THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

# Definitions and Security of Quantum Electronic Voting

**Citation for published version:**
Arapinis, M, Lamprou, N, Kashefi, E & Pappa, A 2021, 'Definitions and Security of Quantum Electronic Voting', *ACM Transactions on Quantum Computing*, vol. 2, no. 1, 4. https://doi.org/10.1145/3450144

**Digital Object Identifier (DOI):**
10.1145/3450144

**Link:**
Link to publication record in Edinburgh Research Explorer

**Document Version:**
Peer reviewed version

**Published In:**
ACM Transactions on Quantum Computing

**General rights**
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 17. Aug. 2021

# Definitions and Security of Quantum Electronic Voting

MYRTO ARAPINIS and NIKOLAOS LAMPROU*, School of Informatics, University of Edinburgh, United Kingdom
ELHAM KASHEFI, LIP6, University Pierre et Marie Curie, France
ANNA PAPPA, Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Germany

Recent advances indicate that quantum computers will soon be reality. Motivated by this ever more realistic threat for existing classical cryptographic protocols, researchers have developed several schemes to resist "quantum attacks". In particular, for electronic voting, several e-voting schemes relying on properties of quantum mechanics have been proposed. However, each of these proposals comes with a different and often not well-articulated corruption model, has different objectives, and is accompanied by security claims which are never formalized and are at best justified only against specific attacks. To address this, we propose the first formal security definitions for quantum e-voting protocols. With these at hand, we systematize and evaluate the security of previously-proposed quantum e-voting protocols; we examine the claims of these works concerning privacy, correctness and verifiability, and if they are correctly attributed to the proposed protocols. In all non-trivial cases, we identify specific quantum attacks that violate these properties. We argue that the cause of these failures lies in the absence of formal security models and references to the existing cryptographic literature.

Additional Key Words and Phrases: quantum electronic voting, quantum cryptography, attacks

**ACM Reference Format:**
Myrto Arapinis, Nikolaos Lamprou, Elham Kashefi, and Anna Pappa. 2018. Definitions and Security of Quantum Electronic Voting. *ACM Trans. Quantum Comput.* 37, 4, Article 111 (August 2018), 33 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Voting is fundamental in democratic societies. With the technological advances of the computer era, voting could benefit to become more secure and efficient and as a result more democratic. For this reason, over the last two decades, several cryptographic protocols for electronic voting were proposed and implemented, see e.g. [1, 11, 16, 28, 31, 44]. The security of all these systems relies on computational assumptions such as the hardness of integer factorization and the discrete logarithm problem. But, these are easy to solve with quantum computers using Shor's algorithm [45]. Although not yet available, recent technological advances indicate that quantum computers will soon be threatening existing cryptographic protocols. In this context, researchers have proposed to use quantum communication to implement primitives like key distribution, bit commitment and oblivious transfer. Unfortunately, perfect security without assumptions has proven to be challenging

Authors' addresses: Myrto Arapinis, marapini@inf.ed.ac.uk; Nikolaos Lamprou, n.lamprou@ed.ac.uk, School of Informatics, University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, United Kingdom; Elham Kashefi, ekashefi@inf.ed.ac.uk, LIP6, University Pierre et Marie Curie, 4 Place Jussieu, Paris, 75005, France; Anna Pappa, annapappa@gmail.com, Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Ernst-Reuter-Platz 7, Berlin, 10587, Germany.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Association for Computing Machinery.
2643-6817/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

| | Privacy | Correctness | Verifiability | Corruption |
|---|---|---|---|---|
| Dual basis (Section 4) | ✗ | ? | ? | $\epsilon$ fraction of voters |
| Travelling ballot (Section 5) | ✗ | ✗ | ✗ | two voters |
| Distributed ballot (Section 6) | ? | ✗* | ✗ | $\epsilon$ fraction of voters |
| Conjugate coding (Section 7) | ✗ | ? | ? | election authority |

Table 1. Properties of the different categories of quantum e-voting protocols. ✗: Insecure, ?: Unexplored Area, *:Protocol runs less than $exp(\Omega(N))$ rounds.

in the quantum setting [34, 36], and the need to study different corruption models has emerged. This includes limiting the number of dishonest parties and introducing different non-colluding authorities.

More than a decade of studies on quantum electronic voting has resulted in several protocols that use the properties of quantum mechanical systems. However, all these new protocols are studied against different and not well-articulated corruption models, and claim security using ad-hoc proofs that are not formalized and backed only against limited classes of quantum attacks. In particular, none of the proposed schemes provides rigorous definitions of privacy and verifiability, nor formal security proofs against specific, well-defined (quantum) attacker models. When it comes to electronic voting schemes, it is particularly hard to ensure that all the, somehow conflicting, properties hold [12]; it is therefore important that these new quantum protocols be rigorously and mathematically studied and the necessary assumptions and limitations formally established.

This is precisely what we set to address in this paper. We first give formal definitions for verifiability and vote privacy in the quantum setting considering adaptive corruption. Subsequently, we systematize and assess the security of existing e-voting protocols based on quantum technology. We specifically examine the claims of each of these solutions concerning the above-mentioned well-defined properties. Unfortunately our analyses uncover vulnerabilities in all the proposed schemes. While some of them suffer from trivial attacks due to inconsistencies in the security definitions, the main contribution of the paper is to argue that sophisticated attacks can exist even in protocols that "seem secure" if the security is proven ad hoc, and not in a formal framework. We argue that the cause of these failures is the absence of an appropriate security framework in which to establish formal security proofs, which we have now introduced.

Therefore, this paper follows previous works [4, 22, 38, 43, 48] in their effort to highlight the importance of formally defining and proving security in the relatively new field of quantum cryptography. This also includes studying classical protocols that are secure against unbounded attackers [8], as well as ones based on problems believed to be hard even for quantum computers e.g. lattice-based [13]. However, it is out of the scope of this study to review such classical protocols, as we are focusing on the possible contribution of quantum computers to the security of e-voting.
**Contributions:** We propose the first formal definitions for vote privacy and universal verifiability in the quantum setting considering adaptive corruption, and show that none of the proposed quantum protocols so far satisfy them. To this end, we systematize the proposed quantum e-voting approaches according to key technical features. To our knowledge, our study covers all relevant research in the field, identifying four main families. Table 1 summarises our results.

- *Two measurement bases protocols* - These protocols rely on two measurement bases to verify the correct distribution of an entangled state. We specifically prove that the probability that a number of corrupted states are not tested and used later in the protocol, is non-negligible, which leads to a violation of voters' privacy. Furthermore, even if the states are shared by a trusted authority, we show that privacy can still be violated in case of abort.

- *Traveling ballot protocols* - In these protocols the "ballot box" circulates among all voters who add their vote by applying a unitary to it. We show how colluding voters can break honest voters' vote privacy just by measuring the ballot box before and after the victim has cast their ballot. These protocols further suffer as we will see from double voting attacks, whereby a dishonest voter can simply apply multiple time the voting operator.
- *Distributed ballot protocols* - These schemes exploit properties of entangled states that allow voters to cast their votes by applying operations on parts of them. We present an attack that allows the adversary to double-vote and therefore change the outcome of the voting process with probability at least 0.25, if the protocol runs fewer than exponentially many rounds in the number of voters. The intuition behind this attack is that an adversary does not need to find exactly how the ballots have been created in order to influence the outcome of the election; it suffices to find a specific relation between them from left-over voting ballots provided by the corrupted voters.
- *Conjugate coding protocols* - These protocols exploit BB84 states adding some verification mechanism. The main issue with these schemes, as we show, is that ballots are malleable, allowing an attacker to modify the part of the ballot which encodes the candidate choice to their advantage.

## 2 PRELIMINARIES

We use the term quantum bit or qubit [39] to denote the simplest quantum mechanical object we will use. We say that a qubit is in a pure state if it can be expressed as a linear combination of other pure states:

$$|x\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,\ \text{where}\ |0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

where $|\alpha|^2 + |\beta|^2 = 1$ for any $\alpha, \beta \in \mathbb{C}$. The states $|0\rangle$ and $|1\rangle$ are called the computational basis vectors. Sometimes it is also helpful to think of a qubit as a vector in the two-dimensional Hilbert space $\mathcal{H}$. If a qubit cannot be written in the above form, then we say it is in a mixed state. The generalization of a qubit to an $m$-dimensional quantum system is called *qudit*:

$$|y\rangle = \sum_{j=0}^{m-1} a_j\,|j\rangle\,,\ \text{where}\ \sum_{j=0}^{m-1} |a_j|^2 = 1$$

Let's now suppose that we have two qubits; we can write the state vector as:

$$|\psi\rangle = \sum_{i,j\in\{0,1\}} \alpha_{ij}\,|ij\rangle$$

where $\sum_{i,j\in\{0,1\}} |\alpha_{ij}|^2 = 1$. If the total state vector $|\psi\rangle$ cannot be written as a tensor product of two qubits (*i.e.* $|x_1\rangle \otimes |x_2\rangle$), then we say that qubits $|x_1\rangle$ and $|x_2\rangle$ are entangled. An example of two-qubit entangled states, are the four *Bell states*, which form a basis of the two-dimensional Hilbert space:

$$|\Phi^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), |\Psi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$$

A quantum system that is in one of the above states is also called an EPR pair [21]. The way we obtain information about a quantum system is by performing a measurement using a family of linear operators $\{M_j\}$ acting on the state space of the system, where $j$ denotes the different

outcomes of the measurement. It holds for the discrete and the continuous case respectively that:

$$\sum_j M_j^\dagger M_j = \int M_j^\dagger M_j dj = \mathbb{I}$$

where $M_j^\dagger$ is the conjugate transpose of matrix $M_j$, and $\mathbb{I}$ the identity operator. For qudit $|y\rangle$, the probability that the measurement outcome is $w$ is: $\Pr(w) = \langle y| M_w^\dagger M_w |y\rangle$ and in the continuous case $\Pr(w \in [w_1, w_2]) = \int_{w_1}^{w_2} \langle y| M_j^\dagger M_j |y\rangle \, dj$.

For a single qubit $|x\rangle = \alpha |0\rangle + \beta |1\rangle$, measurement in the computational basis will give outcome zero with probability $|\alpha|^2$ and outcome one with probability $|\beta|^2$. If our state is entangled, a partial measurement (i.e. a measurement in one of the entangled qudits), not only reveals information about the measured qudit, but possibly about the remaining state. For example, let us recall the Bell state $|\Phi^+\rangle$. A measurement of the first qubit in the computational basis will give outcome 0 or 1 with equal probability and the remaining qubit will collapse to the state $|0\rangle$ or $|1\rangle$ respectively.

In quantum cryptography, the correlations in the measurement outcomes of entangled states are frequently exploited. Another entangled state of interest used in Section 4, gives measurement outcomes that sum up to zero when measured in the computational basis, and equal outcomes when measured in the Fourier basis (denoted by $|\rangle_F$). In the three-qubit case, the state is the following:

$$|D\rangle = \frac{1}{\sqrt{3}}\big( |0\rangle_F |0\rangle_F |0\rangle_F + |1\rangle_F |1\rangle_F |1\rangle_F \big) = \frac{1}{2}\big( |000\rangle + |011\rangle + |101\rangle + |110\rangle \big)$$

Finally, the evolution of a closed quantum system can be described by the application of a unitary operator. Unitary operators are reversible and preserve the inner product. Recall our first example, and let's say we would like to swap the amplitudes on state $|x\rangle$, then we can apply the operator $X$ (known as NOT-gate):

$$X |x\rangle = \beta |0\rangle + \alpha |1\rangle, \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The $X$-gate is one of the Pauli operators, which together with $Z$ and $Y$, as well as the identity operator $\mathbb{I}$, form a basis for the vector space of $2 \times 2$ Hermitian matrices. These operators are unitaries, and as such preserve the inner product.

A very important difference between quantum and classical information, is that there is no mechanism to create a copy of an unknown quantum state [39]. This result, known as the *no-cloning theorem*, is one of the fundamental advantages and at the same time limitations of quantum information. It becomes extremely relevant for cryptography, since brute-force types of attacks cannot be applied on quantum channels that carry unknown information. When verifying quantum resources however, it is necessary to apply a cut-and-choose technique in order to test that the received quantum states are correctly produced. The quantum source would therefore need to send exponentially many copies of the quantum state [29], in order for the verifier to measure most of them and deduce that with high probability, the remaining ones are correct.

## 3 DEFINITIONS OF SECURE QUANTUM ELECTRONIC VOTING

Electronic voting protocols are multiparty protocols typically involving talliers, voters and bulletin boards [1, 28, 31]. In this work, we will be dealing with protocols involving one election authority $EA$, a set of voters $\mathcal{V} = \{V_k\}_{k=1}^N$ and a set of talliers $T$ (which may overlap with $\mathcal{V}$). $EA$ sets the parameters of the protocol, each voter $V_k \in \mathcal{V}$ casts vote $v_k$ and $T$ gathers the ballots, computes the election outcome and announces it. A voting protocol $\Pi$ has three distinct phases (*setup*, *casting*, and *tally*) and running time that is polynomial to the security parameter $\delta_0$. It specifies three algorithms $\{\mathsf{CastBallot}, \mathsf{Tally}, \mathsf{Verify}\}$, the setup procedure and the communication channels

between participants (which is captured in our formal definition with the communication oracle $O$). For formalising security we adopt the standard game-based security framework. The security of a protocol is captured by a game between a challenger $C$ that models the honest parties, and a quantum polynomial-time adversary $\mathcal{A}$ that captures the corrupted parties. $\mathcal{A}$ can adaptively corrupt a fraction $\epsilon$ of the voters. We assume that the eligibility list is provided a priori in a trusted manner and that $\mathcal{A}$ chooses the votes of all voters, in order to provide stronger definitions [28]. We denote with $\mathcal{X}_a$ the local register of each party $a$. Communication between parties is done using *communication oracles* [20]; a call to the oracle $O_{\Pi,\mathcal{A}}(\mathcal{X}_S, \mathcal{X}_R)$ takes contents of the sender's register $\mathcal{X}_S$ and copies them to the receiver's register $\mathcal{X}_R$, according to $\Pi$. In between, it allows the communication to be processed by $\mathcal{A}$, and it also erases the quantum information from the respective 'sent' registers, to respect the no-cloning theorem (classical information can still be transmitted without being erased from the sender). Note that the way $\mathcal{A}$ can treat the transmitted information is specified by the protocol $\Pi$ and quantum mechanics, e.g. in case of a quantum authentication channel between parties, $\mathcal{A}$ is only allowed to erase the content of the quantum register but nothing more.

***Setup phase*:** $\mathcal{A}$ defines the voting choices of all voters. $C$ and $\mathcal{A}$ generate the protocol parameters and store them in the global register $\mathcal{X}$ (comprising of all local registers $\mathcal{X}_a$). This is done according to $\Pi$, therefore if the latter specifies that the parameters are generated by a trusted third party, then this interaction is void. Instead, in protocols like the one in section 4 where the parameters are generated interactively among the parties, this interaction illustrates the fact that some of these parties might be corrupt. In order to also capture protocols that use anonymous channels, we ask $C$ to randomly choose a permutation $\rho$ from the set $\mathcal{F}$ of all permutations of $N$ elements (the uniformly at random choice is denoted with $\rho \in_R \mathcal{F}$). Permutation $\rho$ specifies the casting order of the voters and is initially unknown to $\mathcal{A}$; information about $\rho$ could however be leaked during the next phase.

