Non-interactive Zero-Knowledge Arguments for Voting

Jens Groth*

Dept. of Computer Science, UCLA, USA
jg@cs.ucla.edu

Abstract. In voting based on homomorphic threshold encryption, the voter encrypts his vote and sends it in to the authorities that tally the votes. If voters can send in arbitrary plaintexts then they can cheat. It is therefore important that they attach an argument of knowledge of the plaintext being a correctly formed vote. Typically, these arguments are honest verifier zero-knowledge arguments that are made non-interactive using the Fiat-Shamir heuristic. Security is argued in the random oracle model.

The simplest case is where each voter has a single vote to cast. Practical solutions have already been suggested for the single vote case. However, as we shall see homomorphic threshold encryption can be used for a variety of elections, in particular there are many cases where voters can cast multiple votes at once. In these cases, it remains important to bring down the cost of the NIZK argument.

We improve on state of the art in the case of limited votes, where each voter can vote a small number of times. We also improve on the state of the art in shareholder elections, where each voter may have a large number of votes to spend. Moreover, we improve on the state of the art in Borda voting. Finally, we suggest a NIZK argument for correctness of an approval vote. To the best of our knowledge, approval voting has not been considered before in the cryptographic literature.

1 Introduction

Voting based on homomorphic encryption. A popular paradigm for constructing e-voting protocols is based on homomorphic threshold encryption. The homomorphic property is $E(m_1 + m_2; r_1 + r_2) = E(m_1; r_1)E(m_2; r_2)$. The authorities publish a public key and voters send in encrypted votes. Digital signatures or other means of authentication ensure that only eligible voters vote.

As an example consider an election where voters encode yes-votes as 1 and no-votes as 0. Holding encrypted votes $E(v_1), \ldots, E(v_m)$ the authorities can use the homomorphic property of the cryptosystem to compute $E(\sum_{i=1}^{m} v_i)$. They jointly decrypt this ciphertext to get out the number of yes-votes, $\sum_{i=1}^{m} v_i$. It is important that they have to cooperate to decrypt, if any single authority held the decryption key then the voters' privacy might be at risk.

More advanced encoding methods allow elections where voters have a wide range of options. In the paper, we treat the following possibilities:

* Part of the work done while at Cryptomathic, Denmark and BRICS, Dept. of Computer Science, University of Aarhus, Denmark.

J. Ioannidis, A. Keromytis, and M.Yung (Eds.): ACNS 2005, LNCS 3531, pp. 467–482, 2005.

- Limited vote: N out of L candidates.
- Approval vote: Any number out of L candidates.
- Divisible vote: A huge number of votes distributed among the candidates.
- Borda vote: A preference vote where the best candidate receives L votes, the second best $L - 1$ votes, etc.

The advantage of voting based on homomorphic encryption is that it combines efficiency with a reasonable amount of flexibility. In particular, in comparison with other voting paradigms such as mix-nets, it seems like a superior choice for divisible votes that occur quite frequently in shareholder elections.

Zero-knowledge arguments. We have to ensure that voters do not cheat. Consider for instance in the previous example a voter that sends in $E(-100)$. Effectively this voter is taking 100 yes-votes out of the ballot box. To avoid such attacks we let each voter submit a zero-knowledge argument of correctness of his vote.

In practice, we want to minimize interaction between voters and authorities when casting votes. The common approach is therefore to find an efficient honest verifier zero-knowledge argument for correctness of the vote and make it non-interactive using the Fiat-Shamir heuristic. This yields efficient non-interactive zero-knowledge (NIZK) arguments. Security is proved in the random oracle model[1].

Related work. The idea of using homomorphic encryption to construct voting protocols was suggested by Cohen and Fischer [CF85] and further developed in [BY86, Ben87]. Cramer, Gennaro and Schoenmakers [CGS97] suggested a reasonable efficient yes/no-voting scheme based on ElGamal encryption. Unfortunately, these schemes cannot handle large elections with many candidates.

Concurrently Baudron et al. [BFP$^+$01] and Damgård and Jurik [DJ01][2] suggest voting schemes based on Paillier encryption [Pai99]. Their zero-knowledge arguments involve many encryptions and are therefore close to practical but still a little expensive.

Lipmaa, Asokan and Niemi [LAN02] propose the first practical zero-knowledge argument based on homomorphic integer commitments. Using integer commitments means that they can take advantage of integer properties such as unique prime factorization and get a practical zero-knowledge argument. Damgård, Groth and Salomonsen [DGS03] improve on this scheme and also propose a zero-knowledge argument for a limited vote.

Ishida, Matsuo and Ogata [IMO03] consider the case of shareholder elections and suggest a zero-knowledge argument for correctness of a divisible vote.

Wang and Leung [WfL04] investigate the case of Borda voting. They wish to construct a protocol that only reveals the winner, but not how many votes each candidate got. At a considerable efficiency cost, they proceed to construct such a multi-party computation protocol. Unlike them, we do not try to hide the number of votes candidates receive. Because of this difference, they are satisfied with letting each voter send a ciphertext for each candidate containing the number of votes on that candidate. Nonetheless, while not the focus of their paper they do need a NIZK argument for correctness

[1] See Section 2 for more details.

[2] Damgård, Jurik and Nielsen [DJN03] correct some flaws in this voting scheme.

of a Borda vote. They give a sketch of a NIZK argument for correctness of a Borda vote, however, it turns out the NIZK argument is not sound as it stands [Wan05]. The NIZK argument for correctness of a Borda vote we suggest in the paper can be adapted to their setting and solve their problem in a simple way.

We do not know of any work addressing approval voting in connection with homomorphic threshold encryption based voting schemes.

Our contributions. We observe that approval voting and Borda voting can be implemented efficiently using homomorphic threshold voting and offer corresponding NIZK arguments. We improve the NIZK argument for a limited vote of [DGS03] by simplifying the protocol. We suggest a NIZK argument for a divisible vote that is a factor $\log N$ more efficient, where N is the number of votes the shareholder can cast.

Table 1. Comparison of voting arguments.

Vote	Argument	Verification	Prior art	Argument	Verification
Limited	1 $6N+4$	1 $3N+3$	[DGS03]	1 $8N+2$	1 $7N+2$
Approval	1 $2L+4$	1 $L+3$	No prior work		
Divisible	1 $10L+4$	1 $5L+2$	[IMO03]	$(5/2)L\log N$	$2L\log N$
Borda known shuffle [Gro03]	1 $4L+2$	1 $2L+3$	[WfL04]	Not sound	

For all arguments, the top line contains the number of encryptions, the bottom line the number of exponentiations to make commitments. For all verifications, the top line contains the number of encryptions and the number of exponentiations of ciphertexts (always identical numbers), the bottom line the number of exponentiations to verify the commitments.

