, \pi, t)$ and returns 1 if π and π_{PoK} are valid proofs, $id \in I$ and $t \in [T_s, T_e]$, and $\perp$ otherwise.

• $\mathsf{VerifyVote}(\mathsf{BB}, b)$ is used by the voter to ensure that her ballot b is properly stored on the bulletin board. It outputs 1 if $b \in \mathsf{BB}$ and $\mathsf{ValidateBB}(\mathsf{BB})$ holds, otherwise $\perp$.

• $\mathsf{VoteDummy}(id)$ is used by the posting trustee to cast dummy ballots for a given voter id. The posting trustee samples a random number $m \leftarrow_{\$} \mathbb{P}_d$ and random timestamps $t_1, ..., t_m \leftarrow_{\$} \mathbb{P}_t$, and returns a set of ballots

$$(\mathsf{Vote}((\hat{id}, 0), id, 0, t_1), ..., \mathsf{Vote}((\hat{id}, 0), id, 0, t_m))$$

• $\mathsf{Valid}(\mathsf{BB}, b)$ is run by the board before appending a new ballot. It checks that $\mathsf{Validate}(b) = 1$ and that the ciphertext c in b does not appear in any ballot already on the board. If this holds it returns 1, otherwise $\perp$.

• $\mathsf{ValidateBB}(\mathsf{BB})$ checks that a board is valid. It is run by the tabulation teller as part of the tallying process and by voters verifying the board. It creates an empty board B' and for each ballot $b \in \mathsf{BB}$ runs "if $\mathsf{Valid}(B', b)$ then append b to B'". If any ballot gets rejected it returns $\perp$, otherwise 1.

• $\mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$ is used by the tabulation teller to calculate the election result. It returns a tuple (R, Π) where R is the election result and Π is auxiliary data (proofs of correct tallying). In more detail:

(1) Run $\mathsf{ValidateBB}(\mathsf{BB})$ and return $\perp$ if this fails.
(2) Parse each ballot $b \in \mathsf{BB}$ as $(id, c, \pi_{PoK}, \pi, t)$.
(3) For each id appearing in the ballots, set $c_{id} = \prod_{c \in C(id)} c$ where $C(id)$ is the set of ciphertexts c in ballots belonging to voter id.
(4) Mix the ballots $(c_1, \ldots, c_N)$ (where N is the number of distinct identities who cast a ballot) to get a new list of ballots $(\bar{c}_1, \ldots, \bar{c}_N)$ and a proof π_{mix} of correct mixing.
(5) For each $i \in \{1, \ldots, N\}$ and each valid voting option $v \in \mathbb{V}_{valid}$, use the PET to create a decryption factor $d_{i,v}$ and proof $\pi_{PET,i,v}$.
(6) The result R is the number of votes for each voting option, i. e. $R(v) = |\{i : d_{i,v} = 1\}|$ for all $v \in \mathbb{V}_{valid}$. The auxiliary data Π contains the mixing proofs π_{mix}, the mixed ciphertexts $(\bar{c}_1, \ldots, \bar{c}_N)$, the decryption factors $d_{i,v}$ and the PET proofs $\pi_{PET,i,v}$ for $i \in \{1, \ldots, N\}$ and $v \in \mathbb{V}_{valid}$.

• $\mathsf{ValidateTally}(\mathsf{BB}, (R, \Pi))$ takes a bulletin board BB and the output (R, Π) of Tally and returns 1 if $\mathsf{ValidateBB}(\mathsf{BB}) = 1$ and all the proofs π_{mix} and π_{PET} are valid, otherwise $\perp$. It is used to verify an election.

These functions are combined in order to build the KTV Helios scheme. The corresponding description of the KTV Helios scheme is given in the following paragraphs along the line of the three phases of an election.

Setup phase: The election organizers set up an empty bulletin board BB and publish a set of valid non-null voting options $\mathbb{V}_{valid} = (v_1, ..., v_L)$ with $0 \notin \mathbb{V}_{valid}$. If there is no existing PKI encompassing the eligible voters, the eligible voters from the voting register I register themselves by running $\mathsf{RegisterVoter}(1^{\lambda}, id)$. After the voters have registered, or if there is an existing PKI already established among the voters, the registration authority prepares the list of the eligible voters' public keys by running $\mathsf{RegisterRA}(id, \mathsf{pk}_{id})$ for each voter id and publishing the list $I_{pk} = \{(id_1, \mathsf{pk}_{id_1}), ..., (id_N, \mathsf{pk}_{id_N})\}$. The tabulation teller runs $\mathsf{Setup}(1^{\lambda})$.

Voting phase: The posting trustee runs $\mathsf{VoteDummy}(id)$ for each registered eligible voter $id \in I$. The posting trustee then submits each resulting dummy ballot $b = (id, c, \pi_{PoK}, \pi, t)$ to the bulletin board at a time corresponding to the timestamp t. The bulletin board appends b to BB. The voter id runs $\mathsf{Vote}((id, \mathsf{sk}_{id}), id, v, t)$ in order to cast her ballot for a voting option v at a time denoted by timestamp t. The bulletin board appends b to BB. The voter can run $\mathsf{VerifyVote}(\mathsf{BB}, b)$ to check whether her ballot is properly stored.

Tallying phase: The tabulation teller runs $\mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$ on the contents of the bulletin board, and publishes the resulting output (R, Π). Everyone who wants to verify the correctness of the tally runs $\mathsf{ValidateTally}(\mathsf{BB}, (R, \Pi))$.

3 PARTICIPATION PRIVACY

In this section we provide a general definition of participation privacy and apply to KTV-Helios.

3.1 Defining (δ, k)-Participation Privacy

We first describe the idea and the intuition behind our definition, followed by the definition itself.

3.1.1 Definition Idea. Since one may consider participation privacy an extension of vote privacy, seeing abstention as one of the possible voting options, we decided to consider modifying an existing definition of vote privacy for defining participation privacy. As such, our definition of participation privacy is inspired by the idea of vote swapping that has been used, in particular, in [6] to provide a game-based definition of vote privacy. The vote swapping approach considers two voters, id_0 and id_1 and two different votes v_0 and v_1, so that the adversary has to distinguish between the election where id_0 votes for v_0 and id_1 votes for v_1, or vice versa. While more advanced definitions for vote privacy have been developed (see [8] for an overview), the concepts that they use would not be suitable for defining participation privacy, since the techniques that obfuscate the content of the ballot (i.e. encryption) are generally

[7] As the timestamp t denotes the time at which b is submitted to the bulletin board, we assume that it is chosen in $[T_s, T_e]$.

different from the techniques that obfuscate the identities of the voters who cast their ballots. Hence, based on the vote swapping idea, we consider *voter swapping* in our definition: given two voters id_0, id_1, the adversary should be unable to distinguish whether id_0 has abstained and id_1 participated in the election, or vice versa.

According to our definition, a voting system that reveals nothing but the number of voters who participated in the election and the election result ensures participation privacy. Note that such a scenario is often the case in practice, in both Internet voting and traditional elections. While other voting systems might either refuse to publish anything but the name of the winner, or encode the votes in such a way, that the presentation of the final result does not reveal the number of the voters who cast their ballot, these systems are out of scope of this work.

In order to enable the evaluation of participation privacy in KTV-Helios, we propose a quantitative definition, inspired by the coercion resistance definition in [34] and the verifiability definition in [16]. Similar to the notion of (γ_k, δ)-verifiability with quantitative goal γ_k in [16], we speak of (δ, k)-participation privacy, where δ denotes the advantage of the adversary who tries to tell whether a given voter has abstained from casting her ballot in the election, or cast her ballot at most k times.