***Casting phase*:** The protocol $\Pi$ specifies the algorithm CastBallot for generating the ballots. $C$ generates ballots according to the CastBallot algorithm on behalf of honest voters and $\mathcal{A}$ on behalf of the corrupted ones.

***Tally phase*:** The protocol $\Pi$ specifies the tallying algorithm Tally and the verification algorithm Verify. The Verify algorithm is the protocol-specific public test parties can run to verify the election. Note that if such an algorithm is not explicitly provided in the protocol's description, then it can be modelled by the True predicate (the algorithm will always return 1). If this test is successful (return 1), the tally is then computed. If all talliers are honest, $C$ computes the election result on behalf of them by running the Tally algorithm. If some of the talliers are corrupt, $\mathcal{A}$ and $C$ produce the tally in an interactive way. If none of the talliers is honest, $\mathcal{A}$ computes the tally instead. We capture all these cases with function Tally′, where honest talliers controlled by $C$ act according to $\Pi$ and dishonest talliers act as $\mathcal{A}$ dictates.

Ideally, an e-voting protocol will satisfy at least the following properties [6, 15, 18, 22]. Correctness: in the absence of an adversary, the correct outcome is computed; additionally, corrupted voters cannot modify the honest votes, all voters vote at most once and the protocol does not abort[1]. Privacy: keep the vote of a voter private. Verifiability: allow for verification of the results by voters and external auditors. We focus on privacy and verifiability type properties. We note that our tally function is defined similarly to [6]. Our definitions therefore capture only tally functions where the election result corresponds to a unique set of votes (e.g it doesn't capture privacy for 'weighted

[1]All of the quantum protocols we analyse in this work are subject to abort by a single party except the one presented in [40, 53]. Despite that, we present and focus on attacks that violate properties such as *privacy* and *correctness* by attacking the one-voter-one-vote policy.

voting'); however, this restriction is a natural assumption based on the state-of-the-art on e-voting based protocols [2].

**Universal Verifiability.** Our definition of *universal verifiability* is similar to [15] and is captured by the experiment $\mathbf{EXP}^{\Pi}_{\mathsf{Qver}}$.

*The experiment* $\mathbf{EXP}^{\Pi}_{\mathsf{Qver}}(\mathcal{A}, \epsilon, \delta_0)$

- **Setup phase:** $C$ and $\mathcal{A}$ generate the parameters in register $\mathcal{X}$ as specified by $\Pi$ and the adversarial model. Furthermore, $\mathcal{A}$ chooses the votes for all voters $\{v_k\}_{V_k \in \mathcal{V}}$ and $C$ the casting order $\rho \in_R \mathcal{F}$.
- **Casting phase:** For $k = 1, \ldots, N$
  - If $\mathcal{A}$ chooses to corrupt $V_{\rho(k)}$, they are added to $\mathcal{V}_{\mathcal{A}}$.
  - If $V_{\rho(k)} \notin \mathcal{V}_{\mathcal{A}}$, then $C$ runs $(\mathcal{X}_{\rho(k)}, \perp) \leftarrow \mathsf{CastBallot}(v_{\rho(k)}, \mathcal{X}_{\rho(k)}, \delta_0)$. If not $\perp$, $C$ calls $O_{\Pi,\mathcal{A}}(\mathcal{X}_{\rho(k)}, \mathcal{X}_R)$, where $R$ is the receiver designated by $\Pi$.
  - If $V_{\rho(k)} \in \mathcal{V}_{\mathcal{A}}$, $\mathcal{A}$ performs some operation on register $\mathcal{X}_{\mathcal{A}}$ and calls $O_{\Pi,\mathcal{A}}(\mathcal{X}_{\mathcal{A}}, \mathcal{X}_R)$, where $R$ is any receiver designated by $\mathcal{A}$.
- **Tally phase:** $\mathcal{A}$ and $C$ call the $\mathsf{Tally}'$ function, which depends on $\Pi$ and the adversarial model, to compute the election outcome $X \leftarrow \mathsf{Tally}'(\mathcal{X}_C, \mathcal{X}_{\mathcal{A}}, \delta_0)$.
- If $(\mathsf{Verify}(X, \mathcal{X}_C, \delta_0) = 1)$ and $(\mathsf{P}^{\Pi}_{\mathsf{VCounted}}(\{v_k\}_{V_k \notin \mathcal{V}_{\mathcal{A}}}, X) = 0 \vee \mathsf{Nballots}^{\Pi}(X) > N)$ then `output 1`, else `output 0`.

First, $\mathcal{A}$ defines how honest voters vote. Then, $C$ and $\mathcal{A}$ generate the protocol parameters $\mathcal{X}$ according to $\Pi$ and the corruption model of $\mathcal{A}$. Moreover, $C$ chooses at random a permutation $\rho$ that specifies the casting order of all voters. In the *casting phase*, $\mathcal{A}$ can choose to corrupt voters adaptively. For honest voters, $C$ follows the CastBallot algorithm as specified by $\Pi$ to generate the ballot, and sends it to $\mathcal{A}$. Depending on the protocol specification, $\mathcal{A}$ might then perform some quantum operation on the received ballot and further forward it to the designated receiver. The operation should always be consistent with the protocol, e.g. if $\Pi$ uses quantum authenticated channels, $\mathcal{A}$ will not be able to modify the ballot. This process is captured by calling the communication oracle $O$. For corrupted voters, $\mathcal{A}$ casts the ballot on their behalf. After all votes have been cast, the election outcome is computed by $\mathsf{Tally}'$ defined above, which depends on the protocol $\Pi$ and the adversarial model. $\mathbf{EXP}^{\Pi}_{\mathsf{Qver}}$ outputs 1 if the election outcome is accepted by $C$, while either an honest vote has not been counted in the final outcome, or the number of cast votes exceeds the number of voters; Otherwise the experiment outputs 0. To account for these events, we define three predicates; Verify which is the protocol-specific public test parties can run to verify the election, the predicate $\mathsf{P}^{\Pi}_{\mathsf{VCounted}}$ reveals if honest votes are discarded from or altered in the final outcome[2] and $\mathsf{Nballots}^{\Pi}$ reveals the number of votes accounted for the election result $X$. Specifically, $\mathsf{P}^{\Pi}_{\mathsf{VCounted}}$ captures exactly the two security properties of definition 2, p.9 in [15]. If $\mathcal{A}$ has tampered with the election outcome, the predicate Verify should return `false`.

*Definition 3.1.* A quantum e-voting protocol $\Pi$ satisfies $\epsilon$-**quantum verifiability** if for every quantum polynomial-time $\mathcal{A}$ the probability of winning experiment $\mathbf{EXP}^{\Pi}_{\mathsf{Qver}}(\mathcal{A}, \epsilon, \delta_0)$ is negligible with respect to $\delta_0$:

$$\Pr[1 \leftarrow \mathbf{EXP}^{\Pi}_{\mathsf{Qver}}(\mathcal{A}, \epsilon, \delta_0)] = negl(\delta_0).$$

[2]We require in particular that $\mathsf{P}^{\Pi}_{\mathsf{VCounted}}(\{v_k\}_{V_k \notin \mathcal{V}_{\mathcal{A}}}, \perp) = 0$

**Vote privacy.** The experiment $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ captures vote privacy which ensures that the adversary $\mathcal{A}$ cannot link honest voters to their votes. We build upon definition 1 in [6]; however a problem with this definition is that it requires the honest voters controlled by the challenger to send their ballot in two separate ballot boxes. Such a process is not possible with quantum information due to the no-cloning theorem; we solve this issue with our definition. Moreover, we also capture self-tallying protocols, where [6] states explicitly that a secret key is required for the production of the tally. Finally, our definition also treats protocols that use anonymous channels in order to provide privacy, while [6] leaves out such protocols, since it states that the adversary can call the oracle $O_{\text{cast}}$ with the ballot and the voter's ID. These fundamental differences lead to a completely new experiment.

The goal of $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ is to capture that $\mathcal{A}$ cannot distinguish between 'two worlds', one where the voters vote as $\mathcal{A}$ tells them, and another where their votes have been permuted.

*The experiment* $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}(\mathcal{A}, \epsilon, \delta_0)$

- **Setup phase:** $C$ and $\mathcal{A}$ generate the parameters in register $\mathcal{X}$ as specified by $\Pi$ and the adversarial model. $C$ chooses the casting order $\rho \in_R \mathcal{F}$ and a bit $\beta \in_R \{0,1\}$. Furthermore, $\mathcal{A}$ chooses the votes for all voters $\{v_k\}_{V_k \in \mathcal{V}}$ and a permutation $\mathsf{F} \in \mathcal{F}$.
- **Casting phase:** For $k = 1, \ldots, N$
  - If $\mathcal{A}$ chooses to corrupt $V_{\rho(k)}$, $V_{\rho(k)}$ is added to $\mathcal{V}_{\mathcal{A}}$.
  - If $V_{\rho(k)} \notin \mathcal{V}_{\mathcal{A}}$, then $C$ runs $\{\mathcal{X}_{\rho(k)}, \perp\} \leftarrow \mathsf{CastBallot}(v^{\beta}_{\mathsf{F}(\rho(k))} \cdot v^{1-\beta}_{\rho(k)}, \mathcal{X}_{\rho(k)}, \delta_0)$. If not $\perp$, $C$ calls $O_{\Pi,\mathcal{A}}(\mathcal{X}_{\rho(k)}, \mathcal{X}_R)$, , where $R$ is the receiver designated by $\Pi$.
  - If $V_{\rho(k)} \in \mathcal{V}_{\mathcal{A}}$, $\mathcal{A}$ performs some operations on register $\mathcal{X}_{\mathcal{A}}$ and calls $O_{\Pi,\mathcal{A}}(\mathcal{X}_{\mathcal{A}}, \mathcal{X}_R)$, where $R$ is any receiver designated by $\mathcal{A}$.
- **Tally phase:** If $\{v_k : V_k \notin \mathcal{V}_{\mathcal{A}}\} = \{v_{\mathsf{F}(k)} : V_k \notin \mathcal{V}_{\mathcal{A}}\}$, $C$ announces the election outcome $X \leftarrow \mathsf{Tally}(\mathcal{X}_C, \delta_0)$ to $\mathcal{A}$. Else `output -1`.

$\mathcal{A}$ guesses bit $\beta^*$. If $\beta^* = \beta$ then `output 1`, else `output 0`.

$\mathcal{A}$ defines how honest voters vote and chooses a permutation $\mathsf{F} \in \mathcal{F}$ over the voting choices of all voters in $\mathcal{V}$. After the parameters of the protocol $\mathcal{X}$ are generated, $C$ chooses a random bit $\beta$ which defines two worlds; when $\beta = 0$, the honest voters vote as specified by $\mathcal{A}$, while when $\beta = 1$, the honest voters swap their votes according to permutation $\mathsf{F}$ again specified by $\mathcal{A}$. Again, $C$ chooses at random a permutation $\rho$ which defines the casting order of all voters. If the choices of the honest voters during the `casting phase` are still a permutation of their initial choices the experiment proceeds to the next phase, else it outputs $-1$. In the tally phase, $C$ computes the election outcome. Finally, $\mathcal{A}$ tries to guess if the honest voters controlled by $C$ have permuted their votes ($\beta = 1$) or not ($\beta = 0$), by outputting the guess bit $\beta^*$. If $\mathcal{A}$ guessed correctly $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ outputs 1; otherwise $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ outputs 0.

*Definition 3.2.* A quantum e-voting protocol $\Pi$ satisfies $\epsilon$-**quantum privacy** if for every quantum polynomial-time $\mathcal{A}$ the probability of winning the experiment $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}(\mathcal{A}, \epsilon, \delta_0)$ is negligibly close to $1/2$ with respect to $\delta_0$ under the condition event $\neg$**False_Attack** happens, where **False_Attack** $= \{-1 \leftarrow \mathbf{EXP}^{\Pi}_{\text{Qpriv}}(\mathcal{A}, \epsilon, \delta_0)\}$:

$$\Pr[1 \leftarrow \mathbf{EXP}^{\Pi}_{\text{Qpriv}}(\mathcal{A}, \epsilon, \delta_0) | \neg \textbf{False\_Attack}] \leq 1/2 + negl(\delta_0)$$

The most established game-based definitions for privacy in the classical setting [6] assume two ballot boxes, where one holds the real tally and the other holds either the real or the fake tally. In

the quantum case, the adaptation is not straightforward mainly because of the no-cloning theorem. The existence of two such boxes assumes that information is copyable, which is not the case with quantum information. Similarly, we can't assume that the experiment runs two times because $\mathcal{A}$ could correlate the two executions by entangling their parameters, something that a classical adversary cannot do. We address this difficulty by introducing quantum communication oracles to capture the network activity and model the special handling of quantum information (e.g entangled states). Moreover, the election result is produced on the actual ballots rather than the intended ones. With this, we capture a broader spectrum of attacks (e.g Helios replay attack) and at the same time introduce trivial distinctions corresponding to false attacks. We tackle this by allowing the experiment to output −1 in such undesired cases which are mainly artifacts of the model. So an advantage of our privacy definition is that it allows the analysis of self-tallying type protocols in contrast with previous definitions of privacy [6]. In self-tallying elections the adversary is able to derive the election outcome on their own without the need of secret information.

**Note.** Our definitions of verifiability and privacy capture both classical and quantum protocols. For the classical case, the quantum registers will be used for storing and communicating purely classical information. Devising our definitions for the quantum setting was not a trivial task as there are many aspects that are hard to define, like bulletin boards, and others that need to be introduced, like quantum registers potentially containing entangled quantum states. Moreover, our experiments capture protocols that use anonymous channels, by assuming that the casting order is unknown to $\mathcal{A}$, as well as self-tallying protocols. We leave as future work the investigation of soundness of our definitions. To this end, one would define an ideal functionality capturing privacy and verifiability, and prove the soundness relation between our game-based definitions and this ideal functionality similarly to [6].

In the rest of the paper, we examine all existing proposals for quantum e-voting. For each of them, we identify attacks that violate the previously defined properties. Note that since the proposed protocols do not involve any verifiability mechanism, we need to define an experiment that involves honest Talliers, and that captures security against double voting and vote deletion/alteration against malicious voters. We term this property *correctness* and therefore need to consider experiment $\mathbf{EXP}^{\Pi}_{\mathsf{Qcorr}}$ which is the same as $\mathbf{EXP}^{\Pi}_{\mathsf{Qver}}$ but without the predicate Verify. The experiment $\mathbf{EXP}^{\Pi}_{\mathsf{Qcorr}}$ is detailed in Section A of the Supplementary Material.

