Quantum Cryptography applied to Electronic-Voting Protocols

Miguel Maria Rodrigues Perlico da Cruz Sabino

Instituto Superior Técnico – Universidade de Lisboa

December, 2016

Abstract

Remote electronic voting provides the citizens an easier way to participate in decision making processes. Some remote e-voting protocols have already been used in real voting events, such as the Neuchâtel's federal referendum on March 8th 2015, (Galindo et al., 2015). The major breakthrough presented in the previous e-voting ballot casting protocol is its *Cast-as-Intended* verification mechanism, allowing voters to confirm their voting intents (making use of return codes) while keeping the single vote casting.

In this work we analyse Neuchâtel's e-voting protocol and propose some improvements (mostly related to security and verifiability concerns), achieved through some cryptography techniques (including quantum cryptography). Therefore, we include the required quantum mechanics' concepts and laws to understand the employed quantum cryptography.

Keywords: electronic voting; quantum cryptography; Neuchâtel's protocol; quantum key distribution; quantum bit commitment.

1 Introduction

The possibility of turning the physical voting processes into an electronic and remote operation is of great interest in democratic societies. However, electronic-democracy brings up new important issues, namely the transparency and privacy of the e-voting process, and implementations' reliability.

The three major verification aspects are *Cast-as-Intended*, which provide the voter with methods to check if the encrypted vote prepared by the voting client application actually corresponds to the selected voting options, (it is the central mechanism addressed in Neuchâtel's protocol), *Recorded-as-Cast* and *Counted-as-Recorded*, whose definitions can be found in (Galindo et al., 2015, Section 1).

In this work, we focus our attention on the e-voting protocol implemented on Neuchâtel,[1] assuring cast-as-intended verifiability based on Return Codes, with the particularity of providing the voter with a confirmation phase which allows him to verify its voting intention before the vote is officially cast. Also, in this protocol the voter's privacy does not rely on the need for two server-side entities not to collude. We intend to take advantage of some quantum cryptography applications (cryptographic tasks that are accomplished using quantum mechanical properties), among others, in order to suggest improvements on Neuchâtel's protocol.

In Section 2 we present some quantum mechanics' notions and laws, in Section 3 we focus on quantum key distribution and quantum bit commitment. Next, in Section 4, we study Neuchâtel's protocol, based on (Galindo et al., 2015). Finally, in Section 5, we propose some improvements on this protocol (using some suitable cryptographic techniques such as multi-party computation and quantum key distribution) to achieve more security requisites and at Section 6 we shall conclude our work.

2 Quantum Mechanics

Quantum mechanics is a mathematical framework or set of rules for the construction of physical theories. The use of quantum systems to perform computations gave birth to quantum computation. In order to bring quantum cryptography to e-voting, we must know first some basic ideas on quantum mechanics and understand the major laws that support quantum cryptography.

2.1 Basic Concepts

Definition 2.1. Qubit
A qubit (short for quantum bit) is physical object, such as a photon or an electron, which is the basic unit for quantum computation and quantum information.

[1] An evolution of the Norwegian protocol used in 2011 and 2013 (Gjøsteen, 2011).

Despite being a physical object, we shall use an abstract mathematical point of view to describe it. The qubit has a state, that can be a superposition, i.e, a linear combination of classical states, which is described by

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$. The basis $\mathcal{B} = \{|0\rangle, |1\rangle\}$ is called the computational basis.

One cannot determine a qubit's state (values of α and β), we can only extract one bit of information from the state of a qubit. Measuring the state of the qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with respect to the computational basis, we obtain either the result 0 with probability $|\alpha|^2$ or the result 1 with probability $|\beta|^2$. Furthermore, measuring a qubit will generally change its state: after the measurement, the system is in the measured state. One may perform other measurements according to the chosen basis states for a qubit. Generally, having a basis $\{|a\rangle, |b\rangle\}$, any state of a qubit can be expressed through that basis: $|\psi\rangle = \alpha|a\rangle + \beta|b\rangle$. Moreover, if the basis is orthonormal one may perform a measurement with respect to that basis, obtaining a with probability $|\alpha|^2$ or b with probability $|\beta|^2$.

Definition 2.2. Quantum Observable
A quantum observable is a measurable quantity, expressed as an operator, such that the property of the system state can be determined by means of an operational definition.

Performing a measurement yields a value to some quantum observable. While quantum states are represent by vectors in a Hilbert space V, observables are represented by Hermitian operators on V. In quantum theory, each variable (such as position or velocity) corresponds to an observable.

Lemma 2.3. *If A is a Hermitian operator the following holds:*

1. *The eigenvalues of A are all real;*
2. *The eigenvectors of A associated to different eigenvalues are orthogonal;*
3. *The eigenstates of A form a complete set of basis states for the state space of the system.*

Therefore, one may write

$$A = a_1 \cdot |a_1\rangle\langle a_1| + a_2 \cdot |a_2\rangle\langle a_2|, \tag{2}$$

where A is an Hermitian operator, $|a_1\rangle$ and $|a_2\rangle$ are the eigenvectors of A, and a_1 and a_2 are the associated eigenvalues, respectively. From this point on, unless otherwise stated or derived from the context, we will use the qubit's basis states $|0\rangle$ and $|1\rangle$.

We can rewrite the qubit's state through its geometric representation, see (Nielsen and Chuang, 2010).