In Table 1, we list computational complexities for each NIZK argument. Since a ciphertext containing a vote must remain secure also some time into the future, we often need a long security parameter for the cryptosystem. On the other hand, the NIZK arguments are usually verified by interested parties right after the election, and since they can be made statistical zero-knowledge we can use a much shorter security parameter for the commitment scheme. For the purpose of creating this table, we have assumed that to commit to n elements, one uses $n+1$ exponentiations. In general, the expensive operations are those that involve ciphertexts.

One should be careful when using this table. For instance, the approval vote argument uses short exponents, while the limited vote argument may use longer exponents. The commitment exponentiations may therefore be cheaper for the approval vote argument in a setting with a similar number of voters and candidates. In the case of limited voting one should note that our NIZK argument unlike the [DGS03] NIZK argument is well suited for the use of multi-exponentiation techniques, so our gain is larger than what is indicated by Table 1. Finally, the verification process for most protocols may be sped up using batch verification techniques when verifying many votes at the same time.

Efficient range proof. Proving that a committed number x lies in some interval $[a, b]$ is useful in many protocols. Typically, we do that by proving that both $x - a$ and $b - x$ are non-negative. We can use either Boudot's method [Bou02] or prove that the number can be written as the sum of four squares [Lip03]. The two methods have comparable efficiency. In Section 5 we suggest a little trick to speed up the latter argument. Namely, to prove that y is non-negative we prove that $4y + 1$ is the sum of three squares. We highlight the trick here, since it may have independent interest.

2 Preliminaries

2.1 Voting Based on Homomorphic Encryption

Election parameters M, L, N. Throughout the paper, we assume that we have a group of voters that can choose between L candidates, which may include choices such as a blank vote or an invalid vote. A drawback of this type of election scheme is that the number of candidates is fixed; we do not allow write-in votes. We denote by M a strict upper bound on the number of votes any candidate can receive. In particular, if each voter has one vote then M is a strict upper bound on the number of voters. As will become apparent later, there is much to gain by selecting $M = p^2$, where p is a prime. A third parameter characterizing the elections is the number of votes the voter can cast, denoted by N.

Encoding votes. In the introduction, we sketched how to base voting protocols on homomorphic encryption. Let us offer some more details. The basic ingredient is a homomorphic threshold public-key cryptosystem. We will generate a public key for this cryptosystem, and the secret key is threshold secret shared amongst the authorities.

We assume that the message space is on the form $\mathbb{Z}_n$. We require that n does not have prime factors smaller than 2^{ℓ_e}, where ℓ_e is the length of the output of a suitable hash-function, and that $M^L \leq n$. We represent candidates with numbers $0, \ldots, L - 1$ and encode a vote on candidate i as M^i [3] Summing many such encodings gives us an M-addic representation of the result, $\sum_{i=0}^{L-1} v_i M^i$, where v_i is the number of votes on candidate i.

Representing votes this way, it is straightforward to encrypt a vote on candidate i as $E(M^i)$. Having received many such encrypted votes we may by the homomorphic property of the cryptosystem multiply all the ciphertexts and get a new ciphertext $C = E(\sum_{i=0}^{L-1} v_i M^i)$. We threshold decrypt this ciphertext and now it is straightforward to extract the result from the plaintext.

We shall see in the following sections that in a somewhat similar way it is possible to encode limited votes, approval votes, divisible votes and Borda votes, and therefore such types of elections can also be handled using this approach.

As mentioned in the introduction we need NIZK arguments for correctness of votes to avoid cheating and tampering with the result. In these NIZK arguments, we make use of homomorphic integer commitments. In the security proof of these NIZK arguments, we make use of a property of the integer commitment scheme and of the homomorphic

[3] As an alternative Lipmaa [Lip03] has suggested to encode votes as Lucas numbers.

cryptosystem known as root extraction. We also make use of the random oracle model. We will explain these concepts in the following.

2.2 Setup and Parameters

Throughout the paper, we make use of a semantically secure homomorphic threshold cryptosystem. We assume that the message space is $\mathbb{Z}_n$ for a suitable $n < M^L$ and the randomizer space is $\mathbb{Z}$. The latter assumption is purely out of notational convenience, there would be no problem in using a cryptosystem where the randomness is some finite group, for instance to use threshold Paillier encryption.

We also make use of a homomorphic integer commitment scheme. We always use randomizers from $\mathbb{Z}$. Again, there would be no problem to use other randomizer spaces but we do not yet know any such commitment scheme. The keys for both the cryptosystem and the commitment scheme are public and known to all parties.

We define the following parameters: $\ell_V = 2\lceil L(\log M)/2 \rceil$ is the maximal bit-length of a vote. We assume that the distribution of the randomizer space of the cryptosystem is to pick a random ℓ_R-bit randomizer. Similarly for integer commitments we pick a random ℓ_r-bit number as randomizer. Public keys are chosen with suitable security parameters. In large elections with many candidates, we may be forced to choose a large security parameter to accommodate this size of votes.

We need a couple of extra security parameters. We use a cryptographic hash-function that outputs an ℓ_e-bit number e. For instance, using SHA-256 we have $\ell_e = 256$. Furthermore, we need a security parameter ℓ_s, such that for any value a we have that $a + r_a$ and r_a are indistinguishable, where r_a is a random $|a| + \ell_s$-bit number. We suggest $\ell_s = 80$, this being large enough to ignore the off chance that $|a + r| > |a| + \ell_s$.

2.3 Homomorphic Integer Commitment and Homomorphic Cryptosystem

Integer commitment. We know only few homomorphic integer commitment schemes [FO97, DF02, Gro05], and they are all very similar in structure. As an example, we offer the following variant. We choose a modulus n as a product of two safe primes and random generators $g_1, \ldots, g_k, h$ of QR_n. To commit to integers $m_1, \ldots, m_k$ using randomness $r = (r_1, r_2) \in \{-1, 1\} \times \mathbb{Z}$ we compute $c = \mathrm{com}(m_1, \ldots, m_k; (r_1, r_2)) = r_1 g_1^{m_1} \cdots g_k^{m_k} h^{r_2} \bmod n$. To open the commitment we reveal $(m_1, \ldots, m_k, r)$. A typical choice is $r_1 = 1, r_2 \leftarrow \{0,1\}^{\ell_r}$, where $\ell_r = |n| + \ell_s$, which makes the commitment statistically hiding.

Root extraction property. When proving soundness and knowledge in our protocols we need the following root extraction property. If an adversary comes up with a commitment c, an opening $m_1, \ldots, m_k, r$ and $e \neq 0$, so $c^e = \mathrm{com}(m_1, \ldots, m_k; r)$, then we must have $e|m_1, \ldots, e|m_k$ and be able to compute an opening $\mu_1, \ldots, \mu_k, \rho$ so $c = \mathrm{com}(\mu_1, \ldots, \mu_k; \rho)$, where $\mu_i = m_i/e$.