## 4 DUAL BASIS MEASUREMENT BASED PROTOCOLS

In this section we discuss protocols that use the dual basis measurement technique [27, 51], and use as a blank ballot an entangled state with an interesting property: when measured in the computational basis, the sum of the outcomes is equal to zero, while when measured in the Fourier basis, all the outcomes are equal. Both of these protocols use cut-and-choose techniques in order to verify that the state was distributed correctly. This means that a large amount of states are checked for correctness and a remaining few are kept at the end unmeasured, to proceed with the rest of the protocol. Although a cut-and-choose technique with just one verifying party is secure if the states that are sampled are exponentially many and the remaining ones are constant, it is not clear how this generalizes to multiple verifying parties. Specifically, we show that if the corrupted parties sample their states last, then the probability with which the corrupted states are not checked and remain after all the honest parties sample, is at least a constant with respect to the security parameter of the protocol.

### 4.1 Protocol Specification

We will now present the self-tallying protocol of [51], which is based on the classical protocol of [30]. The voters $\{V_k\}_{k=1}^{N}$, without the presence of any trusted authority or tallier, need to verify that they share specific quantum states. At the end of the verification process, the voters share a classical matrix; every cast vote is equal to the sum of the elements of a row in the matrix.

**Setup phase**

(1) One of the voters, not necessarily trusted, prepares $N + N2^{\delta_0}$ states:

$$|D_1\rangle = \frac{1}{\sqrt{m^{N-1}}} \sum_{\sum_{k=1}^{N} i_k = 0 \mod c} |i_1\rangle |i_2\rangle \dots |i_N\rangle$$

where $m$ is the dimension of the qudits' Hilbert space, $c$ is the number of the possible candidates such that $m \geq c$ and $\delta_0$ the security parameter. The voter also shares $1 + N2^{\delta_0}$ states of the form:

$$|D_2\rangle = \frac{1}{\sqrt{N!}} \sum_{(i_1, i_2, \dots, i_N) \in \mathcal{P}_N} |i_1\rangle |i_2\rangle \dots |i_N\rangle$$

where $\mathcal{P}_N$ is the set of all possible permutations with $N$ elements. Each $V_k$ receives the $k^{th}$ particle from each of the states.

(2) The voters agree that the states they receive are indeed $|D_1\rangle, |D_2\rangle$ by using a cut-and choose technique. Specifically, voter $V_k$ chooses at random $2^{\delta_0}$ of the $|D_1\rangle$ states and asks the other voters to measure half of their particles in the computational and half in the Fourier basis. Whenever the chosen basis is the computational, the measurement results need to add up to 0, while when the basis is the Fourier, then the measurement results are all the same. All voters simultaneously broadcast their results and if one of them notices a discrepancy, the protocol aborts. The states $|D_2\rangle$ are similarly checked.

(3) The voters are left to share $N$ copies of $|D_1\rangle$ states and one $|D_2\rangle$ state. Each voter holds one qudit for each state. They now all measure their qudits in the computational basis. As a result, each $V_k$ holds a "blank ballot" of dimension $N$ with the measurement outcomes corresponding to parts of $|D_1\rangle$ states:

$$B_k = [\xi_k^1 \cdots \xi_k^{sk_k} \cdots \xi_k^N]^\intercal$$

and a unique index, $sk_k \in \{1, \dots, N\}$, from the measurement outcome of the qudit that belongs to $|D_2\rangle$. The set of all the blank ballots has the property $\sum_{k=1}^{N} \xi_k^j = 0 \mod c$ for all $j = 1, \dots, N$.

**Casting phase**

(4) Based on $sk_k$, all voters add their vote, $v_k \in \mathbb{Z}_c$, to the corresponding row of their "secret" column. Specifically, $V_k$ applies $\xi_k^{sk_k} \to \xi_k^{sk_k} + v_k$.

(5) All voters simultaneously broadcast their columns, resulting in a public $N \times N$ table, whose $k$-th column encodes $V_k$'s candidate choice.

$$B = \begin{bmatrix} & & \xi_k^1 & & \\ & & \vdots & & \\ B_1^{v_1} & \cdots & \xi_k^{sk_k} + v_k & \cdots & B_N^{v_N} \\ & & \vdots & & \\ & & \xi_k^N & & \end{bmatrix}$$

**Tally phase**

(6) Each $V_k$ verifies that their vote is counted by checking that the corresponding row of the matrix adds up to their vote. If this fails, the protocol aborts.
(7) Each voter can tally the final outcome of the election by computing the sum of the elements of each row of the public $N \times N$ table. The resulting $N$ elements are the result of the election.

### 4.2 Vulnerabilities Of Dual Basis Measurement Protocols

In this section we present an attack on the cut-and-choose technique of the protocol in the setup phase, that can be used to violate privacy. We consider a static adversary that corrupts $t$ voters, including the one that distributes the states. Suppose that the adversary corrupts $N$ out of $N + N2^{\delta_0}$ states $|D_1\rangle$. We denote with *Bad*, the event that all the corrupted voters choose last which states they want to test, and with *Win*, the event that the $N$ corrupted states are not checked. We want to compute the probability that event *Win* happens, given event *Bad*, *i.e.* the probability none of the $N$ corrupted states is checked by the honest voters, and therefore remain intact until the corrupted voters' turn. The corrupted voters will of course not sample any of the corrupted states and therefore the corrupted states will be accepted as valid.

The number of corrupted states that an honest voter will check, follows a mixture distribution with each mixture component being one of the hypergeometric distributions $\{\mathrm{HG}(L_{i_k}, b_{i_k}, 2^{\delta_0}) : 0 \leq b_{i_k} \leq N\}$ , where $L_{i_k}$ is the number of states left to sample from the previous voter and $b_{i_k}$ the number of the remaining corrupted states. We can therefore define the random variable $X_{i_k}$ that follows the above mixture distribution, where $i_1, \ldots, i_{N-t}$ is a permutation of the honest voters' indices (by slightly abusing notation, we consider the first $N - t$ voters to be honest). The following lemma is proven by induction:

Lemma 4.1. *Let $X_{i_k}$ be a random variable that follows the previous mixture distribution. Then,*

$$\Pr[\sum_{k=1}^{N-t} X_{i_k} = 0] = \prod_{k=1}^{N-t} \Pr[X^*_{i_k} = 0] \text{ where } X^*_{i_k} \sim \mathrm{HG}(L_{i_k}, N, 2^{\delta_0}).$$

We are now ready to prove that with at least a constant probability, the corrupted states will remain intact until the end of the verification process.

Proposition 4.2. *For $0 < \varepsilon < 1$, let $t = \varepsilon N$ be the fraction of voters controlled by the adversary. It holds that :*

$$\Pr[Win \mid Bad] > \left(\frac{\varepsilon}{2}\right)^N$$

Proof.

$$\begin{aligned}
\Pr[Win \mid Bad] &= \Pr[\sum_{k=1}^{N-t} X_{i_k} = 0] = \prod_{k=1}^{N-t} P[X^*_{i_k} = 0] \\
&= \prod_{k=0}^{N-t-1} \binom{N + N2^{\delta_0} - N - k2^{\delta_0}}{2^{\delta_0}} \Big/ \binom{N + N2^{\delta_0} - k2^{\delta_0}}{2^{\delta_0}} \\
&= \frac{(N + t2^{\delta_0} - N + 1) \cdot \ldots \cdot (N + t2^{\delta_0})}{(N + N2^{\delta_0} - N + 1) \cdot \ldots \cdot (N + N2^{\delta_0})} \\
&> \left(\frac{t2^{\delta_0} + 1}{N + N2^{\delta_0}}\right)^N = \left(\frac{t2^{\delta_0}}{N + N2^{\delta_0}} + \frac{1}{N + N2^{\delta_0}}\right)^N \\
&> \left(\frac{t2^{\delta_0}}{N + N2^{\delta_0}}\right)^N = \left(\frac{\varepsilon}{2^{-\delta_0} + 1}\right)^N > \left(\frac{\varepsilon}{2}\right)^N
\end{aligned}$$

□

The question now is with what probability event *Bad* occurs, *i.e* how likely is the fact that voters controlled by the adversary are asked to sample last? The answer is irrelevant, because this probability depends on $N$ and $t$, and are both independent of $\delta_0$. As a result,

$$\Pr[Win] > \Pr[Win \mid Bad] \Pr[Bad] = (\varepsilon/2)^N f(N,t)$$

where $f(N,t)$ is a constant function with respect to the security parameter $\delta_0$, making $\Pr[Win]$ non-negligible in $\delta_0$. As a matter of fact, a static adversary will corrupt the voters that maximize $\Pr[Bad]$. Therefore, we can assume that the honest voters sample the states at random, in order to not favor sets of corrupted voters. Now let us examine how this affects the privacy of the scheme.

Theorem 4.3. *Let $\Pi(N,t,\delta_0)$ be an execution of the self-tallying protocol with $N$ voters, $t$ of them corrupted, and $\delta_0$ the security parameter. We can construct an adversary $\mathcal{A}$, which with non-negligible probability in $\delta_0$ violates privacy.*

Proof. Let $C_{\mathcal{A}}$ be the set of indices of the corrupted voters with $|C_{\mathcal{A}}| = t$. Suppose the voter distributing the states is also corrupted, and prepares $1 + N2^{\delta_0}$ states of the form of $|D_2\rangle$, $N2^{\delta_0}$ states of the form $|D_1\rangle$ and $N$ states of the form:

$$|D_{\text{Corrupt}}\rangle = |\xi_1\rangle \otimes \ldots \otimes |\xi_N\rangle$$

where $\xi_k \in_R \{0,\ldots,c-1\}$ for all $k \in \{2,\ldots,N\}$, $\xi_1 \in \{0,\ldots,c-1\}$ such that[3]:

$$\xi_1 + \ldots + \xi_N = 0 \mod c$$

From Proposition 4.2 and the previous observations we know that the probability that states $|D_{\text{Corrupt}}\rangle$ remain intact after the verification procedure in step 2 (*i.e.* event *Win*), happens with non-negligible probability in the security parameter $\delta_0$. Therefore, with non-negligible probability, the remaining states in step 3 are: one of the form $|D_2\rangle$ and $N$ of the form $|D_{\text{Corrupt}}\rangle$. All honest voters $V_k$ measure their qudits in the computational basis and end up with a secret number $sk_k$ (from measuring the corresponding part of $|D_2\rangle$) and a column

$$B_k = [\xi_k^1 \ \ldots \ \xi_k^{sk_k} \ \ldots \ \xi_k^N]^\intercal$$

(from measuring states $|D_{\text{Corrupt}}\rangle$), that is known to the adversary. Now all voters apply their vote $v_k$ to the $B_k$ according to $sk_k$. As a result:

$$B_k^{v_k} = [\xi_k^1 \ \ldots \ \xi_k^{sk_k} + v_k \ \ldots \ \xi_k^N]^\intercal$$

At this point all voters simultaneously broadcast their $B_k^{v_k}$, as the protocol specifies, and end up with the matrix $B = (B_1^{v_1} \ldots B_N^{v_N})$. Each $V_k, k \notin C_{\mathcal{A}}$ checks that

$$\sum_{j=1}^{N} B[sk_k, j] = v_k \mod c$$

which happens with probability 1 from the description of the attack in the previous steps. As a result, each voter accepts the election result. The adversary knowing both the pre-vote matrix and the post-vote matrix can therefore extract the vote of all honest voters. □

A similar attack can be mounted if the adversary instead of corrupting $N$ out of $N + N2^{\delta_0}$ $|D_1\rangle$ states, corrupts just 1 of the $|D_2\rangle$ states. The attack is similar to the one mentioned above but in this case the adversary knows the row in which each voter voted instead of the pre-vote matrix. Moreover, the probability of theorem 4.2 is improved from $(\varepsilon/2)^N$ to $\varepsilon/2$ (the proof works in a similar way).

[3] $\in_R$ denotes that the element is chosen uniformly at random from a specific domain.

Based on the previous observations we can construct an adversary that violates $\epsilon$-**quantum privacy** for any $0 < \epsilon < 1$ and any non-trivial permutation.

Theorem 4.4. *The quantum self tallying protocol* $\Pi(N, N/\epsilon, \delta_0)$ *doesn't satisfy the* $\epsilon$-**quantum privacy** *property for any* $0 < \epsilon < 1$.

Proof. (sketch) First $\mathcal{A}$ picks a non-trivial permutation $\mathsf{F}^{\mathcal{A}}$. It is easy to see that $\mathcal{A}$ can corrupt the quantum states in experiment $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ and $C$ will accept with probability at least $\alpha(\delta_0)$ the corrupted parameters, where $\alpha()$ a non negligible function, based on proposition 4.2. Next, $\mathcal{A}$ can read all the quantum registers $\mathcal{X}_j$ one by one when the oracle query $O_{\Pi,\mathcal{A}}(\mathcal{X}_j, \mathcal{X}_R)$ is issued, and therefore find out how each voter voted as in theorem 4.3. As a result, $\mathcal{A}$ can find out whether the honest voters have permuted their votes and guess the challenge bit $\beta$ with probability at least $1/2 + \alpha(\delta_0)$. Note that in this case, $\Pr[\mathbf{EXP}^{\Pi}_{\text{Qpriv}} \to -1] = 0$ , because $\mathcal{A}$ can choose which voters to corrupt such that both the cut-and-choose attack in the `Setup phase` is successful and the condition $\{v_k : V_k \notin \mathcal{V}_{\mathcal{A}}\} = \{v_{\mathsf{F}(k)} : V_k \notin \mathcal{V}_{\mathcal{A}}\}$ holds. □

So far we have seen how voters' privacy can be violated if an adversary distributes the quantum states in the protocol. However, even if the sharing of the states is done honestly by a trusted authority, still an adversary $\mathcal{A}$ can violate the privacy of a voter. This is done by replacing one element in a column of one of the players controlled by $\mathcal{A}$ with a random number. As a result, in step 6, the honest voter whose row doesn't pass the test, will abort the protocol by broadcasting it. $\mathcal{A}$ will therefore know the identity of the voter aborting and their corresponding vote, since it knows the matrix before the modification of the column element. Similarly, in experiment $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ the adversary can find out if the voters permute their vote or not.

A possible solution might be to use a classical anonymous broadcast channel, so that the voters can anonymously broadcast abort if they detect any misbehaviour at step 6. However, this might open a path to other types of attacks, like denial-of-service, and requires further study in order to be a viable solution.

## 5 TRAVELING BALLOT BASED PROTOCOLS

In this section we discuss the traveling ballot family of protocols for referendum type elections. Here, $T$ also plays the role of $EA$, as it sets up the parameters of the protocol in addition to producing the election result. Specifically, it prepares two entangled qudits, and sends one of them (the *ballot qudit*) to travel from voter to voter. When the voters receive the ballot qudit, they apply some unitary operation according to their vote and forward the qudit to the next voter. When all voters have voted, the ballot qudit is sent back to $T$ who measures the whole state to compute the result of the referendum. The first quantum scheme in this category was introduced by Vaccaro *et al.* [50] and later improved [7, 25, 33].