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle. \tag{3}$$

for some $0 \leqslant \theta \leqslant \pi, 0 \leqslant \phi < 2\pi$ and $\gamma, \gamma' \in \mathbb{R}$, such that $\gamma + \phi = \gamma'$.

2.2 Fundamental Laws and Postulates

Now we shall present a few principles and laws of quantum mechanics which will be the central key for the security of many quantum protocols.

Quantum Indeterminacy: *There exists a necessary incompleteness in the description of a physical system, resulting from the fact that quantum particles do not have simultaneous determinate positions and momentums.*

Quantum indeterminism (derived from Heisenberg Uncertainty Principle) is a physics' theory asserting that a certain kind of events are actually indeterministic.

Theorem 2.4. ***(No-Cloning Theorem)*** *It is impossible to make a copy of an unknown quantum state.*

The proof can be found in (Nielsen and Chuang, 2010), and essentially shows that it is impossible to perfectly clone an unknown quantum state using unitary evolution.

Quantum Entanglement: *Quantum mechanical phenomenon in which the quantum states of two or more quantum particles have to be described with reference to each other, even though the individual particles may be spatially separated.*

This leads to correlations between observable physical properties of the systems.

Quantum Nonlocality: *The apparent ability of entangled quantum objects to instantaneously know about each other's state, even when separated by large distances, i.e, the ability of one particle when measured to instantly determine the state of its conjoined twin at an arbitrary distance.*

As one would expect, nonlocality occurs due to the phenomenon of entanglement, namely due to the correlations and dependencies existent between the entangled particles.

Principle of Causality: *The same cause or set of causes always produces the same effects (other things being equal) and the causes temporally precedes, or is simultaneous with, its effects.*

In the field of quantum mechanics, causality is closely related to the principle of locality.

Postulate 1: *Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.*

Postulate 2: *The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on times t_1 and t_2, $|\psi'\rangle = U|\psi\rangle$.*

Postulate 3: *Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by*

$$p(m) = \langle\psi|(M_m^T)^* M_m|\psi\rangle, \tag{4}$$

and the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|(M_m^T)^* M_m|\psi\rangle}}. \tag{5}$$

The measurement operators satisfy the completeness equation

$$\sum_m (M_m^T)^* M_m = I \Longrightarrow \sum_m \langle\psi|(M_m^T)^* M_m|\psi\rangle = \sum_m p(m) = 1. \tag{6}$$

Postulate 4: *The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$.*

3 Quantum Cryptography

The exploitation of the principles of quantum mechanics for the completion of cryptographic tasks (many conjectured to be impossible using only classical computation) is the focus of quantum cryptography. It also provides methods to accomplish provably secure protocols that allow for example private information sharing, by taking advantage of the quantum mechanical principle that observation, in general, disturbs the system being observed.

3.1 Quantum Key Distribution

One of the major applications of quantum cryptography is quantum key distribution (QKD), which allows private keys to be created between two parties over a public channel, based on the fundamental idea that an eavesdropper is unable to gain any information from the intercepted qubits without disturbing their state. It can be proven to be unconditionally secure (even against quantum computers), because it relies on physical principles instead of mathematical complexity. The protocol described next is known as the *BB84 method* and it was proposed by Charles Bennett and Gilles Brassard in 1984.

Quantum Key Distribution, BB84: In order to execute the protocol, Alice and Bob must be connected by a quantum communication channel which allows qubits to be transmitted.

1. Alice and Bob agree in two pairs of quantum states, being each pair conjugate to the other pair, and the two states within a pair orthogonal to each other, such that each pair is a basis. The usual polarization state pairs used are the rectilinear basis (0 and $\frac{\pi}{2}$) and the diagonal basis ($\frac{\pi}{4}$ and $\frac{3\pi}{4}$), which are indeed conjugate to each other.
2. For the quantum transmission Alice produces random bit (0 or 1) and then randomly chooses one basis (rectilinear or diagonal) to encode it in. Depending on the bit and the basis chosen, she prepares a photon polarization state, as exemplified in Figure 1.
3. Alice transmits the polarized photon in the specified state to Bob, and records the state, basis and time of each photon sent. This process is repeated from the random bit stage.
4. For each received photon, Bob randomly selects a basis to measure it in (since he does not know the basis used for encoding the photons). He records the time, measurement basis and measurement result.

Figure 1: Photon polarization state according to the bit and basis selected.

Basis	1	0
Rectilinear (+)	90°	0°
Diagonal (×)	45°	135° (or -45°)

5. Alice and Bob communicate over a classical channel (such as the Internet): Alice provides the basis used in the polarization of each photon, and Bob the basis in which each photon was measured in. The photon measurements (bits) where the basis didn't match are discarded by both.
6. The remaining bits constitute the shared key.

Unconditional Security: Let's see what makes this method secure, and how it allows the detection of a potential eavesdropper. Note that there is no possible measurement capable of distinguish between the four different polarization states, consequence of quantum indeterminacy. To make it clear suppose that the photon measurement will be performed using the rectilinear basis: if the photon was created with rectilinear basis the measure will be the correct state (0 or $\frac{\pi}{2}$), but if it was created with diagonal basis, then the rectilinear basis will outcome 0 or $\frac{\pi}{2}$ at random.

