Open Mathematics

Research Article

David Yeregui Marcos del Blanco*, Luis Panizo Alonso, and Jose Angel Hermida Alonso

Review of Cryptographic Schemes applied to Remote Electronic Voting systems: remaining challenges and the upcoming post-quantum paradigm

https://doi.org/10.1515/math-2018-0013
Received April 21, 2017; accepted December 7, 2017.

Abstract: The implantation of Remote Electronic Voting (REV) systems to Electoral Processes is happening at a slower pace than anticipated. One of the relevant factors explaining that reality is the lack of studies about the Cryptographic Schemes and Primitives applied to the existing REV solutions. In this paper, the authors review the main cryptographic schemes applied to date, as well as the most relevant Post Quantum research in the field. The aim is twofold: contribute to clarify the strengths and weaknesses of each scheme as well as expose the remaining challenges, as a necessary step towards a broader introduction of REV solutions in binding elections.

Keywords: e-democracy, Internet Voting, Remote Electronic Voting, Cryptography

MSC: 06B20, 65A05, 81P94

1 Introduction

The Information and Communications Technologies (ICT) have had a huge impact in the day-to-day lives of billions of citizens around the globe in recent years. In the early 2000s, it was widely anticipated that its range would also include public elections and other democratic processes, as an integral part of what was labeled "e-democracy".

More than a decade has passed, and the forecast has not turned into a reality. Some countries such as Estonia, Australia, Norway, Spain, Switzerland, or Canada have implemented e-voting pilots for legally binding elections totalling more than 6 million people. Others, including Germany, Norway, UK, the Netherlands or the US have decided to discontinue their respective e-voting projects.

Certainly, e-voting introduces features making it an especially demanding discipline within the plethora of ICT applications [1]:

– The need to comply simultaneously with two antagonistic properties: Integrity and Privacy [2].

***Corresponding Author: David Yeregui Marcos del Blanco:** Mechanical, Computing and Aerospace Engineering Department, Universidad de Leon, Spain, E-mail: dmarcb01@estudiantes.unileon.es
Luis Panizo Alonso: Mechanical, Computing and Aerospace Engineering Department, Universidad de Leon, Spain, E-mail: luis.panizo@unileon.es
Jose Angel Hermida Alonso: Mathematics Department, Universidad de Leon, Spain, E-mail: jahera@unileon.es

– The outcome gives tremendous power and money resources to the winner, thus attracting very powerful attackers.
– The results can be very difficult to revert in the event of a fraud discovered after the elections ended.

An additional hindrance are the attack vectors, which have been repeatedly used:
– The voter's device, with 30-40 percent of them infected by malware and located in non-controlled environments [2]
– The network itself, with a relevant track record of attacks on the associated cryptographic protocols [3,4]
– The i-voting (or e-voting) system, which oftentimes carries vulnerabilities serious enough to put the elections in jeopardy [5-7]

All the aforementioned could sound alarming, but nothing further from reality: there are several examples of attacks coming from foreign nations during i-voting pilots in the US, Australia and Ukraine among others [8-11].

Additionally, in the latest US General Elections, the Office of the Director of National Intelligence stated: "Putin and the Russian Government aspired to help President-elect Trump's election chances ... by discrediting Secretary Clinton...CIA and FBI have big confidence in this judgement". The original document is publicly available: "Assessing Russian Activities and Intentions in Recent US Elections" [12,13].

In order to confront the aforementioned security issues, there have been innovative research approaches such as the development of evaluation methodologies based on the Council of Europe's recommendations [14-16].

Despite all the efforts, REV implantation was facing a gradual slowdown, partly because of a lack of long-term secure cryptographic approaches. Fortunately, the landscape has started to change thanks to the lattice-based cryptography and other very promising post-quantum schemes [17,18] which are reviewed in the present article.

1.1 Contribution

In the present paper, the authors analyze the main cryptographic schemes and their application to REV systems, including the main security flaws. Subsequently, the most relevant post-quantum cryptographic schemes applied to REV systems are presented, with a focus on the current development stage and pending issues.

The principal contribution is the analysis itself, since there are very limited examples of such research exercises for both "traditional" and post quantum schemes. The authors also sum up the most significant open problems related to cryptograhic schemes applied to REV solutions.

1.2 Structure of this article

Section 1 introduces cybersecurity applied to e-democracy, including recent attacks and new approaches facing recent tasks. Section 2 deals with main definitions, protocols and requirements for a REV systems to be used in this article.

In Section 3, the main REV cryptographic schemes currently applied to REV systems are presented. Some limitations and open problems are also identified.

Section 4 introduces an analysis of the main post quantum schemes applied to e-voting with a focus on the existing limitations and future developments.

Lastly, in Section 5 the main conclusions of this article are presented, together with an outline of the issues to be solved before incorporating e-voting as a fully-secure option for legally-binding national elections.

2 Definitions

As a prior clarification, it is worth noting that in this article, a Remote Electronic Voting (REV), e-voting or i-voting system is defined as: "*A voting system used in a remote, non-controlled environment, through electronic means, in which the vote is sent partially or totally via an internet connection from a personal computer or mobile device which has not been specifically designed as a specialized electronic voting machine*".

Therefore, and according to the previous definition, in this paper REV does not include e-voting systems in controlled environments and/or using specifically designed machines to vote such as DRE voting devices. Those kind initiatives have been thoroughly studied in [19].

Zero Knowledge Proof (ZKP): A ZKP is a protocol between two parties: the prover and the verifier with a language $L = \left\{x \in A^* : \exists w : R(x, w)\right\}$, R being a polynomial-time predicate in a parameter k and x and w strings of length k. The ZKP protocol allows the prover to convince the verifier that she is in possession of a witness w about the fact that $x \in L$. The properties required for ZKP protocols are: **completeness, soundness and zero-knowledge** [20].

When the protocols have a three-move structure (commitment, challenge and response), they are called sigma protocols or Σ **protocols**. They have several interesting properties:

- Can be repeated in parallel.
- Can be combined to prove I know a witness for x OR/AND for x'.
- Can be transformed into Non-Interactive Zero Knowledge Proofs (see below).

Regarding e-voting, ZKPs have several applications depending on the element involved:

- ZKP by the **Voter**:
  1. Proves that the encrypted vote in fact contains a valid value (usually 0 or 1).
  2. It does not reveal the value itself.
- ZKP by the **Authorities**:
  1. Proves that the decryption is correct, namely:
     (a) The correct private key, corresponding to the election public key has been used.
     (b) The result of the election corresponds to the tally of votes present in the Bulletin Board.
  2. It does not reveal the election private key.
- ZKP for **mix-nets** (see Section 3),
  1. Proves that the re-encryption has been performed correctly.
  2. It reveals neither the randomness used nor the permutation applied.

An especial variation of ZKP are the non-interactive ZKP or NIZKP. In this type, the prover is able to produce a string π in one move, which can by itself convince the verifier about the status of $x \in L$.

Therefore, a NIZKP requires a public parameter p produced by an independent third party [2]. Probably the most efficient construction of NIZKP is the Fiat-Shamir heuristic [21]. It is widely used in many well known e-voting systems like Helios Voting [22], although in [23], Bernhard et al. identify the use of weak Fiat-Shamir proofs as a source of attacks against the ballot privacy in Helios.

2.1 REV protocol

For the definition of a standard e-voting/REV protocol, Helios Voting [21] is selected, since it is one of the best-known tools, widely accepted by the research community. The entities are:

- A: the authority that handles the registration of users and updates the public list of legitimate voters.
- BB: the Bulletin Board. It checks the well-formedness of received ballots before they are cast.
- T: a set of trustee(s) in charge of setting up their own decryption keys, and computing the final tally function.