### 5.1 Protocol Specification

Here we present the travelling ballot protocol of [25]; an alternative form [50] encodes the vote in a phase factor rather than in the qudit itself.

**Set up phase** $T$ prepares the state $|\Omega_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_V |j\rangle_T$, keeps the second qudit and passes the first (the ballot qudit) to voter $V_1$.

**Casting phase** For $k = 1, \ldots, N$, $V_k$ receives the ballot qudit and applies the unitary $U^{v_k} = \sum_{j=0}^{N-1} |j+1\rangle \langle j|$, where $v_k = 1$ signifies a "yes vote and $v_k = 0$ a "no" vote (i.e. applying the identity operator). Then, $V_k$ forwards the ballot qudit to the next voter $V_{k+1}$ and $V_N$ to to $T$.

**Tally phase** The global state held by $T$ after all voters have voted, is:

$$|\Omega_N\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j+m\rangle_V |j\rangle_T$$

where $m$ is the number of "yes" votes. $T$ measures the two qudits in the computational basis, subtracts the two results and obtains the outcome $m$.

### 5.2 Vulnerabilities Of Traveling Ballot Based Protocols

The first obvious weakness of this type of protocols is that they are subject to double voting. A corrupted voter can apply the "yes" unitary operation many times without being detected (this issue is addressed in the next session, where we study the distributed ballot voting schemes). As a result we can easily construct an adversary $\mathcal{A}$ that wins in the correctness experiment described in the Appendix (Figure A) with probability 1. Furthermore, these protocols are subject to privacy attacks, when several voters are colluding. In what follows, we describe such an attack on privacy, in the case of two colluding voters. Figure 1 depicts this attack.

Let us assume that the adversary corrupts voters $V_{k-1}$ and $V_{k+1}$ for any $k$. Upon receipt of the ballot qudit, instead of applying the appropriate unitary, $V_{k-1}$ performs a measurement on the traveling ballot in the computational basis. As a result the global state becomes $|\Omega_{k-1}\rangle = |h+m\rangle_V \otimes |h\rangle_T$, where $|h+m\rangle_V$ is one of the possible eigenstates of the observable $O = \sum_{j=0}^{N-1} |j\rangle \langle j|$, and $m$ is the number of "yes" votes cast by the voters $V_1, \ldots, V_{k-2}$ (note that $V_{k-1}$ does not get any other information about the votes of the previous voters, except number $h+m$). Then $V_{k-1}$ passes the ballot qudit $|h+m\rangle_V$ to $V_k$, who applies the respective unitary for voting "yes" or "no". As a result the ballot qudit is in the state $|h+m+v_k\rangle_V$. Next, the ballot qudit is forwarded to the corrupted voter $V_{k+1}$, who measures it again in the computational basis and gets the result $h+m+v_k$. $\mathcal{A}$ can now infer vote $v_k$ from the two measurement results and figure out how $V_{k+1}$ voted. Similarly, $\mathcal{A}$ can guess the correct bit $\beta$ in $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ with probability 1 by measuring the quantum registers $\mathcal{X}_{k-1}$ and $\mathcal{X}_{k+1}$, where $V_k$ an honest party ($\mathcal{A}$ might need to corrupt two more voters such that $\mathbf{EXP}^{\Pi}_{\text{Qpriv}}$ will not output −1). The same attack can also be applied in the case where there are many voters between the two corrupted parties. In this case the adversary can't learn the individual votes but only the total votes. One suggestion presented in [50] is to allow $T$ to perform extra measurements to detect a malicious action during the protocol's execution. However, this only identifies an attack and does not prevent the adversary from learning some of the votes, as described above. Furthermore, the probability of detecting a deviation from the protocol is constant and as such does not depend on the security parameter and does not lead to a substantial improvement of security. It should also be noted that verifiability of the election result is not addressed in any of these works, since $T$ is assumed to generate the initial state honestly. In the case where $T$ is corrupted, privacy is trivially violated.

All traveling ballot protocols proposed ([7, 25, 33, 50]) suffer from the above privacy attack. Next, we discuss how this issue has been addressed by revisiting the structure of the protocols. Unfortunately, as we will see, new issues arise.

## 6 DISTRIBUTED BALLOT BASED PROTOCOLS

Here we describe the family of quantum distributed ballot protocols [7, 25, 50]. In these schemes, $T$ prepares and distributes to each voter a blank ballot, and gathers it back after all voters have cast their vote in order to compute the final outcome. This type of protocols give strong guarantees for privacy against other voters but not against a malicious $T$ which is trusted to prepare correctly

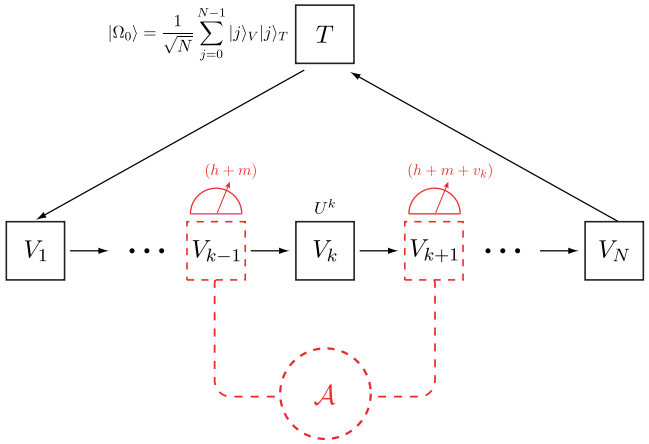


Fig. 1. $\mathcal{A}$ corrupts voters $V_{k-1}$,$V_{k+1}$ and learns how voter $V_k$ voted with probability 1.

specific states. So it is not hard to see that if the states are not the correct ones, then the privacy of a voter can be violated.

A first attempt presented in [50] suffers from double voting similarly to the discussion in the previous section. The same problem also appears in [19]. Later works [7, 25] address this issue with a very elaborate countermeasure. The intuition behind the proposed technique is that $T$ chooses a secret number $\delta$ according to which it prepares two different quantum states: the "yes" and the "no" states. This $\delta$ value is hard to predict due to the non-orthogonality of the shared states and the no-cloning theorem. The authors suggest that many rounds of the protocol be executed. As a result, any attempt of the adversary to learn $\delta$ gives rise to a different result in each round. However, the number of required rounds, as well as a rigorous proof are not presented in the study.

More importantly, a careful analysis reveals that the proposed solution is still vulnerable to double voting. As we will see, an adversary can mount what we call a $d$-transfer attack, and transfer $d$ votes for one option of the referendum election to the other. To achieve this attack, the adversary does not need to find the exact value of $\delta$ (as the authors believed), but knowing the difference of the angles used to create the "yes" and "no" states suffices. We construct a quantum polynomial-time adversary that performs the $d$-transfer attack with probability at least 0.25, if the number of rounds is smaller than exponential in the number of voters. As a result this makes the protocol practically unrealistic for large scale elections.

### 6.1 Protocol Specification

We first present the protocol from [7, 25]:

**Setup phase**

(1) $T$ prepares an $N$-qudit ballot state: $|\Phi\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |j\rangle^{\otimes N}$, where the states $|j\rangle$, $j = 0, ..., D-1$, form an orthonormal basis for the $D$-dimensional Hilbert space, and $D > N$. The $k$-th qudit of $|\Phi\rangle$ corresponds to $V_k$'s blank ballot.

(2) $T$ sends to $V_k$ the corresponding blank ballot together with two option qudits, one for the "yes" and one for the "no" option:

$$\text{yes:} \quad |\psi(\theta_y)\rangle = \tfrac{1}{\sqrt{D}} \textstyle\sum_{j=0}^{D-1} e^{ij\theta_y} |j\rangle, \text{no:} \quad |\psi(\theta_n)\rangle = \dfrac{1}{\sqrt{D}} \displaystyle\sum_{j=0}^{D-1} e^{ij\theta_n} |j\rangle$$

For $v \in \{y, n\}$ we have $\theta_v = (2\pi l_v/D) + \delta$, where $l_v \in \{0, \ldots, D-1\}$ and $\delta \in [0, 2\pi/D)$. Values $l_y$ and $\delta$ are chosen uniformly at random from their domain and $l_n$ is chosen such that $N(l_y - l_n \mod D) < D$ . These values are known only to $T$.

**Casting phase**

(3) Each $V_k$ decides on "yes" or "no" by appending the corresponding option qudit to the blank ballot and performing a 2-qudit measurement $R = \sum_{r=0}^{D-1} rP_r$, where:

$$P_r = \sum_{j=0}^{D-1} |j+r\rangle \langle j+r| \otimes |j\rangle \langle j|$$

According to the result $r_k$, $V_k$ performs a unitary correction $U_{r_k} = \mathbb{I} \otimes \sum_{j=0}^{D-1} |j + r_k\rangle \langle j|$ and sends the 2-qudits ballot along with $r_k$ back to $T$.

**Tally phase**

(4) The global state of the system (up to normalization) is:

$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} \prod_{k=1}^{N} \alpha_{j,r_k} |j\rangle^{\otimes 2N}$$

where,

$$\alpha_{j,r_k} = \begin{cases} e^{i(D+j-r_k)\theta_v^k}, & 0 \le j \le r_k - 1 \\ e^{i(j-r_k)\theta_v^k} & r_k \le j \le D-1 \end{cases}$$

(5) For every $k$, using the announced results $r_k$, T applies the unitary operator:

$$W_k = \sum_{j=0}^{r_k-1} e^{-iD\delta} |j\rangle \langle j| + \sum_{j=r_k}^{D-1} |j\rangle \langle j|$$

on one of the qudits in the global state (it is not important on which one, since changes to the phase factor of a qudit that is part of a bigger entangled state take effect globally). Now $T$ has the state:

$$|\Omega_m\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ij(m\theta_y + (N-m)\theta_n)} |j\rangle^{\otimes 2N}$$

where $m$ is the number of "yes" votes.

(6) By applying the unitary operator $\sum_{j=0}^{D-1} e^{-ijN\theta_n} |j\rangle \langle j|$ on one of the qudits and setting $q = m(l_y - l_n)$, we have:

$$|\Omega_q\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{2\pi ijq/D} |j\rangle^{\otimes 2N}$$

We note here that $q$ must be between 0 and $D-1$, so that the different outcomes be distinguishable. Now with the corresponding measurement $T$ can retrieve $q$. Since $T$ knows values $l_y$ and $l_n$, it can derive the number $m$ of "yes" votes. Note that if a voter does not send back a valid ballot, the protocol execution aborts.

### 6.2 Vulnerabilities Of Distributed Ballot Protocols

In this section, we show how the adversary can perform the $d$-transfer attack in favor of the "yes" outcome. We proceed as follows. We first show that this is possible if the adversary knows the difference $l_y - l_n$. We then show how the adversary can find out this value, and conclude the section with the probabilistic analysis of our attack which establishes that it can be performed with overwhelming probability in the number of voters.

**The d-transfer attack:** Given the difference $l_y - l_n$, a dishonest voter can violate the no-double-voting. From the definition of $l_y$ and $l_n$ it holds that:

$$2\pi(l_y - l_n)/D = \theta_y - \theta_n \tag{1}$$

If a corrupted voter (e.g. $V_1$) knows $l_y - l_n$, then they proceed as follows (w.l.o.g. we assume that they want to increase the number of "yes" votes by $d$):

(1) $V_1$ applies the unitary operator: $C_d = \sum_{j=0}^{D-1} e^{ijd(\theta_y-\theta_n)} |j\rangle\langle j|$ to the received option qudit $|\psi(\theta_y)\rangle$. As a result, the state becomes:

$$C_d |\psi(\theta_y)\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ijd(\theta_y-\theta_n)} e^{ij\theta_y} |j\rangle$$

(2) $V_1$ now performs the 2-qudit measurement specified in the Casting phase of the protocol and obtains the outcome $r_1$.

(3) $V_1$ performs the unitary correction $U_{r_1}$. For $\tilde{\theta} = d(\theta_y - \theta_n) + \theta_y$, the global state now is:

$$U_{r_1} P_{r_1} \big( |\Phi\rangle \otimes C_d |\psi(\theta_y)\rangle \big) =$$
$$\frac{1}{\sqrt{D}} \Big[ \sum_{j=0}^{r_1-1} e^{i(D+j-r_1)\tilde{\theta}} |j\rangle^{\otimes N+1} + \sum_{j=r_1}^{D-1} e^{i(j-r_1)\tilde{\theta}} |j\rangle^{\otimes N+1} \Big]$$

(4) Before sending the two qudit ballot and the value $r_1$ to $T$, $V_1$ performs the following operation to the option qudit:

$$\textbf{Correct}_{r_1} = \begin{cases} e^{-iDd(\theta_y-\theta_n)} |j\rangle\langle j|, & 0 \le j \le r_1 - 1 \\ |j\rangle\langle j| & r_1 \le j \le D-1 \end{cases}$$

(5) After all voters have cast their ballots to $T$, the global state of the system (up to normalization) is:

$$\frac{1}{\sqrt{D}} \Big( \sum_{j=0}^{r_1-1} e^{i(j-r_1)d(\theta_y-\theta_n)} e^{i(D+j-r_1)\theta_y} \prod_{k=2}^{N} \alpha_{j,r_k} |j\rangle^{\otimes 2N}$$
$$+ \sum_{j=r_1}^{D-1} e^{i(j-r_1)d(\theta_y-\theta_n)} e^{i(j-r_1)\theta_y} \prod_{k=2}^{N} \alpha_{j,r_k} |j\rangle^{\otimes 2N} \Big)$$

where,

$$\alpha_{j,r_k} = \begin{cases} e^{i(D+j-r_k)\theta_v^k}, & 0 \le j \le r_k - 1 \\ e^{i(j-r_k)\theta_v^k} & r_k \le j \le D-1 \end{cases}$$

and $\theta_v^k$ describes the vote of voter $V_k$, where $v \in \{y, n\}$. $T$ just follows the protocol specification. It applies some corrections on the state given the announced results $r_k$ and finally the state becomes:

$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{i(j-r_1)d(\theta_y-\theta_n)} e^{i(j-r_1)\theta_y} \cdot \ldots \cdot e^{i(j-r_n)\theta_v^n} |j\rangle^{\otimes 2N}$$

which under a global phase factor is equivalent to:

$$\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ijd(\theta_y-\theta_n)} e^{ij(m\theta_y+(N-m)\theta_n)} |j\rangle^{\otimes 2N}$$

(6) $T$ removes the unwanted factor $e^{ijN\theta_n}$ as prescribed by the protocol, and the final state is:

$$|\Omega_{m+d}\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ijd(\theta_y-\theta_n)} e^{ijm(\theta_y-\theta_n)} |j\rangle^{\otimes 2N}$$
$$= \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{2\pi ij(m+d)(l_y-l_n)/D} |j\rangle^{\otimes 2N}$$

(7) After measuring the state, the result is $m + d$ instead of $m$.