Root extraction property of homomorphic cryptosystem. In the voting protocol, we use a semantically secure homomorphic threshold cryptosystem. Like the integer commitment scheme, it must have a root extraction property. If we create a ciphertext C

and $e \neq 0$ so $|e| < \ell_e$ and $C^e = E(M; R)$, then it must be possible to find μ, ρ so $M = e\mu, R = e\rho$ and $C = E(\mu; \rho)$.

ElGamal encryption [ElG84], Paillier encryption [Pai99] and several other homomorphic cryptosystems are semantically secure, have the root extraction property and admit threshold decryption.

2.4 NIZK Arguments and the Random Oracle Model

Consider a typical 3-move honest verifier zero-knowledge argument. The prover has some statement x that he wants to prove, and he knows a witness w. He sends an initial message a, receives a random challenge e and responds with an answer z. Given (x, a, e, z) the verifier can now choose whether to accept the argument or not.

Using Fiat-Shamir heuristic we let the prover compute the challenge e as a hash-function of x, a. I.e., the prover computes an argument (a, e, z), where $e = \text{hash}(x, a)$ [4]. This way we can make the argument non-interactive. Of course, the same methodology can be applied to arguments that use more than 3 moves.

As a heuristic argument of security of such protocols Bellare and Rogaway [BR93] suggest the random oracle model. The hash-function is modeled as a random function that pairs inputs (x, a) with a random output e. Furthermore, to argue zero-knowledge they allow the random oracle to be programmed. The simulator can choose inputs (x, a) and corresponding outputs e and the random oracle will on such an input return the corresponding output.

As a simple example, consider proving knowledge of the plaintext of a ciphertext C. We will present a well-known argument for this statement. Using the notation of [CS97] we write

$$\text{SPK}[(\mu, \rho) : C = E(\mu; \rho)].$$

We use Greek letters for the unknown variables we are proving something about and provide the statement that we are proving. This way we can quickly describe the goal of a NIZK argument without specifying the actual protocol. The following argument of plaintext knowledge is used as a subprotocol in most of our protocols.

Theorem 1. *In the random oracle model, the protocol in Figure 1 is a NIZK argument of plaintext knowledge.*

Proof. In the above argument, it is easy to see that we have completeness.

To argue zero-knowledge we pick e at random. We choose $\boxed{m} \leftarrow \{0, 1\}^{|m|+\ell_e+\ell_s}$, $\boxed{R} \leftarrow \{0, 1\}^{\ell_R+\ell_e+\ell_s}$. We set $C_R = E(\boxed{m}; \boxed{R})C^{-e}$. Finally, we program the random oracle to output e on input (C, C_R). We leave it to the reader to see that this is indeed a good simulation of an argument.

To argue knowledge we consider an adversary that has made a query (C, C_R) to the random oracle. If it is in a state where it has noticeable probability of using it in a valid argument, then we can upon seeing such an argument rewind it and feed it

[4] Sometimes some auxiliary information will be included in the hash-function. For instance, we might include the identity of the prover to avoid duplication of the proof. So we would write $e = \text{hash}(x, a, aux)$.

NIZK Argument for Plaintext Knowledge

Common input: Ciphertext C and public keys.
Prover's input: Message m and randomizer R so $C = E(m; R)$.

Argument: Choose $R_m \leftarrow \{0,1\}^{|m|+\ell_e+\ell_s}$ and $R_R \leftarrow \{0,1\}^{\ell_R+\ell_e+\ell_s}$. Set $C_R = E(R_m; R_R)$.
Compute the challenge $e = \mathrm{hash}(C, C_R)$.
Set $\boxed{m} = em + R_m, \boxed{R} = eR + R_R$.
The argument is $(C_R, \boxed{m}, \boxed{R})$.

Verification: Compute e as above. Verify $C^e C_R = E(\boxed{m}; \boxed{R})$.

Fig. 1. Plaintext Knowledge Argument.

with different random answers to the query. In expected polynomial time, we will get another acceptable argument. We now have two acceptable arguments $C_R, e, \boxed{m}, \boxed{R}$ and $C_R, e', \boxed{m}', \boxed{R}'$. With overwhelming probability, we have $e \neq e'$. From the verifying equations we have $C^e C_R = E(\boxed{m}; \boxed{R}$ and $C^{e'} C_R = E(\boxed{m}'; \boxed{R}')$. This means $C^{e-e'} = E(\boxed{m} - \boxed{m}'; \boxed{R} - \boxed{R}')$. From the root extraction property we can extract $\mu = (\boxed{m} - \boxed{m}')/(e - e')$ and ρ so $C = E(\mu; \rho)$. □

Remark 1. We routinely use the notation $\boxed{a} = ea + r_a$ throughout the paper. As a reminder one can think of it as putting a in a box that hides a. As we shall see, the random factor e allows us to make computations with the hidden variable a. For instance, if an equation $\boxed{a}\boxed{b} = e\boxed{c}$ holds with non-negligible probability over e, then the secret variables a, b, c satisfy $c = ab$ with overwhelming probability. The box-notation is intended to show on one hand that the variable is hidden, one the other hand indicate that we can perform standard algebraic operations on the hidden variables and under the hood the expected results come out. We hope this notation can serve as a helping guide in complex zero-knowledge arguments using many hidden variables.

3 Limited Vote

In some elections, voters can vote multiple times, say, N times. It may be a requirement that they use all their votes on different candidates, or alternatively they may be permitted to spend several votes on the same candidates. We will present a protocol for the former case; it is easy to modify the protocol into one that admits multiple votes on the same candidate.

The voter encodes his vote as $V = \sum_{j=1}^{N} M^{i_j}$, where $0 \leq i_1 < \cdots < i_N < L$. He then encrypts the vote and has to form a NIZK argument that the plaintext is on the right form. In other words, we wish to make the following argument of knowledge

$$\mathrm{SPK}[(\upsilon, \rho, \iota_1, \cdots, \iota_N) : C = E(\upsilon; \rho) \text{ and } \upsilon = \sum_{j=1}^{N} M^{\iota_j} \text{ and } 0 \leq \iota_1 < \cdots < \iota_N < L].$$

To make this argument of knowledge we actually use

$$\text{SPK}[(v,\rho,\alpha_1,\ldots,\alpha_N,\beta_1,\ldots,\beta_N):$$
$$C = E(v;\rho) \text{ and } v = \sum_{j=1}^{N}\alpha_j^2 \text{ and } \bigwedge_{j=1}^{N}\alpha_{j+1} = p\alpha_j\beta_j],$$

where p is a prime so $M = p^2$ and $\alpha_{N+1} = p^L$.