And the phases, according to [22,24] are:

1. The election is created by naming an election officer, selecting a set of trustees and introducing the list of authorized voters.
2. The Ballot Preparation System (BPS) generates the ballot as well as a distributed key pair pk, sk (public and private key respectively).
3. Each voter receives an email containing her user ID, password and election URL. Upon clicking, the Javascript starts and downloads the parameters.
4. The voter selects her option and the BPS encrypts it with pk. The vote also contains a NIZKP to verify that the vote is well formed (preventing a malicious voter to introduce an integer i value instead of 1, allowing the ballot to incorrectly represent i votes), because votes are not decrypted individually to be tallied. As a replacement, the homomorphic properties of exponential ElGamal [25] are used.
5. The Software client shows the voter a hash of her encrypted vote. The voter then has two options:
   (a) Audit the ballot: the voter receives the nonce used to encrypt her vote. She can use it to verify that her vote has been included and that it represents her elected option. However, the audited ballot is no longer valid and the voter has to restart with the voting process. Therefore, Helios Voting is a cast-or-audit type of i-voting tool. The voter can verity her vote as many times as she wants; until she is convinced that Helios is trustable.
   (b) Seal the vote: in order to send it, the BPS will ask to provide her user ID, password to be identified. The voter sends her user ID, password, encrypted ballot and Zero Knowledge Proof to the server, which verifies that all the information is correct.
6. Once the voting phase is over, the server publishes the Bulletin Board with all the encrypted votes, together with the voter's name.
7. Each of the trustees publishes a partial decryption of the encrypted tally, together with a signature of knowledge proving the partial decryption's correct construction. Anyone can verify those proofs.
8. The election officer decrypts the tally and publishes the result. Anyone can check the decryption.

Using a formal description and for the simplified case of a yes/no type of election: In the case of three voters: Alice, Bob and Charlie, if Alice wants to vote for option a, her vote is represented as v_a which is encrypted with the public key p_k, obtaining $\{v_a\}_{p_k}$. There is a Zero Knowledge Proof attached to the vote in order to verify that the vote is valid; which means that either $v_a = 0$ or $v_a = 1$. The verification is critical because if it didn't exist, a malicious voter could send $v_a = i$, being i a positive or negative integer, making her vote amount for i valid votes instead of 1. The Zero Knowledge Proof for v_a is identified as ZKP_a. Subsequently, Alice sends $\{v_a\}_{p_k}$, ZKP_a to the Bulletin Board (BB). Since the BB is public, Alice can check whether her vote has arrived or not.

2.2 Security requirements of a REV system

For this paper, the authors include classifications of security requirements for REV systems from two different standpoints:

– Semantic and Active Security [26]:
  1. **Semantic Security**, or indistinguishability under chosen-plaintext attacks (IND-CPA). Given a bit $b \in \{0, 1\}$ generate a public/secret key pair, make the public key available to the attacker, who must reply with two valid messages m_0, m_1 , encrypt m_b using the public key and give the resulting "challenge ciphertext" to the attacker. The scheme is said to be or IND-CPA secure if it is not feasible for an attacker to distinguish between $b = 0$ and $b = 1$, i.e., the probability of accepting both cases should be negligible.
  2. **Active Security**, indistinguishability under chosen-ciphertext attacks (IND-CCA). It provides the attacker with a decryption oracle, which runs the decryption algorithm (with the secret key) on any

ciphertext the attacker queries, except for the challenge ciphertext itself. The scheme is said to be actively (or IND-CCA) secure if it is not feasible for an attacker to distinguish between $b = 0$ and $b = 1$.

- The REV requirements according to the most widely accepted definitions of **End to End Verifiability (E2Ev)** [27] and **Privacy** [28] for e-voting systems. The correlation between E2Ev, privacy and the five requirements for a democratic election according to the Council of Europe (universal, equal, free, direct and secret) was defined in [15].
  - **End to End Verifiability (E2Ev)** : Unfortunately, there does not exist a formal, universal definition for (E2Ev). The main reason is the lack of automated proofs. There are several tools for the symbolic analysis of security protocols such as ProVerif [29] and other prototype tools accepting equivalence properties such as AKISS [30] or APTE [31]. For e-voting systems, they all face the same unsolved challenge: associative and commutative operators are out of reach for all of them, making it impossible to analyze the following homomorphic property [32]:

$$enc\,(pk; v_1) * enc\,(pk; v_2) = enc\,(pk; v_1 + v_2)$$

    Therefore, the challenge of formally defining verifiability remains unsolved and its analysis must be performed case by case. Currently, the most widely accepted definition for E2Ev [27] includes the following three properties:
    * Cast as Intended: voters can get convincing evidence that their encrypted votes accurately reflect their choices.
    * Recorded as Cast: voters can check that their encrypted votes have been correctly included by finding exactly the encrypted value they cast on a public Bulletin Board (BB).
    * Tallied as Recorded: Any member of the public can check that all the published encrypted votes are correctly included in the tally, without knowing how any individual voted.

    The first two are usually referred to as individual verifiability and the last one as universal verifiability. In the case of Helios Voting, for the Cast as Intented property, it introduces the *cast-or-audit* approach [21], shared by most of the current e-voting tools. The voter can audit her vote as many times as she wants, until she is convinced that Helios is trustable. Regarding Recorded as Cast, the voter receives a hash of her encrypted vote, which can later check on the BB. Finally, for the Tallied as Recorded condition, ElGamal [25] together with the Sako-Kilian mixnet (see Section 3) are implemented. They also include a ZKP as previously explained. A complete analysis of the E2Ev in Helios Voting system is performed in [33].
  - **Privacy** It can be categorized in 3 increasing demanding levels:
    * Voter's privacy: The voter's vote is not revealed to anyone.
    * Receipt-freeness: The voter cannot obtain any information (like a receipt) which could prove a coercer how she voted.
    * Coercion Resistance: a voter cannot cooperate with a coercer to prove him that she voted in a certain way (even if she is willing to).

    Hirt and Sako proved in [34] that even receipt-freeness does not suffice for privacy in e-voting systems. Therefore, the required level is the highest one: Coercion Resistance (CR). Originally, the CR concept was introduced by Juels et al. in [28].

    In order to prevent coercion during the voting, it is mandatory to provide a system where re-voting is fully secret. Here, Plaintext Equivalence Tests apply [35]. On the other hand, several threads remain unchanged: imperfect private channels, dishonest Bulletin Board and DDoS attacks are still not considered. An interested reader is thus referred to [35] for a detailed explanation.

    Currently, no e-voting system is coercion-resistant according to the definition in [28]. Additonally, each country has a different legal approach regarding privacy. In Australia, the Authorities consider the coercion risk as low, and therefore do not include coercion-resistance as a requirement for an e-voting system [36]. In Estonia and Norway (two of the countries with a longest tradition of e-voting in binding-elections), despite the existing bibliography, the Governments decided to implement a voting scheme with receipts, thus failing to comply even with the second degree of privacy [37,38].

3 Main REV cryptographic schemes

3.1 Mixnets

Introduced in 1981 by D. Chaum [39] for anonymous communications, and subsequently applied to voting systems in 1992 by Fujioka et al. [40]. A mix-net defines a sequence of proxy servers in which each one of them takes as input a set of ciphertexts (encrypted votes in the case of a REV system) obtained through Public Key Protocols, re-encrypts them, shuffles them following a secret permutation and sends the output to the next proxy server, which proceeds in the same way.

In reality, the aforementioned sequence corresponds to a re-encryption mix-net, which is the most popular type in the development of REV solutions. The main reason is that, as opposed to decryption mix-nets, it suffices that just one server is honest to guarantee the vote's anonymity.

In order to verify that every server is honest, ZKP are performed in each one of them, considerably increasing the computational complexity of the scheme.

Formally, it consists of an algorithm $Shuffle()$ which receives as input a public-key pk and a sequence of ciphertexts $\psi = (\psi_1, \ldots \psi_n)$. $Shuffle()$ then produces as output another sequence of ciphertexts $\psi' = (\psi'_1, \ldots \psi'_n)$ and a proof π (usually a NIZKP).

The previous NIZKP ensures the following about the values (p_k, ψ, ψ'):

Let M be the sequence of plaintexts in which the i-th position equals $Dec(\psi_i)$ and M' the sequence of plaintexts in which the i-th position equals $Dec(\psi'_i)$. It should hold that there is a permutation μ over n elements such as the j-th element of M is equal to the $\mu(j)$ element of M' $\forall j = 1, \ldots, n$.

The $Shuffle()$ algorithm can be applied sequentially over the same sequence of ciphertexts. In such case, the correspondence between the original sequence and the final sequence of ciphertexts will be hidden even with just one honest server.