3.1.2 Definition of (δ, k)-Participation Privacy. We consider the following experiment $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$ given the adversary $\mathcal{A} \in C_S$, so that C_S is a set of PPT adversaries, defined according the adversarial model for a particular scheme. There are two bulletin boards BB_0, BB_1 that are set up by the challenger. The adversary only sees the public output for one of these bulletin boards BB_β, $\beta \leftarrow\!\$ \{0, 1\}$. Let Q_S be a set of oracle queries which the adversary has access to. Using these queries, the adversary fills both of the bulletin boards with additional content modeling the voting, so that BB_0 and BB_1 contain the same cast ballots except for the ballots for the voters id_0, id_1: given a number of voting options $v_1, ..., v_{k'}$ chosen by the adversary, $k' \leq k$, for each $i = 0, 1$, the bulletin board BB_i contains the votes for $v_1, ..., v_{k'}$ on behalf of id_i and an abstention from the election is modeled for the voter id_{1-i}.

The oracle computes the tally result R on BB_0. In case a voting scheme provides auxiliary output Π for the tally, the oracle returns (R, Π) in case $\beta = 0$, and simulates the auxiliary output $\Pi' = \mathsf{SimProof}(\mathsf{BB}_1, R)$, returning the tuple (R, Π') in case $\beta = 1$[8]. The oracle further outputs the public content of BB_β to the adversary. The goal of the adversary is to guess whether the provided output corresponds to BB_0 or to BB_1, i.e. to guess β.

The definition of (δ, k)-participation privacy is then as follows:

Definition 3.1. The voting scheme $\mathcal{S}$ achieves (δ, k)-participation privacy given a subset of PPT adversaries C_S, if for any adversary $\mathcal{A} \in C_S$, $k \in \mathbb{N}$ and two honest voter id_0, id_1 holds

$$|\Pr\left[\mathsf{Exp}^{\mathsf{ppriv},0}_{\mathcal{A},\mathcal{S},k} = 0\right] - \Pr\left[\mathsf{Exp}^{\mathsf{ppriv},1}_{\mathcal{A},\mathcal{S},k} = 0\right] - \delta|$$

is negligible in the security parameter.

3.2 Instantiating (δ, k)-Participation Privacy in the KTV-Helios Scheme:

In order to evaluate (δ, k)-participation privacy in the KTV-Helios scheme according to the aforementioned definition, we first need to specify the adversary $\mathcal{A} \in C_S$ we aim to protect against.

We make following assumptions regarding adversarial capabilities: the tabulation teller is trustworthy, both the voting and the verification device are trustworthy, the adversary does not observe the communication channel between the voter, the posting trustee and the voting system, the posting trustee is trustworthy, the bulletin board with which the voter communicates is trustworthy, the honest voters (aside from id_0 and id_1 in $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$) decide to participate or to abstain in the election independently from each other, the adversary is computationally restricted and the voters are not actively trying to prove that they abstained due to coercion.

We define C_S as a set of adversaries that are given access to the queries $Q_S = \{\mathcal{O}\mathsf{Cast}, \mathcal{O}\mathsf{VoteAbstain}, \mathcal{O}\mathsf{VoteLR}, \mathcal{O}\mathsf{Tally}\}$ in the experiment $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$. These queries are defined as follows:

```
$\mathcal{O}$VoteAbstain($v_1, \ldots, v_{k'}$):
if $k' > k$ then
    return $\perp$
endif
$b_{0,1}, \ldots, b_{0,m_0} \leftarrow\!\$$ VoteDummy($id_0$)
$b_{1,1}, \ldots, b_{1,m_1} \leftarrow\!\$$ VoteDummy($id_1$)
for $j = 1, \ldots, k'$ do
    $t'_j \leftarrow\!\$ \mathbb{P}_t$
    $b'_{0,j}$ = Vote(($id_\beta$, sk$_{id_0}$), $id_0$, $v_j$, $t'_j$)
    $b'_{1,j}$ = Vote(($id_\beta$, sk$_{id_1}$), $id_1$, $v_j$, $t'_j$)
endfor
Append $b_{0,1}, \ldots, b_{0,m_0}$ to BB$_0$
Append $b_{1,1}, \ldots, b_{1,m_1}$ to BB$_1$
Append $b'_{0,1}, \ldots, b'_{0,k'}$ to BB$_0$
Append $b'_{1,1}, \ldots, b'_{1,k'}$ to BB$_1$
```

```
$\mathcal{O}$Cast($b$):
if Valid(BB$_\beta$, $b$) then
    Append $b$ to BB$_0$
    Append $b$ to BB$_1$
endif
```

```
$\mathcal{O}$VoteLR($id$, $v_0$, $v_1$, $t$):
$b_0$ = Vote(($id$, sk$_{id}$), $id$, $v_0$, $t$)
$b_1$ = Vote(($id$, sk$_{id}$), $id$, $v_1$, $t$)
if Valid(BB$_\beta$, $b_\beta$) = 0 then
    return $\perp$
endif
Append $b_0$ to BB$_0$
Append $b_1$ to BB$_1$
```

```
$\mathcal{O}$Tally():
if $\beta = 0$ then
    return Tally(sk, BB$_0$)
else
    ($R$, $\Pi$) = Tally(sk, BB$_0$)
    $\Pi'$ = SimTally(BB$_1$, $R$)
endif
return ($R$, $\Pi'$)
```

The adversary may query $\mathcal{O}\mathsf{Tally}$ and $\mathcal{O}\mathsf{VoteAbstain}$ only once.

3.3 Proving (δ, k)-participation privacy for KTV-Helios

We further use the definition of (δ, k)-participation privacy and its instantiation for KTV-Helios for the evaluation of participation privacy. Namely, given k ballots, our goal is to find δ, so that KTV-Helios satisfies (δ, k)-participation privacy against an adversary with access to the queries in Q_S. In this we proceed as follows. First, we consider the fact that the participation privacy in KTV-Helios relies on the inability of the adversary to distinguish between dummy ballots and non-dummy ballots cast by the voters. Hence, as the dummy ballots encrypt a null-vote, as opposed to non-dummy ballots, the participation privacy is strongly connected to the ballot privacy in KTV-Helios. Therefore, we first define and prove the fulfillment of ballot privacy in KTV-Helios in Section 3.3.1. Afterwards, we consider further sources of information that can be used

[8] The tally result should be the same, if the vote of each voter is equally included in the result. However, in order to be able to model the voting schemes where the weight of the vote might depend on the voter's identity, we chose to simulate the auxiliary output in our definition.

by the adversary to win $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$, namely, the number of ballots near the voters name. Finally, we determine the value of δ so that KTV-Helios ensures (δ, k)-participation privacy for a given k and provide a corresponding proof.

3.3.1 *Ballot Privacy.*

We show that KTV-Helios has ballot privacy and two auxiliary properties called strong correctness and strong consistency in the sense of [8]. Ballot privacy is the property that an adversary cannot learn more from the election data than from the election result alone. Here the adversary may be an observer or a coalition of a subset of voters and the election data is the bulletin board with voters' ballots and any auxiliary data published by the election officials such as proofs of correct tallying. We assume as in [8] that both the tabulation teller and the bulletin board that the voter communicates with are trustworthy, the voting device does not leak private information and the adversary is computationally restricted.

The ballot pricacy (BPRIV) notion in [8] is a security experiment with two bulletin boards, one of which (chosen at random by sampling a bit β) is shown to the adversary. For each voter, the adversary may either cast a ballot themselves or ask the voter to cast one of two votes v_0, v_1 of the adversary's choice. In this case a ballot for v_0 is sent to the first board and a ballot for v_1 is sent to the second board. The adversary thus sees either a ballot for v_0 or a ballot for v_1; in a scheme with privacy a PPT adversary must be unable to distinguish the two cases with more than a negligible advantage[9].