To see that the two arguments of knowledge are equivalent notice that $\bigwedge_{j=1}^{N}\alpha_{j+1} = p\alpha_j\beta_j$ implies $p\alpha_N | p^L, \ldots, p\alpha_1|\alpha_2$. I.e., we can write $\alpha_N = \pm p^{\iota_N}, \cdots, \alpha_1 = \pm p^{\iota_1}$, for some $0 \le \iota_1 < \cdots < \iota_N < L$. The second equation gives us

$$v = \sum_{j=1}^{N}\alpha_j^2 = \sum_{j=1}^{N}(\pm p^{\iota_j})^2 = \sum_{j=1}^{N} M^{\iota_j}.$$

The argument of knowledge is presented in Figure 2. In the protocol we argue knowledge of $\alpha_j, \rho_{a_j}, \beta_j, \rho_{b_j}, \Delta_j, \rho_{\Delta_j}$ so $\boxed{a_j} = e\alpha_j + \rho_{a_j}$, $\boxed{b_j} = e\beta_j + \rho_{b_j}$, $\boxed{\Delta_j} = e\Delta_j + \rho_{\Delta_j}$. We check that $\boxed{\Delta_j} = p\boxed{a_j}\boxed{b_j} - e\boxed{a_{j+1}}$, i.e.,

$$e\Delta_j + \rho_{\Delta_j} = e^2(p\alpha_j\beta_j - \alpha_{j+1}) + e(p\alpha_j\rho_{b_j} + p\beta_j\rho_{a_j} - \rho_{a_{j+1}}) + p\rho_{a_j}\rho_{b_j}.$$

The idea is that with overwhelming probability over e this equation can only hold if $p\alpha_j\beta_j - \alpha_{j+1} = 0$. Combine all these equalities to get $\bigwedge_{j=1}^{N}\alpha_{j+1} = p\alpha_j\beta_j$.

Included in the argument is an argument of plaintext knowledge of v, ρ_V so $\boxed{V} = ev + \rho_V$, as well as Δ, ρ_Δ so $\boxed{\Delta} = e\Delta + \rho_\Delta$. We check that $\boxed{\Delta} = \sum_{j=1}^{N}\boxed{a_j}^2 - e\boxed{V}$, giving us $e\Delta + \rho_\Delta = e^2(\sum_{j=1}^{N}\alpha_j^2 - v) + e(2\sum_{j=1}^{N}\alpha_j\rho_{a_j} - \rho_V) + \sum_{j=1}^{N}\rho_{a_j}^2$. With overwhelming probability over e this tells us that $v = \sum_{j=1}^{N}\alpha_j^2$. Finally, in the process we also argue knowledge of ρ so $C = E(v;\rho)$ in a similar way to the argument of plaintext knowledge in Section 2.4.

Theorem 2. *In the random oracle model, the protocol in Figure 2 is a NIZK argument of knowledge for C encrypting a correctly formed limited vote. If the commitment scheme is statistically hiding then the argument is statistical zero-knowledge.*

Proof. It is straightforward to verify that the protocol is complete. It remains to argue zero-knowledge and soundness and knowledge.

Zero-knowledge. To simulate an argument we pick a challenge $e \leftarrow \{0,1\}^{\ell_e}$ at random. Given the challenge e, we make a simulation like this. We pick $\boxed{V} \leftarrow \{0,1\}^{\ell_V+\ell_e+\ell_s}$ and $\boxed{R} \leftarrow \{0,1\}^{\ell_R+\ell_e+\ell_s}$. We pick $\boxed{a_1}, \boxed{b_1}, \ldots, \boxed{a_N}, \boxed{b_N} \leftarrow \{0,1\}^{\ell_V/2+\ell_e+\ell_s}$ and $\boxed{r} \leftarrow \{0,1\}^{\ell_r+\ell_e+\ell_s}$. We set $\boxed{\Delta_j} = p\boxed{a_j}\boxed{b_j} - e\boxed{a_{j+1}}$, using $\boxed{a_{N+1}} = ep^L$. We set $\boxed{\Delta} = \sum_{j=1}^{N}\boxed{a_j}^2 - e\boxed{V}$. We set $C_R = E(\boxed{V};\boxed{R})C^{-e}$. We set $c \leftarrow \text{com}(0,\ldots,0)$ and $c_r = \text{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \ldots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}; \boxed{r})c^{-e}$. Finally, we program the random oracle to return e when queried on (C, C_R, c, c_r).

Zero-Knowledge Argument for Correctness of a Limited Vote

Common input: Ciphertext C and public keys.
Prover's input: $0 \leq i_1 < \cdots < i_N < L$ and $R \in \{0,1\}^{\ell_R}$ such that
$C = E(\sum_{j=1}^{N} M^{i_j}; R)$.

Let $\alpha_{N+1} = p^L$. We prove correctness of the vote by producing

$$\mathrm{SPK}[(\upsilon, \rho, \alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N) : \\ C = E(\upsilon; \rho) \text{ and } \upsilon = \sum_{j=1}^{N} \alpha_j^2 \text{ and } \bigwedge_{j=1}^{N} \alpha_{j+1} = p\alpha_j\beta_j].$$

Argument: Let $V = \sum_{j=1}^{N} M^{i_j}$, choose $R_V \leftarrow \{0,1\}^{\ell_V + \ell_e + \ell_s}$, $R_R \leftarrow \{0,1\}^{\ell_R + \ell_e + \ell_s}$ and set $C_R = E(R_V; R_R)$.
Let $a_j = p^{i_j}, b_j = p^{i_{j+1} - i_j - 1}$, where $i_{N+1} = L$. Let $r_{a_{N+1}} = 0$ and choose $r_{a_1}, \ldots, r_{a_N}, r_{b_1}, \ldots, r_{b_N} \leftarrow \{0,1\}^{\ell_V/2 + \ell_e + \ell_s}$. Let $\Delta_j = pa_i r_{b_j} + pb_j r_{a_j} - r_{a_{j+1}}$ and $\Delta = 2\sum_{j=1}^{N} a_j r_{a_j} - R_V$. Set $c = \mathrm{com}(a_1, b_1, \Delta_1, \ldots, a_N, b_N, \Delta_N, \Delta; r)$. Set $c_r = \mathrm{com}(r_{a_1}, r_{b_1}, pr_{a_1}r_{b_1}, \ldots, r_{a_N}, r_{b_N}, pr_{a_N}r_{b_N}, \sum_{j=1}^{N} r_{a_j}^2; r_r)$.

Compute the challenge as $e \leftarrow \mathrm{hash}(C, C_R, c, c_r)$.