In e-voting, one of the most common ways to re-encrypt with a new randomness is to apply exponential ElGamal [20,25] so that the plaintext (vote) stays unchanged and the new encryption cannot be linked to the old one:

Let $(c, d) = (g^r, g^m \cdot y^r)$ be an encryption of the plaintext g^m using randomness r. We take an encryption (with the same public key y) of 1, using a random value:

$$(c_1, d_1) = (g^{r'}, g \cdot y^{r'}).$$

By multiplying those two ciphertexts:

$$(c, d) \cdot (c_1, d_1) = (g^{r+r'}, g^m \cdot y^{r+r'}),$$

we obtain an encryption of the same plaintext g^m using a different randomness.

Concerning mix-net based e-voting systems, the most important advantages are:

- They effectively break the link between voter and vote, contributing to the privacy.
- Better performance in elections with relevant differences in the number of voting options such as Australia [15].
- The ZKPs are not performed in each server, reducing the burden to the voter's device, usually not powerful enough for those complex operations.

Regarding the main disadvantages:

- The computational burden eased to the voter has to be carried by the REV system. Therefore, the amount of technical resources allocated is higher than in the case of homomorphic encryption based e-voting systems [15].
- The tally cannot start until the ballot box is closed (the last server has re-encrypted and mixed the last vote), which in practice can cause important delays for bigger elections (as happened in Norway in 2013 [38]).

– Mix-net based e-voting systems are more vulnerable to DDoS attacks since all mix-net servers need to be available for the tally.

In conclusion, mix-net based e-voting systems are better suited for elections with a vast array of voting options (many candidates, with preference etc.) but they also require a bigger investment in infrastructure and tend to be more vulnerable to DDoS attacks because of its nature and the fact that the tally cannot start until the voting process is over.

There have been some improvements in the protection againts DDoS attacks, including the development of a distributed version of Helios [39] and implementation of Paxos [40], but a certain degree of vulnerability still remains. In practice, many of the most relevant e-voting programs implement a hybrid system with both mix-net and homomorphic encryption (see Section 3.2) algorithms, such as the ones deployed in Australia and Norway [36,38].

3.2 Homomorphic encryption

Homomorphic encryption allows computing on data while it is encrypted rather than having to decrypt it first. Regarding e-voting, the additively homomorphic encryption has important applications:

The space of plaintexts is an abelian group $(G, +)$: Given the ciphertexts $\psi_1 = Enc(p_k, m_1)$ and $\psi_2 = Enc(p_k, m_2)$, there is a ciphertext ψ which can be computed and corresponds to an encryption of $Enc(p_k, m_1 + m_2)$. It is also required that if at least one of ψ_1, ψ_2 is uniformly shaped over all ciphertexts, the output follows the uniform distribution over all ciphertexts of $m_1 + m_2$.

The two main direct applications for e-voting are:

– Given k ciphertexts $\psi_1, \ldots \psi_k$ encoding k votes $v_1, \ldots, v_k$ and defining $\mathbb{B} \in \{0, 1\}$, it is possible to derive a single ciphertext which encodes $T = \sum_{i=1}^{k} v_i$. In practice, T is used to derive the tally of an election if each plaintext vote v_i corresponds to the choice made by the i-th voter.
– It is possible to refresh the randomness of ψ for $\psi_1 = Enc(p_k, m)$ by processing ψ with $Enc(p_k, 0)$. This application is very uselful for re-encryption mix-nets as explained in Section 3.1.

Specifically, one of the most widely used additively homomorphic-based cryptosystem is ElGamal [25]. In this case, the group is $(G, +) = (\mathbb{Z}_q^*, \cdot)$. The main functions of the expotential ElGamal (implementing additive homomorphism) are:

Given parameters (p, q, g) where p and q are large primes such that $q|p - 1$, g is a generator of the multiplicative group $\mathbb{Z}_q^*$ and a number n of trustees, ElGamal defines the following operations:

– Distributed key generation: Each trustee $i \in n$ selects a private key share $x_i \in \mathbb{Z}_q^*$ and computes a public key share $h_i = g^{x_i} mod\ p$. The public key is $h = h_1 \cdot \ldots \cdot h_n\ mod\ p$.
– Encryption: Given a message m and a public key h, selects a random nonce $r \in \mathbb{Z}_q^*$ and derives the ciphertext $(a, b) = (g^r\ mod\ p, g^m \cdot h^r\ mod\ p)$.
– Re-encryption: Given a ciphertext (a, b) and a public key h, selects a random nonce $r' \in_R \mathbb{Z}_q^*$ and derives the re-encrypted ciphertext $(a', b') = (a \cdot g^{r'}\ mod\ p, b \cdot h^{r'}\ mod\ p)$.
– Homomorphic addition: Given two ciphertexts (a, b) (a', b'), the homomorphic addition of plaintexts is computed by multiplication $(a \cdot a'\ mod\ p, b \cdot b'\ mod\ p)$
– Distributed decryption: Given a ciphertext (a, b), each trustee $i \in n$ computes the partial decryption $k_i = a^{x_i}$. The plaintext $m = log_g M$ is recovered from $M = b/(k_1 \cdot \ldots \cdot k_n)\ mod\ p$.

The authors have selected ElGamal to illustrate the homomorphic encryption subsection because it is the most widely used cryptosystem in the REV field. Prominent real-life e-voting deployments implementing ElGamal include: Helios Voting for the IACR elections [43] and the Scytl e-voting tool [44] in the binding elections of Norway [38], Australia [36], Switzerland [45] and France [46]. Other relevant homomorphic cryptosystems include RSA [47] or Paillier [48], but they have not been so profusely implemented in REV systems for encryption.

Regarding **security issues** with e-voting systems implementing homomorphic encryption schemes, there have been reported vulnerabilities at least in the Norwegian [49], Estonian [50], and Australian [10] elections, exploiting export-level RSA deficiencies in the latter. Although no massive attack could be proved in any case, it probably affected in the Norwegian Government decision of cancelling the e-voting implantation in 2014.

Another very relevant issue in the homomorphic encryption schemes, is that they are **malleable** by definition, which means that they do not provide active security (see Section 2.2): an attacker could apply a known operation to the challenge ciphertext, send the result to the oracle, and apply the inverse of the operation to reveal the plaintext selected by the challenger. The problem is shared by partially homomorphic systems, as the ones introduced in the present section 3.2.

Additionally, ElGamal, Paillier and RSA cryptosystems are believed to be vulnerable to quantum attacks, as exposed by Shor [51]:

- ElGamal's security is broken if dicrete algorithms can be computed efficiently (which is the case with Shor's algorithm). This is due to the fact the private part can be recovered from the public key by finding the x such that $\alpha^x \equiv \beta \, (mod \, p)$ given everything but x.
- Paillier and RSA security is broken if large primes can be efficiently factored (which is the case with Shor's algorithm). This is caused by the fact that the private part can be recovered from the public key by factoring n.

In order to solve the vulnerabilities related to quantum attacks, Gentry introduced in 2009 the first plausible construction for a fully homomorphic encryption scheme (FHE), using lattice-based cryptography [17]. Gentry's revolutionary research has led to innovative proposals of e-voting systems resistant to quantum attacks. The most relevants are reviewed in Section 4.

To sum up, partially homomorphic schemes such as ElGamal are implemented in most of the e-voting systems like Helios Voting or Scytl. The problem is that they are malleable by definition and vulnerable to quantum attacks. Therefore, FHE alternatives are necessary to guarantee sufficient levels of long term security for e-voting systems.

From a user's standpoint, homomorphic encryption schemes require that the voters provide evidence that their cast votes are valid through ZKP, usually very demanding operations for user level PC's.

The biggest advantages of e-voting systems based on homomorphic encryption schemes are:

- They are very effective in the tally process because the votes do not need to be decrypted individually.
- The tally can start before the end of the election, which in practice is a big advantage (voters demand speed in result publication).
- There is no need for an anonymous channel (unlike for blind signature schemes).

3.3 Blind Signature

The blind signature scheme was originally introduced by D. Chaum in [39] and designed to be used in telematic payments. In 1992 Fujioka et al. [40] applied it to voting systems. It implements a type of digital signature in which the authority signs the message without having access to its content. The analogy with the carbon paper exemplifies it: the sender encloses the message in a carbon paper envelope. If the sender is successfully identified, the authority signs the envelope without opening it (hence without accessing the message). A message is valid only if it includes the authority's signature. An example of a voting application of the blind signature scheme based on RSA would be [52]:

Let (n, e) and (n, d) be respectively the authority's public and private signature.