Further, the adversary can close the election and ask for the result. Here we cannot simply tally the visible board as the results on the two boards may differ which would trivially let the adversary distinguish. Instead, [8] says that the adversary shall always see the result for the first board. If the first board is the visible one, the experiment tallies it normally; if the second board is visible then the experiment tallies the first board to get the election result and provides simulated election data (e. g. proofs of correct tallying) to make it seem that the second board (which the adversary can see) tallies to the result of the first board. A scheme is BPRIV secure if there is an algorithm SimTally that can provide simulated election data such that no PPT (probabilistic polynomial time) adversary can guess better than at random whether they saw the first or the second board in this experiment. We give the BPRIV experiment with minor syntax adjustments, e. g. our Vote algorithm takes a voter private signing key and a timestamp too.

Definition 3.2. A voting scheme $\mathcal{S}$ has ballot privacy (BPRIV) if there is a PPT algorithm SimProof such that for every PPT adversary $\mathcal{A}$ the following quantity is negligible in the security parameter

$$\mathsf{Adv}^{\mathsf{bpriv}}_{\mathcal{A},\mathcal{S}} := \left|\Pr\left[\mathsf{Exp}^{\mathsf{bpriv},0}_{\mathcal{A},\mathcal{S}} = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{bpriv},1}_{\mathcal{A},\mathcal{S}} = 1\right]\right|$$

Where the BPRIV experiment is defined as follows and BB_β is visible to $\mathcal{A}$

```
$\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$:
$(\mathsf{pk}, \mathsf{sk}) = \mathsf{Setup}()$
$I$ = register of voters
initialize $\mathsf{BB}_0$, $\mathsf{BB}_1$
$g = \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, I)$
```

```
$\mathcal{O}\mathsf{Cast}(b)$:
if $\mathsf{Valid}(\mathsf{BB}_\beta, b)$ then
    Append $b$ to $\mathsf{BB}_0$
    Append $b$ to $\mathsf{BB}_1$
endif
```

```
$\mathcal{O}\mathsf{VoteLR}(id, v_0, v_1, t)$:
$b_0 = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_0, t)$
$b_1 = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_1, t)$
if $\mathsf{Valid}(\mathsf{BB}_\beta, b_\beta) = 0$ then
    return $\perp$
endif
Append $b_0$ to $\mathsf{BB}_0$
Append $b_1$ to $\mathsf{BB}_1$
```

```
$\mathcal{O}\mathsf{tally}()$:
if $\beta = 0$ then
    return $\mathsf{Tally}(\mathsf{sk}, \mathsf{BB}_0)$
else
    $(R, \Pi) = \mathsf{Tally}(\mathsf{sk}, \mathsf{BB}_0)$
    $\Pi' = \mathsf{SimTally}(\mathsf{BB}_1, R)$
endif
return $(R, \Pi')$
```

KTV-Helios has ballot privacy. Since KTV decrypts ballots with PETs, the ballot privacy proof is actually easier than for existing Helios. The SimTally algorithm checks the board, sums the ballots for each voter and mixes them just like Tally. The result R that SimTally takes as input shows how many votes it needs to simulate for each valid choice $v \in V$ so it makes a list L containing a random permutation of these votes. It then produces simulated decryption factors $d_{i,j}$ which are 1 if $L[i] = j$ and random in $\mathbb{Z}_q$ otherwise. Here i ranges over the ballots output by the mix and j ranges over votes $v \in V$. Since the encryption scheme is NM-CPA, a PPT adversary cannot tell real from simulated decryption factors. The EqProof proofs for the PET are Chaum-Pedersen proofs which SimTally can simulate (in the random oracle model) for any inputs. The full proof is provided in the full version of our paper [19].

Strong correctness and strong consistency. These two properties from [8] prevent an adversary from breaking privacy by encoding instructions in its own ballots on the board. Strong correctness requires that an adversary cannot manipulate the board such that the board would reject honest ballots and strong consistency ensures that ballots are independent in the sense that one ballot (cast by the adversary) cannot influence how another ballot (from a honest voter) is counted.

For both security games we need to make minor syntactical changes to include e.g. timestamps in ballots. The proofs of both properties are then routine and reduce to observing that (1) ballot-weeding with Validate and Valid do not use the private election key, so they cannot depend on the votes encrypted in the ballots and (2) the tallying algorithm works correctly. The full proofs are again in the full version of our paper [19].

3.3.2 *Number of Ballots.*

One source of information that can be used by the adversary to win in $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$ are the $k' \leq k$ additional ballots on BB_1 as the output of $\mathcal{O}\mathsf{VoteAbstain}$. In order to account for the adversarial advantage gained from this number, we define the following experiment $\mathsf{Exp}^{\mathsf{num},\beta}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t,k'}$: the challenger chooses a random $\beta \in 0, 1$. She then outputs two numbers m_0, m_1, so that $m_\beta = m + k'$, with $m \leftarrow\$ \mathbb{P}_d$, and $m_{1-\beta} \leftarrow\$ \mathbb{P}_d$. The oracle additionally returns the set of timestamps $t_1, ..., t_{m_0+m_1}$ that are independently sampled from $\mathbb{P}_t$ to the adversary. Hence, the experiment models the number of ballots next to id_0, id_1 in the election in which the voter $id_{1-\beta}$ abstains and the voter id_β casts k' ballots. The adversary has to guess β. Let $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}$ denote an advantage in this experiment, so that $|\Pr\left[\mathsf{Exp}^{\mathsf{num},0}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t,k} = 0\right] - \Pr\left[\mathsf{Exp}^{\mathsf{num},1}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t,k} = 0\right] - \delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}|$ is negligible.

[9] To be precise, the adversary outputs a guess g for β and the probability that $g = \beta$ must not be more than $1/2 + \varepsilon$ for some negligible ε.

3.3.3 *Determining the Optimal Value for δ.*

We are now ready to determine an optimal value δ, so that the KTV-scheme achieves (δ, k)-participation privacy, but does not achieve (δ', k)-participation privacy for any lower values of δ'.

THEOREM 3.3. *KTV-Helios, instantiated with the probability distributions $\mathbb{P}_d, \mathbb{P}_t$ achieves (δ, k)-participation privacy for a given $k > 0$ given the subset of adversaries C_S, with $\delta = \max_{k' \leq k} \delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$. It further does not achieve (δ', k)-participation privacy for any $\delta' < \delta$.*

We base our proof on the idea that the the number of ballots next to id_0 and id_1 is the only thing that give advantage to the adversary. The rest of the public election data does not provide any advantage to the adversary. Our proof strategy is as follows. We consider a sequence of games that starts from $\mathsf{Exp}^{\mathsf{ppriv},0}_{\mathcal{A},\mathcal{S}k}$ and ends with $\mathsf{Exp}^{\mathsf{ppriv},1}_{\mathcal{A},\mathcal{S},k}$ and show that the adversary $\mathcal{A}$ with the access to the queries in Q_S distinguishes the transition through all those games with the advantage of at most $\delta := \max_{k' \leq k} \delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$. We define $\mathsf{BB}_{0,i}$ as the content of the bulletin board and (R_i, Π_i) as the tally output at the end of the game G_i, $i = 1, ..., 4$. The game sequence is as follows:

- G_1. The first game G_1 is equivalent to the experiment $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$ with $\beta = 0$, and $v_1, ..., v_{k'} \neq 0$ (hence, it is equivalent to the election where the voter id_0 abstains, and the voter id_1 casts $k' \leq k$ ballots with the votes $v_1, ..., v_{k'}$). Thus, the content of $\mathsf{BB}_{0,1}$ and the tally output (R_1, Π_1) correspond to the content of BB_0 and the output of $\mathcal{O}\mathsf{Tally}$ at the end of $\mathsf{Exp}^{\mathsf{ppriv},0}_{\mathcal{A},\mathcal{S},k}$.
- G_2. The second game G_2 is equivalent to the election, where the voter id_0 abstains, and the voter id_1 casts $k' \leq k$ ballots with null-votes. The contents of the bulletin board $\mathsf{BB}_{0,2}$ is equivalent to the content of the bulletin board BB_1 at the end of $\mathsf{Exp}^{\mathsf{ppriv},1}_{\mathcal{A},\mathcal{S},k}$ for the adversary using the query $\mathcal{O}\mathsf{VoteAbstain}(v_1, ..., v_{k'})$ with $v_l = 0\ \forall l = 1, ..., k'$. The tally result R, however, is calculated on the contents of the bulletin board $\mathsf{BB}_{0,1}$ in the game G_1, and the auxiliary output Π_2 is simulated as $\Pi_2 = \mathsf{SimProof}(R_1, \mathsf{BB}_{0,2})$.