**Finding the difference between $l_y$ and $l_n$:** What remains in order to complete our attack is to find the difference $l_y - l_n$. We now show how an adversary can learn this difference with overwhelming probability in $N$. We assume that the adversary controls a fraction $\varepsilon$ of the voters ($0 < \varepsilon < 1$), who are (all but one) instructed to vote half the times "yes" and the other half "no". Instead of destroying the remaining option qudits (exactly $\varepsilon N/2$ "yes" and $\varepsilon N/2$ "no" votes), the adversary keeps them to run Algorithm 1. In essence, the algorithm is executed twice - once for each set of option qudits

**Algorithm 1** Adversary's algorithm

**Input:** $D, |\psi(\theta_v)\rangle_1, \cdots, |\psi(\theta_v)\rangle_{\varepsilon N/2}$
**Output:** $\tilde{l} \in \{0, \ldots, D-1\}$
1: `Record` $= [0, \ldots, 0] \in \mathbb{N}^{1\times D}$; ▷ This vector shows us how many values are observed in each interval
2: `Solution` $= [$"$Null$","$Null$"$] \in \mathbb{N}^{1\times 2}$;
3: $i, l, m = 0$;
4: **while** $i \leq \varepsilon N/2$ **do**
5: Measure $|\psi(\theta_v)\rangle_i$ by using POVM operator $E(\theta)$ from Eq.(2), the result is $y_i$;
6: Find the interval for which $\frac{2\pi j}{D} \leq y_i \leq \frac{2\pi(j+1)}{D}$;
7: `Record`$[j]$ =++;
8: $i$++;
9: **end while**
10: **while** $l < D$ **do**
11: **if** `Record`$[l] \geq 40\%(\varepsilon N/2)$ **then**
12: `Solution`$[m] = l$;
13: $m++$;
14: **end if**
15: $l++$;
16: **end while**
17: **if** `Solution` $== [0, D-1]$ **then**
18: `Solution` $= [$`Solution`$[1],$ `Solution`$[0]]$;
19: **end if**
20: **return** $\tilde{l} =$ `Solution`$[0]$;

$\{|\psi(\theta_v)\rangle\}_{\varepsilon N/2}$, where $v \in \{y, n\}$. It measures the states in each set and attributes to each one an integer. After all states have been measured, the algorithm creates a vector `Record`, which contains the number of times each integer appeared during the measurements. Finally, Algorithm 1 creates the vector `Solution` in which it registers the values that appeared at least 40% of times during the measurements, equivalently the values for which the `Record` vector assigned a number greater or equal than 40% of times. The algorithm outputs the first value in the `Solution` vector. As we see in Figure 2, with high probability the value that algorithm outputs is either $l_v$ or $l_v - 1$, for both values of $v$. Hence, we can find the difference $l_y - l_n$. After having acquired knowledge of $l_y - l_n$, the adversary can instruct the last corrupted voter to change the outcome of the voting process as previously described.

**Probabilistic analysis** We prove here that the adversary's algorithm succeeds with overwhelming probability in $N$, where $N$ is the number of voters. Therefore, as we later prove in Theorem 6.8, the election protocol needs to run at least exponentially many times with respect to $N$ in order

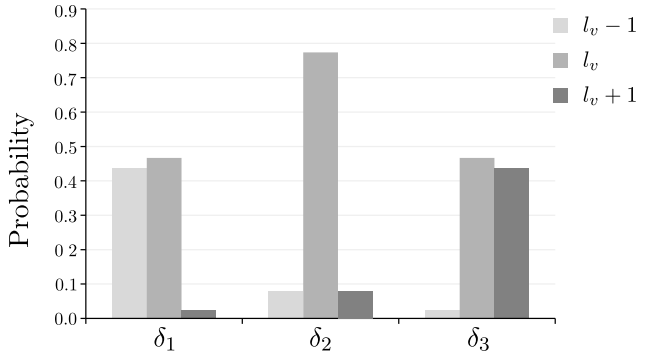


Fig. 2. The probabilities with which Algorithm 1 records a value in $\{l_v - 1, l_v, l_v + 1\}$ after measuring state $|\psi(\theta_v)\rangle$ for $\delta_1 = \frac{\pi}{2^{35}}$, $\delta_2 = \frac{\pi}{2^{30}}$, and $\delta_3 = \frac{\pi(2^6-1)}{2^{35}}$.

to guarantee that the success probability of the adversary is at most 0.25. We present here the necessary lemmas and give the full proofs in the Supplementary Material.

In order to compute the success probability of the attack, we first need to compute the probability of measuring a value in the interval $(x_l, x_{l+w})$, where $x_l = \dfrac{2\pi l}{D},\ l \in \{0, 1, \ldots, D-1\}$[4].

Lemma 6.1. *Let $\Theta^v_{D,\delta} \in [0, 2\pi]$ be the continuous random variable that describes the outcome of the measurement of an option qudit $|\psi(\theta_v)\rangle, v \in \{y, n\}$ using operators:*

$$E(\theta) = \frac{D}{2\pi} |\Phi(\theta)\rangle \langle\Phi(\theta)| \tag{2}$$

*where $|\Phi(\theta)\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ij\theta} |j\rangle$. It holds that:*

$$\Pr[x_l < \Theta^v_{D,\delta} < x_{l+w}] = \frac{1}{2\pi D} \int_{x_l}^{x_{l+w}} \frac{\sin^2[D(\theta - \theta_v)/2]}{\sin^2[(\theta - \theta_v)/2]} d\theta$$

According to Algorithm 1, an option qudit is attributed with the correct value $l_v$ when the result of the measurement is in the interval $[x_{l_v}, x_{l_v+1}]$. Using Lemma 6.1, we can prove the following:

Lemma 6.2. *Let $|\psi(\theta_v)\rangle$ be an option qudit of the protocol. Then it holds:*

$$\Pr[x_{l_v} < \Theta^v_{D,\delta} < x_{l_v+1}] > 0.405$$

Lemma 6.2 shows that with probability at least 0.405, the result of the measurement is in the interval $(x_{l_v}, x_{l_v+1})$. Since Algorithm 1 inserts an integer to the `Solution` vector if it corresponds to at least 40% of the total measured values, $l_v$ will most likely be included in the vector (we formally prove it later). Furthermore, we prove now that with high probability, there will be no other values to be inserted in `Solution`, except the neighbours of the value $l_v$ (namely $l_v \pm 1$).

Lemma 6.3. *Let $|\psi(\theta_v)\rangle$ be an option qudit of the protocol. Then it holds:*

$$Pr[x_{l_v-1} < \Theta^v_{D,\delta} < x_{l_v+2}] > 0.9$$

Here we need to note that we are aware of the cases $l_v \in \{0, D-1\}$ where the members $x_{l_v-1}$ and $x_{l_v+2}$ are not defined. It turns out not to be a problem and the same thing can be proven for these values (see Supplementary Material).

[4]It is convenient to think of $l$ as the $D^{th}$ roots of unity.

We have shown that the probability the measurement outcome lies in the interval $(x_{l_v-1}, x_{l_v+2})$, and therefore gets attributed with a value of $l_v - 1$, $l_v$ or $l_v + 1$, is larger than 0.9. If we treat each measurement performed by Algorithm 1 on each option qudit $|\psi(\theta_v)\rangle$, as an independent Bernoulli trial with success probability $p_l = Pr[x_l < \Theta^v_{D,\delta} < x_{l+1}]$, we can prove the following theorem:

Theorem 6.4. *With overwhelming probability in the number of voters N, Algorithm 1 includes $l_v$ in the* `Solution` *vector*

$$\Pr[\texttt{Solution}[0] = l_v \vee \texttt{Solution}[1] = l_v] > 1 - 1/exp(\Omega(N))$$

We have proven that with overwhelming probability in $N$, integer $l_v$ occupies one of the two positions of vector `Solution`, but what about the other value? In the next theorem, we show that with overwhelming probability in $N$, the other value is one of the neighbours of $l_v$, namely $l_v + 1$ or $l_v - 1$.

Theorem 6.5. *With negligible probability in the number of voters N, Algorithm 1 includes a value other than $(l_v - 1, l_v, l_v + 1)$ in the* `Solution` *vector,* i.e. $\forall w \in \{0, \ldots, l_v - 2, l_v + 2, \ldots, D - 1\}$ *:*

$$\Pr[\texttt{Solution}[0] = w \vee \texttt{Solution}[1] = w] < 1/exp(\Omega(N))$$

Lemma 6.6. *With overwhelming probability in N, the* `Solution` *vector in Algorithm 1, is equal to $[l_v - 1, l_v]$, $[l_v, "Null"]$ or $[l_v, l_v + 1]$. Specifically,*

$$\Pr[\texttt{Solution} \in \{[l_v - 1, l_v], [l_v, "Null"], [l_v, l_v + 1]\}] > 1 - 1/exp(\Omega(N))$$

Now consider we have two executions of the Algorithm 1, one for the "yes" and one for the "no" option qudits. It turns out that the values in the positions $l_y - 1$ and $l_n - 1$ of the vector `Record`, follow the same Binomial distribution (it is easy to see that $p_{l_y-1} = p_{l_n-1}$). Also, each of them can be seen as a function of $\delta$ which is a monotonic decreasing function that takes a maximum value for $\delta = 0$ (the proof technique is similar to Lemma 6.2). At this point the probability is equal to $p_{l_v}$, which is at least 0.405 as we have proven in Lemma 6.2[5]. Armed with this observation we can prove the next theorem.

Theorem 6.7. *If we define the event "Cheat" as:*

$$Cheat = \left[Algo(y) - Algo(n) = l_y - l_n\right]$$

*where $Algo(v)$ is the execution of Algorithm 1 with $v \in \{y, n\}$, then it holds that:*

$$Pr["Cheat"] > 1 - 1/exp(\Omega(N))$$

Proof. (sketch) We have seen that there exists a $\delta_0$ such that the probability $p_{l_v-1}$ is equal to 0.4 for both values of $v$. It holds that:

$$\begin{aligned} Pr["Cheat"] &= Pr["Cheat"|\delta \in [0, \delta_0)] \cdot Pr[\delta \in [0, \delta_0)] \\ &+ Pr["Cheat"|\delta = \delta_0] \cdot Pr[\delta = \delta_0] \\ &+ Pr["Cheat"|\delta \in (\delta_0, 2\pi/D)] \cdot Pr[\delta \in (\delta_0, 2\pi/D)] \end{aligned}$$

For the first interval, for both values of $v$, Algorithm 1 registers $\texttt{Solution} = [l_v - 1, l_v]$ with overwhelming probability in $N$. This holds because of Theorem 6.6 and the previous observation. Therefore, for both values of $v$ the algorithm outputs the values $l_v - 1$. As a result, $l_y - 1 - (l_n - 1) = l_y - l_n$.

For the second term, $Pr[\delta = \delta_0] = 0$, because $\delta$ is a continuous random variable. Finally, in the last term, the probability that the algorithm registers $\texttt{Solution} = [l_v - 1, l_v]$ is negligible in $N$,

[5]The same holds for the $p_{l_y+1}, p_{l_n+1}$ except that probability is a monotonic increasing function with maximum value at point $\delta = 2\pi/D$ and value equal to $p_{l_v}$.

and by Theorem 6.6, `Solution` has the form $[l_v]$ or $[l_v, l_v + 1]$. So for both values of $v$, the printed values are $l_y$ and $l_n$.

□

At this point we have proven that the adversary succeeds with overwhelming probability in $N$ to perform the $d$-transfer attack in one round. But how many rounds should the protocol run in order to prevent this attack?

In the next theorem we prove that if the number of rounds is at most $exp(\Omega(N))$, the adversary succeeds with probability at least 0.25. Although in a small election these numbers might not be big, in a large scale election it is infeasible to run the protocol as many times, making it either inefficient or insecure. We also note that the probabilistic analysis for one round is independent of the value $D$, so cannot be used to improve the security of the protocol.

Theorem 6.8. *Let* $(|\Phi\rangle, |\psi(\theta_y)\rangle, |\psi(\theta_n)\rangle, \delta, D, N)$ *define one round of the protocol. If the protocol runs* $\rho$ *rounds, where* $2 \leq \rho \leq exp(\Omega(N))$ *, the d-transfer attack succeeds with probability at least* 0.25.

Proof. According to Theorem 6.7 the probability that an adversary successfully performs the $d$-transfer attack is:

$$Pr[\text{"}Cheat\text{"}] > 1 - 1/exp(\Omega(N))$$

Now, for $\rho$ protocol runs, where $2 \leq \rho \leq exp(\Omega(N))$, this probability becomes:

$$(Pr[\text{"}Cheat\text{"}])^{\rho} > (1 - 1/exp(\Omega(N)))^{\rho} \geq (1 - 1/\rho)^{\rho} > 0.25$$ □

Now, based on Theorem 6.8, we can create an adversary such that $\mathcal{A}$ wins the $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}$ with probability at least 25% if the protocol runs fewer than exponential number of rounds with respect to the number of voters.

Theorem 6.9. *The adversary from section 6.2 wins the experiment* $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}$*, where* $\Pi$ *is the protocol as described in section 6.1, with probability at least* 0.25% *for every* $\epsilon > 0$ *and number of rounds* $2 \leq \rho \leq exp(\Omega(N))$.

Proof. (sketch) When $\mathcal{A}$ calls the oracle $O_{\Pi,\mathcal{A}}$ on behalf of a corrupted voter $V_k$ in **Casting phase**, $\mathcal{A}$ applies the operations as described in section 6.2 at register $\mathcal{X}_k$. Next, in **Tally phase** $C$ computes the election outcome. If in a round the result is different from a previous round, the election result $X$ will be equal to "$\perp$". However, from Theorem 6.8, we know that $X \neq \perp$ with probability at least 0.25%, in which case the predicate Verify is always 1. □

## 7 QUANTUM VOTING BASED ON CONJUGATE CODING

This section looks at protocols based on conjugate coding [40, 53]. The participants in this family of protocols are one or more election authorities, the tallier and the voters. The election authorities are only trusted for the purpose of eligibility; privacy should be guaranteed by the protocol against both malicious $EA$ and $T$. Unlike the previous protocols, here the voters do not share any entangled states with neither $EA$ nor $T$ in order to cast their ballots. One of the main differences between the two protocols is that [40] does not provide any verification of the election outcome, while [53] does, but at the expense of receipt freeness, which [40] satisfies. Specifically, in [53] each $V_k$ establishes two keys with $T$ in an anonymous way by using part of protocol [40] as a subroutine. It's worth to mention that in order for these keys to be established, further interaction between the voters and $EA$ is required and $EA$ is assumed trusted for that task. At the end of an execution, $V_k$ encrypts the ballot with one of the keys and sends it to $T$ over a quantum anonymous channel. $T$ announces the result of each ballot accompanied with the second key so that the voters can verify that their ballot

has been counted. This makes it also possible for a coercer to verify how a voter voted, by showing them the second key used as a receipt. It is worth mentioning that protocol [40] could easily be made to satisfy the same notion of verifiability.