The sender generates a random value r such as $GCD(r, n) = 1$ and sends it to the authority:

$$v' = v \cdot r^e \; mod \; n$$

Therefore the value r is used to hide or "blind" the vote v to the authority. The authority signs the blinded vote and returns s'

$$s' = \left(v'\right)^d \; mod \; n \; = \; v^d \cdot \left(r^e\right)^d \; mod \; n$$

Since the sender knows r, she can obtain the signature s by computing:

$$s = s' \cdot r^{-1} \; mod \; n \; = \; v^d \cdot r \cdot r^{-1} \; = \; v^d \; mod \; n.$$

Once the sender receives the vote signed by the authority, she sends it to a mix-net (Section 3.1) to break the link between the vote and voter.

To avoid potential RSA malleability-related problems, hash functions [2] must be applied to the vote v. Otherwise, the system would be vulnerable to RSA *binding* attacks and variants [53].

Blind Signature schemes are the most efficient for ballot tallying but they present serious weaknesses:

- Verifiability limitations, especially regarding mix-nets because no external party can verify that only (and all) the valid votes had been counted.
- Blind Signature based e-voting systems do not include any protocol for voters who decide not to vote. Therefore, any attacker (or even a corrupt authority) can send all those "non-claimed votes" and no party can even verify whether that *"ballot stuffing"* happened or not.
- They demand anonymous communication channels, very difficult to achieve in reality.

Consequently, blind signature based schemes are currently not extensively used for the development of REV systems. There has been one relevant proposal in 2003, but it never reached a fully-developed stage [54].

4 Post Quantum schemes

As previously exposed in this article, the security of all probably secure cryptochemes still rely on classical assumptions. It means that they could be compromised in polynomial time by a quantum computer, as originally detailed in 1994 by Peter Shor in the case of RSA [51].

Regarding e-voting, REV systems based on such cryptoschemes are also vulnerable and therefore it is crucial to develop quantum-safe protocols in order to assure a promising future for internet voting.

4.1 Full Homomorphic Encryption and Lattices

Full Homomorphic Encryption (FHE) was first defined in 1978 [55], without even knowing if it was solvable or not. As previously presented, the first fully homomorphic encryption scheme was presented by Gentry using lattice-based cryptography [17].

Currently, despite interesting improvements, in one of the most efficient implementations [56], the homomorphic evaluation of a single AES-128 encryption operation takes 4 minutes with a state of the art computer with an amortized rate of 2 seconds per block.

FHE allows for arbitrary computations on encrypted data, that is to say, if a user has a function f and wants to obtain $f(m_1, \ldots, m_n)$ for some inputs $m_1, \ldots, m_n$, it is possible to compute on encryptions of these inputs, $c_1, \ldots, c_n$ instead; obtaining a result which decrypts to $f(m_1, \ldots, m_n)$.

In FHE, the plaintext is masked with inner and outer randomness: the former is labeled as noise. Subsequently, additions and multiplications can be performed; although in certain occasions a NAND gate has to be used.

Every operation increases noise level. When the noise level reaches the same size as the outer randomness, the ciphertext will not be decryptable anymore. Multiplications insert a lot more noise than additions. The noise problem can sometimes be solved using bootstrapping (the ciphertext is encrypted again) and the inner encryption is removed by running the decryption circuit in an encrypted state. The parameters can be adjusted so that the resulting ciphertext has a lower noise level than the original one. Nonetheless, bootstrapping is computationally expensive, so it must be handled carefully.

Regarding lattices, formally a lattice is a set of points in an n -dimensional space with a periodic structure. An n-dimensional lattice L is any subset of $\mathbb{R}^n$ that is both:

- an **additive subgroup**: $0 \in L$ and $-x, x + y \in L$ for every $x, y \in L$; and
- **discrete**: every $x \in L$ has a neighborhood in $\mathbb{R}^n$ in which x is the only lattice point.

Given k-linearly independent vectors $b_1, \ldots b_k \in \mathbb{R}^n$, the rank k lattice generated by them is the set of vectors [17]:

$$L(b_{1,\ldots,b_n}) = \left\{ \sum_{i=1}^{k} z_i b_i : z_i \in \mathbb{Z} \right\} = \left\{ Bz : z \in \mathbb{Z}^n \right\}$$

Most of the lattice problems exist in two different versions:

- **The exact problem**: it is a particular instance of the approximation problem where the approximation factor is $\gamma(n) = 1$,
- **The approximation problem**: it is a generalization defined with an approximation factor $\gamma(n)$ in terms of the lattice dimension.

Lattice cryptography became very popular since Regev [57] discovered a quantum reduction from the natural lattice shortest vector problem (SVP) or finding a short basis of independent vectors (SIVP). Lattice problems such as (ring) learning with errors ((R)LWE) are considered to be hard to solve, even for a large quantum computer. In more detail, some of the main lattice problems are:

- **Approximate Shortest Vector Problem** $(\gamma - SVP)$ Given a lattice basis B, find a non-zero vector $v \in L(B)$ such that $\|v\| \leq \gamma \cdot \lambda_1(L)$ with $\lambda_1(L)$ being the the minimum distance between two points belonging to the lattice.
- **Approximate Closest Vector Problem** $(\gamma - CVP)$ Given a lattice basis B and a target vector t (not necessarily in the lattice), find a lattice point $u \in L(B)$ such that:

$$v = \underset{v \in L}{argmin} \|t - v\|, \|u - v\| \leq \gamma \|t - v\|$$

- **Shortest Independent Vector Problem** $(SIVP)$
  Given a lattice basis B and a parameter $q \in \mathbb{Z}$, find a set of shortest q linearly independent lattice vectors (i.e., a set of linearly independent vectors $s_1, \ldots, s_q \in L(B)$ such that $s_1, \ldots, s_q \leq \lambda_q(B)$).
- **Learning with errors problem** (LWE): Originally introduced by Regev [57] and further improved by Lyubashevsky et al. to obtain ring variants [58] and by Ducas et al. among others to obtain homomorphic encryption [59].
  Let $\alpha \in \mathbb{R}^+$ be a noise parameter, $(s_1, \ldots, s_n)$ be a uniformly distributed binary secret in $\mathbb{B}^n$, and $G \subseteq \mathbb{T}^n$ a sufficiently dense finite discretization group. LWE (s, α, G) is the scaling-invariant instance.
  A random LWE sample of a message $\mu \in \mathbb{T}$ is defined as an element $(a, b) \in G \times \mathbb{T}$ where $a = (a_1, \ldots, a_n) \in G \subset \mathbb{T}^n$ is a uniform sample of G, and b is equal to $\sum_{i=1}^{n} s_i a_i + \mu + e \in \mathbb{T}$, where e is statistically close from a zero-centered continuous Gaussian sample of $\mathbb{T}$ over α.
  The phase of a LWE sample $(a, b) \in \mathbb{T}^n \times \mathbb{T}$ is defined as:

$$\varphi_s((a, b)) = b - \sum_{i=1}^{n} s_i a_i \in \mathbb{T}$$

  The security in LWE relies on the other two parameters: the Gaussian error parameter α and the number n of bits corresponding to the entropy in the secret. According to standard lattice reduction estimates [60], LWE is 128-bit secure for α equal to 2^{-10}, 2^{-30} or 2^{-50} if $n \geq 300$, 800 and 1500 respectively. Additionally, for any α and $n = \Omega(log(\frac{1}{\alpha}))$, LWE benefits asymptotically from the worst case to average case reduction, depending on the shape of G [61]. Any reader interested in the complete definitions of the aforementioned variables, formulae and problems can refer to [57-61].

Once the definitions related to FHE, lattices and their problems have been exposed, the properties applied to e-voting schemes are explained in the following Sections 4.2 and 4.3 for the Helios Voting and Norwegian cases.

4.2 LWE-based E-Voting Scheme (Helios)

The FHE and lattice-based cryptosystems are still a fairly new research topic and, therefore, there are not many implementations in the e-voting field.