We prove, that the adversarial advantage in distinguishing between the output of G_1 and G_2 is at most the adversarial advantage in the ballot privacy experiment (Section 3.3.1). Consider an adversary $\mathcal{B}$ in the ballot privacy experiment $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$, who simulates the games G_1 and G_2 for the adversary $\mathcal{A}$. The adversary $\mathcal{B}$ returns the output of $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$ for the queries $\mathcal{O}\mathsf{Cast}$ and $\mathcal{O}\mathsf{Tally}$. For simulating the output of $\mathcal{O}\mathsf{VoteAbstain}(v_1, ..., v_{k'})$, $\mathcal{B}$ proceeds as follows: First, she simulates the dummy ballots for each voter id_i, $i \in \{0, 1\}$ by choosing a random values $m_i \leftarrow\!\!\$\, \mathbb{P}_d$, and a set of random timestamps $t_1, ..., t_{m_i} \leftarrow\!\!\$\, \mathbb{P}_t$. The dummy ballots $b_{i,1}, ..., b_{i,m_i}$ are computed as $b_{i,j} = \mathsf{Vote}((\hat{id}, 0), id_i, 0, t_j)$, $j = 1, ..., m_i$. Aftwewards, she simulates casting the votes $v_1, ..., v_{k'}$: For each of the votes v_l, $l = 1, ..., k'$, she uses the query $\mathcal{O}\mathsf{VoteLR}(id_1, id_1, 0, v_l, t)$ for a random $t_l \in \mathbb{P}_t$ in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$. The output of the queries $\mathcal{O}\mathsf{VoteLR}$ and the dummy ballots $b_{i,1}, ..., b_{i,m_i}$ is returned to $\mathcal{A}$. At the end, $\mathcal{B}$ returns the value β output by $\mathcal{A}$ as the guess in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$. Thus, it follows that the adversarial advantage in distinguishing G_1 from G_2 is at most equal to the adversarial advantage in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$, denoted as δ_{BPRIV}.

- G_3. The third game G_3 is equivalent to the election, where the voter id_0 casts $k' \leq k$ ballots with null-vote, and the voter id_1 abstains from the election. Namely, the content of the bulletin board $\mathsf{BB}_{0,3}$ is equivalent to the content of the bulletin board BB_1 at the end of $\mathsf{Exp}^{\mathsf{ppriv},1}_{\mathcal{A},\mathcal{S},k}$ for the adversary using the query $\mathcal{O}\mathsf{VoteAbstain}(v_1, ..., v_{k'})$ with $v_l = 0\ \forall l = 1, ..., k'$, $k' \leq k$. The tally outputs the result R_1 computed on $\mathsf{BB}_{0,1}$ and simulated auxilary data $\Pi_3 = \mathsf{SimProof}(R2, \mathsf{BB}_{0,3})$.

We prove, that the adversary has an advantage of $\max_{k' \leq k} \delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$ of distinguishing between the output of G_2 and G_3. The tally result does not change, hence the tally output (R_1, Π_2) is equivalent to the tally output (R_1, Π_3). The only difference between the contents of $\mathsf{BB}_{0,1}$ and $\mathsf{BB}_{0,2}$ is the presence of k' additional ballots with the encryption of 0 on $\mathsf{BB}_{0,3}$. Therefore, we conclude that the challenge in distinguishing between the outputs of G_2 and G_3 is equivalent to the challenge in distinguishing between the output of $\mathsf{Exp}^{\mathsf{num},0}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t,k'}$ and $\mathsf{Exp}^{\mathsf{num},1}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t,k'}$ for every $k' \leq k$ chosen by the adversary, and therefore the adversarial advantage of distinguishing between the output of G_1 and G_2 is at most $\max_{k' \leq k} \delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$.

- G_4. The fourth game G_4 is equivalent to the election where the voter id_0 casts k' ballots with the votes $v_1, ..., v_{k'}$, and the voter id_1 abstains. The tally is computed on $\mathsf{BB}_{0,1}$, and the auxiliary output is simulated as $\Pi_4 = \mathsf{SimProof}(R_1, \mathsf{BB}_{0,4})$. Following the argument for the indistinguishability of G_1 and G_2, it holds that adversary distinghuishes between the outputs of G_3 and G_3 with the same advantage δ_{BPRIV} as in the ballot privacy experiment.

It follows, that the in transition through the game sequence $G_1 \to G_2 \to G_3 \to G_4$, the outputs of each game are distinguished from the outputs of a previous game with the advantage either δ_{BPRIV} (for the games G_1 and G_2, and for the games G_3 and G_4) or $\delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$ for $k' \leq k$ (for the games G_1 and G_2). Since δ_{BPRIV} is negligible, as proven in Section 3.3.1, it holds that the adversary distinguishes between the output in $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},k}$ with the advantage only negligibly larger than $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}$ for each $k' < k$ that she chooses in the experiment. Thus, given that an adversary chooses k' so that $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t} \geq \delta_{num,k''}\ \forall k'' \neq k', k'' \leq k$, the adversarial advantage in $\mathsf{Exp}^{\mathsf{ppriv},\beta}_{\mathcal{A},\mathcal{S},k}$ is negligibly larger than $\delta_k := \max_{k' \leq k} \delta^{\mathsf{num}}_{k',\mathbb{P}_d,\mathbb{P}_t}$. □

We provide an example of how to quantify (δ, k)-participation privacy given a particular distribution for the number of dummy ballots $\mathbb{P}_d$. Let $\mathbb{P}_d$ be a geometric distribution with the parameter $p \in (0, 1]$, so that the probability $\Pr[X = m] = (1 - p)^m p$ for $m \geq 0$ and $\Pr[X = m] = 0$ for $m < 0$. Since the probability distribution for times of casting the dummy ballots corresponds to the distribution of times at which the voters cast their ballots, the timestamps on the ballots do not provide any additional information to the adversary. Hence, we only consider the adversary seeing the total number of cast ballots next to the voter.

Let $k > 0$, $M_c \subset \mathbb{N}_0^2$ be a set of all pairs (m_0, m_1) output in $\mathsf{Exp}^{\mathsf{num},\beta}_{\mathcal{A},k}$, for which an adversary guesses $\beta = 0$ (i.e. that $m_0 = m + k$ with $m \leftarrow\!\!\$\, \mathbb{P}_d$, $m_1 \leftarrow\!\!\$\, \mathbb{P}_d$. It holds for $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}$ as defined in Section 3.1.2:

$$\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t} := \Pr\left[\mathsf{Exp}^{\mathsf{num},0}_{\mathcal{A},k} = 0\right] - \Pr\left[\mathsf{Exp}^{\mathsf{num},1}_{\mathcal{A},k} = 0\right]$$
$$= \sum_{(m_0,m_1)\in M_c} \Pr[X = m_0 - k] \cdot \Pr[X = m_1]$$
$$- \Pr[X = m_0] \cdot \Pr[X = m_1 - k]$$

Let $M_+ := \{(m_0, m_1) \in \mathbb{N}_0^2 : \Pr[X = m_0 - k] \cdot \Pr[X = m_1] - \Pr[X = m_0] \cdot \Pr[X = m_1 - k] > 0\}$. It further holds,

$$\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t} \geq \sum_{(m_0,m_1)\in M_+} \Pr[X = m_0 - k] \cdot \Pr[X = m_1]$$
$$-\Pr[X = m_0] \cdot \Pr[X = m_1 - k] = \sum_{m_1=0}^{k-1} (1-p)^{m_1} p \sum_{m_0=0}^{\infty} (1-p)^{m_0} p$$
$$= 1 - (1-p)^k$$

It further follows, that an adversary who is instructed to always output $\beta = 0$ if for the output pair (m_0, m_1) if it holds that $\Pr[X = m_0 - k] \cdot \Pr[X = m_1] - \Pr[X = m_0] \cdot \Pr[X = m_1 - k] > 0$, guesses β correctly with an advantage of $1-(1-p)^k$. Hence, it holds that $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t} = 1-(1-p)^k$. It further holds, that $\max_{k' \leq k} \delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t} = \delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}$. Thus, the KTV-Helios scheme with $\mathbb{P}_d$ as a geometric distribution with parameter p achieves (δ, k)-participation privacy with $\delta = 1 - (1-p)^k$.