Suppose there is an eavesdropper, Eve, trying to learn the secret key that Alice and Bob are sharing. Due to the No-Cloning Theorem (2.4), Eve cannot duplicate those photons so she must measure the intercepted ones, but according to Postulate 3, measuring an unknown quantum state will change that state. Consider the following scenario:

1. Alice encodes bit 0 with rectilinear basis, thus resulting in horizontal polarization (0);
2. Eve intercepts the photon and measures it, but she guesses the wrong basis, i.e, she measures this photon with diagonal basis and returns bit 1.[2] Furthermore, the photon becomes polarized in the state it was measured in, so it is now polarized in state ($\frac{\pi}{4}$).
3. Bob receives the photon and randomly chooses the rectilinear basis to measure it (with $\frac{1}{2}$ probability). The measurement yields value 1 (with $\frac{1}{2}$ probability).
4. Alice and Bob compare their basis (used to encode and measure the photon, respectively) and since they coincide, they keep the bit. But the bit Alice encoded differs from the bit Bob learned!

The previous plot shows that Alice and Bob can compare some substring of the key, searching for errors in Bob's measurements. If those errors are found they had detected the presence of an eavesdropper, so they abort the key.

3.2 Quantum Bit Commitment

In a very simplistic way, a bit commitment scheme defines a two-party method such that a party can commit to a chosen value at *commitment* phase, keeping it secret to everyone else (*concealing* property), with the capacity to reveal its commitment later at *opening* phase, while forcing the commitment to remain unchanged from the commitment moment until it is revealed (*binding* property). Beside these two phases, quantum bit commitment usually begins with an extra phase, the *initialization* phase, which will be explained later.

No quantum bit commitment protocol (QBC protocol) can be unconditionally *binding* and *concealing* at the same time (see Mayers, 1997), but considering the technological restrictions existent nowadays (and expected in the upcoming years) we will approach a solution for bit commitment protocols using quantum properties, proposed by Loura et al. (2014).

Recalling quantum complementarity (it is impossible to simultaneously measure two non-commuting observables[3] of a physical system), one can think of the choice on which observable to measure as the commitment to a bit value c, and even on the measurement result as a proof of the commitment. Before anything, some setup has to be done during the *initialization* phase, which will be specified later. Bob prepares a number of identical qubits, which will be sent to Alice in order for her to perform the measurements. These qubits should be randomly generated in one out of two quantum states, $|0\rangle$ or $|1\rangle$, with the two following requirements:

- The states $|0\rangle$ and $|1\rangle$ cannot be orthogonal.
- $\langle 0|1\rangle = \cos\theta$, with $\theta \in \left(0; \frac{\pi}{2}\right)$.

We define the states orthogonal to $|0\rangle$ and $|1\rangle$ by $|0^{\perp}\rangle$ and $|1^{\perp}\rangle$, respectively, establishing two orthogonal bases $\mathcal{B}_0 = \{|0\rangle, |0^{\perp}\rangle\}$ and $\mathcal{B}_1 = \{|1\rangle, |1^{\perp}\rangle\}$ which will produce the following two orthogonal observables:

$$\hat{C}_0 = 0 \cdot |0\rangle\langle 0| + 1 \cdot |0^{\perp}\rangle\langle 0^{\perp}|, \tag{7}$$

$$\hat{C}_1 = 1 \cdot |1\rangle\langle 1| + 0 \cdot |1^{\perp}\rangle\langle 1^{\perp}|. \tag{8}$$

[2] Note that this event is totally plausible, since the probability that it occurs is Pr("Eve selects wrong basis") × Pr("measure returns 1" | "basis is wrong")= $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$.

[3] Two noncommuting observables are said to be incompatible and they cannot be measured simultaneously due to the Heisenberg Uncertainty Principle.

Let $(|b_1\rangle, \ldots, |b_N\rangle)$ be the qubits sent by Bob. Alice may not receive all of them, say she only receives n ($\leq N$). Then, denoting by $(\tau_1, \ldots \tau_n)$ the Alice's measurement times and $(r_1, \ldots, r_n)$ the respective outcomes, there exists an index $k(i)$, for each $i = 1, \ldots, n$, identifying it with Bob's qubit emission time $t_{k(i)}$ and respective bit $b_{k(i)}$. We can think of Alice's measurement r_i as an attempt to match the state of Bob's qubit $|b_{k(i)}\rangle$. Using Postulate 3 we derive the conditional probability that measuring the observable $\hat{C}_c$ on state $|b\rangle$ one obtains result r, denoted by $p_c(r|b)$.

The eigenvectors of $\hat{C}_0$ expressed in basis $\mathcal{B}_1$ and the eigenvectors of $\hat{C}_1$ expressed in basis $\mathcal{B}_0$ are the following, respectively:

$$\begin{cases} |0\rangle = \cos\theta|1\rangle + \sin\theta|1^\perp\rangle \\ |0^\perp\rangle = e^{-i\phi}(\sin\theta|1\rangle - \cos\theta|1^\perp\rangle) \end{cases} \quad \text{and} \quad \begin{cases} |1\rangle = \cos\theta|0\rangle + e^{i\phi}\sin\theta|0^\perp\rangle \\ |1^\perp\rangle = \sin\theta|0\rangle - e^{i\phi}\cos\theta|0^\perp\rangle \end{cases}, \tag{9}$$

for some $\phi \in [0; 2\pi[$. Throughout this discussion we will use the orthogonal basis $\{|0\rangle, |0^\perp\rangle\}$ whenever expressing states or constructing measurement operators, so $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} \cos\theta \\ e^{i\phi}\sin\theta \end{bmatrix}$.