Nonetheless, Chillotti et al. proposed in 2016 a tentative post-quantum protocol [18] based on the open-source e-voting reference Helios Voting [22]. The construction is based on LWE fully homomorphic encryption, as presented in the previous Section 4.1.

Compared to the traditional Helios, LWE-based Helios presents the following differences:

- It is built on post-quantum primitives: unforgeable lattice-based signatures, LWE-based homomorphic encryption and trapdoors for lattices.
- It does not implement ZKP. The original Helios includes two: one so the voter can prove that her vote is valid and another when the trustees decrypt the final result.
  In the post-quantum protocol, the first ZKP is substituted by Ducas et al. FHE-based bootstrapping [59]. The second one is replaced by publicly verifiable ciphertext trapdoors generated with GPV lattice signatures [62].
- In order to guarantee privacy even if some authorities were corrupt, they rely on concatenated LWE schemes rather than Fiat Shamir secret sharing [21].

4.2.1 Post quantum cryptographic schemes

Regarding **ballot privacy**, it is considered that LWE-Helios *Vote* protocol fulfills it if there exist an efficient simulator such that, for any Probabilistic Polynomial Timing (PPT) adversary A, the following holds [18]:

$$\left| Pr\left[Exp^{bpriv,\beta}_{A,Vote}(\lambda) = \beta \right] - \frac{1}{2} \right|.$$

With respect to **verifiability**, the *Vote* protocol is considered verifiable if for any adversary A:

$$Succ^{ver} = \left[Exp^{ver}_{A,Vote}(\lambda) = 1 \right]$$

is negligible in λ.

Following past Section 4.1 lattice-based properties applied to e-voting, LWE samples satisfy a straightforward linear **homomorphism property**, deriving from continuous Gaussian convolution [59]:

Let $c_1, \ldots, c_p$ be p independent LWE samples of messages $\mu_1, \ldots, \mu_p \in \mathbb{T}$; $\alpha_1, \ldots, \alpha_p$ noise paramenters, and $x_1, \ldots, x_p \in \mathbb{Z}$ be p integer coefficients. It holds that the sample $c = \sum_{i=1}^{p} x_i c_i$ is a valid encryption of the message $\mu = \sum_{i=1}^{p} x_i \mu_i$ with square noise parameter $\alpha^2 \leq \sum_{i=1}^{p} x_i^2 \alpha_i^2$.

For non-linear operations, the Bootstrapping theorem by Ducas et al. [59] can be implemented:

$$Bootstrap_{BK}(c, \mu_1', \mu_0') = \begin{cases} LWESymEncrypt_{s',\alpha,G'}(\mu_1') \textbf{ if } d(\varphi_s(c), \frac{1}{2}) < d(\varphi_s(c), 0) \\ LWESymEncrypt_{s',\alpha,G'}(\mu_0') \qquad\qquad \textbf{otherwise}. \end{cases}$$

Ducas et al. imply that the output of the aforementioned function is indistinguishable from a fresh LWE sample of μ_0' or μ_1'. In the event that it is preferred to control de randomness for verifiability purposes, in the previous theorem, the Bootstrap function can be assimilated into a perfect random oracle which returns a fresh LWE sample of μ_b' where $b = 1$ if $d(\varphi_s(c), \frac{1}{2}) < d(\varphi_s(c), 0)$.

With respect to the second ZKP implemented in the original version of Helios for the **distributed decryption**, in LWE-Helios it is replaced by a **publicly verifiable ciphertext trapdoor** inspired by the GPV Lattice-based signature [62]:

Let $LWE(s, \alpha, G)$ be a LWE instance, $pk = [M \mid y] \in (G \times \mathbb{T})^m$ a public key and M a discrete message space of packing radius $\geq d$. Let also $c = (a, b)$ be a sample with noise amplitude $\leq \delta$ and $\beta = \sqrt{(d^2 - \delta^2)/\bar{\alpha}^2}$; then $x = (x_1, \ldots, x_m) \in \mathbb{Z}^m$ is a ciphertext trapdoor of c **if** $\|x\| \leq \beta$ and $x \cdot M = a$ in G.

The **distributed** property is introduced with concatenated LWEs rather than using the Fiat-Shamir Heuristic [21], such being the case for the original Helios. The reason is that the authors prefer to propose a trustable scheme, even if all but one trustees are corrupt and leak their private key.

The definition is as follows:

Let $LWE(s_i, \alpha, G)$, where $1 \leq i \leq t$, be λ-bit secure instances of LWE. Let $pk_i = [M_i \mid y_i] \in (G \times \mathbb{T})^m$ be the corresponding public keys with associated trapdoors R_i. A **concatenated LWE** is the LWE instance whose private key is $s = (s_1 \mid \ldots \mid s_t)$, discretization group is $G = G_1 \times \ldots \times G_t$ and public key is:

$$pk = \begin{bmatrix} M_1 & 0 & 0 & y_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & M_t & y_t \end{bmatrix}.$$

To decrypt the LWE ciphertext $c = (a_1 \mid \ldots \mid a_t, b) \in G \times \mathbb{T}$, each of the t trustees uses his master trapdoor R_i to provide a ciphertext trapdoor Π_i of $(a_i, 0)$. The concatenated ciphertext trapdoor $\Pi = (\Pi_1 \mid \ldots \mid \Pi_t)$ is a ciphertext trapdoor for c. As explained in [18], even in the case of collusion (see [2] for the definition) of all trustees but one, the decryption is still λ-bit secure.

4.2.2 LWE-based Helios E-voting protocol

As it has been detailed in the present section 4, there are very relevant differences in the cryptographic approach between the traditional Helios Voting protocol and LWE Helios.

It is not the case for the voting protocol itself because the main phases and functions are the same in both cases. Although a full explanation of the protocol can be found in [18], the main phases are as follows:

- **Setup**: The Bulletin Board Manager generates a pair of keys $(pk_{BB}, sk_{BB}) = KeyGenE_{BB}(1^\lambda)$ and publishes pk_{BB}. The trustees setup the concatenated LWE scheme and each one generates its own LWE secret key $s_i \in \mathbb{B}^n$, master trapdoor R_i and a corresponding public key $pk_i \in (G \times \mathbb{T})^m$. They also provide three bootstrapping keys $BK_1 := BK_{\left[(s^{(f)}, \frac{1}{4}) \to (s^{(m)}, \frac{1}{4})\right]}$, $BK_2 := BK_{\left[(s^{(m)}, \frac{1}{4}) \to (s^{(m)}, \frac{1}{16})\right]}$ and a larger one for low noise amplitude $BK_3 := BK_{\left[(s^{(m)}, \frac{1}{4}) \to (s, \frac{1}{L^{3/2}})\right]}$. Bootstrapping can be performed with BK_1 and BK_2 in less than 700ms [59]. For BK_3 it is expected a slowdown by a constant factor of ≈ 16.
- **Voter Registration**: $Register(1^\lambda, id)$. The authority runs $(upk_{id}, usk_{id}) \leftarrow KeyGenS(1^\lambda)$, adds upk_{id} in L_U and outputs (upk_{id}, usk_{id}).
- **Voting Phase**: $Vote(pk, usk, upk, v)$: Every voter computes the binary decomposition $(v_0, \ldots, v_{k-1}) \in \{0,1\}^k$, such that $v = \sum_{j=0}^{k-1} v_j 2^j$. Each bit is encrypted as $c_j = LWEPubEncrypt_{pk^{(f)}}(\hat{v}_j)$. c'_j is bootstrapped as $c'_j = Bootstrap_{BK_1}(c_j, \frac{1}{2}, 0)$ and returns the final ballot $b = (content, \sigma)$, where $content = (aux, upk, (c'_0, \ldots, c'_{k-1}), num)$ and $\sigma = Sign(usk, content)$.
- **Ballot Processing**: $ProcessBB(BB, b, sk_{BB})$: Upon receiveing a ballot b, it is analized as $(content, \sigma)$, where $content = (aux, upk, (c'_0, \ldots, c'_{k-1}), num)$. The Bulletin Board verifies that $upk \in L_U$ and checks whether $VerifyS(upk_{id}, content)$. Unlike the classical Helios, no ZKP is needed, since all binary messages are valid choices.

  Subsequently, BB applies a sequence of public homomorphic operations which do not require the presence of the voter (see [18] for the full explanation):

  1. Pre-Bootstrapping:

  $$Bootstrap_{BK_2}(c'_j, \frac{1}{4}, 0)$$

  is applied to each c'_j.