Set $\boxed{V} = eV + R_V = e\sum_{j=1}^{N} M^{i_j} + R_V$ and $\boxed{R} = eR + R_R$.
Set $\boxed{a_j} = ea_j + r_{a_j} = ep^{i_j} + r_{a_j}, \boxed{b_j} = eb_j + r_{b_j} = ep^{i_{j+1} - i_j - 1} + r_{b_j}$ and $\boxed{r} = er + r_r$,

The argument is $(C_R, c, c_r, \boxed{V}, \boxed{R}, \boxed{a_1}, \boxed{b_1}, \ldots, \boxed{a_N}, \boxed{b_N}, \boxed{r})$.

Verification: Compute e as above. Let $\boxed{a_{N+1}} = ep^L$ and set
$\boxed{\Delta_j} = p\boxed{a_j}\boxed{b_j} - e\boxed{a_{j+1}}$ and $\boxed{\Delta} = \sum_{j=1}^{N} \boxed{a_j}^2 - e\boxed{V}$.
Verify that $C^e C_R = E(\boxed{V}; \boxed{R})$ and
$c^e c_r = \mathrm{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \ldots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}; \boxed{r})$.

Fig. 2. Limited Vote Argument.

To argue that the simulated argument is indistinguishable from a real argument, consider the following hybrid argument. Let $i_1, \ldots, i_N$ be the chosen candidates and define $i_{N+1} = L, a_j = p^{i_j}, b_j = p^{i_{j+1} - i_j - 1}$. We proceed as in the simulation except when computing c. We set $R_V = \boxed{V} - e\sum_{j=1}^{N} M^{i_j}, r_{a_j} = \boxed{a_j} - ea_j, r_{b_j} = \boxed{b_j} - eb_j$. We let $a_{N+1} = p^L$ and $\Delta_j = pa_j r_{b_j} + pb_j r_{a_j} - a_{j+1}$. Compute $c \leftarrow \mathrm{com}(a_1, b_1, \Delta_1, \ldots, a_N, b_N, \Delta_N, 2\sum_{j=1}^{N} a_j r_{a_j} - R_V)$. The rest of the hybrid argument is carried out as in the simulation.

The hybrid argument is statistically indistinguishable from a real argument, all that is changed is the order in which we choose the elements. On the other hand, the only difference from a simulated argument is in the computation of the commitment c. The commitment scheme's hiding property shows that the hybrid argument is indistinguishable from a simulated argument of knowledge. Moreover, if the commitment scheme is statistically hiding then the hybrid argument is statistically indistinguishable from the simulated argument of knowledge.

Soundness and knowledge. Suppose an adversary produces a valid argument for ciphertext C containing a valid limited vote. We wish to extract a witness $(\upsilon, \rho, \iota_1, \ldots, \iota_N)$. To do so we rewind the adversary to the point where it queries the random oracle with C, C_R, c, c_r. We then give it random challenges until we get a new acceptable argument. This takes expected polynomial time. Let us call the two acceptable arguments $(C_R, c, c_r, e, \boxed{V}, \boxed{R}, \boxed{a_1}, \boxed{b_1}, \ldots, \boxed{a_N}, \boxed{b_N}, \boxed{r})$ and $(C_R, c, c_r, e', \boxed{V}', \boxed{R}', \boxed{a_1}', \boxed{b_1}', \ldots, \boxed{a_N}', \boxed{b_N}', \boxed{r}')$. We compute the corresponding $\boxed{\varDelta_1}, \ldots, \boxed{\varDelta_N}, \boxed{\varDelta}$ and $\boxed{\varDelta_1}', \ldots, \boxed{\varDelta_N}', \boxed{\varDelta}'$ as in the verification.

Since the arguments are acceptable we have $C^e C_R = E(\boxed{V}; \boxed{R})$ and $C^{e'} C_R = E(\boxed{V}'; \boxed{R}')$. This gives us $C^{e-e'} = E(\boxed{V} - \boxed{V}'; \boxed{R} - \boxed{R}')$. With overwhelming probability we have $e \neq e'$ and using the root extraction property of the cryptosystem we can extract (υ, ρ) so $C = E(\upsilon; \rho)$.

It remains to argue that υ is a message on the form $\sum_{j=1}^N M^{\iota_j}$ for $0 \leq \iota_1 < \cdots < \iota_N < L$. From $c^{e-e'} = \mathrm{com}(\boxed{a_1} - \boxed{a_1}', \boxed{b_1} - \boxed{b_1}', \boxed{\varDelta_1} - \boxed{\varDelta_1}', \ldots, \boxed{a_N} - \boxed{a_N}', \boxed{b_N} - \boxed{b_N}', \boxed{\varDelta_N} - \boxed{\varDelta_N}', \boxed{\varDelta} - \boxed{\varDelta}'; \boxed{r} - \boxed{r}')$ we get an opening $(\alpha_1, \beta_1, \varDelta_1, \ldots, \alpha_N, \beta_N, \varDelta_N, \varDelta, \rho_c)$ of c. From $c_r = \mathrm{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\varDelta_1}, \ldots, \boxed{a_N}, \boxed{b_N}, \boxed{\varDelta_N}\boxed{\varDelta}; \boxed{r})c^{-e}$ we then get an opening $(\rho_{a_1}, \rho_{b_1}, \rho_{\varDelta_1}, \ldots, \rho_{a_N}, \rho_{b_N}, \rho_{\varDelta_N}, \rho_\varDelta, \rho_r)$ of c_r. Moreover, define $\rho_V = \boxed{V} - e\upsilon$, $\rho_R = \boxed{R} - e\rho$ and we have $C_R = E(R_V; \rho_R)$.

Consider now an adversary having noticeable probability of making an acceptable argument of knowledge using C, C_R, c, c_r. It must use $\boxed{a_j} = e\alpha_j + \rho_{a_j}$, $\boxed{b_j} = e\beta_j + \rho_{b_j}$, $\boxed{\varDelta_j} = e\varDelta_j + \rho_{\varDelta_j}$. We have equations $\boxed{\varDelta_j} = p\boxed{a_j}\boxed{b_j} - e\boxed{a_{j+1}}$, where by definition $\boxed{a_{N+1}} = e\alpha_{N+1} = ep^L$. This means $e^2(p\alpha_j\beta_j - \alpha_{j+1}) + e(p\alpha_j\rho_{b_j} + \beta_j\rho_{a_j} - \rho_{a_{j+1}} - \varDelta_j) + p\rho_{a_j}\rho_{b_j} - \rho_{\varDelta_j} = 0$. With overwhelming probability over the choice of e we then have $\bigwedge_{j=1}^N \alpha_{j+1} = p\alpha_j\beta_j$. This means $p\alpha_1|\alpha_2, \ldots, p\alpha_N|p^L$, so there exists $0 \leq \iota_1 < \cdots < \iota_N < L$ so $\alpha_j = \pm p^{\iota_j}$.