### 7.1 Protocol Specification

**Set up phase**

(1) *EA* picks a vector $\bar{b} = (b_1, \ldots, b_{n+1}) \in_R \{0,1\}^{n+1}$, where $n$ is the security parameter of the protocol. This vector will be used by *EA* for the encoding of the ballots and it will be kept secret from $T$ until the end of the ballot casting phase.

(2) For each $V_k$, *EA* prepares $w = poly(n)$ blank ballot fragments each of the form $|\phi_{\bar{a}_j,\bar{b}}\rangle = |\psi_{a_j^1,b_1}\rangle \otimes \ldots \otimes |\psi_{a_j^{n+1},b_{n+1}}\rangle$, $j \in \{1,\ldots,w\}$, where $\bar{a}_j = (a_j^1, \ldots, a_j^{n+1})$ such that:

$$(a_j^1, \ldots, a_j^n) \in_R \{0,1\}^n, a_j^{n+1} = a_j^1 \oplus \ldots \oplus a_j^n$$

and: $|\psi_{0,0}\rangle = |0\rangle, |\psi_{1,0}\rangle = |1\rangle, |\psi_{0,1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_{1,1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
These $w$ fragments will constitute a blank ballot (e.g the first row of Fig. 3 is a blank ballot fragment).

(3) *EA* sends one blank ballot to each $V_k$ over an authenticated channel.

**Casting phase**

(4) After reception of the blank ballot, each $V_k$ re-randomizes it by picking for each fragment a vector $\bar{d}_j = (d_j^1, \ldots, d_j^{n+1})$ such that:

$$(d_j^1, \ldots, d_j^n) \in_R \{0,1\}^n, d_j^{n+1} = d_j^1 \oplus \ldots \oplus d_j^n.$$

$\forall j \in \{1,\ldots,w\}$, $V_k$ applies unitary $U_j^{\bar{d}_j} = Y^{d_j^1} \otimes \ldots \otimes Y^{d_j^{n+1}}$ to the blank ballot fragment $|\phi_{\bar{a}_j,\bar{b}}\rangle$ ,, where:

$$Y^1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, Y^0 = \mathbb{I}$$

(5) $V_k$ encodes the candidate of choice in the $(n+1)^{th}$-qubit of the last blank ballot fragments[6]. For example, if we assume a referendum type election, $V_k$ votes for $c \in \{0,1\}$ by applying to the blank ballot fragment $|\phi_{\bar{a}_w,\bar{b}}\rangle$ the unitary operations $U_w^{\bar{c}}$ respectively, where: $\bar{c} = (0, \ldots, 0, c)$ (see Fig. 3).

(6) $V_k$ sends the ballot to $T$ over an anonymous channel.

**Tally phase**

(7) Once the ballot casting phase ends, *EA* announces $\bar{b}$ to $T$.

(8) With this knowledge, $T$ can decode each cast ballot in the correct basis. Specifically, $T$ decodes each ballot fragment by measuring it in the basis described by vector $\bar{b}$ and XORs the resulting bits. After doing this to each ballot fragment, $T$ ends up with a string, which is the actual vote cast.

(9) $T$ announces the election result.

[6]Candidate choices are encoded in binary format.

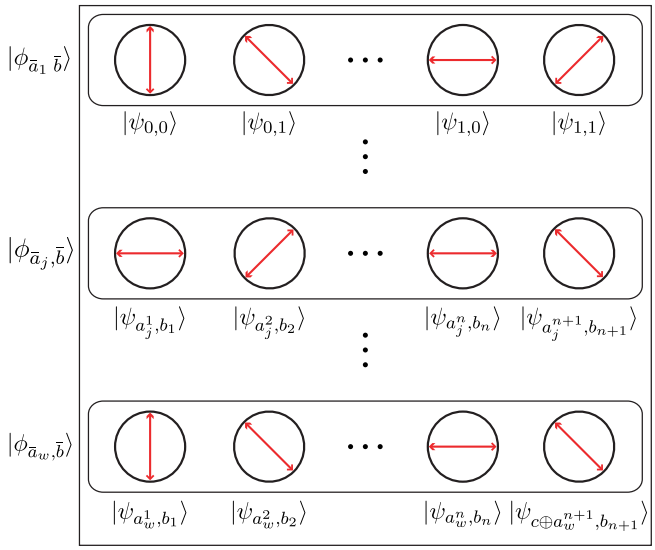


Fig. 3. The ballot consisting of $w$ ballot fragments, which encode the binary choice "0...01" in a referendum type election example.

### 7.2 Vulnerabilities of Conjugate Coding Protocols

The technique underlying this protocol is closely related to the one used in the first quantum key distribution protocols [5, 46]. However, it has some limitations in the context of these voting schemes.

**Malleable blank ballots:** An adversary can change the vote of an eligible voter, when the corresponding ballot is cast over the anonymous channel. Assume $V_k$ has applied the appropriate unitary on the blank ballot in order to vote for the candidate of their choice. And let us consider that the last $m$ ballot fragments encode the candidate. When the adversary sees the cast ballot over the quantum anonymous channel, they apply the unitary $U_{w-(m-1)}^{\tilde{c}_1}, \ldots, U_w^{\tilde{c}_m}$, where $c_r$ is either 0 or 1, depending on their choice to flip the candidate bit or not. As a result the adversary modifies the ballot of $V_k$ such that it decodes to a different candidate than the intended one. This is possible because the adversary is aware of the ballot fragments used to encode the candidate choice. Furthermore, if the adversary has side channel information about the likely winning candidate (from pre-election polls for instance), they will be able to change the vote encoded in the ballot into one of their desire. This is possible because the adversary is aware of which bits are encoded in the ballot more frequently and knows exactly which unitary operator to apply in order to decode to a specific candidate.
**Violation of privacy:** It is already acknowledged by the authors of [40] that the *EA* can introduce a "serial number" in a blank ballot to identify a voter, *i.e* some of the blank ballot fragments in the head of the ballot decode to "1" instead of "0". This allows the *EA* to decode any ballot cast over the quantum anonymous channel, linking the identity of the voters with their choice.
**One-more-unforgeability:** The security of the protocol relies on a quantum problem introduced in [40], named *one-more-unforgeability* and the assumption that it is computationally hard for a quantum adversary. The game that captures this assumption goes as follows: a challenger encodes $w$ blank ballot fragments in a basis $\bar{b}$ and gives them to the adversary. The adversary wins the game if they produce $w + 1$ valid blank ballot fragments in the basis $\bar{b}$. The authors claim the probability of the adversary of winning this game is at most $1/2 + 1/2(negl(n))$.

**On the security parameter:** Because of the ballots' malleability, an adversary could substitute the parts of the corrupted voters' blank ballot fragments that encode a candidate, with blank ballot fragments in a random base. Of course these ballots would open into random candidates in a specific domain but would still be valid, since the leading zeros would not be affected by this change. This is because blank ballots contain no entanglement. Now the adversary can keep these valid spare blank ballot fragments to create new valid blank ballots. To address this problem, the size of blank ballots needs to be substantially big compared to the number of voters and the size of the candidate space ($Nm << w$).

## 8 OTHER PROTOCOLS

Other protocols have also been proposed, with main characteristic that the $EA$ controls when ballots get counted. This can be achieved with either the use of shared entangled states between $EA$, $T$, and $V_k$ [47, 52] or Bell pairs [47] between $T$ and $V_k$ with $EA$ knowing the identity of the holder of each pair particle. However, we do not fully analyse these protocols in this review, as they have many and serious flaws making even the correctness arguable. The protocol of [52] claims to provide verifiability of the election outcome, but without explaining how this can be achieved. From our understanding of the protocol this seems unlikely to be the case. From the description of the protocol each voter can change their mind and announce a different vote from the originally cast one. This is possible because the function every voter uses to encode their vote is not committed in any way. Two protocols introduced in [47] have similar limitations. For instance, there is no mechanism for verifiability of the election outcome. In addition, privacy against $T$ is not satisfied in contrast to protocols we saw in section 6. This is because each voter's vote is handled individually and not in a homomorphic manner. All of these could be achieved just by a classical secure channel. Last, the protocol appearing in [26] shares many of the limitations of the former protocols as well as some further ones. The method introduced for detecting eavesdropping in the election process is insecure, as trust is put into another voter in order to detect any deviation from the protocol specification. Moreover, the way each voter casts their vote is not well defined in the protocol, which makes privacy and correctness trivially violated.

Finally, we note that there exist protocols that consider elections with quantum input, see e.g. [3]. This type of protocols is more relevant to quantum game theory and less to election schemes with classical input/voting choice, and we have not considered them in this study.

## 9 DISCUSSION

In this work, we have examined the current state of the art in quantum e-voting, by presenting the most prominent proposals and analyzing their security. What we have found is that all the proposed protocols fail to satisfy the necessary security definitions for future implementations. Despite this, these protocols open the way to new avenues of research, specifically on whether quantum information can solve some long-standing issues in e-voting and cryptography in general. By studying them, we can identify several interesting ideas for further development as well as possible bottlenecks in future quantum protocols.

For instance, we saw that, unless combined with some new technique, the traveling ballot protocols do not provide a viable solution, as double-voting is always possible, and there is no straightforward way to guarantee privacy. On the other hand, the distributed ballot protocols give us very strong privacy guarantees because of the entanglement between the ballot states, but verifiability against malicious talliers might be hard to achieve. In fact, one of the most intriguing questions in quantum e-voting is whether we can achieve all desired properties simultaneously. For instance, every classical definition of verifiability [15] assumes a trusted bulletin board that the participants can read, write on, and finally verify the outcome of the election. However,

implementing a quantum bulletin board to achieve the same properties is not straightforward, since reading a quantum state can 'disturb' it in an irreversible way. For that reason we have defined in our experiment $\mathbf{EXP}^{\Pi}_{\mathsf{QVer}}$ the public quantum register $\mathcal{B}$, and the predicate Verify. Of course, an implementation of such predicate seems hard to realize in the quantum setting and more research is needed.

We have also shown that the cut-and-choose technique used by the protocols in Section 4 is both inefficient and insecure. A solution could be to provide some type of randomness to the voters (e.g. in the form of a common random string), which would determine if a state should be verified or used for the voting phase (a similar process is shown in [37, 41]). However, even if the problem with the cut-and-choose technique is addressed, privacy can still be violated as we have seen, and fixing this might require the use of more advanced techniques. Notwithstanding these limitations, we believe that our analysis opens new research directions for the study of the quantum cut-and-choose technique, which plays a fundamental role in the secure distribution of quantum information.

The general aim of studying quantum cryptographic protocols, is to provide better guarantees than classically possible, be that in security or efficiency. This has been achieved for primitives like coin flipping and oblivious transfer, against unbounded adversaries [10, 42], and against bounded ones that are more relevant to practical implementations (e.g. with limited storage [17], noisy storage [32], or bounded by relativistic constraints [35]). The question whether quantum technology could enhance electronic voting as well has not yet been answered, and requires further study of both the existing classical and quantum literature. First, bottlenecks in classical election protocols that could potentially be solved by using quantum subroutines, need to be identified. Then, quantum protocols need to be designed, that satisfy well articulated definitions of all the required properties in composable frameworks. In classical cryptography, this was pursued with the help of automated provers and model checkers such as EasyCrypt [14], game based definitions [6, 15], and by employing the Universal Composability Framework [9, 24]. However, in quantum cryptography, it remains unclear how these techniques can be adopted. An interesting approach appears in [23], where the authors provide an automated verification tool that enables checking properties of systems which can be expressed within the quantum stabilizer formalism. Finally, a recent work by Unruh [49] on quantum relational Hoare logic might open new avenues and help provide a solution to this problem.

## REFERENCES

[1] Ben Adida. 2008. Helios: Web-based Open-Audit Voting. In *USENIX security symposium*, Vol. 17. 335–348.
[2] Myrto Arapinis, Véronique Cortier, and Steve Kremer. 2016. When Are Three Voters Enough for Privacy Properties?. In *Computer Security – ESORICS 2016*, Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows (Eds.). Springer International Publishing, Cham, 241–260.
[3] Ning Bao and Nicole Yunger Halpern. 2017. Quantum voting and violation of Arrow's impossibility theorem. *Physical Review A* 95 (2017), 062306. Issue 6.
[4] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. 2002. Authentication of quantum messages. In *Proceedings of the the 43rd Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 449–458.
[5] Charles H. Bennett and Gilles Brassard. 1984. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, Vol. 175. New York, 8.
[6] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. 2015. SoK: A comprehensive analysis of game-based ballot privacy definitions. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 499–516.
[7] Marianna Bonanome, Vladimír Bužek, Mark Hillery, and Mário Ziman. 2011. Toward protocols for quantum-ensured privacy and secure voting, Vol. 84. APS, 022331.
[8] Anne Broadbent and Alain Tapp. 2007. Information-Theoretic Security Without an Honest Majority. In *Advances in Cryptology – ASIACRYPT 2007*, Kaoru Kurosawa (Ed.). Springer Berlin Heidelberg, 410–426.

[9] Ran Canetti. 2001. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*. 136–145.

[10] André Chailloux and Iordanis Kerenidis. 2009. Optimal Quantum Strong Coin Flipping. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. 527–533.

[11] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. 2009. Scantegrity II: end-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE Trans. Information Forensics and Security* 4, 4 (2009), 611–627.

[12] Benoît Chevallier-Mames, Pierre-Alain Fouque, David Pointcheval, Julien Stern, and Jacques Traoré. 2010. *On Some Incompatible Properties of Voting Schemes*. Springer Berlin Heidelberg, Berlin, Heidelberg, 191–199.

[13] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. 2016. A homomorphic lwe based e-voting scheme. In *International Workshop on Post-Quantum Cryptography*. Springer, 245–265.

[14] Véronique Cortier, C. C. Dragan, F. Dupressoir, B. Schmidt, P. Strub, and B. Warinschi. 2017. Machine-Checked Proofs of Privacy for Electronic Voting Protocols. In *2017 IEEE Symposium on Security and Privacy (SP)*. 993–1008.

[15] Véronique Cortier, D. Galindo, R. Küsters, J. Müller, and T. Truderung. 2016. SoK: Verifiability Notions for E-Voting Protocols. In *2016 IEEE Symposium on Security and Privacy (SP)*. 779–798.

[16] Véronique Cortier, P. Gaudry, and S. Glondu. 2019. Belenios: A Simple Private and Verifiable Electronic Voting System. In *Foundations of Security, Protocols, and Equational Reasoning (Lecture Notes in Computer Science)*, Meseguer J. Guttman J., Landwehr C. and Pavlovic D. (Eds.), Vol. 11565. Springer, Cham.

[17] I. B. Damgard, S. Fehr, L. Salvail, and C. Schaffner. 2005. Cryptography in the bounded quantum-storage model. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005*. 24–27.

[18] Stephanie Delaune, Steve Kremer, and Mark Ryan. 2006. Coercion-resistance and receipt-freeness in electronic voting. In *19th IEEE Computer Security Foundations Workshop*. 28–42.