  2. Homomorphic Binary Expansion:

  $$\begin{aligned} HomAND(c_1, c_2) &= Bootstrap_{BK_2}((0, -\tfrac{1}{8}) + c_1 + c_2, \tfrac{1}{4}, 0) \\ HomANDNot(c_1, c_2) &= Bootstrap_{BK_2}((0, \tfrac{1}{8}) + c_1 - c_2, \tfrac{1}{4}, 0) \end{aligned}.$$

  3. Generalized Bootstrapping:

  $$Bootstrap_{BK_2}((0, -\frac{1}{8}) + c_1 + c_2, \frac{1}{4}, 0).$$

4. Homomorphic Addition: BB homomorphically adds all ciphertexts without bootstrapping. It yields the final LWE ciphertexts $(C_0, \ldots, C_{l-1})$ of $(\frac{n_0}{L}, \ldots, \frac{n_{l-1}}{L})$ with noise $\Omega(L^{-1})$.

– **Tally and Verification**:
  1. $Tally(BB, sk = (sk_1, \ldots, sk_t))$: for each C_j, the trustees perform the distributed encryption as defined earlier in the present section and subsequently publish a ciphertext trapdoor $\Pi_{i,j} \in \mathbb{Z}^m$, which is revealed to anyone.
  2. $VerifyTally(BB, (\Pi_1, \ldots, \Pi_t))$. If a trapdoor $\Pi_{i,j}$ is invalid, it proves that the i-th trustee is not honest and thus $VerifyTally$ returns $\perp$. If all the trapdoors are valid, anyone can use $(\Pi_{1,j}, \ldots, \Pi_{t,j})$ to decrypt C_j and recover n_j for all j, therefore revealing the number of votes to candidate j . $VerifyTally$ then returns the result $(n_o, \ldots, n_{l-1})$.

Lastly, regarding **Correctness, Verifiability and Privacy**, and based on the LWE cryptographical concepts and E-voting protocols introduced in the present section, the authors in [18] affirm that:

– **Correctness**: assuming that: the signature scheme S is unforgeable, ε is a non-malleable encryption scheme, and the public homomorphic operations performed by the Bulletin Board are correct, LWE-based Helios is correct.
– **Verifiability**: Assuming that the Bulletin Board accepts only valid votes and that all operations performed by BB are public, LWE-based Helios is verifiable according to the following definition introduced in section 4.2.1:

$$Succ^{ver} = \left[Exp^{ver}_{A,Vote}(\lambda) = 1\right]$$

is negligible in λ.
– **Privacy**: assuming that the publicly verifiable decryption for LWE holds (see section 4.2.1), LWE-based Helios fulfills privacy according to following definition, introduced in the same section 4.2.1 [18]:

$$\left|Pr\left[Exp^{bpriv,\beta}_{A,Vote}(\lambda) = \beta\right] - \frac{1}{2}\right|.$$

4.2.3 LWE-based Helios conclusion

Chillotti et al. have introduced in [18] a proposal of a post-quantum LWE and Helios Voting-based e-voting protocol. It is the most complete exercise of this type to date, and they have achieved remarkable milestones:

– A successful aplication of post quantum primitives to an e-voting protocol: unforgeable lattice-based signatures, LWE-based homomorphic encryption and trapdoors for lattices.
– Achieving verifiability without ZKP. The authors implemented FHE [59] and trapdoor-based lattice signatures [62] instead.
– The protocol is correct, verifiable and fulfills privacy, according to [28,32].

It is necessary to point out that, while conforming an excellent starting point, LWE-based Helios is (as the authors themselves acknowledge), a theoretical protocol, still far from being a ready-to-use e-voting system. Additionally, the current version leaves open problems, namely:

– The proper definition and adaptation of strong correctness, strong consistency and privacy models against a malicious Bulletin Board and or/corrupted registration authority. In short, the **authors assume** that the **Bulletin Board is not dishonest or corrupt**. Depending of the typology of elections and the country, such assumption could be considered risky.
– The proof of privacy relies on a rather strong assumption: a properly **randomized bootstrapping function** is modelled as a perfect random oracle. It is necessary to successfully simulate the tally. The equivalent problem in the **standard model** is **still an open problem**.

4.3 Road to fully Homomorphic Elections in Norway

The present subsection is based on the preliminary work by Gjosteen and Strand in [63].Compared to LWE-based Helios, the current status of the Norwegian approach is clearly at an earlier stage, with no deep cryptographic research made thus far. Nonetheless, it certainly constitutes an interesting first step and shows that the Norwegian researchers are already starting to work in a potential post-quantum e-voting system.

Norway implemented an e-voting system created by Scytl [44] for the local elections in 2011 and the parliamentary elections in 2013. Altogether, almost 100.000 votes were cast with the REV tool. Despite there were no attacks reported, certain technical issues, together with a change in the ruling coalition to a less "e-voting friendly" one in 2014, contributed to a cancellation of the e-voting initiative that year [38].

Taking into account the experiences learned during the e-voting pilot deploymet, the authors tried to address the following Norwegian e-voting issues in [63]:

- An insufficient degree of verifiability.
- The cryptographic schemes were not quantum-safe.
- Voter verifiability is not considered an issue by the Norwegian electorate [64].
- Norwegian ballots are very complex, because the voter can give list votes, person votes or even write in a certain number of candidates. This complexity has important implications as to what cryptographic primitive is more suitable [38].

The main **contribution** of the article is the introduction of a possible FHE application to the Norwegian electoral system.

Regarding **security**, the authors informally introduce the following requirements:

- **D-privacy**: The decryption service should not be able to correlate its input to voter identities.
- **B-privacy**: The ballot box should not learn anything from the ciphertexts.
- **R-privacy**: The receipt generator should not be able to correlate return codes to the voter's vote.
- **A-privacy**: The auditor should not learn anything about how anyone votes.
- **B-integrity**: The ballot box must not be able to create an encrypted ballot such that its decryption is inconsistent with the related information that is sent to the receipt generator.
- **D-integrity**: The decryption service must not be able to alter the election outcome.

Regarding the **cryptographic primitives**, there is no in-depth research and the authors simply suggest equality checking with [65], the Smart-Vercauteren scheme for sorting and division by rational numbers [66], and BGV [67] as the FHE cryptosystem. They also state that some of the potentially required primitives have not yet been described.

Regarding **voting instantiation**, BGV encrypts tuples with the following format:

$$(p_1, \ldots, p_m, p'_1, \ldots, p'_m, p''_1, \ldots, p''_m, \ldots, c_1 \ldots c_n)$$

with the tuple requiring $3m + n$ slots. Assuming that the voter's ciphertext encodes the vector

$$b = (p, s_1, \ldots, s_{n_p}, e_1, \ldots, e_{n'})$$

where p is the index of the chosen party list, s_i is a bit indicating whether candidate i receives a person vote and $e_1, \ldots, e_{n'}$ are the indices of the write-in representatives.

In order to adapt the ballot to the format accepted by BGV, the authors define a function Eq which returns 1 whenever the two input values are equal. They also define $ln(a, S) = \sum_{s \in S} Eq(a, s)$, which will return 1 if a is a member of the set S.

Let $\{P_i\}$ be all parties in the election and P''_i the number of list votes tranferred to party P_i. Let also P_i denote the set of indexes for the candidates on the party list of party i. Then, P''_i (the number of list votes transferred) is computed as:

$$p''_i \leftarrow (ln(e_1, P_i) + ln(e_2, P_i) + \ldots + ln(e_{n'}, P_i))$$

Subsequently, in order to compute person's votes to candidates from other lists, let P_j be the party that candidate c_j belongs to, with p being the party selected by the voter:

$$c_j \leftarrow (1 - Eq(p, P_j))(Eq(j, e_1) + Eq(j, e_2) + \ldots + Eq(j, e_{n'})).$$

Finally, to verify that the vote is valid, the following polynomial is computed:

$$\prod_{i \leq j} (e_i - e_j).$$

The other interesting contribution included in [63] is the **parameter election exercise** made by the authors: They consider Oslo, with 500.000 eligible voters, 17 party lists with a total of 659 candidates. The city council consists of 59 members and each voter can list at most 15 names from other parties to her ballot. Therefore, the biggest number to handle is $7500000 \approx 2^{23}$ so equality checks will need a depth-23 circuit.