The computed probabilities are:

$$\begin{cases} p_0(0|0) = 1 \\ p_0(0|1) = \cos^2\theta \\ p_0(1|0) = 0 \\ p_0(1|1) = \sin^2\theta \end{cases} \quad \text{and} \quad \begin{cases} p_1(1|1) = 1 \\ p_1(1|0) = \cos^2\theta \\ p_1(0|0) = \sin^2\theta \\ p_1(0|1) = 0 \end{cases}. \tag{10}$$

If Alice committed to the value c, the statistics that characterize her results $\{r_i\}$ (corresponding to $\{b_{k(i)}\}$) must follow the distribution $\{p_c(r_i|b_{k(i)})\}$, so Alice's results are actually a proof of her commitment.

Now a schematic formulation of the protocol is displayed:

1. Bob generates a random string of identical qubits $(|b_1\rangle, \ldots, |b_N\rangle)$, with $b_k \in \{0, 1\}$, for $k = 1, \ldots, N$, and sends them to Alice at times $(t_1, \ldots, t_N)$, respectively. This correspinds to the *initialization* phase.
2. Alice commits to $c \in \{0, 1\}$ and according to that she measures the observable $\hat{C}_c$ on all the n qubits received. Also, she discloses the time each measurement was performed, $\tau_1 < \ldots < \tau_n$.
3. For each time measurement τ_i, with $i = 1, \ldots, n$, Bob verifies if there exists any sending time $t_{k(i)} \in \{t_1, \ldots, t_N\}$ such that $\tau_i = t_{k(i)} + vl$, where v is the speed of the qubits and l is the distance between Bob and Alice.
4. Alice reveals her commitment c and her measurement outcomes $(r_1, \ldots, r_n)$.
5. For each measurement result r_i, with $i = 1, \ldots, n$, Bob compares it with the encoded bits sent $(b_1, \ldots, b_N)$ to verify that:

$$\begin{cases} b_{k(i)} = c \Rightarrow r_i = c \\ b_{k(i)} \neq c \Rightarrow p_c(r_i|b_{k(i)}) \simeq \frac{n(r_i|b_{k(i)})}{n(b_{k(i)})} \end{cases}, \tag{11}$$

   where $n(b_{k(i)})$ is the total number of qubits received in the state $|b_{k(i)}\rangle$ and $n(r_i|b_{k(i)})$ is the number of outcomes r_i obtained from Alice's measurements on qubits received in state $|b_{k(i)}\rangle$.

Security Analysis: We start by showing that the commitment must be done before the *Opening* phase, since Alice is forced to measure the qubits as soon as she receives them from Bob, due to the non-existence of long-term stable quantum memories, meaning that Alice is obligated to choose which observable to measure at the time she receives the first qubit.[4]

The *binding* property is achieved by making use of quantum mechanics' laws, that show to be impossible for Alice to perform a measurement capable of providing her the information relative to the states of all qubits received from Bob. In other words, she cannot simultaneously get knowledge of $p_0(r_i|0)$ and $p_1(r_i|1)$ for all i's, otherwise she could always obtain the correct state of the qubit.[5]

At last, we focus on the *concealing* property. Bob cannot learn anything about Alice's results $(r_1, \ldots, r_n)$ from the measurements themselves, quantum non-locality and causality imply that even if Bob measures his portion of an entangled pair, he cannot discover Alice's local measurement (Alice's commitment), as long as they are spatially distant. Therefore, Bob he is unable to obtain any knowledge from her commitment until the *opening* phase.

4 Neuchâtel's Protocol

The e-voting protocol presented in (Galindo et al., 2015) was implemented in Neuchâtel, in the March 8^{th} 2015 federal referendum, and provides *cast-as-intended* verification, which is achieved with the use of return codes that

[4] Also, for Alice to learn the qubits' arrival times she must measure the photons on their arrival: one can not detect the photon without measuring it and it is the currently infeasible to perform non-demolition measurements.

[5] Obviously it is mandatory that the test cases include qubits whose states correspond to Alice's commitment and qubits whose states do not correspond, in order to make an appropriate validation statistic.

allow voters to check if their voting options were properly recorded by the voting server (obviously keeping their privacy) and only validate them if they are correct. Also, this protocol uses single vote casting, i.e, a vote is only considered to be cast after being confirmed by the voter, which avoids the need of multi-vote casting.

4.1 Cryptographic Background

The cryptography behind Neuchâtel's protocol is based on some important cryptographic methods: ElGamal cryptosystem[6] used several times throughout the protocol (we will denote the ElGamal encryption and decryption functions by e_K and d_K respectively, where K relates to the key space), Mix-Networks to eliminate the correlation between encrypted votes and decrypted ones, RSA Full Domain Hash Signature scheme (RSA-FDH signature) to sign specific information and Non-Interactive Zero Knowledge proofs (NIZK proofs) which are mainly used for verification purposes. More details on the used cryptography techniques can found on the full dissertation.