[19] Shahar Dolev, Itamar Pitowsky, and Boaz Tamir. 2006. A quantum secret ballot. *arXiv preprint quant-ph/0602087* (2006).

[20] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. 2012. Actively Secure Two-Party Evaluation of Any Quantum Operation. In *Advances in Cryptology – CRYPTO 2012*, Reihaneh Safavi-Naini and Ran Canetti (Eds.). Springer Berlin Heidelberg, 794–811.

[21] A. Einstein, B. Podolsky, and N. Rosen. 1935. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Physics Review* 47 (1935), 777–780. Issue 10.

[22] Gina Gallegos-Garcia, Vincenzo Iovino, Alfredo Rial, Peter B. Roenne, and Peter Y. A. Ryan. 2016. (Universal) Unconditional Verifiability in E-Voting without Trusted Parties. arXiv:cs.CR/1610.06343

[23] Simon J. Gay, Rajagopal Nagarajan, and Nikolaos Papanikolaou. 2008. QMC: A Model Checker for Quantum Systems. In *Proceedings of CAV 2008*, Vol. LNCS 5123. Springer-Verlag, Berlin, Heidelberg, 543–547.

[24] Jens Groth. 2004. Evaluating Security of Voting Schemes in the Universal Composability Framework. In *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 46–60.

[25] Mark Hillery, Mário Ziman, Vladimír Bužek, and Martina Bieliková. 2006. Towards quantum-based privacy and voting. In *Physics Letters A*, Vol. 349. Elsevier, 75–81.

[26] Dmitri Horoshko and Sergei Kilin. 2011. Quantum anonymous voting with anonymity check. In *Physics Letters A*, Vol. 375. Elsevier, 1172–1175.

[27] Wei Huang, Qiao-Yan Wen, Bin Liu, Qi Su, Su-Juan Qin, and Fei Gao. 2014. Quantum anonymous ranking. In *Physical Review A*, Vol. 89. APS, 032325.

[28] Ari Juels, Dario Catalano, and Markus Jakobsson. 2005. Coercion-resistant electronic elections. In *Proc. of the ACM workshop on privacy in the electronic society*. 61–70.

[29] Elham Kashefi, Luka Music, and Petros Wallden. 2017. The Quantum Cut-and-Choose Technique and Quantum Two-Party Computation. *arXiv preprint arXiv:1703.03754* (2017).

[30] Aggelos Kiayias and Moti Yung. 2002. Self-tallying elections and perfect ballot secrecy. In *Public Key Cryptography*, David Naccache and Pascal Paillier (Eds.), Vol. 2274. Springer Berlin Heidelberg, 141–158.

[31] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 2015. End-to-end verifiable elections in the standard model. In *Advances in Cryptology - EUROCRYPT 2015*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer Berlin Heidelberg, 468–498.

[32] R. Konig, S. Wehner, and J. Wullschleger. 2012. Unconditional Security From Noisy Quantum Storage. *IEEE Transactions on Information Theory* 58, 3 (2012), 1962–1984.

[33] Yuan Li and Guihua Zeng. 2008. Quantum anonymous voting systems based on entangled state. In *Optical review*, Vol. 15. Springer, 219–223.

[34] Hoi-Kwong Lo and H.F. Chau. 1998. Why quantum bit commitment and ideal quantum coin tossing are impossible. In *Physica D: Nonlinear Phenomena*, Vol. 120. 177 – 187.

[35] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, S. Wehner, and H. Zbinden. 2015. Practical Relativistic Bit Commitment. *Physical Review Letters* 115 (2015), 030502. Issue 3.
[36] Dominic Mayers. 1997. Unconditionally Secure Quantum Bit Commitment is Impossible. *Physical Review Letters* 78 (1997), 3414–3417. Issue 17.
[37] Will McCutcheon, Anna Pappa, BA Bell, A McMillan, André Chailloux, Tom Lawson, M Mafu, Damian Markham, Eleni Diamanti, and Iordanis Kerenidis. 2016. Experimental verification of multipartite entanglement in quantum networks. *Nature Communications* 7 (2016), 13251.
[38] Tal Moran and Moni Naor. 2006. Receipt-Free Universally-Verifiable Voting with Everlasting Privacy. In *Advances in Cryptology - CRYPTO 2006*, Cynthia Dwork (Ed.). Springer Berlin Heidelberg, 373–392.
[39] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information*. Cambridge University Press, New York, NY, USA.
[40] Koutarou Suzuki Tatsuaki Okamoto and Yuuki Tokunaga. 2008. Quantum voting scheme based on conjugate coding. *NTT Technical Review* 6, 1 (2008), 1–8.
[41] Anna Pappa, André Chailloux, Stephanie Wehner, Eleni Diamanti, and Iordanis Kerenidis. 2012. Multipartite entanglement verification resistant against dishonest parties. *Physical Review Letters* 108, 26 (2012), 260502.
[42] Anna Pappa, Paul Jouguet, Thomas Lawson, André Chailloux, Matthieu Legré, Patrick Trinkler, Iordanis Kerenidis, and Eleni Diamanti. 2014. Experimental plug and play quantum coin flipping. *Nature Communications* 5, 3717 (2014).
[43] Christopher Portmann. 2017. Quantum authentication with key recycling. In *Advances in Cryptology – EUROCRYPT 2017*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.). Springer International Publishing, 339–368.
[44] Peter Y. A. Ryan and Steve A. Schneider. 2006. Prêt-à-voter with re-encryption mixes.. In *11th European Symp. On Research In Computer Security (ESORICS'06)*, Vol. 4189. Springer, 313–326.
[45] P. W. Shor. 1994. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, USA, 124–134.
[46] Peter W Shor and John Preskill. 2000. Simple proof of security of the BB84 quantum key distribution protocol. In *Physical review letters*, Vol. 85. APS, 441.
[47] Kishore Thapliyal, Rishi Dutt Sharma, and Anirban Pathak. 2017. Protocols for quantum binary voting. In *International Journal of Quantum Information*, Vol. 15. 1750007.
[48] Dominique Unruh. [n.d.]. Universally Composable Quantum Multi-party Computation.. In *Advances in Cryptology (EUROCRYPT 2010)*, Vol. 6110. 486–505.
[49] Dominique Unruh. 2019. Quantum Relational Hoare Logic. *Proc. ACM Program. Lang.* 3, Article 33 (2019), 31 pages.
[50] Joan Alfina Vaccaro, Joseph Spring, and Anthony Chefles. 2007. Quantum protocols for anonymous voting and surveying. In *Physical Review A*, Vol. 75. APS, 012333.
[51] Qingle Wang, Chaohua Yu, Fei Gao, Haoyu Qi, and Qiaoyan Wen. 2016. Self-tallying quantum anonymous voting. In *Physical Review A*, Vol. 94. APS, 022333.
[52] Peng Xue and Xin Zhang. 2017. A simple quantum voting scheme with multi-qubit entanglement. In *Scientific Reports*, Vol. 7. Nature Publishing Group, 7586.
[53] Rui-Rui Zhou and Li Yang. 2013. Distributed quantum election scheme. *arXiv preprint arXiv:1304.0555* (2013).

## SUPPLEMENTARY MATERIAL

In the supplementary material we include some extra technical details that due to lack of space could not be included in the main body of the paper, as well as our response to reviews received on previous submissions of this work.

## A FORMAL DEFINITION OF QUANTUM CORRECTNESS

The correctness experiment $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}$ is the same as $\mathbf{EXP}^{\Pi}_{\text{Qver}}$ with the only exceptions that there isn't a predicate $\mathrm{P}^{\Pi}_{\text{Verify}}$ and we don't allow $\mathcal{A}$ to corrupt the tallier (if there exist in the protocol $\Pi$). As a result, we don't capture universal verifiability in $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}$, but only double voting and vote deletion/alteration of honest ballots. Moreover, in the case that the election result $X$ is equal "$\perp$" the adversary wins the experiment.

*The experiment* $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}(\mathcal{A}, \epsilon, \delta_0)$

- **Set up phase:** $C$ and $\mathcal{A}$ generate the parameters in register $\mathcal{X}$ as specified by $\Pi$ and the adversarial model. Furthermore, $\mathcal{A}$ chooses the votes for all voters $\{v_k\}_{V_k \in \mathcal{V}}$ and $C$ the casting order $\rho \in_R \mathcal{F}$.
- **Casting phase:** For each $k = 1, \ldots, N$
  - If $\mathcal{A}$ chooses to corrupt $V_{\rho(k)}$, they are added to $\mathcal{V}_{\mathcal{A}}$.
  - If $V_{\rho(k)} \notin \mathcal{V}_{\mathcal{A}}$, then $C$ runs $(\mathcal{X}_{\rho(k)}, \perp) \leftarrow \mathsf{CastBallot}(v_{\rho(k)}, \mathcal{X}_{\rho(k)}, \delta_0)$. If not $\perp$, $C$ calls $\mathcal{O}_{\Pi,\mathcal{A}}(\mathcal{X}_{\rho(k)}, \mathcal{X}_R)$, , where $R$ is the receiver designated by $\Pi$.
  - If $V_{\rho(k)} \in \mathcal{V}_{\mathcal{A}}$, $\mathcal{A}$ performs some operation on register $\mathcal{X}_{\mathcal{A}}$ and calls $\mathcal{O}_{\Pi,\mathcal{A}}(\mathcal{X}_{\mathcal{A}}, \mathcal{X}_R)$, where $R$ is any receiver designated by $\mathcal{A}$.
- **Tally phase:** $C$ computes $X \leftarrow \mathsf{Tally}(\mathcal{X}_C, \delta_0)$:
  - If $(X = \perp)$ or $(\mathrm{P}^{\Pi}_{\mathsf{VCounted}}(\{v_k\}_{V_k \notin \mathcal{V}_{\mathcal{A}}}, X) = 0 \vee \mathsf{Nballots}^{\Pi}(X) > N)$ then `output 1`, else `output 0`.

*Definition A.1.* We say that a quantum e-voting protocol $\Pi$ satisfies $\epsilon$-**quantum correctness** if for every quantum polynomial-time $\mathcal{A}$, the probability to win the experiment $\mathbf{EXP}^{\Pi}_{\text{Qcorr}}(\mathcal{A}, \epsilon, \delta_0)$ is negligible with respect to $\delta_0$:

$$\Pr[1 \leftarrow \mathbf{EXP}^{\Pi}_{\text{Qcorr}}(\mathcal{A}, \epsilon, \delta_0)] = negl(\delta_0).$$

## B PROOF OF ATTACK ON DISTRIBUTED BALLOT PROTOCOLS

Now we give detailed proofs of the theorems and lemmas of Section 6.

**Lemma** 6.1. *Let* $\Theta^{v}_{D,\delta} \in [0, 2\pi]$ *be the continuous random variable that describes the outcome of the measurement of a vote state* $|\psi(\theta_v)\rangle, v \in \{y, n\}$ *using operators*

$$E(\theta) = \frac{D}{2\pi} |\Phi(\theta)\rangle \langle\Phi(\theta)| \tag{3}$$

*where* $|\Phi(\theta)\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} e^{ij\theta} |j\rangle$. *It holds that:*

$$Pr[x_l < \Theta^{v}_{D,\delta} < x_{l+w}] = \frac{1}{2\pi D} \int_{x_l}^{x_{l+w}} \frac{\sin^2[D(\theta - \theta_v)/2]}{\sin^2[(\theta - \theta_v)/2]} d\theta \tag{4}$$

Proof.

$$\begin{aligned}
Pr[x_l < \Theta^{v}_{D,\delta} < x_{l+w}] &= \langle\phi(\theta_v)| \int_{x_l}^{x_{l+w}} E(\theta) d\theta \, |\phi(\theta_v)\rangle \\
&= \int_{x_l}^{x_{l+w}} \langle\phi(\theta_v)| E(\theta) |\phi(\theta_v)\rangle \, d\theta \\
&= \frac{D}{2\pi D^2} \int_{x_l}^{x_{l+w}} |\sum_{j=0}^{D-1} e^{(\theta-\theta_v)ij}|^2 d\theta \\
&= \frac{1}{2\pi D} \int_{x_l}^{x_{l+w}} ([\sum_{j=0}^{D-1} \cos[(\theta - \theta_v)j]]^2 \\
&+ [\sum_{j=0}^{D-1} \sin[(\theta - \theta_v)j]]^2) d\theta
\end{aligned}$$

For any $x \in \mathbb{R}$, the following two equations hold:

$$\sum_{j=0}^{D-1} \cos[jx] = \frac{\sin[Dx/2]}{\sin[x/2]} \cos[(D-1)x/2]$$

$$\sum_{j=0}^{D-1} \sin[jx] = \frac{\sin[Dx/2]}{\sin[x/2]} \sin[(D-1)x/2]$$

So finally we have:

$$Pr[x_l < \Theta^v_{D,\delta} < x_{l+w}] = \frac{1}{2\pi D} \int_{x_l}^{x_{l+w}} \frac{\sin^2[D(\theta-\theta_v)/2]}{\sin^2[(\theta-\theta_v)/2]} d\theta$$

□

**Lemma** 6.2. *Let $|\psi(\theta_v)\rangle$ be a voting state of the protocol. Then it holds:*

$$Pr[x_{l_v} < \Theta^v_{D,\delta} < x_{l_v+1}] > 0.405$$

Proof. A simple change of variables in Eq.(4) gives us:

$$Pr[x_{l_v} < \Theta^v_{D,\delta} < x_{l_v+1}] = \frac{1}{2\pi D} \int_0^{2\pi/D} \frac{\sin^2[D(\theta-\delta)/2]}{\sin^2[(\theta-\delta)/2]} d\theta$$

By setting $(\theta-\delta)/2 = y$, we get:

$$Pr[x_{l_v} < \Theta^v_{D,\delta} < x_{l_v+1}] = \frac{1}{\pi D} \int_{-\delta/2}^{(2\pi/D-\delta)/2} \frac{\sin^2[Dy]}{\sin^2[y]} dy$$

The above is just a function of $\delta$, which we denote as $F(\delta)$. In order to lower-bound $F(\delta)$ we need to find its derivative:

$$\frac{dF(\delta)}{d\delta} = \frac{1}{2\pi D} \left( \frac{\sin^2[D\delta/2]}{\sin^2[\delta/2]} - \frac{\sin^2[D\delta/2]}{\sin^2[(2\pi/D-\delta)/2]} \right)$$

It is easy to check that:

$$\frac{dF(\delta)}{d\delta} = 0, \text{ when } \delta = 0 \text{ or } \delta = \pi/D$$
$$\frac{dF(\delta)}{d\delta} > 0, \text{ when } 0 < \delta < \pi/D$$
$$\frac{dF(\delta)}{d\delta} < 0, \text{ when } \pi/D < \delta < 2\pi/D$$