Likewise, if the adversary has noticeable probability of making an acceptable argument of knowledge with C, C_R, c, c_r it must use $\boxed{\varDelta} = e\varDelta + \rho_\varDelta$ and $\boxed{V} = e\upsilon + \rho_V$. We verify that $\boxed{\varDelta} = \sum_{j=1}^N \boxed{a_j}^2 - e\boxed{V}$, i.e., $e^2(\sum_{j=1}^N \alpha_j^2 - \upsilon) + e(\sum_{j=1}^N \alpha_j\rho_{a_j} - \rho_V - \varDelta) + \sum_{j=1}^N \rho_{a_j}^2 - \rho_\varDelta = 0$. With overwhelming probability over e we must therefore have

$$\upsilon = \sum_{j=1}^N \alpha_j^2 = \sum_{j=1}^N (\pm p^{\iota_j})^2 = \sum_{j=1}^N M^{\iota_j}. \qquad \square$$

4 Approval Vote

In approval voting the voter can vote for as many different candidates as he likes. The advantage of this kind of voting system is that the voter does not risk wasting votes by selecting his preferred candidate. Compare this to other voting systems where it may be foolish to cast a vote for a candidate who has little chance of winning. In this kind of election the number of votes cast by the voter may be anywhere between 0 and L.

Define $a_i = 1$ if the voter wishes to vote for candidate i and $a_i = 0$ if he does not. The plaintext vote is $V = \sum_{i=0}^{L-1} a_i M^i$. The voter encrypts this to get a ciphertext $C = E(\sum_{i=0}^{L-1} a_i M^i; R)$. He now needs to prove that indeed the plaintext is on the right form.

We commit to $a_0, \ldots, a_{L-1}$. In order to prove that the hidden $a_i \in \{0,1\}$ we use the fact that $x^2 \geq x$ for any integer, obtaining only equality if $x = 0$ or $x = 1$. This means that if we can prove $\sum_{i=0}^{L-1}(a_i^2 - a_i) = 0$, then all a_i's belong to $\{0,1\}$.

Using standard techniques, we get out hidden variables $\boxed{a_i} = ea_i + r_{a_i}$ as well as $\boxed{\Delta} = e\Delta + r_\Delta$, where Δ is a committed value. In the verification, we end up with an equation $\boxed{\Delta} = \sum_{i=0}^{L-1}(\boxed{a_i}^2 - e\boxed{a_i})$. The left hand side is a degree 1 polynomial in e and the right hand side is a degree 2 polynomial in e. With overwhelming probability over e, the equation implies $\sum_{i=0}^{L-1}(a_i^2 - a_i) = 0$ as we wanted.

The other parts of the NIZK argument are a proof of knowledge of the plaintext V, as well as an argument that this plaintext is constructed as described above using the a_i's that we committed to.

Theorem 3. *In the random oracle model, the protocol in Figure 3 is a NIZK argument of knowledge for C containing a correctly formed approval vote. If the commitment scheme is statistically hiding then the argument is statistical zero-knowledge.*

We prove Theorem 3 in the full paper.

Limited vote with large N. It is possible to modify the protocol into an NIZK argument of correctness of an approval vote with the additional condition that $\sum_{i=0}^{L-1} \alpha_i = N$ for some known N. The addition can be made at low computational cost. This variation can be used as an alternative to the limited vote argument from the previous section.

The $\boxed{a_i}$'s are of small size, while the limited vote argument may use very large exponents in large elections with many candidates. The limited vote argument is thus suitable when N is small in comparison with L, while for large N it is better to use the variation of the approval vote argument.

5 Divisible Vote

Consider a shareholder election where each share gives the right to cast one vote. It may be impractical for large shareholders to cast multiple single votes, or even to use the limited vote technique, since it forces them to make a huge number of encryptions. We prefer proving in a direct manner that the ciphertext contains a vote on the form $\sum_{i=0}^{L-1} v_i M^i$, where v_i is the number of votes on candidate i.

In [IMO03] they call this divisible voting and offer zero-knowledge arguments for correctness of a divisible vote. We suggest an alternative NIZK argument that takes full advantage of integer commitments. In comparison with [IMO03] we save a factor $\log N$ in complexity, where N is the number of votes the voter has, and we benefit from using integer commitments instead of encryptions.

Zero-Knowledge Argument for Correctness of an Approval Vote

Common input: Ciphertext C and public keys.
Private input: $a_0, \ldots, a_{L-1} \in \{0,1\}$ and $R \in \{0,1\}^{\ell_R}$ such that
$C = E(\sum_{i=0}^{L-1} a_i M^i; R)$.

We prove correctness of the vote by producing

$$\mathrm{SPK}[(\upsilon, \rho, \alpha_0, \ldots, \alpha_{L-1}) : C = E(\upsilon; \rho) \text{ and } \upsilon = \sum_{i=0}^{L-1} \alpha_i M^i \text{ and } \sum_{i=0}^{L-1} (\alpha_i^2 - \alpha_i) = 0]$$

Argument: Choose $R_V \leftarrow \{0,1\}^{\ell_V+\ell_e+\ell_s}$, $R_R \leftarrow \{0,1\}^{\ell_R+\ell_e+\ell_s}$ and set $C_R = E(R_V; R_R)$.
Choose $r_{a_0}, \ldots, r_{a_{L-1}} \leftarrow \{0,1\}^{1+\ell_e+\ell_s}$ and let $\Delta = \sum_{i=0}^{L-1}(2a_i - 1)r_{a_i}$. Choose $r \leftarrow \{0,1\}^{\ell_r}$ and set $c = \mathrm{com}(a_0, \ldots, a_{L-1}, \Delta; r)$. Choose $r_r \leftarrow \{0,1\}^{\ell_r+\ell_e+\ell_s}$ and set $c_r = \mathrm{com}(r_{a_0}, \ldots, r_{a_{L-1}}, \sum_{i=0}^{L-1} r_{a_i}^2; r_r)$.

Compute a challenge $e \leftarrow \mathrm{hash}(C, C_R, c, c_r)$.

Set $\boxed{R} = eR + R_R$. Set $\boxed{a_i} = ea_i + r_{a_i}$ and $\boxed{r} = er + r_r$.

The argument is $(C_R, c, c_r, \boxed{R}, \boxed{a_0}, \ldots, \boxed{a_{L-1}}, \boxed{r})$.

Verification: Compute e as above. Define $\boxed{V} = \sum_{i=0}^{L-1} \boxed{a_i} M^i$ and
$\boxed{\Delta} = \sum_{i=0}^{L-1}(\boxed{a_i}^2 - e\boxed{a_i})$.
Verify $C^e C_R = E(\boxed{V}; \boxed{R})$ and $c^e c_r = \mathrm{com}(\boxed{a_0}, \ldots, \boxed{a_{L-1}}, \boxed{\Delta}; \boxed{r})$.

Fig. 3. Approval Vote Argument.