It follows some important algorithms' related notation which will be used hereinafter:

- `KGen`: having as input a group G, with generator α of prime order q of elements in $\mathbb{Z}_p^*$, where $p = 2q + 1$ is a safe prime; it outputs an ElGamal key pair (p_k, s_k), such that $pk \in G$ and $sk \in \mathbb{Z}_q$.
- `Enc`: inputting a message $x \in G$ and the public key pk, it chooses a random $k \in \mathbb{Z}_q$ and outputs the ElGamal encryption $e_K(x, k) = (y_1, y_2)$.
- `Dec`: it receives the ElGamal encrypted pair (y_1, y_2) and the private key sk and outputs the decrypted message $d_K(y_1, y_2) = x$.
- `SignKeyGen`: it generates a RSA key pair (pks, sks), such that $pks = \{pk_{RSA}, n_{RSA}\}$ with $n_{RSA} = pq$, where p and q primes, and $sks = sk_{RSA}$.
- `Sign`: for a given hashing function H, it takes as input any message x and the private key sks and outputs $y = H(x)^{sks} \pmod{n_{RSA}}$.
- `SignVerify`: given the hash function H it verifies the signature, by receiving as input the public key $pks = \{pk_{RSA}, n_{RSA}\}$, the message x and the signature y and checking whether $H(x) = y^{pk_{RSA}} \pmod{n_{RSA}}$.

4.2 Process Description

Neuchâtel's protocol has seven agents: Election Authorities, Voters, Registrars, Voting Server, Voting Device, Code Generator and Auditors. Next a step by step execution of the protocol is present, divided by its four main phases: *Configuration*, *Registration*, *Voting* and *Counting* phases.

Configuration phase:

i Election Authorities determine the voters participating in the election, by defining the set ID of their identities.

ii Election Authorities run the `Setup`(1^λ) algorithm: it first chooses a group G and then generates an ElGamal key pair (pk, sk), using `KGen`(G); it chooses a random $x_1 \in G$ which will induce a pseudo-random function f_{x_1}, in the family F of pseudo-random functions (its purpose will become clear later) and picks two hash functions H_1 and H_2. Then, an RSA key pair is generated $(pks = \{pk_{RSA}, n_{RSA}\}, sks = sk_{RSA})$, using `SignKeyGen` and finally it defines the set $V = \{v_1, \ldots, v_k\}$ of voting options, each of them represented by a small-bit length prime in G. In the end, the algorithm outputs the following: the election public/private key $(pk_e, sk_e) = ((pk, G, H_1, H_2), sk)$; the global code generation public/private key $(pk_c, sk_c) = (\perp, x_1)$ (note that $f_{x_1} = f_{sk_c}$); the signing public/private key $(pk_s, sk_s) = (pks, sks)$ and the set of voting options $V = \{v_1, \ldots, v_k\}$.

iii Election Authorities publish pk_e, pk_c, pk_s, ID and V in the Public Bulletin Board (PBB), provide sk_c to both the Registrars and the Code Generator and also provide sk_s to the Registrar.

Registration phase:

i Voter provides his $id \in$ ID to the Registrar, in order to participate in the election.

ii Registrars run the `Register`(id, sk_c, sk_s) algorithm: it generates an ElGamal key pair (pk, sk), using `KGen`(G), and chooses a random $x_2 \in G$. The algorithm outputs: the voter's code generation public/private key $(pk_{id}, sk_{id}) = (pk, sk)$; the voter's confirmation value $CV^{id} = x_2$; the voter's finalization value $FC^{id} = f_{sk_c}((CV^{id})^{sk_{id}})$, with a validity proof $\Pi_{FC^{id}} =$ `Sign`$(FC^{id}, sk_s) = H_1(FC^{id})^{sk_s} \pmod{n_{RSA}}$ for that finalization code; the set of voter's return codes, such that each return code is related to a voting option $\{v_i, RC_i^{id}\}_{i=1}^k = \{v_i, f_{sk_c}(v_i^{sk_{id}})\}_{i=1}^k$, and at last a set of reference values $\{RF_i^{id}\}_{i=1}^k = \{H_1(RC_i^{id})\}_{i=1}^k$.

iii Registrars publish pk_{id}, $\Pi_{FC^{id}}$ and $\{RF_i^{id}\}_{i=1}^k$ in PBB and provide (pk_{id}, sk_{id}), $\{v_i, RC_i^{id}\}_{i=1}^k$, CV^{id} and FC^{id} to the voter. The set $\{\{v_i, RC_i^{id}\}_{i=1}^k, CV^{id}, FC^{id}\}$ constitutes the voter's Verification Card.

[6] ElGamal's homomorphic property provides an alternative way to anonymize votes before decryption, although it is not used in this protocol because it is unfit to deal with multiple voting option scenarios like neuchâtel's referendum.

At this point *Configuration* and *Registration* phases are completed. Before continuing, recall that PBB has now the information

$$\left\{pk_e, pk_c, pk_s, \mathrm{ID}, V, \left\{pk_{id}, \Pi_{FC^{id}}, \{RF_i^{id}\}_{i=1}^{k}\right\}_{id\in \mathrm{ID}}\right\}.$$

Voting phase:

Upon entering this phase the user must already be prepared to authenticate himself. We won't dwell on the authentication subject, since it gets out of the scope of this thesis.

i Voter authenticates through the voting device to the voting server. If the authentication is successfully completed, the values id and pk_{id} are stored in the Voting Device.

ii Voter chooses a set of voting options $\{v_{j_1}, \ldots, v_{j_t}\} \in V$ and enters them into the Voting Device, together with sk_{id}.