The authors conclude that no part of the computation requires a depth > 50. The number of slots needed, as previously explained, is $3m + n$; $3 \cdot 17 + 659 = 710$. Using the test program HElib [68] on a server running Ubuntu 14.04 on Intel Xeon 2.67 GHz, processors with a total of 24 cores and 256 GB of memory. The key generation ran on a single core, 8 cores were used for some ciphertexts operations. The maximum memory usage was 20GB. The whole process took **4:52 minutes**, with the key generation consuming half of it. With that amount of time, it is not feasible for a single voter but it might be acceptable for an election.

To sum up, Gjosteen and Strand have proposed a first initial post-quantum cryptographic approach to e-voting in Norway. Based on the experience accrued during the 2011-2013 e-voting pilots and adapting FHE schemes [65-67] to the Norwegian idiosyncrasy, the authors have introduced an initial approach which would take less than 5 minutes to be launched with affordable computational capacity.

The article constitutes a very remarkable initiative that will be remembered as a pioneering effort in post-quantum cryptography applied to e-voting.

Regarding the open issues, the following aspects should be addressed in order to further improve the scheme towards a functional tool:

- The approach is well adapted to the Norwegian case but the **cryptographic and methodologic aspects need further refinement**.
- **Implementation** of the proposed **algorithms**.
- Define and **develop in detail** the **schemes** and the **cryptographic integration** of the primitives.
- The **coercion resistance** should **not** be addressed by **simply allowing re-voting in paper** ballots.

5 Conclusions

In recent years, the implementation of e-voting systems to electoral processes has been struggling to find continuity. There are still active projects in relevant countries such as Estonia or some Swiss cantons, but many others have decided to discontinue its implementation (Norway, UK, Germany, USA, the Netherlands), or just decided not to start with the foreseen deployments (New Zealand).

The reasons are varied and include:

- E-voting demands simultaneously verifiability and privacy [2].
- Different electoral laws depending on the country/territory.
- Flaws in the cryptographic schemes and the e-voting tools [1,3,8,13].
- Geopolitical tensions and cyber-attacks (including DDoS episodes) [8,9,11].

Regarding the main cryptographic schemes, most of the fully-functioning e-voting tools rely on either mix-nets, homomorphic encryption or a hybrid system. Currently, none of them can assure coercion resistance, even though it should be the required level of privacy [34]. Regarding verifiability, the assumptions still tend to be hard for collusion, even for the latest schemes [32]. Lastly, protection against DDoS attacks has improved [40], but mix-net based solutions are still vulnerable to those cyberattacks.

Overall, the "classic" cryptographic approaches seemed to be limited to face the upcoming security challenges, specially in the long term. Moreso since Shor proved in [51] that the traditional cryptographic schemes are vulnerable to quantum-computing attacks.

Fortunately, in the last two years there has been an increasing number of applications of post-quantum schemes to e-voting. In the present article, the authors have reviewed two of the most relevant:

- The lattice-based learning with errors (LWE) application to a Helios [22] based e-voting protocol [18].
- The initial steps of a post-quantum scheme for the Norwegian elections [63].

Although the aforementioned post-quantum e-voting schemes are still at an early stage (specially the Norwegian one), they certainly provide a promising alternative. In particular, the LWE-based scheme over Helios has achieved several remarkable milestones, namely:

- A successful aplication of unforgeable lattice-based signatures, LWE-based homomorphic encryption and trapdoors for lattices to a e-voting scheme.
- Achieving verifiability without ZKP. The authors implemented FHE [59] and trapdoor-based lattice signatures [62] instead.

On the other hand, the research still relies on strong assumptions, such as an honest Bulletin Board or a perfect random oracle.

There are still important challenges ahead, including the development of a fully-functioning post-quantum e-voting system, but the initial steps are already being taken, and they are very promising.

References

[1] Springall D., Finkenauer T., Durumeric Z., Kitcat J., Hursti H., MacAlpine M., et al. Security Analysis of the Estonian Internet Voting System, Proc 21st ACM Conf Comput Commun Secur., 2014, 703-715. doi:10.1145/2660267.2660315

[2] Foundation USV. The Future of Voting. In: The Future of Voting [Internet]. Available: https://www.usvotefoundation.org/e2e-viv/summary , 2015

[3] The FREAK Attack. In: The FREAK Attack [Internet]. Available: https://censys.io/blog/freak , 2015

[4] Adrian D., Bhargavan K., Durumeric Z., Gaudry P., Green M., Halderman JA., et al. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2015, 5-17 doi:10.1145/2810103.2813707

[5] Wang X., Yu H., How to Break MD5 and Other Hash Functions. Adv Cryptol – EUROCRYPT, 2005, 19-35 doi:10.1007/114266392

[6] Goldwasser S., Tauman Y., On the (In)security of the Fiat-Shamir Paradigm. Focs, 2003

[7] Achenbach D., Kempka C., Lowe B., Muller-Quade J., Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting. USENIX J Elect Technol Syst., 2015, 26-45

[8] Wolchok S., Wustrow E., Isabel D., Halderman JA., Attacking the Washington , D . C . Internet Voting System. System, International Conference on Financial Cryptography and Data Security, 2012, 114-128

[9] M C. Ukraine election narrowly avoided “wanton destruction” from hackers. In: Christian Science Monitor [Internet]. Available: http://www.csmonitor.com/World/Passcode/2014/0617/Ukraine-election-narrowly-avoided-wanton-destruction-from-hackers-video , 2014

[10] Halderman JA., Teague V. The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In: Haenni R., Koenig RE., Wikström D., editors. E-Voting and Identity: 5th International Conference, VoteID 2015, Bern, Switzerland, September 2-4, 2015 35-53. doi:10.1007/978-3-319-22270-73

[11] Nakashima E., Arizona: Russian hackers targeted Arizona election system. The Washington Post., 2016

[12] Background to Assessing Russian Activities and Intentions in Recent US Elections: The Analytic Process and Cyber Incident Attribution, 2017

[13] Bernhard M., Election Recount Hacking Voting Machines. The Guardian, 2016

[14] Neumann SR., Ph.D. Thesis Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements, Technische Universität Darmstadt, Germany, 2016

[15] Marcos del Blanco DY., Panizo Alonso L., and Hermida Alonso JA., The need for Harmonization in the online voting field: Towards an European Standard for edemocracy. The International Conference on Electronic Voting, E-Vote-ID 2016, October 18-21, Bregenz, Austria, 2016, 339-343