It also holds that $F(0) = F(2\pi/D)$, so the minimum extreme points of our function are equal. As a result we have:

$$F(\delta) \geq \lim_{\delta \to 0^-} F(\delta) = F(0) \tag{5}$$

From the fact that:

$$|\sin[x]| \leq |x|, \forall x \in \mathbb{R}$$
$$|\sin[x]| \geq |(2/\pi)x|, \forall x \in [0, \pi/2]$$
$$|\sin[x]| \geq |-(2/\pi)x + 2|, \forall x \in [\pi/2, \pi]$$

It follows:

$$\begin{aligned} F(0) &\geq \frac{1}{\pi D}\int_0^{\frac{\pi}{2D}} \left(\frac{2}{\pi D y}\right)^2 / y^2 dy + \int_{\frac{\pi}{2D}}^{\frac{\pi}{D}} \left(\frac{2}{\pi D y} + 2\right)^2 / y^2 dy \\ &\geq \frac{4}{\pi^2} \\ &> 0.405 \end{aligned}$$

□

Now in order to prove lemma 6.3, we need the following proposition:

Proposition B.1. *$\forall x \in [-2\pi, 2\pi]$ it holds that:*

$$\sin^2[x] > \sum_{n=1}^{20} (-1)^{n+1} \frac{2^{2n-1} x^{2n}}{(2n)!} \tag{6}$$

Proof. From the Taylor series expansion at point 0 of $\cos[x]$, we know that:

$$\cos[x] = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!},\ \forall x \in \mathbb{R}$$

Then:

$$\begin{aligned} \sin^2[x] &= \frac{1}{2} - \frac{\cos[2x]}{2} = \frac{1}{2} - \frac{1}{2}\sum_{n=0}^{\infty} (-1)^n \frac{2^{2n} x^{2n}}{(2n)!} \\ &= \sum_{n=1}^{\infty} (-1)^{n+1} \frac{2^{2n-1} x^{2n}}{(2n)!} \end{aligned}$$

Given the above equation, in order to prove Eq.(6), we simply need to show:

$$\sum_{n=21}^{\infty} (-1)^{n+1} \frac{2^{2n-1} x^{2n}}{(2n)!} > 0$$

If we think of the above as a sum of terms $a_n$ ($n = 21, \ldots, \infty$), for integer $j \geq 10$, it holds that:

$$\begin{aligned} a_n &> 0, \text{ when } n = 2j + 1, \\ a_n &< 0, \text{ when } n = 2j. \end{aligned}$$

We therefore need to prove that $\sum_{n=21}^{\infty} a_n > 0$, which in turn is equivalent to proving that:

$$\begin{aligned} |a_n| > |a_{n+1}| &\iff 2^{2n-1} x^{2n}/(2n)! > 2^{2n+1} x^{2n+2}/(2n+2)! \\ &\iff 1 > 4x^2/((2n+1)(2n+2)) \\ &\iff (2n+1)(2n+2)/4 > x^2 \end{aligned}$$

In this case, the above holds, because the minimum value of n is 21 and the maximum value of $x^2$ is $4\pi^2$. □

**Lemma** 6.3. *Let $|\psi(\theta_v)\rangle$ be a voting state of the protocol. Then it holds:*

$$Pr[x_{l_v - 1} < \Theta^v_{D,\delta} < x_{l_v + 2}] > 0.9$$

Proof. We follow exactly the same procedure as lemma 6.2 and get:

$$\begin{aligned} & Pr[x_{l_v-1} < \Theta^v_{D,\delta} < x_{l_v+2}] \qquad (7)\\ &= \frac{1}{2\pi D}\int_{x_{l_v-1}}^{x_{l_v+2}} \frac{\sin^2[D(\theta-\theta_v)/2]}{\sin^2[(\theta-\theta_v)/2]}d\theta \\ &= \frac{1}{2\pi D}\int_{-2\pi/D}^{4\pi/D} \frac{\sin^2[D(\theta-\delta)/2]}{\sin^2[(\theta-\delta)/2]}d\theta \\ &= \frac{1}{\pi D}\int_{-\pi/D-\delta/2}^{2\pi/D-\delta/2} \frac{\sin^2[Dy]}{\sin^2[y]}dy \qquad (8)\end{aligned}$$

where $(\theta-\delta)/2 = y$. Again the above probability depends only on $\delta$ and can therefore be denoted with $F(\delta)$. In a similar way as before, we can prove that the minimum of this function is at $\delta = 0$ and compute $F(0)$.

$$\begin{aligned} F(0) &= \frac{1}{\pi D}\int_{-\pi/D}^{2\pi/D} \frac{\sin^2[Dy]}{\sin^2[y]}dy \\ &\geq \frac{1}{\pi D}\int_{-\pi/D}^{2\pi/D} \frac{\sum_{n=1}^{20} \frac{(-1)^{n+1}2^{2n-1}(Dy)^{2n}}{(2n)!}}{y^2}dy \\ &= \frac{1}{\pi D}\sum_{n=1}^{20}\int_{-\pi/D}^{2\pi/D} \frac{(-1)^{n+1}2^{2n-1}D^{2n}y^{2n}}{y^2(2n)!}dy \\ &= \frac{1}{\pi D}\sum_{n=1}^{20} \frac{(-1)^{n+1}2^{2n-1}D^{2n}}{(2n)!}\int_{-\pi/D}^{2\pi/D} y^{2(n-1)}dy \\ &= \frac{1}{\pi D}\sum_{n=1}^{20} \frac{(-1)^{n+1}2^{2n-1}D^{2n}}{(2n)!}[y^{2n-1}/(2n-1)]_{-\pi/D}^{2\pi/D} \\ &= \sum_{n=1}^{20} \frac{(-1)^{n+1}2^{2n-1}}{(2n)!}\frac{\pi^{2n-2}(2^{2n-2}+1)}{2n-1} \\ &\approx 0.9263 \qquad (9)\end{aligned}$$

□

**Theorem** 6.4. *With overwhelming probability in the number of voters N, algorithm 1 includes $l_v$ in the* `Solution` *vector (i.e. it measures a value in the interval $[x_{l_v}, x_{l_v+1}]$ more than 40% of the time).*

$$\Pr[\texttt{Solution}[0] = l_v \vee \texttt{Solution}[1] = l_v] > 1 - 1/exp(\Omega(N))$$

Proof. We can see each measurement that algorithm 1 performs at each vote state $|\psi(\theta_v)\rangle$, as an independent Bernoulli trial $X_l$ with probability of success $p_l = \Pr[x_l < \Theta^v_{D,\delta} < x_{l+1}]$. Then the value of `Record`$[l]$ follows the binomial distribution:

$$X_{\texttt{Record}[l]} \sim B(\frac{\varepsilon N}{2}, p_l)$$

We can therefore compute:

$$\begin{aligned}
&\Pr\left[\texttt{Solution}[0] = l_v \vee \texttt{Solution}[1] = l_v\right] \\
&= \Pr\left[\texttt{Record}[l_v] \geq 0.4\varepsilon N/2\right] \\
&\geq 1 - \Pr\left[\texttt{Record}[l_v] \leq 0.4\varepsilon N/2\right] \\
&\overset{1}{=} 1 - \Pr\left[\texttt{Record}[l_v] \leq (1-\gamma)p_{l_v}\varepsilon N/2\right] \\
&\overset{2}{\geq} 1 - exp(-\gamma^2 p_{l_v}\varepsilon N/6) \\
&= 1 - (exp(-\gamma^2 p_{l_v}\varepsilon/6))^N \\
&= 1 - 1/exp(\Omega(N))
\end{aligned}$$

□

**Theorem** 6.5. *With negligible probability in the number of voters N, algorithm 1 includes a value other than* $(l_v - 1, l_v, l_v + 1)$ *in the* Solution *vector, i.e.* $\forall w \in \{0, \ldots, l_v - 2, l_v + 2, \ldots, D - 1\}$*:*

$$Pr[\texttt{Solution}[0] = w \vee \texttt{Solution}[1] = w] < 1/exp(\Omega(N))$$

Proof. Let $w \in \{0, \ldots, D-1\} \setminus \{l_v - 1, l_v, l_v + 1\}$, then it holds:

$$\begin{aligned}
&Pr[\texttt{Solution}[0] = w \vee \texttt{Solution}[1] = w] \\
&= Pr[X_{\texttt{Record}[w]} \geq 0.4\varepsilon N/2]
\end{aligned}$$

We know from lemma 6.3 that $p_w < 0.1$, so $\exists \gamma > 0$ such that:[7]

$$\begin{aligned}
&Pr[X_{\texttt{Record}[w]} \geq 0.4\varepsilon N/2] \\
&= Pr[X_{\texttt{Record}[w]} \geq (1+\gamma)p_w\varepsilon N/2] \\
&< exp(-\gamma p_w \varepsilon N/6) \\
&= (exp(-\gamma p_w \varepsilon/6))^N \\
&= 1/exp(\Omega(N))
\end{aligned}$$

□

**Lemma** 6.6. *With overwhelming probability in N, the* Solution *vector in algorithm 1, is equal to* $[l_v - 1, l_v]$, $[l_v, \text{“}Null\text{”}]$ *or* $[l_v, l_v + 1]$*. Specifically,*

$$\begin{aligned}
&\Pr[\texttt{Solution} \in \{[l_v - 1, l_v], [l_v, \text{“}Null\text{”}], [l_v, l_v + 1]\}] \\
&> 1 - 1/exp(\Omega(N))
\end{aligned}$$

Proof. Let as define the following events:

$$A = \Big[\texttt{Solution}[0] = w \vee \texttt{Solution}[1] = w, \\ w \in \{0, \ldots, l_v - 2, l_v + 2, \ldots, D - 1\}\Big]$$

$$B = \left[\texttt{Solution}[0] = l_v \vee \texttt{Solution}[1] = l_v\right]$$

[1] $p_{l_v} > 0.405 \implies \exists \gamma > 0$ s.t $0.4 = (1-\gamma)p_{l_v}$

[2] The Chernoff bound for a random variable $X \sim B(N, p)$ and expected value $E[X] = \mu$ is: $Pr[X \leq (1-\gamma)\mu] \leq exp(-\gamma^2\mu/3)$

[7] The Chernoff bound for a random variable $X \sim B(N, p)$ and expected value $E[X] = \mu$ is: $Pr[X \leq (1+\gamma)\mu] \leq exp(-\gamma\mu/3), \gamma > 1$

Since the cases $\texttt{Solution} = [l_v, l_v - 1]$ and $\texttt{Solution} = [l_v + 1, l_v]$ are impossible from the construction of the algorithm, from theorems 6.4 and 6.5 it holds:

$$\begin{aligned}&\Pr[\texttt{Solution} \in \{[l_v - 1, l_v], [l_v, \text{“}Null\text{”}], [l_v, l_v + 1]\}]\\&= \Pr[B \wedge \neg A]\\&= \Pr[B] - \Pr[B \wedge A]\\&> 1 - 1/exp(\Omega(N))\end{aligned}$$ □

Lemma B.2. *Let $|\psi(\theta_v)\rangle$ be a voting state with $\delta \in [0, 2\pi/D)$ and $l_v = D-1$,where $\delta$ is a continuous random variable .Then it holds:*

$$\Pr[x_{D-2} < \Theta^v_{D,\delta} < x_D] + Pr[x_0 < \Theta^v_{D,\delta} < x_1] > 0.9$$

Proof.

$$\Pr[x_0 < \Theta^v_{D,\delta} < x_1] \tag{10}$$

$$= 1/(2\pi D) \int_{x_0}^{x_1} (\frac{Sin[D/2(\theta - \theta_v)]}{Sin[1/2(\theta - \theta_v)]})^2 d\theta \tag{11}$$

Now we set $\theta = \theta - x_D$ to 11 and we have:

$$\Pr[x_0 < \Theta^v_{D,\delta} < x_1] \tag{12}$$

$$= 1/(2\pi D) \int_{x_D}^{x_D+x_1} (\frac{Sin[-D\pi + D/2(\theta - \theta_v)]}{Sin[-\pi + 1/2(\theta - \theta_v)]})^2 d\theta \tag{13}$$

$$= 1/(2\pi D) \int_{x_D}^{x_D+x_1} (\frac{Sin[D/2(\theta - \theta_v)]}{Sin[1/2(\theta - \theta_v)]})^2 d\theta \tag{14}$$

Finally we have:

$$\Pr[x_{D-2} < \Theta^v_{D,\delta} < x_D] + \Pr[x_0 < \Theta^v_{D,\delta} < x_1] \tag{15}$$

$$= 1/(2\pi D) \int_{x_{D-2}}^{x_D+x_1} (\frac{Sin[D/2(\theta - \theta_v)]}{Sin[1/2(\theta - \theta_v)]})^2 d\theta \tag{16}$$

$$= 1/(2\pi D) \int_{-2\pi/D}^{4\pi/D} \frac{Sin^2[D(\theta - \delta)/2]}{Sin^2[(\theta - \delta)/2]} d\theta \tag{17}$$

From lemma 6.3 this integral is at least 0.9. The proof is similar for $l_v = 0$. □

Lemma B.3. *Let* $\texttt{Solution}$ *be the matrix of algorithm 1, then it holds:*

$$\begin{aligned}&\Pr[\texttt{Solution} \in \{\{l_v - 1, l_v\}, \{l_v\}, \{l_v, l_v + 1\}\}]\\&= \Pr[\texttt{Solution} \in \{[l_v - 1, l_v], [l_v], [l_v, l_v + 1]\}]\end{aligned}$$

Proof. (sketch)It holds that:

$$\Pr[\texttt{Solution} \in \{\{l_v - 1, l_v\}, \{l_v\}, \{l_v, l_v + 1\}\}] \tag{18}$$

$$= \Pr[\texttt{Solution} \in \{l_v - 1, l_v\}] \tag{19}$$

$$+ \Pr[\texttt{Solution} \in \{l_v\}] \tag{20}$$

$$+ \Pr[\texttt{Solution} \in \{l_v, l_v + 1\}] \tag{21}$$

We need to prove that:

$$\Pr[\texttt{Solution} \in \{l_v - 1, l_v\}] = \Pr[\texttt{Solution} = [l_v - 1, l_v]] \tag{22}$$

From the construction of the algorithm 1 we know that:

$$\Pr[\texttt{Solution} = [l_v, l_v - 1] | \texttt{Solution} \in \{l_v - 1, l_v\}] = 0 \tag{23}$$

This is true because the values of the `Solution` are from the matrix `Record` in a progressive manner. So under the assumption that both $l_v, l_v - 1$ had appeared at least 40% times, they inserted in a progressive order. The only time they will not is the case in which $l_v = 0$ and $l_v - 1 = D - 1$. At first the order is $[0, D - 1]$, but because of the special condition we had in our algorithm the order switches to $[D - 1, 0]$.

It holds that:

$$\Pr[\texttt{Solution} = [l_v, l_v - 1] | \texttt{Solution} \in \{l_v - 1, l_v\}] \tag{24}$$

$$= \Pr[\texttt{Solution} = [l_v, l_v - 1]] + \Pr[\emptyset] \tag{25}$$

$$= \Pr[\texttt{Solution} = [l_v, l_v - 1]] \tag{26}$$

$$= 0 \tag{27}$$

Similar are the other cases. □