iii Voting Device runs the $\texttt{Vote}(id, sk_{id}, \{v_{j_1}, \ldots, v_{j_t}\})$ algorithm: it computes the vote $v = \prod_{l=1}^{t} v_{j_l}$ and then encrypts it using the ElGamal encryption scheme $(c_1, c_2) = \texttt{Enc}(pk, v)$, where the key pk is the one presented in the election public key pk_e. Then, a partial computation of the Return Codes is calculated $(v_{j_1}^{sk_{id}}, \ldots, v_{j_t}^{sk_{id}})$ and $(c_1^{sk_{id}}, c_2^{sk_{id}})$ is computed. Finally, three Non-Interactive Zero Knowledge (NIZK) proofs are computed: π_{enc} proves knowledge of the random element used for encrypting v (using the ElGamal scheme), and obtaining (c_1, c_2); π_{exp} proves that the ciphertext (c_1, c_2) raised to the voter's code generation private key sk_{id} is actually $(c_1^{sk_{id}}, c_2^{sk_{id}})$ and π_{prod} proves that using the election public key pk_e to encrypt the product $\prod_{l=1}^{t} v_{j_l}^{sk_{id}}$ one obtains the ciphertext $(c_1^{sk_{id}}, c_2^{sk_{id}})$. The π_{enc} proof is meant to assure that the encryption of the voting options was actually performed by the Voting Device. The π_{prod} proof guarantees that $(c_1^{sk_{id}}, c_2^{sk_{id}})$ is calculated from the partial return codes corresponding to the correct voting options, and together with π_{exp} they aim to provide an evidence that the voting options contained in the ciphertext (c_1, c_2) are the same as the voting options used for the partial computation of the return codes, i.e, the voting options in the ciphertext are truly the voter's intended ones. The algorithm outputs the ballot $b = \left(id, (c_1, c_2), (v_{j_1}^{sk_{id}}, \ldots, v_{j_t}^{sk_{id}}), (c_1^{sk_{id}}, c_2^{sk_{id}}), pk_{id}, \pi_{enc}, \pi_{exp}, \pi_{prod}\right)$. Note that `Vote` is a probabilistic algorithm, due to the randomness of ElGamal encryption.

iv Voting Device sends (id, b) to the Voting Server.

v Voting Server executes $\texttt{ProcessBallot}(BB, b, id, pk_{id})$ algorithm: first it checks is there exists already a ballot associated to id in BB.

$$\begin{cases} \text{If yes, the algorithm outputs 0 and notifies the Voting Device} \\ \text{about the error.} \\ \\ \text{If not, the algorithm validates the proofs } \pi_{enc}, \pi_{exp}, \pi_{prod} \text{ and} \\ \text{if all are successful the algorithm outputs 1.} \\ \text{In this case the ballot box BB is updated with } (id, b), \text{ with the} \\ \text{status "ballot received" and the Code Generator is notified of} \\ \text{the new update in BB.} \end{cases}$$

vi Code Generator runs $\texttt{RCGen}(b, id, sk_c)$: it generates the final Return Codes $\{\overline{RC_{j_l}^{id}}\}_{l=1}^{t}$, such that $\overline{RC_{j_l}^{id}} = f_{sk_c}(v_{j_l}^{sk_{id}})$ for each $l = 1, \ldots, t$. Then it checks if $\{\overline{RC_{j_l}^{id}}\}_{l=1}^{t}$ is a subset of $\{RF_i^{id}\}_{i=1}^{k}$.

$$\begin{cases} \text{If not, the algorithm outputs } \perp \text{ and notifies the Voting Device} \\ \text{about the error/rejection.} \\ \\ \text{If yes, the algorithm outputs the unordered set of Return Codes } \{\overline{RC_{j_l}^{id}}\}_{l=1}^{t} \\ \text{and sends them to the Voting Server.} \end{cases}$$

vii Voting Server updates the status of ballot b in the BB to "return code generated" and forwards $\{\overline{RC_{j_l}^{id}}\}_{l=1}^{t}$ to the Voting Device.

viii Voting Device shows $\{\overline{RC_{j_l}^{id}}\}_{l=1}^{t}$ to the Voter.

ix Voter runs $\texttt{RCVerif}\left(\{v_{j_1}, \ldots, v_{j_t}\}, \{\overline{RC_{j_l}^{id}}\}_{l=1}^{t}, \{v_i, RC_i^{id}\}_{i=1}^{k}\right)$ algorithm, which outputs:

$$\begin{cases} 1, \text{ if } \{RC_{j_i}^{id}\}_{i=1}^{t} = \{\overline{RC_{j_l}^{id}}\}_{l=1}^{t} \text{ as sets.} \\ 0, \text{ otherwise.} \end{cases}$$

If the output of `RCVerif` is 1 the Voter confirms the ballot cast by providing CV^{id} to the Voting Device.

x Voting Device executes $\texttt{Confirm}(CV^{id}, id, sk_{id})$ which computes and outputs the confirmation message $CM^{id} = (CV^{id})^{sk_{id}}$, that will be sent to the Voting Server (together with id).

xi Voting Server forwards CM^{id} to the Code Generator.

xii Code Generator runs $\texttt{FCGen}(CM^{id}, id, sk_c, \Pi_{FC^{id}})$ algorithm: it uses the $\texttt{SignVerify}(pk_s, \overline{FC^{id}}, \Pi_{FC^{id}})$ algorithm, where $\overline{FC^{id}} = f_{sk_c}(CM^{id})$, outputting:

$$\begin{cases} \text{The finalization code } \overline{FC^{id}}, \text{ in case of success.} \\ \perp, \text{ otherwise. In this case the Voter is notified.} \end{cases}$$

In case of success, the finalization code $\overline{FC^{id}}$ is sent to the Voting Server.

xiii Voting Server stores $\overline{FC^{id}}$ together with the ballot b, updates the ballot's status to "confirmed" and forwards $\overline{FC^{id}}$ to the Voting Device.

xiv Voter should check if $\overline{FC^{id}}$ matches FC^{id} (received during registration). If it matches, $\overline{FC^{id}}$ serves the Voter as a confirmation of the correct submission of his vote. If it does not, the Voter may complain to election administrators, so the vote can be cast using another channel (e.g. a polling station).