[16] Directorate General of Democracy and Political Affairs. Certification of e-voting systems. Guidelines for developing processes that confirm compliance with prescribed requirements and standards. Council of Europe, 2011
[17] Gentry C., Fully homomorphic encryption using ideal lattices., In: Proceedings of the forty-first annual ACM symposium on Theory of computing May 31 - June 02 2009, Bethesda, MD, USA, 2009, 169-178
[18] Chillotti I., Gama N., Georgieva M., Izabachene M. An Homomorphic LWE based E-Voting Scheme In: 7th International Workshop, PQCrypto 2016, February 24-26, 2016, Fukuoka, Japan, 2016, 245-265
[19] Panizo L., Ph.D. Thesis Desarrollo de una metodología para el análisis y la clasificación de los sistemas de voto electrónico , University of Leon, Leon Spain, 2014 (in Spanish)
[20] Ronquillo L., Securing e-voting systems, lecture. 12 May 2015
[21] Fiat A., Shamir A., How to prove yourself: Practical solutions to identification and signature schemes, In: Advances in Cryptology Crypto'86 Springer-Verlag, 1986, 186-194
[22] Adida B., Helios: Web-based Open-audit Voting, In: Proceedings of the 17th Conference on Security Symposium, July 28 - August 1 2008, San Jose, CA, USA, 2008, 335-348
[23] Bernhard D., Pereira O., and Warinschi B. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios, In: Advances in Cryptology ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, December 2-6, Beijing, China, 2012, 626-643
[24] Kusters R., Truderung T., Vogt A., Clash Attacks on the Verifiability of E-Voting Systems, In: Proceedings of the 2012 IEEE Symposium on Security and Privacy, May 20-23 2012, San Francisco, CA, USA, 2012, 395-409
[25] ElGamal T., Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, In: G R Blakley and D Chaum (Eds.) Advances in Cryptology: Proceedings of CRYPTO 84, Berlin, 1985 469-472
[26] Sanou E., MSc Thesis Post-Quantum Cryptography: Lattice-based, Universitat Politecnica de Catalunya, Barcelona Spain, 2016
[27] Benaloh JDC., Rivest R., Ryan PYA., Stark P., Teague V., and Vora P., End-to-end verifiability arXiv e-prints, 2014
[28] Juels A., Catalano D. and Jakobsson M. Coercion-resistant electronic elections, In: Lecture Notes in Computer Science vol. LNCS 6000, 2010 37-63
[29] Blanchet B., An Automatic Security Protocol Verifier based on Resolution Theorem Proving (invited tutorial) In: Nieuwenhuis R. (Ed.) 20th International Conference on Automated Deduction, July 22-27, Tallinn, Estonia, 2005, 3-51
[30] Chadha R., Cheval V., Ciobaca S., and Kremer S. Automated Verification of Equivalence Properties of Cryptographic Protocols, 2012
[31] Cheval V. APTE: An Algorithm for Proving Trace Equivalence In: Abraham E. and Havelund K. (Eds.) Tools and Algorithms for the Construction and Analysis of Systems: 20th International Conference, TACAS 2014, part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, April 5-13, Grenoble, France, 2014 587-592
[32] Cortier V., Formal Verification of e-Voting: Solutions and Challenges, ACM SIGLOG News, vol. 2, 1, 2015, 25-34
[33] Panizo L., Gasco M., Marcos del Blanco DY., Hermida JA. and Alaiz H. E-voting system evaluation based on the Council of Europe recommendations: Helios Voting IEEE Transactions on emerging topics in computing. Special issue on e-government development and applications (SIEGDA) (expected 2018), under review.
[34] Hirt M., Sako K. Efficient Receipt-Free voting based on homomorphic encryption In: Preneel B., editor, EUROCRYPT'00, vol. 1807 LNCS Bruges, Belgium, 2000, 539-556
[35] Achenbach D., Kempka C., Lowe B., Muller-Quade J. Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting In: JETS, The Usenix Journal of Election Tech. and Systems, 12-14 August, Washington, USA, 2015, 26-45
[36] Smith R. Internet Voting and Voting Interference. A report for the New South Wales Electoral Commission https://www.elections.nsw.gov.au/_data/assets/pdf_file/0003/118380/NSWEC_2013_Report_V2.0.pdf ,2013
[37] Heiberg S., Parsovs A., Willemson J. Log Analysis of Estonian Internet Voting 2013 - 2015. Smartmatic – Cybernetica Centre of Excellence for Internet Voting, Software Technology and Applications Competence Centre, Tartu University, 2015
[38] Nore H., Implementing E-Voting in Norwegian Elections New Voting Technology Consulting AS, 2015
[39] Chung D., Bishop M., Peisert S. Distributed Helios – Mitigating Denial of Service Attacks in Online Voting University of California Davis, 2016
[40] Chandra TD., Griesemer R., Redstone J., Paxos made live: An engineering perspective. In: 26th Annual ACM Symposium on Principles of Distributed Computing, August 12-15, Portland, OR, USA, 2007, 398-407
[41] Chaum D. Untraceable electronic mail, return addresses and digital pseudonyms. ACM, 24(2), 1981, 84-90
[42] Fujioka A., Okamoto T., Ohta K. A practical secret voting scheme for large scale elections. ASIACRYPT'92, Workshop on the Theory and Application of Cryptographic Techniques, LNCS 718, Gold Coast, Australia, 1992, 244-251
[43] Haber S., Benaloh J. and Halevi S. The Helios e-Voting Demo for the IACR, 2010,
[44] Scytl, R&D Department. Articles and Publications. https://www.scytl.com/en/articles-and-publications/ , 2017
[45] Perriard B. Vote electronique: the long path towards the digitalization of political rights Swiss Federal Chancellery, 2015
[46] Scytl Report French Ministry of Foreign Affairs. French Expats vote online in 2012 legisla-tive elections Available at: https://www.parliament.uk/documents/speaker/digital-democracy/FR_Successcase.pdf , 2012
[47] Rivest R., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21, 1978, 120-126

[48] Paillier P. Public-Key Cryptosystems based on Composite-Degree Residuosity Classes. In: J Stern (Ed.) EUROCRYPT'99 May 2-6, Prague, Czech Republic, 1999, 223-238
[49] Koenig RE., Locher P, Haenni R. A Security Flaw in the Verification Code Mechanism of the Norwegian Internet Voting System Bern University of Applied Sciences, 2013
[50] Springall D., Finkenauer T., Durumeric Z., Kitcat J., Hursti H., MacAlpine M, Halderman JA. Security Analysis of the Estonian Internet Voting System. In: ACM CCS November 3-7 Scottsdale, Arizona, USA, 2014, 703-715
[51] Shor P. Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science November 20-22 Santa Fe, New Mexico, USA, 1994, 124-134
[52] RSA Laboratories, EMC Corporation. "What is a Blind Signature Scheme, 2016
[53] Witteman M., van Woudenberg J., Menarini F. Defeating RSA multiply-always and message blinding countermeasures Riscure BV. The Netherlands, 2007
[54] Ibrahim S., Kamat M, Salleh M., Aziz SRA. Secure E-voting with Blind Signature. In: NCTT 2003 Proceedings, 4th National Conference on Telecommunication Technology, January 14-15 Shah Alam, Malaysia, 2003, 193-197
[55] Rivest R., Adleman L., and Dertouzos M. On data banks and privacy homomorphisms. Foundations of Secure Computation, Academia Press, 1978, 169-180
[56] Gentry C., Halevi S., and Smart NP. Homomorphic Evaluation of the AES Circuit. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, August 19-23, Santa Barbara, CA, USA, 2012, 850-867
[57] Regev O. On lattices, learning with errors, random linear codes, and cryptography. In: H N Gabow and R Fagin, (Eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, May 21-24, Maryland, USA,2005, 84-93
[58] Lyubashevsky V., Peikert C., and Regev O. On ideal lattices and learning with errors over rings. In: Proc. of EUROCRYPT, vol. 6110 of LNCS May 30 - June 3 Monaco and Nice, Monaco, France,2010, 1-23
[59] Ducas L. and Micciancio D. FHEW: Bootstrapping homomorphic encryption in less than a second. In: Eurocrypt 2015, April 26-30 Sofia, Bulgaria, 2015, 617-640
[60] Chen Y. and Nguyen PQ. BKZ 2.0: Better lattice security estimates. In: Wang X. and Lee DH. (Eds.) Asiacrypt 2011, Seoul (Korea,2011) 1-20
[61] Gama N., Izabachene M., Nguyen PQ., and Xie X. Structural lattice reduction: Generalized worst-case to average-case reductions. In: EUROCRYPT 2016, 2016, 528-558
[62] Micciancio D. and Peikert C. Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval D. and Johansson T. (Eds.) Eurocrypt 2012, April 15-19, Cambridge, UK, 2012, 700-718
[63] Gjosteen K., Strand M. A roadmap to fully homomorphic elections: Stronger security, better verifiability. IACR Cryptology ePrint Archive, 2017, 404-418
[64] OSCE Office for Democratic Institutions and Human Rights. Norway, Parliamentary Elections 9 September 2013, Final Report, 2013
[65] Kim M., Lee HT., Ling S., and Wang H. On the efficiency of FHE-based private queries. IEEE Transactions on Dependable and Secure Computing, 2016
[66] Smart NP. and Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Nguyen PQ. and Pointcheval D. (Eds.) Public Key Cryptography – PKC 2010, vol. 6056 LNCS, 2010, 420-443
[67] Brakerski Z., Gentry C., and Vaikuntanathan V. Fully homomorphic encryption without bootstrapping. Electronic Colloquium on Computational Complexity (ECCC), 2011
[68] Halevi S. and Shoup V. Bootstrapping for HElib. In: Oswald E. and Fischlin M. (Eds.), Advances in Cryptology – EUROCRYPT 2015, vol 9056 of LNCS, 2015, 641-670