4 RECEIPT-FREENESS

In this section we provide the definition for δ-receipt-freeness for deniable vote updating and apply it to evaluate KTV-Helios.

4.1 Defining δ-Receipt-Freeness

As in Section 3, we start with describing the idea and the intuition behind our definition, and then provide the definition itself.

4.1.1 Definition Idea.

The KTV-Helios scheme ensures probabilistic receipt-freeness via deniable vote updating. The principle of deniable vote updating has also been proposed in other e-voting schemes [1, 36, 37] in order to prevent a voter from constructing receipts that show how the voter has voted. As such, the voter can cast her ballot for the voting option the adversary instructs to vote for, but due to deniable vote updating the voter can change her vote without the adversary knowing it. The variant of deniable vote updating used in KTV-Helios is also characterized by enabling the so-called preliminary deniable vote updating. Given two ballots $b_{\mathcal{A}}$, b_v, with $b_{\mathcal{A}}$ as the ballot with the vote for a candidate demanded by the adversary, and b_v the ballot that ”updates” $b_{\mathcal{A}}$ to a vote for a candidate chosen by the voter, the voter can cast $b_{\mathcal{A}}$ and b_v in any order. This approach prevents an attack, where the voter succeeds to cast $b_{\mathcal{A}}$ as the last ballot in the election, ensuring that her vote has not been updated. However, in KTV-Helios, constructing b_v requires the knowledge of a vote that was cast with $b_{\mathcal{A}}$.

We propose a formal definition for probabilistic receipt-freeness for e-voting schemes with deniable vote updating. Our definition is inspired by the definition of coercion resistance by Kuesters et al. in [34] and the definition of receipt-freeness by Cortier et al. [10, 13]. As such, we introduce a game-based definition based on [13] and modified for the support of deniable vote updating. Similar to [34], we employ the δ-notation in order to denote an adversarial advantage δ in finding out whether the voter indeed voted as instructed by the adversary, or whether she faked the receipt and voted for another voting option. Furthermore, similar to [34], we consider vote buying from a single voter, while considering an extension towards multiple voters in future work.

Note that the definition in [10, 13] argues that the receipt-freeness should not rely on the actions of the voter, the so-called “counter-strategy”, that the voter should apply in order to fake her receipt while still voting how she wants to. However, previous research on receipt-freeness (see e.g. [32]) also considers a different approach on whether receipt-freeness should include counter-strategies or not. Hence, we agree that our definition describes a weaker version of receipt-freeness, which is ensured in KTV-Helios and other schemes that rely on deniable vote updating.

Intuitively, the definition encompasses the scenario of vote selling, whereby the adversary tells the voter the name of the candidate the voter has to provide a receipt for, and the voter is able to access the randomness used in creating an adversarial ballot $b_{\mathcal{A}}$. It, however, does not cover the scenarios where the adversary wants to make sure the voter did not cast a valid vote in the election, or to change the voter's vote to a random candidate (forced abstention and randomization as described in [26]). It also does not consider the information leakage from the election result.

4.1.2 Definition of δ-Receipt-Freeness.

We adjust the definition by Cortier et al. by enabling the voter to apply a counter-strategy against an adversary that demands a receipt, namely, to deniably update her vote. The receipt-freeness in our definition relies on the existence of following algorithms:

- $\mathsf{DeniablyUpdate}(id, \mathsf{sk}_{id}, v_0, v_1, t_v)$ as the function for casting a ballot that changes the vote of the voter id from v_0 to v_1. The function further takes as input the voter's private signing key sk_{id} and the timestamp at which the updating ballot is cast.
- $\mathsf{Obfuscate}(id)$ as the function used by the voting system for hiding the presence of ballots cast by the voter id for the purpose of deniable vote updating.
- $\mathsf{SimProof}(\mathsf{BB}, R)$ as the function for simulating the proof of correct tallying given the ballots published on the bulletin board BB and the tally result R.

We define an experiment $\mathsf{Exp}^{\mathsf{rfree},\beta}_{\mathcal{A},\mathcal{S}}$ for a voting scheme $\mathcal{S}$ as follows. The challenger sets up two bulletin boards BB_0, BB_1 by running the setup as described Section 2.2 and randomly chooses $\beta \in \{0, 1\}$, so that the adversary only sees BB_β. The adversary has access to the following queries:

```
$\mathcal{O}\mathsf{Receipt}(id, v_0, v_1, t)$:
if $v_0 \notin \mathbb{V}_{valid}$ or $v_1 \notin \mathbb{V}_{valid}$ then
    return $\perp$
endif
$b_{\mathcal{A}} = \mathsf{Vote}(id, \mathsf{sk}_{id}, v_0, t)$
Append $b_{\mathcal{A}}$ to $\mathsf{BB}_0$
Append $b_{\mathcal{A}}$ to $\mathsf{BB}_1$
$t_v \leftarrow\$ \mathbb{P}_t$
$b_v = \mathsf{DeniablyUpdate}(id, \mathsf{sk}_{id}, v_0, v_1, t_v)$
Append $b_v$ to $\mathsf{BB}_1$
$\mathsf{Obfuscate}(BB_0, id)$
$\mathsf{Obfuscate}(BB_1, id)$
```

```
$\mathcal{O}\mathsf{VoteLR}(id, v_0, v_1, t)$:
$b_0 = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_0, t)$
$b_1 = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_1, t)$
if $\mathsf{Valid}(\mathsf{BB}_\beta, b_\beta) = 0$ then
    return $\perp$
endif
Append $b_0$ to $\mathsf{BB}_0$
Append $b_1$ to $\mathsf{BB}_1$
```

```
$\mathcal{O}\mathsf{Tally}()$:
if $\beta = 0$ then
    return $\mathsf{Tally}(\mathsf{sk}, \mathsf{BB}_0)$
else
    $(R, \Pi) = \mathsf{Tally}(\mathsf{sk}, \mathsf{BB}_0)$
    $\Pi' = \mathsf{SimTally}(\mathsf{BB}_1, R)$
endif
return $(R, \Pi')$
```

The oracle also fills both of the bulletin boards with the content on behalf of honest voters and honest voting system entities. At the end of an experiment, the adversary outputs her guess for β.

We now define δ-receipt-freeness for deniable vote updating:

Definition 4.1. The voting scheme $\mathcal{S}$ achieves δ-receipt-freeness, if there are algorithms SimProof, DeniablyUpdate, Obfuscate so that holds

$$|\Pr\left[\mathsf{Exp}^{\mathsf{rfree},0}_{\mathcal{A},\mathcal{S}} = 0\right] - \Pr\left[\mathsf{Exp}^{\mathsf{rfree},1}_{\mathcal{A},\mathcal{S}} = 0\right] - \delta|$$

is negligible in the security parameter.