Counting phase:

i Election Authorities run $\texttt{Tally}\big(\text{BB}, sk_e, \{\Pi_{FC^{id}}\}_{id\in\text{ID}}\big)$ algorithm: for all ballots in the Ballot Box BB having a finalization code stored together with the ballot, $\texttt{SignVerify}(pk_s, \overline{FC^{id}}, \Pi_{FC^{id}})$ is run, to select the ones which have been confirmed by the voters. The resulting set of ballots is then shuffled for privacy purposes using mix-nets, and for each ballot the decryption process $\texttt{Dec}(\{c_1, c_2\}, sk_e)$ is applied, to obtain the cleartext v. After that, v is factorized to recover from $v = v_1^{\beta_1} \dots v_k^{\beta_k}$ the factors v_i such that $\beta_i = 1$, i.e, the chosen voting options. At last, the voting options v_i are used to compute the final result r, which is, together with a proof Π of the tally correctness, the output of this algorithm.

ii Auditors execute the $\texttt{Verify}(\text{PBB}, r, \Pi)$ algorithm, which outputs:

$$\begin{cases} 1, \text{ in case of success. The result } r \text{ is announced as fair.} \\ 0, \text{ otherwise. Research is made to find the reasons of failure.} \end{cases}$$

4.3 Honesty Requirements

Some trust assumptions are required in order to guarantee *cast-as-intended* verifiability and/or the privacy of the voting process. To ensure *cast-as-intended* verifiability, one must consider that, for each pair presented next, at least one of the agents is honest: Voting Device and Code Generator; Voting Device and Registrars; Voting Device and Voting Server; Registrars and Code Generator. To guarantee the protocol's privacy, the following conditions are required: Voting Device is not compromised; Election Authorities are honest; Verification Card contents are only known to the voter.

5 Improvements on Neuchâtel's Protocol

5.1 Ring Signature

Suppose a group of entities such that each have a pair of public/private keys. A ring signature is a type of digital signature that can be performed by any entity of the group, while maintaining the anonymity, and which can be the verified be checked by anyone.

Definition 5.1. Ring Signature
*A ring signature scheme is a tripe of PPT algorithms (**Gen**, **Sign**, **Vrf**) such that:*

- $\textbf{\textit{Gen}}(1^k)$*, where k is a security parameter, outputs a public key PK and a secret key SK.*
- $\textbf{\textit{Sign}}_{SK_i}(M, R)$*, outputs a signature σ on the message M with respect to the ring $R = (PK_1, \dots, PK_n)$, assuming that:*
  1. *$(R[i], SK_i)$ is a valid key pair output by **Gen**;*
  2. *$|R| \geqslant 2$;*
  3. *each public key in the ring is distinct, without loss of generality.*
- $\textbf{\textit{Vrf}}_R(M, \sigma)$*, verifies a purported signature σ on message M with respect to the ring of pubic keys R.*

The ring signature scheme must satisfy the completeness condition, i.e, for any integer k, any $\{(PK_j, SK_j)\}_{j=1}^n$ output by $\texttt{Gen}(1^k)$, any $i \in [n]$, and any M, we have $\texttt{Vrf}_R(M, \texttt{Sign}_{SK_i}(M, R)) = 1$, where $R = (PK_1, \dots, PK_n)$.

Our suggestion is that we take the group with two entities, Voter and Registrar, considering that $R = (PK_1, PK_2)$, where $i = 1$ (resp. $i = 2$) refers to the Voter (resp. the Registrar), without loss of generality. We introduce a ring signature on the Verification Card (call it M) that the Registrar gives to the Voter at point iii of Registration phase. In other words, the Registrar signs M before giving it to the Voter, outputting σ, $\texttt{Sign}_{SK_2}(M, R) = \sigma$. Thus we show that:

Proposition 5.2. *A ring signature with respect to the ring R=(Voter, Registrar) upon the Verification Card allows the Voter to authenticate the received information, without enabling vote selling.*

Proof(Sketch): When the Voter receives σ, he uses the verification method provided in the ring signature scheme, to validate the Verification Card.[7] If $\texttt{Vrf}_R(M, \sigma) = 1$ then the authentication is successful. At the same time the Voter cannot sell the authenticated vote to another party, because it is impossible to the "buyer" to discover who did sign the Verification Card due to the properties of the ring signature. Then, the Voter is unable to prove that it was the Registrar who performed the signature, which prevents vote selling.