The idea is the following. We commit to $v_0, \ldots, v_{L-1}$. We prove that indeed the ciphertext contains $\sum_{i=0}^{L-1} v_i M^i$. We also prove that all these elements $v_0, \ldots, v_{L-1}$ are non-negative. Finally, we prove that their sum is N.

To prove that an element is positive we could use Boudot's argument [Bou02] or we could use [LAN02]'s argument where v_i is proven to be a sum of four squares. We offer a variation over the latter idea. It is a well-known fact from number theory that the only numbers that cannot be written as the sum of three squares are on the form $4^n(8k+7)$. This means $4v_i + 1$ can be written as a sum of three squares. Obviously, writing $4v_i + 1$ as the sum of three squares implies that v_i is non-negative.

Rabin and Shallit [RS86] offer an efficient and simple algorithm for finding three such squares, for sufficiently large numbers. In our case, the numbers are relatively small though; in few elections do voters have more than a million votes. It is not hard to change their algorithm into something that is suitable for small numbers though, since for small numbers factorization is easy. We discuss in the full paper how to decompose $4v_i + 1$ into three squares.

Theorem 4. *In the random oracle model, the protocol in Figure 4 is a NIZK argument of knowledge for a ciphertext containing a specified number N votes. If the commitment scheme is statistically hiding then the argument is statistical zero-knowledge.*

We prove Theorem 4 in the full paper.

Zero-Knowledge Argument for Correctness of a Divisible Vote

Common input: Ciphertext C, a number of votes N and public keys.
Private input: $0 \leq v_0, \ldots, v_{L-1}$ and $R \in \{0,1\}^{\ell_R}$ such that $N = \sum_{i=0}^{L-1} v_i$ and $C = E(\sum_{i=0}^{L-1} v_i M^i; R)$.

We prove correctness of the vote by producing

$$\mathrm{SPK}[(v, \rho, v_0, \alpha_0, \beta_0, \delta_0, \ldots, v_{L-1}, \alpha_{L-1}, \beta_{L-1}, \delta_{L-1}) : C = E(v; \rho) \text{ and } v = \sum_{i=0}^{L-1} v_i M^i \text{ and } \bigwedge_{i=0}^{L-1} 4v_i + 1 = \alpha_i^2 + \beta_i^2 + \delta_i^2 \text{ and } N = \sum_{i=0}^{L-1} v_i].$$

Argument: Find a_i, b_i, d_i such that $4v_i + 1 = a_i^2 + b_i^2 + d_i^2$. Choose $r_{v_i}, r_{a_i}, r_{b_i}, r_{d_i} \leftarrow \{0,1\}^{\log N + \ell_e + \ell_s}$. Let $\Delta_i = 4r_{v_i} - 2a_i r_{a_i} - 2b_i r_{b_i} - 2d_i r_{d_i}$. Choose $r \leftarrow \{0,1\}^{\ell_r}$ and set $c = \mathrm{com}(v_0, a_0, b_0, d_0, \Delta_0, \ldots, v_{L-1}, a_{L-1}, b_{L-1}, d_{L-1}, \Delta_{L-1}; r)$. Choose $r_r \leftarrow \{0,1\}^{\ell_r + \ell_e + \ell_s}$ and set $c_r = \mathrm{com}(r_{v_0}, r_{a_0}, r_{b_0}, r_{d_0}, -r_{a_0}^2 - r_{b_0}^2 - r_{d_0}^2, \ldots, r_{v_{L-1}}, r_{a_{L-1}}, r_{b_{L-1}}, r_{d_{L-1}}, -r_{a_{L-1}}^2 - r_{b_{L-1}}^2 - r_{d_{L-1}}^2; r_r)$.
Let $R_V = \sum_{i=0}^{L-1} r_{v_i} M^i$ and choose $R_R \leftarrow \{0,1\}^{\ell_R + \ell_e + \ell_s}$. Set $C_R = E(R_V; R_R)$.
Set $r_\sum = \sum_{i=0}^{L-1} r_{v_i}$.

Compute the challenge as $e \leftarrow \mathrm{hash}(C, C_R, c, c_r, r_\sum)$.

Let $\boxed{R} = eR + R_R$. Let $\boxed{v_i} = ev_i + r_{v_i}$, $\boxed{a_i} = ea_i + r_{a_i}$, $\boxed{b_i} = eb_i + r_{b_i}$, $\boxed{d_i} = ed_i + r_{d_i}$ and $\boxed{r} = er + r_r$.

The argument is
$(C_R, c, c_r, r_\sum, \boxed{R}, \boxed{v_0}, \boxed{a_0}, \boxed{b_0}, \boxed{d_0}, \ldots, \boxed{v_{L-1}}, \boxed{a_{L-1}}, \boxed{b_{L-1}}, \boxed{d_{L-1}}, \boxed{r})$.

Verification: Compute the challenge e as in the argument. Define $\boxed{\Delta_i} = e(4\boxed{v_i} + e) - \boxed{a_i}^2 - \boxed{b_i}^2 - \boxed{d_i}^2$. Set $\boxed{V} = \sum_{i=0}^{L-1} \boxed{v_i} M^i$.
Verify $C^e C_R = E(\boxed{V}; \boxed{R})$, $c^e c_r = \mathrm{com}(\boxed{v_0}, \ldots, \boxed{\Delta_{L-1}}; \boxed{r})$ and $\sum_{i=0}^{L-1} \boxed{v_i} = eN + r_\sum$.

Fig. 4. Divisible Vote Argument.

6 Borda Vote

In Borda voting, voters cast weighted votes. The worst candidate gets 1 vote, the second worst 2 votes, and so forth. A valid vote is therefore on the form $\prod_{i=1}^{L} \pi(i) M^{i-1}$ for some permutation $\pi \in \Sigma_L$. We will suggest an efficient argument for correctness of such a vote[5].

[5] Interestingly, it turns out that in Borda voting we do not need an integer commitment scheme; we can use a commitment scheme based on a group of known order q. We just need to take care that q is large enough to avoid overflows, the $\boxed{a_i}$'s in the protocol should come out as unreduced integers.

To prove correctness of a Borda vote corresponding to permutation π we form a commitment $c \leftarrow \mathrm{com}(\pi(1), \ldots, \pi(L))$. Using the Fiat-Shamir heuristic on an argument for correctness of a shuffle [Gro03, Fur04a] we can demonstrate that c has been correctly formed. In [Gro03] there is a shuffle argument for known messages. This means we can take advantage of the fact that we know that the messages are known to be $1, \ldots, L$ to obtain greater efficiency. Once we have formed the commitment and demonstrated that the content is indeed a permutation of $1, \ldots, L$, then it is pretty straightforward to prove knowledge of the plaintext of the encrypted vote as well as show that the content is on the form described above.