4.2 Instantiating δ-Receipt-Freeness in KTV-Helios

We accept following assumptions on adversarial capabilities for receipt-freeness in KTV-Helios: the tabulation teller is trustworthy, both the voting and the verification devices are trustworthy, the adversary does not observe the communication channel between the voter, the posting trustees and the voting system, the posting trustee is trustworthy, the bulletin board with which the voter communicates is trustworthy, the adversary is computationally restricted, the voter can cast a ballot without being observed by the adversary and the voters who are required by the adversary to provide receipts act independent from each other.

In order to evaluate δ-receipt-freeness for the KTV-Helios scheme, we define the algorithms in $\mathsf{Exp}^{\mathsf{rfree},\beta}_{\mathcal{A},\mathcal{S}}$ as follows:

- DeniablyUpdate$(id, \mathsf{sk}_{id}, v_0, v_1, t_v)$ as casting a ballot for $v_1 - v_0$: that is,

$$\mathsf{DeniablyUpdate}(id, \mathsf{sk}_{id}, v_0, v_1, t_v) = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_1 - v_0, t_v)$$

- Obfuscate(id) as casting a random number of dummy ballots distributed according to $\mathbb{P}_d, \mathbb{P}_t$: that is,

$$\mathsf{Obfuscate}(id) = \mathsf{VoteDummy}(id)$$

- SimProof as simulating a proof as described in Section 3.3.1.

4.3 Proving δ-Receipt-Freeness for KTV-Helios

In order to find an appropriate value of δ, so that we can show that KTV-Helios achieves δ-receipt-freeness, we need to account for the adversarial advantage gained from the number of ballots next to voter's id on the bulletin board. For this purpose, we define the following experiment $\mathsf{Exp}^{\mathsf{rfnum},\beta}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t}$: The challenger chooses a random $\beta\{0,1\}$ and outputs the number $m + \beta$, with $m \leftarrow\$ \mathbb{P}_d$, and the set of timestamps $t_1, ..., t_{m+\beta}$ that are independently sampled from $\mathbb{P}_t$ to the adversary. The adversary has to guess β. Hence, the experiment models the voter either obeying the adversary's instructions (for $\beta = 0$) or casting an additional ballot (for $\beta = 1$), whereby the adversary only has access to the number of ballots and their timestamps, but not to the ballots themselves. Let $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$ denote an advantage in this experiment, so that

$$\Pr\left[\mathsf{Exp}^{\mathsf{rfnum},0}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t} = 0\right] - \Pr\left[\mathsf{Exp}^{\mathsf{rfnum},1}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t} = 0\right] - \delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$$

is negligible.

Theorem 4.2. *KTV-Helios, instantiated with probability distributions $\mathbb{P}_d, \mathbb{P}_t$, achieves δ-receipt-freeness given the algorithms* SimProof, DeniablyUpdate, Obfuscate*, with $\delta = \delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$. It further does not achieve δ'-receipt-freeness for any $\delta' < \delta$.*

We base our proof on the idea, that the number of ballots next to the voter is the only source of information that gives advantage to the adversary. We consider a sequence of games, starting from $\mathsf{Exp}^{\mathsf{rfree},0}_{\mathcal{A}}$ and ending with $\mathsf{Exp}^{\mathsf{rfree},1}_{\mathcal{A}}$ and show, that the adversary $\mathcal{A}$ distinguishes the transition through all those games with the advantage of at most $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$. We define $\mathsf{BB}_{0,i}$ as the content of the bulletin board and (R_i, Π_i) as the tally output at the end of the game G_i, $i = 1, ..., 4$. We define the sequence as follows:

- G_1**.** The first game G_1 is equivalent to the experiment $\mathsf{Exp}^{\mathsf{rfree},\beta}_{\mathcal{A}}$ with $\beta = 0$ (hence, it is equivalent to the election where the voter *id* does not try to deniably update her vote). Thus, the content of $\mathsf{BB}_{0,1}$ and the tally output (R_1, Π_1) correspond to the content of BB_0 and the output of $\mathcal{O}\mathsf{Tally}$ at the end of $\mathsf{Exp}^{\mathsf{rfree},0}_{\mathcal{A}}$.
- G_2**.** The second game G_2 is equivalent to the election, where the voter *id* casts an additional ballot with a null-vote. Thus, the content of the bulletin board $\mathsf{BB}_{0,2}$ is equivalent to the content of the bulletin board BB_1 at the end of $\mathsf{Exp}^{\mathsf{rfree},1}_{\mathcal{A}}$ for the adversary using the query $\mathcal{O}\mathsf{Receipt}(id, v_0, v_1, t)$ with $v_0 = v_1$.

We prove, that the adversary has an advantage of δ_{num} of distinguishing between the output of G_1 and G_2. The tally result does not change, hence the tally output (R_2, Π_2) is equivalent to the tally output (R_1, Π_1). The only difference between the contents of $\mathsf{BB}_{0,1}$ and $\mathsf{BB}_{0,2}$ are the ballots next to *id*. Namely, G_1 contains only the ballot $b_{\mathcal{A}}$ and m dummy ballots $b_1, ..., b_m$ generated by the function $\mathsf{VoteDummy}(id)$ next to *id*, with $m \leftarrow\$ \mathbb{P}_d$ and the timestamps for the ballots $b_1, ..., b_m$ randomly sampled from $\mathbb{P}_t$. As for the second game, in addition to the ballots $b_{\mathcal{A}}, b_1, ..., b_m$, the bulletin board $\mathsf{BB}_{0,2}$ further contains an additional non-dummy (i.e. cast by the voter, not by a posting trustee) ballot $b_v = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, 0, t_v)$ cast by the voter at a random timestamp $t_v \leftarrow\$ \mathbb{P}_t$. As b_v, as well as $b_1, ..., b_m$, contains an encryption of 0, and due to the zero-knowledge property of the disjunctive proof π attached to both dummy and non-dummy ballots, it holds that b_v is indistinguishable from the dummy ballots $b_i, ..., b_m$. Furthermore, the timestamp attached to b_v is randomly sampled from the same distribution $\mathbb{P}_t$ as the timestamps for the dummy ballots $b_1, ..., b_m$. Hence, the number of the ballots next to *id* remains the only source of information that the adversary can use to gain advantage in distinguishing between G_1 and G_2. It therefore follows, that in order to distinguish between G_1 and G_2, the adversary has to distinguish, given the number of ballots m', whether m' was sampled from $\mathbb{P}_d$ (in which case the adversary is in G_1), or $m' = m + 1$ with $m \leftarrow\$ \mathbb{P}_d$ (in which case there is an additional non-dummy ballot, and the adversary is in G_2). This distinction corresponds to the definition of the experiment $\mathsf{Exp}^{\mathsf{rfnum},\beta}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t}$. Therefore, we conclude that distinguishing between the outputs of G_1 and G_2 is equivalent to distinguishing between the output of $\mathsf{Exp}^{\mathsf{rfnum},0}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t}$ and $\mathsf{Exp}^{\mathsf{rfnum},1}_{\mathcal{A},\mathbb{P}_d,\mathbb{P}_t}$, and therefore the adversarial advantage of distinguishing between the output of G_1 and G_2 is $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$.

- G_3**.** The third game G_3 is equivalent to the election, where the voter cast a vote for a non-null voting option $v \neq 0$, and the tally result R is calculated on the bulletin board $\mathsf{BB}_{0,2}$ with simulated tally proof $\Pi = \mathsf{SimProof}(\mathsf{BB}_{0,3}, R)$.