■

5.2 Multi-party Computation

Given a number of parties $p_1, \ldots, p_n$ each having their own secret input $x_1, \ldots, x_n$, and a public function F, the MPC creates a protocol such that, by interacting only between each other, the parties can correctly compute the value of $F(x_1, \ldots, x_n)$ without revealing any information about their inputs. MPC must achieve:

- *Input Privacy:* the exchange of messages during the execution protocol reveals no information about the private inputs provided by each party.
- *Correctness:* it is impossible to any proper subset of colluding participants to manipulate the protocol (by sharing information or deviate from the instructions) in such a way that it produces an incorrect output.
- *Independence of Inputs:* corrupted parties must choose their inputs independently of honest parties' inputs.

In Neuchâtel's protocol, if the Voter corrupts the Voting Device to get access to the randomness used to encrypt the vote, he will be able to construct a proof of its vote and then sell it. Then, we propose create a Multi-party computation between the parties Voting Device (p_1) and Registrar (p_2), such that p_1 has private input (k, v), where k is a random element and v is Voter's vote, while p_2 has private input the random element k'. The computation should output the proofs π_{prod}, π_{enc} (similar to the previously explained) and the value of the public function $F((k, v), k') = \texttt{ElGamal}(k \oplus k', pk, v) = \texttt{ElGamal}(k'', pk, v) = (c_1, c_2)$, where $\oplus$ is a determined operation and $\texttt{ElGamal}(k \oplus k', pk, v)$ represents the ElGamal encryption of v using $k \oplus k'$ as the secret random element required for the encryption and pk as the public key. The operation referenced as $k \oplus k'$ can be, for example, the bitwise Exclusive Or logical operation (XOR).

Proposition 5.3. *The implementation of a Multi-party computation between Voting Device and Registrar, as described above, makes the protocol robust against vote selling, without compromising the privacy and verifiability of the vote cast.*

Proof(Sketch): In fact, the presented MPC method takes away from the Voting Device (and consequently from the Voter) the knowledge about the randomness used in the encryption of the vote (due to input privacy and independence of inputs), eliminating the possibility to construct a proof of the vote cast and so, the possibility to sell the vote. At the same time, it keeps the privacy of the voting options due to the lack of knowledge about the randomness used in the encryption of the vote, and also the verifiability of the correctness of the encryption (through the NIZK proofs, π_{prod} and π_{enc}, and the correctness of MPC).

■

5.3 Bringing in Quantum Cryptography

Informally speaking, our goal now will be to reinforce the already existing e-voting protocol through the properties of the previously mentioned quantum methods. We shall replace the normal exchange of messages (namely keys) by quantum key distribution, implement quantum bit commitment wherever an agent must commit to some group element/key and introduce multi-party computation as explained previously. For the sake of brevity, this work won't display the protocol again. For a detailed presentation of the protocol with the specified variants, the reader should see the full dissertation.

The benefits of the implementation of quantum cryptography methods within the e-voting protocol and in the proposed schemes are stressed in the next property:

Proposition 5.4. *With the introduction of quantum cryptography to implement quantum key distribution, quantum bit commitment and multi-party computation within the e-voting protocol, we attain information security in the public exchanged messages for outside eavesdropper. In other words, the protocol is perfectly secure against an eavesdropper that does not take part of the protocol, since it has to break the key generated by the quantum key distribution.*

[7] Obviously the Voter knows that the signature was performed by the Registrar, since it is the only entity capable of doing so, apart from himself.

6 Conclusion

In this last chapter we should present some final reasoning about the analysis performed on the e-voting protocol and point directions for future researching.

A significant amount of the security increase could be achieved with the implementation of quantum methods, as exhibited in Section 5.3, namely in terms of the exchange of secret information/keys, which became shielded to unnoticed eavesdropping, and robust with respect to setup (commitment) of values with a relevant role throughout the protocol's operations.

Another valuable effort was addressed to correct some vulnerabilities found in the protocol, such as vote selling issues, or, as its counterpart, the required no-corruption assumptions. Those frailties are also exposed in Section 5, followed by the solutions to overcome those problems. Built upon Ring Signature schemes and Multi-party Computation, the presented ideas bring robustness to the protocol, against the mentioned topics.

Focusing now on the next steps to take towards the strengthening of the e-voting protocol, one could work on the formalization and rigorous construction of proofs on the security properties achieved by the proposed methods. Also, it is of interest to address the design and optimization of the required infrastructures for the quantum interactions between agents, aiming to construct a practical and feasible e-voting system, usable by the target remote voters. Finally, and as a long-term project, this protocol could be upgraded to provide the verification mechanisms that make it suitable for a higher percentage of the electorate.

References

Galindo, David, Guasch, Sandra, and Puiggalí, Jordi (2015). 2015 Neuchâtel's Cast-as-Intended Verification Mechanism. In: *E-Voting and Identity*, pp. 3–18. Springer.

Gjøsteen, Kristian (2011). The norwegian internet voting protocol. In: *E-voting and identity*, pp. 1–18. Springer.

Loura, Ricardo, Almeida, Álvaro J, André, Paulo S, Pinto, Armando N, Mateus, Paulo, and Paunković, Nikola (2014). Noise and measurement errors in a practical two-state quantum bit commitment protocol. In: *Physical Review A* 89.5.

Mayers, Dominic (1997). Unconditionally secure quantum bit commitment is impossible. In: *Physical review letters* 78.17, p. 3414.

Nielsen, Michael A and Chuang, Isaac L (2010). *Quantum computation and quantum information*. Cambridge university press.

Stinson, Douglas R (2005). *Cryptography: theory and practice*. CRC press.