Theorem 5. *In the random oracle model, the protocol in Figure 5 is a NIZK argument of knowledge of C containing a correctly formed Borda vote. If the commitment scheme is statistically hiding and the shuffle argument is statistical zero-knowledge then the argument is statistical zero-knowledge.*

We prove Theorem 5 in the full paper.

Zero-Knowledge Argument for Correctness of a Borda Vote

Common input: Ciphertext C and public keys.
Private input: $\pi \in \Sigma_L$ and $R \in \{0,1\}^{\ell_R}$ such that $C = E(\sum_{i=1}^{L} \pi(i) M^{i-1}; R)$.

We argue correctness of the vote by making the following signature of knowledge

$$\mathrm{SPK}[(v, \rho, \pi \in \Sigma_L, \alpha_1, \ldots, \alpha_L) : C = E(v;\rho) \text{ and } v = \sum_{i=1}^{L} \alpha_i M^{i-1} \text{ and } \bigwedge_{i=1}^{L} \alpha_i = \pi(i)]$$

Argument: Define $a_i = \pi(i)$, choose $r \leftarrow \{0,1\}^{\ell_r}$ and set $c = \mathrm{com}(a_1, \ldots, a_L; r)$. In addition make a signature of knowledge of c being a commitment to a permutation of $1, \ldots, L$. I.e., set

$$p \leftarrow \mathrm{SPK}[(\rho_c, \pi \in \Sigma_L) : c = \mathrm{com}(\pi(1), \ldots, \pi(L); \rho_c)].$$

Choose $r_{a_1}, \ldots, r_{a_L} \leftarrow \{0,1\}^{\log L + \ell_e + \ell_s}$ and $r_r \leftarrow \{0,1\}^{\ell_r + \ell_e + \ell_s}$ and set $c_r = \mathrm{com}(r_{a_1}, \ldots, r_{a_L}; r_r)$.
Define $R_V = \sum_{i=1}^{L} r_{a_i} M^{i-1}$. Choose $R_R \leftarrow \{0,1\}^{\ell_R + \ell_e + \ell_s}$ and set $C_R = E(R_V; R_R)$.

Compute a challenge as $e \leftarrow \mathrm{hash}(C, C_R, c, c_r, p)$.

Set $\boxed{R} = eR + R_R$. Set $\boxed{a_i} = e a_i + r_{a_i} = e\pi(i) + r_{a_i}$ and $\boxed{r} = er + r_r$.

The argument is $(C_R, c, c_r, \boxed{R}, \boxed{a_1}, \ldots, \boxed{a_L}, \boxed{r}, p)$.

Verification: Verify the shuffle argument p. Compute e as in the argument. Set $\boxed{V} = \sum_{i=1}^{L} \boxed{a_i} M^{i-1}$
Verify $C^e C_R = E(\boxed{V}; \boxed{R})$ and $c^e c_r = \mathrm{com}(\boxed{a_1}, \ldots, \boxed{a_L}; \boxed{r})$.

Fig. 5. Borda Vote Argument.

Acknowledgment

Thanks to one of the referees for pointing us to [WfL04].

References

Ben87. Josh Cohen Benaloh. Verifiable secret ballot elections. Technical Report 561, Yale University, 1987. PhD thesis. x+123 pp.

BFP^{+}01. Oliver Baudron, Pierre-Alain Fouque, David Pointcheval, Guillaume Poupard, and Jacques Stern. Practical multi-candidate election scheme. In *proceedings of PODC '01*, pages 274–283, 2001.

Bou02. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *proceedings of EUROCRYPT '00, LNCS series, volume 1807*, pages 431–444, 2002.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS '93*, pages 62–73, 1993.

BY86. Josh Cohen Benaloh and Moti Yung. Distributing the power of a government to enhance the privacy of voters. In *proceedings of PODC '86*, pages 52–62, 1986.

CF85. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme. In *proceedings of FOCS '85*, pages 372–382, 1985.

CGS97. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally eficient multi-authority election scheme. In *proceedings of EUROCRYPT '97, LNCS series, volume 1233*, pages 103–118, 1997.

CS97. Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In *proceedings of CRYPTO '97, LNCS series, volume 1294*, pages 410–424, 1997.

DF02. Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *proceedings of ASIACRYPT '02, LNCS series, volume 2501*, pages 125–142, 2002.

DGS03. Ivan Damgård, Jens Groth, and Gorm Salomonsen. The theory and implementation of an electronic voting system. In D. Gritzalis, editor, *Secure Electronic Voting*, pages 77–100. Kluwer Academic Publishers, 2003.

DJ01. Ivan Damgård and Mads J. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *proceedings of PKC '01, LNCS series, volume 1992*, 2001.

DJN03. Ivan Damgård, Mads J. Jurik, and Jesper Buus Nielsen. A generalization of paillier's public-key system with applications to electronic voting. Manuscript, 2003. `http://www.brics.dk/∼ivan/GenPaillierfinaljour.ps`.

ElG84. Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *proceedings of CRYPTO '84, LNCS series, volume 196*, pages 10–18, 1984.

FO97. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *proceedings of CRYPTO '97, LNCS series, volume 1294*, pages 16–30, 1997.

Fur04a. Jun Furukawa. Efficient, verifiable shuffle decryption and its requirement of unlinkability. Manuscript, 2004. Full version of [Fur04b].

Fur04b. Jun Furukawa. Efficient, verifiable shuffle decryption and its requirement of unlinkability. In *proceedings of PKC '04, LNCS series, volume 2947*, pages 319–332, 2004.

Gro03. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. In *proceedings of PKC '03, LNCS series, volume 2567*, pages 145–160, 2003.

Gro05. Jens Groth. Cryptography in subgroups of $\mathbb{Z}_n^*$. In *proceedings of TCC '05, LNCS series, volume 3378*, pages 50–65, 2005.

IMO03. Natsuki Ishida, Shin'ichiro Matsuo, and Wakaha Ogata. Divisible voting scheme. In *ISC '03, LNCS series, volume 2851*, pages 137–150, 2003.

LAN02. Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure vickrey auctions without threshold trust. In *proceedings of Financial Cryptography '02, LNCS series, volume 2357*, pages 87–101, 2002.

Lip03. Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *proceedings of ASIACRYPT '03, LNCS series, volume 2894*, pages 398–415, 2003.

Pai99. Pascal Paillier. Public-key cryptosystems based on composite residuosity classes. In *proceedings of EUROCRYPT '99, LNCS series, volume 1592*, pages 223–239, 1999.

RS86. Michael Rabin and Jeffrey Shallit. Randomized algorithms in number theory. *Commun. Pure and Appl. Math*, 39, suppl.:S240–S256, 1986.

Wan05. Changjie Wang. Personal communication, 2005.

WfL04. Changjie Wang and Ho fung Leung. A secure and fully private borda voting protocol with universal verifiability. In *proceedings of COMPSAC '04*, pages 224–229, 2004.