We now prove, that the adversarial advantage in distinguishing between the output of G_2 and G_3 is negligible. Consider an adversary $\mathcal{B}$ in the ballot privacy experiment $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$ who simulates the games G_2 and G_3 for the adversary $\mathcal{A}$. The adversary $\mathcal{B}$ returns the output of $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A}}$ for the queries $\mathcal{O}\mathsf{VoteLR}$, $\mathcal{O}\mathsf{Tally}$. For simulating the output of $\mathcal{O}\mathsf{Receipt}(id, v_0, v_1, t)$, $\mathcal{B}$ proceeds as follows: first, she computes a ballot $b_v = \mathsf{Vote}((id, \mathsf{sk}_{id}), id, v_0, t)$. She then chooses a random value $m \leftarrow_{\$} \mathbb{P}_d$, and a set of and random timestamps $t_1, ..., t_m \leftarrow_{\$} \mathbb{P}_t$, and computes a set of ballots $b_1, ..., b_m$ with $b_i = \mathsf{Vote}((\hat{id}, 0), id, 0, t_i)$. She then uses the query $\mathcal{O}\mathsf{VoteLR}(id, id, 0, v_1/v_0, t')$ for a random $t' \in \mathbb{P}_t$ in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A}}$ and returns its output together with the ballots $b_v, b_1, ..., b_m$ to $\mathcal{A}$. At the end, $\mathcal{B}$ returns the value β output by $\mathcal{A}$ as the guess in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A},\mathcal{S}}$. Thus, it follows that the adversarial advantage in distinguishing G_2 from G_3 is at most equal to the adversarial advantage in $\mathsf{Exp}^{\mathsf{bpriv},\beta}_{\mathcal{A}}$, denoted as δ_{BPRIV}.

It follows, that in the transition through the game sequence $G_1 \rightarrow G_2 \rightarrow G_3$ the outputs of each game are distinguished from the outputs of a previous game with the advantage either $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$ (for games G_1 and G_2) or δ_{BPRIV} (for games G_2 and G_3). Hence, the adversary distinguishes between the output in $\mathsf{Exp}^{\mathsf{rfree},\beta}_{\mathcal{A}}$ with the advantage of at most $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t} + \delta_{BPRIV}$, with δ_{BPRIV} negligible as proven in Section 3.3.1. □

The value of $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$ can be calculated similar to $\delta^{\mathsf{num}}_{k,\mathbb{P}_d,\mathbb{P}_t}$ in Section 3. We provide an example for $\delta^{\mathsf{rfnum}}_{\mathbb{P}_d,\mathbb{P}_t}$ for some choices of $\mathbb{P}_d$ and $\mathbb{P}_t$ in the full version of our paper [19].

5 RELATED WORK

Several definitions of security requirements in electronic voting and underlying assumptions have been developed. An overview of game-based ballot privacy definitions was proposed in [8], and a framework that proposes a uniform treatment of the verifiability definitions from [7, 15, 28, 33, 44] is described in [17]. Other approaches for defining and evaluating the security of voting schemes include applied pi-calculus [5, 20, 29], process algebra [39], k-resilience terms [43], a taxonomy of different levels of security requirements [35] or a formal model based on the Common Criteria Protection Profile [22]. These approaches have been applied to evaluate various voting schemes [3, 20, 30, 43]. In particular, the formal security analysis of Helios has been the topic of [8, 15, 30, 35].

A number of formal definitions for receipt-freeness in electronic voting have been proposed. A game-based definition by Kiayias et al [28] ensures, that the voters do not not get any information from the voting system that can serve as a receipt. Their definition, however, excludes the scenarios where the voters obtain a receipt by following the instructions of the adversary. Cortier et al. [10, 13] provide another game-based definition, which, however, does not consider counter-strategies available to the voter. The simulation-based definition of Moran et al. in [40], as well as the definition in [32] based on epistemic logic, on the contrary, allow the voter to apply counter-strategies to fake her receipts. Further symbolic definitions of receipt-freeness include [5, 20, 25, 32, 40] (see also an overview of such definitions in [38]), and a framework for expressing the existing definitions of receipt-freeness in the modal logics of strategic ability method has been proposed in [46].

For now, participation privacy electronic voting has not been in the focus of research on formal security proofs. Hence, although the definitions of vote privacy (see e.g. an overview of such definitions in [8]) can be adjusted to address participation privacy, no formal definitions of this requirement have been proposed specifically.

Various modifications of Helios have been proposed, fixing its vulnerabilities [9], introducing new security properties such as receipt-freeness [13], long-term privacy [21] or verifiability against malicious bulletin board [15] or proposing alternatives to the verification mechanism [23]. Other research focused on improving the usability of Helios [27, 41].

6 CONCLUSION AND FUTURE WORK

We have proposed a probabilistic abstract definition of (δ, k)-participation privacy, with δ representing the adversarial advantage in distinguishing whether a particular honest voter has cast up to k ballots in the election. We also proposed a probabilistic abstract definition of δ-receipt-freeness for voting schemes based on deniable vote updating. We used both of these definitions to evaluate the security of the KTV-Helios extension proposed in [31].

We plan to extend the proofs in this paper for the case where the tabulation teller is implemented in a distributed way. We further plan to address the existing security and efficiency issues of KTV-Helios, such as the possibility of board flooding and the necessity of trusting the device that holds the private signing key for integrity.

ACKNOWLEDGEMENTS

We would like to thank Vincent Cheval, Véronique Cortier, Marc Fischlin, Vanessa Teague and Tomasz Truderung for their helpful feedback at various stages. The research that led to these results has been partially funded from a project in the framework of Hessen ModellProjekte (HA project no. 435/14-25), financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence). It has also been partially developed within the project 'VALID' - Verifiable Liquid Democracy - which is funded by the Polyas GmbH. This work has also been supported by the German Federal Ministry of Education and Research (BMBF) as well as by the Hessen State Ministry for Higher Education, Research and the Arts within CRISP.

REFERENCES

[1] Dirk Achenbach, Carmen Kempka, Bernhard Löwe, and Jörn Müller-Quade. 2015. Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting. *JETS 2015: USENIX Journal of Election Technology and Systems* (2015), 26–45.

[2] Ben Adida. 2008. Helios: Web-based Open-audit Voting. In *SS 2008: 17th Conference on Security Symposium*. USENIX, 335–348.

[3] Mathilde Arnaud, Véronique Cortier, and Cyrille Wiedling. 2013. Analysis of an Electronic Boardroom Voting System. In *VoteID 2013: 4th International Conference on E-Voting and Identify*. Springer, 109–126.

[4] N. Asokan, Victor Shoup, and Michael Waidner. 1998. Optimistic fair exchange of digital signatures. In *EUROCRYPT 1998: International Conference on the Theory and Application of Cryptographic Techniques*. Springer, 591–606.

[5] M. Backes, C. Hritcu, and M. Maffei. 2008. Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-Calculus. In *CSFS 2008: 21st IEEE Computer Security Foundations Symposium*. IEEE, 195–209.

[6] Josh C Benaloh and Moti Yung. 1986. Distributing the power of a government to enhance the privacy of voters. In *PODC 1986: 5th annual ACM symposium on Principles of distributed computing*. ACM, 52–62.

[7] Josh Daniel Cohen Benaloh. 1987. Verifiable Secret-Ballot Elections. *PhD thesis, Yale University, Department of Computer Science* (1987).

[8] D. Bernhard, V. Cortier, D. Galindo, O. Pereira, and B. Warinschi. 2015. SoK: A Comprehensive Analysis of Game-Based Ballot Privacy Definitions. In *SP 2015: IEEE Symposium on Security and Privacy*. IEEE, 499–516.

[9] David Bernhard, Olivier Pereira, and Bogdan Warinschi. 2012. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In *ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 626–643.

[10] Pyrros Chaidos, Véronique Cortier, Georg Fuchsbauer, and David Galindo. 2016. Beleniosrf: A non-interactive receipt-free electronic voting scheme. In *CCS 2016: ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1614–1625.

[11] David Chaum and Torben Pryds Pedersen. 1993. Wallet Databases with Observers. In *CRYPTO 1992: 12th Annual International Cryptology Conference*. Springer, 89–105.

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

[13] Véronique Cortier, Georg Fuchsbauer, and David Galindo. 2015. BeleniosRF: A Strongly Receipt-Free Electronic Voting Scheme. Cryptology ePrint Archive, Report 2015/629. (June 2015). http://eprint.iacr.org/.

[14] Véronique Cortier, David Galindo, Stéphane Glondu, and Malika Izabachène. 2013. Distributed ElGamal à La Pedersen: Application to Helios. In *WPES 2013: 12th ACM Workshop on Workshop on Privacy in the Electronic Society*. ACM, 131–142.

[15] Véronique Cortier, David Galindo, Stéphane Glondu, and Malika Izabachène. 2014. Election Verifiability for Helios under Weaker Trust Assumptions. In *ESORICS 2014: 19th European Symposium on Research in Computer Security, Part II*. Springer, 327–344.

[16] Veronique Cortier, David Galindo, Ralf K üsters, Johannes Mueller, and Tomasz Truderung. 2016. Verifiability Notions for E-Voting Protocols. (March 2016). Cryptology ePrint Archive, Report 2016/287 http://eprint.iacr.org/.

[17] Veronique Cortier, David Galindo, Ralf Kuesters, Johannes Mueller, and Tomasz Truderung. 2016. Verifiability Notions for E-Voting Protocols. Cryptology ePrint Archive, Report 2016/287. (March 2016). http://eprint.iacr.org/.

[18] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. 1994. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO 1994: 14th Annual International Cryptology Conference*. Springer, 174–187.

[19] Melanie Volkamer David Bernhard, Oksana Kulyk. 2017. Security Proofs for Participation Privacy, Receipt-Freeness, Ballot Privacy, and Verifiability Against Malicious Bulletin Board for the Helios Voting Scheme. Cryptology ePrint Archive, Report 2016/431. (March 2017). http://eprint.iacr.org/2016/431.

[20] Stéphanie Delaune, Steve Kremer, and Mark Ryan. 2009. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security* 17, 4 (2009), 435–487.

[21] Denise Demirel, Jeroen Van De Graaf, and Roberto Samarone dos Santos Araújo. 2012. Improving Helios with Everlasting Privacy Towards the Public. In *EVTW/WTE 2012: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX.

[22] Rüdiger Grimm, Melanie Volkamer, and others. 2008. Development of a formal IT-security model for remote electronic voting systems. In *3nd International Workshop on Electronic Voting, Lecture Notes in Informatics*. 185–196.

[23] Sandra Guasch and Paz Morillo. 2016. How to challenge and cast your e-vote. In *FC 2016: 20th international conference on Financial Cryptography and Data Security*. IFCA.

[24] IACR. 2016. IACR Elections. http://www.iacr.org/elections. (2016). [Online; accessed 19-January-2017].

[25] H.L. Jonker and W. Pieters. 2006. Receipt-freeness as a special case of anonymity in epistemic logic. In *WOTE 2006: IAVoSS Workshop On Trustworthy Elections*. Robinson College.

[26] Ari Juels, Dario Catalano, and Markus Jakobsson. 2005. Coercion-resistant electronic elections. In *WPES 2005: ACM workshop on Privacy in the electronic society*. ACM, 61–70.

[27] Fatih Karayumak, Michaela Kauer, M Maina Olembo, Tobias Volk, and Melanie Volkamer. 2011. User study of the improved Helios voting system interfaces. In *STAST 2011: 1st Workshop on Socio-Technical Aspects in Security and Trust*. IEEE, 37–44.

[28] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 2015. End-to-End Verifiable Elections in the Standard Model. In *EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Part II*. Springer, 468–498.

[29] Steve Kremer and Mark Ryan. 2005. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In *ESOP 2005: 14th European Symposium on Programming, Programming Languages and Systems*. Springer, 186–200.

[30] Steve Kremer, Mark Ryan, and Ben Smyth. 2010. Election Verifiability in Electronic Voting Protocols. In *ESORICS 2010: 15th European Symposium on Research in Computer Security*. Springer, 389–404.

[31] Oksana Kulyk, Vanessa Teague, and Melanie Volkamer. 2015. Extending Helios Towards Private Eligibility Verifiability. In *VoteID 2015: 5th International Conference on E-Voting and Identity*. Springer, 57–73.

[32] Ralf Küsters and Tomasz Truderung. 2009. An epistemic approach to coercion-resistance for electronic voting protocols. In *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 251–266.

[33] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. 2010. Accountability: Definition and Relationship to Verifiability. In *CCS 2010: 17th ACM Conference on Computer and Communications Security*. ACM, 526–535.

[34] R. Küsters, T. Truderung, and A. Vogt. 2010. A Game-Based Definition of Coercion-Resistance and Its Applications. In *CSFS 2010: 23rd IEEE Computer Security Foundations Symposium*. IEEE, 122–136.

[35] Lucie Langer, Axel Schmidt, Johannes Buchmann, and Melanie Volkamer. 2010. A taxonomy refining the security requirements for electronic voting: analyzing Helios as a proof of concept. In *ARES 2010: International Conference on Availability, Reliability, and Security*. IEEE, 475–480.

[36] Philipp Locher and Rolf Haenni. 2016. Receipt-free remote electronic elections with everlasting privacy. *Annals of Telecommunications* 71, 7 (2016), 323–336.

[37] Philipp Locher, Rolf Haenni, and Reto E Koenig. 2016. Coercion-Resistant Internet Voting with Everlasting Privacy. In *FC 2016: International Workshops, BITCOIN, VOTING, and WAHC, Revised Selected Papers*. Springer.

[38] Bo Meng. 2009. A critical review of receipt-freeness and coercion-resistance. *Information Technology Journal* 8, 7 (2009), 934–964.

[39] Murat Moran, James Heather, and Steve Schneider. 2012. Verifying anonymity in voting systems using CSP. *Formal Aspects of Computing* 26, 1 (Dec. 2012), 63–98.

[40] Tal Moran and Moni Naor. 2006. Receipt-free universally-verifiable voting with everlasting privacy. In *CRYPTO 2006: Annual International Cryptology Conference*. Springer, 373–392.

[41] Stephan Neumann, M. Maina Olembo, Karen Renaud, and Melanie Volkamer. 2014. Helios Verification: To Alleviate, or to Nominate: Is That the Question, or Shall we Have Both?. In *EGOVIS 2014: 3rd International Conference on Electronic Government and the Information Systems Perspective*. Springer, 246–260.

[42] Peter Y A Ryan, Peter B Roenne, and Vincenzo Iovino. 2015. Selene: Voting with Transparent Verifiability and Coercion-Mitigation. Cryptology ePrint Archive, Report 2015/1105. (Nov. 2015). http://eprint.iacr.org/.

[43] Guido Schryen, Melanie Volkamer, Sebastian Ries, and Sheikh Mahbub Habib. 2011. A Formal Approach Towards Measuring Trust in Distributed Systems. In *SAC 2011: ACM Symposium on Applied Computing*. ACM, 1739–1745.

[44] Ben Smyth, Steven Frink, and Michael R. Clarkson. 2015. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. Cryptology ePrint Archive, Report 2015/233. (March 2015). http://eprint.iacr.org/.

[45] Oliver Spycher, Reto Koenig, Rolf Haenni, and Michael Schläpfer. 2011. A new approach towards coercion-resistant remote e-voting in linear time. In *FC 2011: 15th international conference on Financial Cryptography and Data Security*. Springer, 182–189.

[46] Masoud Tabatabaei, Wojciech Jamroga, and Peter YA Ryan. 2016. Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability: Preliminary Attempt. In *Proceedings of the 1st International Workshop on AI for Privacy and Security*. ACM, 1.

[47] Björn Terelius and Douglas Wikström. 2010. Proofs of Restricted Shuffles. In *AFRICACRYPT 2010: 3rd International Conference on Cryptology in Africa*. Springer, 100–113.

[48] Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. 2013. From Helios to Zeus. In *EVTW/WTE 2013: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX.